ГОСУДАРСТВЕННЫЙ СТАНДАРТ
СОЮЗА ССР

ЯЗЫК ПРОГРАММИРОВАНИЯ КОБОЛ

ГОСТ 22558—89
(СТ СЭВ 6184—88, ИСО 1989—85)

ЧАСТИ 1—7

Издание официальное

5 р. 90 к.

ГОСУДАРСТВЕННЫЙ КОМИТЕТ СССР ПО УПРАВЛЕНИЮ
КАЧЕСТВОМ ПРОДУКЦИИ И СТАНДАРТАМ
Москва

GOST EXPERT ГОСТ 22558-89,

УДК 681.3.06:006.354 Группа П85

ГОСУДАРСТВЕННЫЙ СТАНДАРТ СОЮЗА ССР

ЯЗЫК ПРОГРАММИРОВАНИЯ КОБОЛ

Programming language COBOL

ГОСТ 22558—89 (СТ СЭВ 6184—88, ИСО 1989—85)

ОКСТУ 4002

Дата введения 01.01.91

Часть 1. ОСНОВНЫЕ ПОЛОЖЕНИЯ

1. ВВЕДЕНИЕ К СТАНДАРТУ

1.1. Область действия и назначение

Настоящий стандарт распространяется на форму и интерпретацию программ, выраженных в русской или английской нотации языка Кобол. Он предназначен для обеспечения высокой степени машинной независимости Кобол-программ и их совместимости в различных системах автоматической обработки данных.

1.2. Структура спецификаций языка

Организация спецификаций Кобола в настоящем документе базируется на понятии функционального обрабатывающего модуля. В языке выделены одиннадцать функциональных обрабатывающих модулей:

ядро;
последовательный ввод-вывод;
относительный ввод-вывод;
индексный ввод-вывод;
межпрограммные связи;
сортировка-слияние;
обработка исходных текстов;
генератор отчетов;
коммуникации;
отладка;
сегментация.

Девять из модулей, как указано ниже, содержат элементы, которые в модуле разделены на элементы уровня 1 и элементы уров-

Издание официальное

★

© Издательство стандартов, 1991

Настоящий стандарт не может быть полностью или частично воспроизведен, тиражирован и распространен без разрешения Госстандарта СССР

ня 2. Элементы уровня 1 модуля являются подмножеством элементов уровня 2 того же модуля; два модуля содержат только элементы уровня 1.

Модуль ядра содержит элементы языка для внутренней обработки данных в базисной структуре четырех разделов программ. Ядро также содержит элементы языка для определения таблиц и доступа к ним. Элементы ядра разделены на два уровня. Уровень 1 ядра содержит элементы, необходимые для выполнения основных внутренних операций, т. е. элементарные варианты различных фраз и операторов. Уровень 2 ядра обеспечивает более широкие возможности внутренней обработки данных.

Модуль последовательного ввода-вывода содержит элементы языка для определения файлов с последовательной организацией и доступа к ним. Элементы модуля последовательного ввода-вывода разделены на два уровня. Уровень 1 модуля последовательного ввода-вывода включает элементы для основных возможностей определения и доступа к последовательным файлам. Уровень 2 модуля послсдовательного ввода-вывода содержит элементы для полных возможностей определения и доступа к последовательным файлам.

Модуль относительного ввода-вывода содержит элементы языка для определения файлов массовой памяти, записи которых идентифицируются своими относительными номерами, доступа к ним. Элементы модуля относительного ввода-вывода разделены на два уровня. Уровень 1 относительного ввода-вывода содержит элементы для основных возможностей определения файлов с относительной организацией и доступа к ним. Уровень 2 относительного ввода-вывода содержит элементы для более полных возможностей, включающих возможность как произвольного, так и последовательного доступа к файлу в одной и той же программе Кобола.

Модуль индексного ввода-вывода содержит элементы языка для определения файлов массовой памяти и доступа к ним. Записи этих файлов идентифицируются значением ключа и доступны посредством индекса. Элементы модуля индексного ввода-вывода разделены на два уровня. Уровень 1 индексного ввода-вывода содержит элементы для основных возможностей определения и доступа к индексным файлам. Уровень 2 индексного ввода-вывода содержит элементы для более полных возможностей, включающих дополнительные ключи и возможность как произвольного, так и последовательного доступа к файлу в одной и той же программе Кобола.

Модуль межпрограммных связей содержит элементы языка, позволяющие программе поддерживать связь с одной или несколькими другими программами. Элементы модуля межпрограммных связей разделяются на два уровня. Уровень 1 межпрограммных

GOST EXPERT ГОСТ 22558-89,

связей предоставляет элементы для передачи управления другой программе, известной во время компиляции, и для доступа к определенным общим данным в обеих программах. Уровень 2 межпрограммных связей предоставляет элементы для передачи управления другой программе, не идентифицированной во время компиляции, и для обеспечения вложенности программ.

Модуль сортировки-слияния содержит элементы языка для упорядочения одного или нескольких файлов. Модуль сортировки-слияния также содержит элементы языка для соединения двух или более идентично упорядоченных файлов. По желанию пользователь может применить некоторую специальную обработку к каждой из отдельных записей посредством процедур ввода или вывода. Модуль сортировки-слияния содержит элементы только одного уровня.

Модуль обработки исходного текста содержит элементы языка для вставки и замены текста исходной программы как части компилируемой исходной программы. Элементы модуля обработки исходного текста разделены на два уровня. Уровень 1 модуля предоставляет возможность копирования текста в исходную программу из единственной библиотеки. Уровень 2 данного модуля предоставляет дополнительные возможности изменения библиотечного текста в процессе копирования, определения нескольких библиотек Кобола во время компиляции и изменения текста исходной программы.

Модуль генератора отчетов содержит элементы языка для полуавтоматического производства печатных отчетов. Модуль генератора отчетов содержит элементы только одного уровня.

Модуль коммуникаций содержит элементы языка для получения, обработки и создания сообщений или их частей и для связи с коммуникационными устройствами при помощи системы управления сообщениями. Элементы модуля коммуникаций разделены на два уровня. Уровень 1 коммуникаций предоставляет элементы для основных возможностей передачи или получения полных сообщений. Уровень 2 коммуникаций предоставляет элементы для более широких возможностей, включающих возможность посылать или получать сегменты сообщения.

Модуль отладки предоставляет средства, при помощи которых пользователь может указывать свой алгоритм отладки — условия, при которых данные или процедуры контролируются во время выполнения программы. Элементы модуля отладки разделяются на два уровня. Уровень 1 модуля отладки предоставляет основные возможности отладки, включающие возможность указывать выборочное или полное контролирование параграфов. Уровень 2 модуля отладки предоставляет полные возможности отладки в языке Кобол.

GOST EXPERT ГОСТ 22558-89,

Модуль сегментации обеспечивает совмещение памяти для секций раздела процедур во время выполнения. Элементы модуля сегментации разделены на два уровня. Уровень 1 сегментации предусматривает номера сегментов секций и фиксированные границы сегментов. Уровень 2 сегментации добавляет средства изменения границы сегментов.

1.3. **Структура документа**

Настоящий документ состоит из 17 частей.

Часть 1 содержит справочные сведения о языке и состоит из введения к стандарту и списка элементов языка.

Часть 2 представляет концепции, имеющие отношение к использованию и организации средств языка Кобол.

Часть 3 состоит из глоссария, определяющего термины в соответствии с их значением в Коболе.

Часть 4 содержит общие сведения.

Часть 5 содержит сводки форматов в английской и русской нотациях.

Части 6—16 содержат спецификации одиннадцати функциональных обрабатывающих модулей. В этих частях спецификация уровня 2 выделены рамкой.

В частях 2—16 включены подробные спецификации стандарта языка Кобол. В части 17 содержатся приложения.

1.4. **Рекомендации по использованию текста стандарта**

Очевидно, что стандарт будет изучаться с нескольких различных точек зрения.

Ключом к изучению стандарта, кроме оглавления, служит также список элементов по модулям, в котором содержится подробная конкретизация каждого элемента стандарта Кобола в данном модуле. Например, установить содержание уровня 1 модуля последовательного ввода-вывода можно, найдя список элементов Кобола, включающий общие сведения, статьи раздела оборудования, статьи раздела данных и глаголы раздела процедур, имеющие отношение к модулю последовательного ввода-вывода.

Спецификация каждого из элементов языка производится как для русской, так и для английской нотаций. Там, где это требуется в тексте настоящего стандарта, приводятся форматы. Между конструкциями английской и русской нотаций языка Кобол, специфицированных настоящим стандартом, существует полное семантическое соответствие, если исходные данные объектной программы подготовлены без использования русского алфавита. Там, где в тексте требуется ссылка на лексические элементы языка, первым приводится лексический элемент английской нотации, за ним в скобках следует элемент русской нотации. Форматы языковых конструкций приводятся также для английской и русской нотаций. Спецификации особенностей использования данных, представлен-

GOST EXPERT ГОСТ 22558-89,

ных в русской лексике, снабжаются пометкой «для русской нотации».

Определение модулей и их уровней, на которых появляются определенные средства языка, приводится в списке элементов Кобола по разделам Кобола. В этом списке приведены в деталях все элементы стандарта языка Кобол и указано их появление в различных модулях. Элементы, не содержащиеся полностью на одном уровне модуля, приведены в деталях, достаточных для определения нахождения каждого подэлемента. Например, оператор READ (ЧИТАТЬ) появляется на уровне 1 модуля последовательного ввода-вывода, относительного ввода-вывода и индексного ввода-вывода. Так как определенные фразы оператора READ (ЧИТАТЬ) появляются только на уровне 2 этих модулей, подэлементы оператора READ (ЧИТАТЬ) вынесены в отдельный список.

Графическое представление 11 функциональных обрабатывающих модулей, образующих стандарт Кобола, показано в табл. 1, в которой используются сокращения (например, 2 ИПД 0,2), указывающие иерархическую позицию каждого уровня в функциональном обрабатывающем модуле, также как и число уровней, на которые разделены элементы модуля. Сокращенная запись состоит (слева направо) из однозначного числа, указывающего позицию уровня в иерархии, трехбуквенного обозначения модуля и двух однозначных чисел, указывающих минимальный и максимальный уровни модуля. Число нуль указывает, что допустимо пустое подмножество, соответствующее наименьшему уровню в модуле. Например, 2 ИПД 0,2 указывает, что этот уровень является вторым уровнем модуля индексного ввода-вывода, который содержит нулевой уровень и два ненулевых уровня (уровень 1 и уровень 2). 2 ЯДР 1,2 указывает, что этот уровень является вторым ненулевым уровнем ядра, которое состоит из двух ненулевых уровней (уровень 1 и уровень 2).

Для указания модулей используются следующие обозначения.

Ядро	ЯДР
Последовательный ввод-вывод	ПОД
Относительный ввод-вывод	ОТД
Индексный ввод-вывод	ИПД
Межпрограммные связи	МПС
Сортировка-слияние	СРТ
Обработка исходных текстов	ОИТ
Генератор отчетов	ГОТ
Коммуникации	КОМ
Отладка	ОТЛ
Сегментация	СЕГ

GOST EXPERT ГОСТ 22558-89,

Подмножества Кобола	Обязательные (обязательны в под				
	Ядро	Последовательный ввод-вывод	Относительный ввод-вывод	Индексный ввод-вывод	Межпрограммные связи
Максимальное	2 ЯДР 1,2	2 ПОД 1,2	2 ОТД 0,2	2 ИИД 0,2	2 МПС 1,2
Промежуточное	1 ЯДР 1,2	1 ПОД 1,2	1 ОТД 0,2	1 ИПД 0,2	1 МПС 1,2
Минимальное	1 ЯДР 1,2	1 ПОД 1,2	Нуль (пустое подмножество)	Нуль (пустое подмножество)	1 МПС 1,2

1.5. Определение реализации стандарта языка Кобол

В этом документе представлены описания средств языка, образующих стандарт языка Кобол. Стандарт языка Кобол состоит из 11 модулей, семь из которых обязательны и четыре — необязательны. В п. 1.5 и его подпунктах определены критерии для правильного определения того, насколько реализация соответствует стандарту языка Кобол.

1.5.1. Определение подмножеств

Имеются три подмножества стандарта языка Кобол: максимальное подмножество, промежуточное подмножество и минимальное подмножество. Каждое подмножество состоит из уровня семи обязательных модулей: ядра, последовательного ввода-вывода, относительного ввода-вывода, индексного ввода-вывода, межпрограммных связей, сортировки-слияния и обработки исходных текстов. В табл. 1 подмножество стандарта языка Кобол представлено одной из трех горизонтальных строк в столбцах обязательных модулей. Четыре модуля (генератор отчетов, коммуникации, отладка и сегментация) являются необязательными в трех подмножествах стандарта языка Кобол.

Максимальное подмножество стандарта языка Кобол содержит все элементы наивысшего уровня всех обязательных модулей, а именно:

элементы уровня 2 ядра, последовательного ввода-вывода, относительного ввода-вывода, индексного ввода-вывода, межпрограммных связей и обработки исходных текстов;

элементы уровня 1 сортировки-слияния.

Промежуточное подмножество стандарта языка Кобол содержит все элементы уровня 1 всех обязательных модулей:

GOST EXPERT ГОСТ 22558-89,

Таблица 1

модули множествах)		Необязательные модули (не обязательны в подмножествах)			
Сортировка-слияние	Обработка исходных текстов	Генератор отчетов	Коммуникации	Отладка	Сегментация
1 СРТ 0,1	2 ОИТ 0,2	1 ГОТ 0,1	2 КОМ 0,2	2 ОТЛ 0,2	2 СЕГ 0,2
1 СРТ 0,1	1 ОИТ 0,2		1 КОМ 0,2	1 ОТЛ 0,2	1 СЕГ 0,2
Нуль (пустое подмножество)	Нуль (пустое подмножество)				

элементы уровня 1 ядра, последовательного ввода-вывода, относительного ввода-вывода, индексного ввода-вывода, межпрограммных связей, сортировки-слияния и обработки исходных текстов.

Минимальное подмножество стандарта языка Кобол содержит все элементы уровня 1 ядра, последовательного ввода-вывода и межпрограммных связей.

1.5.2. Определение соответствия реализации стандарту

Реализация, соответствующая стандарту языка Кобол, должна полностью поддерживать любое из трех подмножеств, определенных в п. 1.5.1, и может включать все, любую комбинацию или ни одного из уровней необязательных модулей.

Соответствующая стандарту реализация данного подмножества стандарта языка Кобол должна полностью поддерживать все элементы языка этого подмножества (п. 1.5.2.5 настоящей части).

Кроме того, каждая реализация должна удовлетворять требованиям, изложенным ниже (пп. 1.5.2.1—1.5.2.4).

1.5.2.1. *Замена элементов или дополнительные элементы языка*

Реализация не должна для выполнения функций, идентичных функциям элементов стандарта языка Кобол, допускать замену или включение дополнительных элементов языка в исходной программе. Однако в спецификациях стандарта языка Кобол имеется ряд элементов языка, синтаксис или действие которых определяется реализацией (приложение 2, п. 2); но и при определении синтаксиса или правил для таких элементов не разрешается замена элементов или включение дополнительных элементов в исходную программу.

GOST EXPERT ГОСТ 22558-89,

1.5.2.2. *Соглашения о стандартных элементах языка*

Реализация должна поддерживать синтаксис и обеспечивать функции всех элементов стандарта языка Кобол, как это определено данным уровнем модуля, включаемым в реализацию, за исключением элементов языка, зависящих от специфических компонент оборудования, которые приведены в п. 1.5.2.5.1 настоящей части. Если реализация поддерживает синтаксис элементов стандарта языка Кобол для данного уровня модуля, отличного от того, поддержка которого объявлена, реализация должна обеспечить функции, определенные стандартом языка Кобол для данного синтаксиса, или идентифицировать эти элементы языка как нестандартные расширения (п. 1.5.2.5.2 настоящей части).

1.5.2.3. *Устаревшие элементы языка*

Устаревшие элементы языка — элементы в стандарте языка Кобол, которые будут изъяты при следующем пересмотре стандарта (приложение 2, п. 1). Устаревшие элементы языка не были ни усилены, ни модифицированы в процессе пересмотра. Взаимодействие между устаревшими и остальными элементами языка не определено, если не оговорено специально в стандарте. Элементы языка, которые предполагается изъять из стандарта, будут сначала идентифицированы как устаревшие элементы языка.

От реализации стандарта Кобола требуется поддержка устаревших элементов в подмножестве и уровнях необязательных модулей, поддержка которых объявлена. Документация, связанная с реализацией, должна идентифицировать все устаревшие элементы языка в данной реализации.

Реализация, соответствующая стандарту языка Кобол, должна предусмотреть средство, которое по желанию программиста может быть подключено к компилятору для определения, содержит ли программа устаревшие элементы языка.

1.5.2.4. *Действия, обеспечиваемые вне исходной программы*

Если любая функция, которая выполняет функции элемента стандарта Кобола, содержащегося в данном уровне модуля, который объявлен включенным в реализацию, обеспечивается вне исходной программы, реализация не должна требовать спецификации внешних функций вместо или в дополнение к элементу стандарта языка.

Реализация может требовать спецификации вне исходной программы интерфейса с операционной средой для поддержки функций, определенных в исходной программе.

1.5.2.5. *Уточнения*

К реализации спецификаций стандарта Кобола применяются приведенные ниже уточнения.

1.5.2.5.1. Элементы языка, зависящие от оборудования

GOST EXPERT ГОСТ 22558-89,

Ряд элементов языка рассчитан на определенный тип технических средств (приложение 2, п. 3). Для того, чтобы реализация удовлетворяла требованиям этого стандарта, разработчик должен указать технические средства, которые поддерживает реализация. Более того, когда объявлена поддержка специфических технических средств, должны быть реализованы все элементы языка, которые зависят от этих средств, если модуль, в котором они появляются, включен в реализацию. Элементы языка, имеющие отношение к специфическим компонентам оборудования, для которых не объявлена поддержка реализации, нет необходимости реализовать. Отсутствие таких элементов должно быть указано реализацией стандарта языка Кобол.

1.5.2.5.2. Расширение элементов языка

Реализация, включающая элементы языка в дополнение к подмножеству и уровням необязательных модулей, для которых объявлена поддержка, удовлетворяет требованиям стандарта языка Кобол. Это истинно, хотя и может означать расширение списка зарезервированных слов данной реализацией, и, таким образом, может помешать правильной компиляции некоторых программ, которые удовлетворяют требованиям стандарта языка Кобол.

Реализация должна идентифицировать каждое стандартное расширение (элементы языка, не указанные в поддерживаемом подмножестве или поддерживаемых уровнях необязательных модулей, но определенные в другом месте стандарта языка Кобол) или нестандартные расширения (элементы языка или действия, не определенные в стандарте языка Кобол) в соответствующей ей документации.

Реализация, соответствующая стандарту языка Кобол, должна предусмотреть средство, которое по желанию программиста может быть подключено к компилятору для определения, содержит ли данная программа нестандартные расширения, включенные в реализацию.

1.5.2.5.3. Зарезервированные слова

Реализация стандарта языка Кобол должна распознавать в качестве зарезервированных слов все зарезервированные слова Кобола, встречающиеся в спецификациях семи обязательных модулей и четырех необязательных модулей (ч. 4, п. 8).

1.5.2.5.4. Замена литер

Определение набора литер Кобола (ч. 3) представляет полный набор литер стандарта языка Кобол. Когда реализация не обеспечивает графическое представление для всего набора литер Кобола, реализацией может быть указана заменяющая графика для непредставимых литер.

1.5.2.5.5. Оператор ENTER (ВОЙТИ)

По усмотрению разработчика, реализация стандарта языка Кобол может включать или не включать оператор ENTER (ВОЙТИ).

1.6. **Соответствие исходной программы стандарту**

Соответствующая стандарту исходная программа — это программа, не нарушающая явно установленных условий и спецификаций стандарта языка Кобол. Чтобы исходная программа соответствовала стандарту языка Кобол, она не должна включать никаких элементов языка, не определенных в этом стандарте. Выполнение программы, исходный текст которой соответствует стандарту языка Кобол, предсказуемо только в пределах, определенных в этом стандарте. Результаты нарушения форматов или правил стандарта языка Кобол не определены, если противное не оговорено в стандарте.

Для того, чтобы исходная программа соответствовала определенному подмножеству стандарта языка Кобол, она должна включать в себя только элементы языка этого подмножества.

В стандарте языка Кобол имеется ряд ситуаций, в которых результат выполнения оператора не определен или непредсказуем (приложение 2, п. 4). Исходная Кобол-программа, допускающая такие ситуации, не соответствует стандарту, поскольку результат выполнения не определен стандартом языка Кобол.

1.7. **Сочетание соответствующей стандарту программы и соответствующей стандарту реализации**

Компиляция соответствующей стандарту исходной программы соответствующей стандарту реализацией и последующее выполнение результирующей объектной программы определены только в пределах, указанных стандартом. Однако это еще не значит, что программа будет скомпилирована или выполнена успешно, так как это зависит от других факторов, таких как использование элементов языка, определяемых реализацией, логическая правильность программы и данных, которыми оперирует программа.

В общем случае, стандарт языка Кобол не указывает верхние количественные ограничения числа операторов в программе и числа операндов в некоторых операторах. Такие ограничения могут изменяться от реализации к реализации.

2. СПИСОК ЭЛЕМЕНТОВ ПО МОДУЛЯМ

2.1. **Общее описание**

Ниже содержится список всех элементов стандарта Кобола, составленный в соответствии с функциональными обрабатывающими модулями.

Столбец, озаглавленный «Уровень 1», определяет уровень 1 элементов модуля. Столбец, озаглавленный «Уровень 2», определяет уровень 2 элементов модуля.

GOST EXPERT ГОСТ 22558-89,

Символ «×» в столбце означает наличие указанного элемента в указанном уровне модуля.

Символ «—» в столбце означает отсутствие указанного элемента в указанном уровне модуля.

Символ «+» в столбце означает наличие указанного элемента в указанном уровне модуля; однако этот элемент является устаревшим элементом в стандарте Кобола, поэтому он будет удален в следующей редакции стандарта.

2.2. **Список элементов в модуле ядра**

Элемент	Уровень 1	Уровень 2
ПОНЯТИЯ ЯЗЫКА		
Набор литер		
Литеры, используемые для слов в английской нотации 0—9, A—Z, - (дефис)	×	×
в русской нотации А—Я, D, F, G, I, J, L, N, Q, R, S, U, V, W, Y, Z, 0—9, - (дефис)	×	×
Литеры, используемые для пунктуации " ( ) . , ; пробел	×	×
Литеры, используемые для пунктуации : (двоеточие)	—	×
Литеры, используемые для редактирования B + — . , Z(П) * $(¤) 0 CR(КР) DB(ДБ) /	×	×
Литеры, используемые в арифметических операциях + — * / **	—	×
Литеры, используемые в отношениях = >= <= > <	×	×
Литеры, используемые при индексировании + —	×	×
Разрешена замена одной литерой	×	×
Разрешена замена парами литер	+	+
Разделители		
» ( ) . , ; пробел	×	×
: (двоеточие)	—	×
Строки-литер		
Слова Кобола		
Не более 30 литер	×	×
Слова, определенные пользователем		
имя-алфавита	×	×
имя-класса	×	×
имя-условия	—	×
имя-данного	×	×
имя-индекса	×	×
номер-уровня	×	×
мнемоническое-имя	×	×
имя-параграфа	×	×
имя-программы	×	+
имя-программного-модуля	+	×
имя-секции	×	×

GOST EXPERT ГОСТ 22558-89,

Элемент	Уровень 1	Уровень 2
символическая-литера	—	X
Системные-имена		
имя-машины	X	X
имя-реализации	X	X
имя-языка	+	+
Зарезервированные слова		
Обязательные слова	X	X
Ключевые слова	X	X
Слова специальные литеры		
Знаки арифметических операций + — * / **	—	X
Знаки арифметических операций при индексировании + —	X	X
Литеры отношения = > < >= <=	X	X
Необязательные слова	X	X
Слова специального назначения		
Стандартные константы: ZERO (НУЛЬ), ZEROS, ZEROES (НУЛИ), SPACE (ПРОБЕЛ), SPACES (ПРОБЕЛЫ), HIGH-VALUE (НАИБОЛЬШЕЕ-ЗНАЧЕНИЕ), HIGH-VALUES (НАИБОЛЬШИЕ-ЗНАЧЕНИЯ), LOW-VALUE (НАИМЕНЬШЕЕ-ЗНАЧЕНИЕ), LOW-VALUES (НАИМЕНЬШИЕ-ЗНАЧЕНИЯ), QUOTE (КАВЫЧКА), QUOTES (КАВЫЧКИ)	X	X
Стандартные константы: символическая-литера, ALL литерал (ВСЕ литерал), ALL стандартная-константа (ВСЕ стандартная-константа), ALL символическая-литера (ВСЕ символическая-литера)	—	X
Литералы		
Числовые литералы: от 1 до 18 цифр	X	X
Нечисловые литералы: от 1 до 160 литер	X	X
PICTURE строка-литер (ШАБЛОН строка-литер)	X	X
Статья-комментарий	+	+
Однозначность ссылки		
Уточнение		
Уточнение недопустимо, имена должны быть однозначны при ссылке	X	—
50 уточнителей	—	X
Индексирование		
3 уровня индексов	X	—
7 уровней индексов	—	X
Индексирование литералом	X	X
Индексирование именем-данного	X	X
Индексирование именем индекса	X	X
Относительное индексирование	X	X
Модификация ссылки	—	X
Формат представления		
Порядковый номер	X	X

GOST EXPERT ГОСТ 22558-89,

Элемент	Уровень 1	Уровень 2
Продолжение строк		
Нечисловой литерал	X	X
Слова Кобола, числовой литерал, строка-литер шаблона	—	X
Строки пробелов (пустые строки)	X	X
Строки комментария		
Строки комментария со звездочкой (*)	X	X
Строки комментария с дробной чертой (/)	X	X
Отладочная строка с литерой D (T) в поле индикатора	X	X
Структура исходной программы		
Раздел идентификации обязателен	X	X
Раздел оборудования необязателен	X	X
Раздел данных необязателен	X	X
Раздел процедур необязателен	X	X
Заголовок конца программы	—	X
РАЗДЕЛ ИДЕНТИФИКАЦИИ		
Параграф PROGRAM-ID (ПРОГРАММА)	X	X
имя-программы	X	X
Параграф AUTHOR (АВТОР)	+	+
Параграф INSTALLATION (ПРЕДПРИЯТИЕ)	+	+
Параграф DATE-WRITTEN (ДАТА-НАПИСАНИЯ)	+	+
Параграф DATE-COMPILED (ДАТА-ТРАНСЛЯЦИИ)	—	+
Параграф SECURITY (ПОЛНОМОЧИЯ)	+	+
РАЗДЕЛ ОБОРУДОВАНИЯ		
Секция конфигурации		
Параграф SOURCE-COMPUTER (ИСХОДНАЯ-МАШИНА)	X	X
имя-машины	X	X
фраза WITH DEBUGGING MODE (В РЕЖИМЕ ОТЛАДКИ)	X	X
Параграф OBJECT-COMPUTER (РАБОЧАЯ-МАШИНА)	X	X
имя-машины	X	X
фраза MEMORY SIZE (РАЗМЕР ПАМЯТИ)	+	+
фраза PROGRAM COLLATING SEQUENCE (ПРОГРАММНЫЙ АЛФАВИТ)	X	X
Параграф SPECIAL-NAMES (СПЕЦИАЛЬНЫЕ-ИМЕНА)	X	X
Фраза ALPHABET (АЛФАВИТ)	X	X
вариант STANDARD-1 (СТАНДАРТ-А)	X	X
вариант STANDARD-2 (СТАНДАРТ-М)	X	X
вариант NATIVE (ВНУТРЕННИЙ)	X	X
вариант имя-реализации	X	X
вариант литерал	—	X
Фраза CLASS (КЛАСС)	X	X
Фраза CURRENCY SIGN (ВАЛЮТНЫЙ ЗНАК)	X	X
Фраза DECIMAL-POINT (ДЕСЯТИЧНАЯ ТОЧКА)	X	X

GOST EXPERT ГОСТ 22558-89,

Элемент	Уровень 1	Уровень 2
Фраза имя-реализации	X	X
вариант IS мнемоническое-имя	X	X
вариант ON STATUS IS имя-условия (ВКЛЮЧЕНО имя-условия)	X	X
вариант OFF STATUS IS имя-условия (ВЫКЛЮЧЕНО имя-условия)	X	X
Фраза SYMBOLIC CHARACTER (СИМВОЛИЧЕСКАЯ ЛИТЕРА)	—	X
РАЗДЕЛ ДАННЫХ		
Секция рабочей памяти		
Статья-описания записи	X	X
Статья описания уровня 77	X	X
Статья описания данного		
Фраза BLANK WHEN ZERO (ПРОБЕЛ КОГДА НУЛЬ)	X	X
Фраза имя-данного	X	X
Фраза FILLER (ЗАПОЛНИТЕЛЬ)	X	X
Фраза JUSTIFIED (СДВИНУТО)	X	X
Фраза номер-уровня	X	X
от 01 до 49; одна или две цифры	X	X
66	—	X
77	X	X
88	—	X
Фраза OCCURS (ПОВТОРЯЕТСЯ)	X	X
целое TIMES (целое РАЗ)	X	X
фраза ASCENDING/DESCENDING KEY (ПО ВОЗРАСТАНИЮ/УБЫВАНИЮ КЛЮЧА)	—	X
фраза INDEXED BY (ИНДЕКСИРУЕТСЯ)	X	X
фраза целое-1 TO целое-2 TIMES DEPENDING ON (целое-1 ДО целое-2 РАЗ В ЗАВИСИМОСТИ ОТ)	—	X
Фраза PICTURE (ШАБЛОН)	X	X
строка-литер содержит не более 30 литер	X	X
литеры данных: X 9 A	X	X
операционные символы: S(3) V(T) P(M)	X	X
литеры фиксированной вставки B + — . 0 $ (¤) CR (КР) DB (ДБ) /	X	X
литеры замещения или плавающей вставки * + — Z(П) $ (¤)	X	X
замена валютного знака	X	X
замена десятичной точки	X	X
Фраза REDEFINES (ПЕРЕОПРЕДЕЛЯЕТ)	X	X
не может быть вложенной	X	—
может быть вложенной	—	X
Фраза RENAMES (ПЕРЕИМЕНОВЫВАЕТ)	—	X
Фраза SIGN (ЗНАК)	X	X

GOST EXPERT ГОСТ 22558-89,

Элемент	Уровень 1	Уровень 2
Фраза SYNCHRONIZED (ВЫДЕЛЕНО)	X	X
Фраза USAGE (об использовании)	X	X
BINARY (ДВОИЧНОЕ)	X	X
COMPUTATIONAL (ДЛЯ ВЫЧИСЛЕНИЙ)	X	X
DISPLAY (ДЛЯ ВЫДАЧИ)	X	X
INDEX (ДЛЯ ИНДЕКСА)	X	X
PACKED-DECIMAL (ДЕСЯТИЧНОЕ)	X	X
Фраза VALUE (ЗНАЧЕНИЕ)	X	X
литерал	X	X
несколько литералов	—	X
литерал-1 THROUGH литерал-2 (литерал-1 ПО литерал-2)	—	X
несколько диапазонов литералов	—	X
РАЗДЕЛ ПРОЦЕДУР		
Арифметическое выражение	—	X
Знаки бинарных арифметических операций + — * / **	—	X
Знаки унарных арифметических операций + —	—	X
Условные выражения	X	X
Простое условие	X	X
Условие отношения	X	X
Знаки операций отношения		
[NOT] GREATER THAN ([НЕ] БОЛЬШЕ)	X	X
[NOT] > ([НЕ] >)	X	X
[NOT] LESS THAN ([НЕ] МЕНЬШЕ)	X	X
[NOT] < ([НЕ] <)	X	X
[NOT] EQUAL TO ([НЕ] РАВНО)	X	X
[NOT] = ([НЕ] =)	X	X
GREATER THAN OR EQUAL TO (БОЛЬШЕ ИЛИ РАВНО)	X	X
>=	X	X
LESS THAN OR EQUAL TO (МЕНЬШЕ ИЛИ РАВНО)	X	X
<=	X	X
Сравнение числовых операндов	X	X
Сравнение нечисловых операндов	X	X
Сравнение имен индексов и (или) индексных данных	X	X
Условие класса	X	X
NUMERIC (ЧИСЛОВОЕ)	X	X
ALPHABETIC (БУКВЕННОЕ)	X	X
ALPHABETIC-LOWER (СТРОЧНЫЕ)	X	X
ALPHABETIC-UPPER (ПРОПИСНЫЕ)	X	X
имя-класса	X	X
Условие имени-условия	—	X
Условие знака	—	X
Условие состояния переключателя	X	X
Сложное условие	—	X
Знаки логических операций AND (И) OR (ИЛИ) NOT (НЕ)	—	X
Отрицание условия	—	X

GOST EXPERT ГОСТ 22558-89,

Элемент	Уровень 1	Уровень 2
Комбинированное условие	—	X
Условие в скобках	X	X
Сокращенные комбинированные условия отношений	—	X
Арифметические операторы	X	X
Арифметические операнды имеют длину до 18 цифр	X	X
Композиция операндов содержит не более 18 цифр	X	X
Оператор ACCEPT (ПРИНЯТЬ)	X	X
идентификатор	X	X
только одна передача данных	X	—
число передач данных не ограничивается	—	X
фраза FROM мнемоническое-имя (С мнемоническое-имя)	—	X
фраза FROM DATE/DAY/DAY-OF-WEEK/TIME (ДАТУ/ДЕНЬ/ДЕНЬ/-НЕДЕЛИ/ВРЕМЯ)	—	X
Оператор ADD (СЛОЖИТЬ)	X	X
идентификатор/литерал	X	X
несколько идентификаторов/литералов	X	X
ТО идентификатор (С идентификатор)	X	X
ТО несколько идентификаторов (С несколько идентификаторов)	X	X
ТО идентификатор/литерал GIVING идентификатор (С идентификатор/литерал ПОЛУЧАЯ идентификатор	X	X
ТО идентификатор/литерал GIVING несколько идентификаторов (С идентификатор/литерал ПОЛУЧАЯ несколько идентификаторов	X	X
фраза ROUNDED (ОКРУГЛЯЯ)	X	X
фраза ON SIZE ERROR (ПРИ ПЕРЕПОЛНЕНИИ)	X	X
фраза NOT ON SIZE ERROR (БЕЗ ПЕРЕПОЛНЕНИЯ)	X	X
фраза END-ADD (КОНЕЦ-СЛОЖИТЬ)	X	X
фраза CORRESPONDING (СООТВЕТСТВЕННО)	—	X
Оператор ALTER (ИЗМЕНИТЬ)	+	+
только одно имя-процедуры	+	—
несколько имен-процедур	—	+
Оператор COMPUTE (ВЫЧИСЛИТЬ)	—	X
арифметическое выражение	—	X
несколько идентификаторов	—	X
фраза ROUNDED (ОКРУГЛЯЯ)	—	X
фраза ON SIZE ERROR (ПРИ ПЕРЕПОЛНЕНИИ)	—	X
фраза NOT ON SIZE ERROR (БЕЗ ПЕРЕПОЛНЕНИЯ)	—	X
фраза END-COMPUTE (КОНЕЦ-ВЫЧИСЛИТЬ)	—	X
Оператор CONTINUE (ПРОДОЛЖИТЬ)	X	X
Оператор DISPLAY (ВЫДАТЬ)	X	X
только одна передача данных	X	—

GOST EXPERT ГОСТ 22558-89,

Элемент	Уровень 1	Уровень 2
число передач данных не ограничено	—	X
идентификатор/литерал	X	X
несколько идентификаторов/литералов	X	X
фраза UPON мнемоническое-имя (НА мнемоническое-имя)	—	X
фраза WITH NO ADVANCING (БЕЗ ПРОДВИЖЕНИЯ)	—	X
Оператор DIVIDE (РАЗДЕЛИТЬ)	—	X
BY идентификатор/литерал (НА идентификатор/литерал)	X	X
INTO идентификатор/литерал (НА идентификатор/литерал)	X	X
INTO несколько делимых (НА несколько делимых)	X	X
GIVING идентификатор (ПОЛУЧАЯ идентификатор)	X	X
GIVING несколько идентификаторов (ПОЛУЧАЯ несколько идентификаторов)	X	X
фраза ROUNDED (ОКРУГЛЯЯ)	X	X
фраза REMAINDER (ОСТАТОК)	—	X
фраза ON SIZE ERROR (ПРИ ПЕРЕПОЛНЕНИИ)	X	X
фраза NOT ON SIZE ERROR (БЕЗ ПЕРЕПОЛНЕНИЯ)	X	X
фраза END-DIVIDE (КОНЕЦ-РАЗДЕЛИТЬ)	X	X
Оператор ENTER (ВОЙТИ)	+	+
Оператор EVALUATE (ОЦЕНИТЬ)	—	X
идентификатор/литерал	—	X
арифметическое выражение	—	X
условное выражение	—	X
TRUE/FALSE (ИСТИНА/ЛОЖЬ)	—	X
фраза ALSO (ТАКЖЕ)	—	X
фраза WHEN (КОГДА)	—	X
фраза ALSO (ТАКЖЕ)	—	X
фраза WHEN OTHER (ИНАЧЕ)	—	X
фраза END-EVALUATE (КОНЕЦ-ОЦЕНИТЬ)	—	X
Оператор EXIT (ВЫЙТИ)	X	X
Оператор GO TO (ПЕРЕЙТИ К)	X	X
имя-процедуры обязательно	X	—
имя-процедуры необязательно	—	+
фраза DEPENDING ON (В ЗАВИСИМОСТИ ОТ)	X	X
Оператор IF (ЕСЛИ)	X	X
только повелительные операторы	X	—
повелительные и(или) условные операторы	—	X
вложенные операторы IF (ЕСЛИ)	X	X
необязательное слово THEN (ТО)	X	X
фраза NEXT SENTENCE (СЛЕДУЮЩЕЕ ПРЕДЛОЖЕНИЕ)	X	X
фраза ELSE (ИНАЧЕ)	X	X
фраза END-IF (КОНЕЦ-ЕСЛИ)	X	X

GOST EXPERT ГОСТ 22558-89,

Элемент	Уровень 1	Уровень 2
Оператор INITIALIZE (ИНИЦИИРОВАТЬ)	—	X
несколько идентификаторов	—	X
фраза REPLACING (ЗАМЕНЯЯ)	—	X
несколько REPLACING (ЗАМЕНЯЯ)	—	X
Оператор INSPECT (ПРОСМОТРЕТЬ)	X	X
на вхождение одной литеры	X	—
на вхождение нескольких литер	—	X
фраза TALLYING (СЧИТАЯ)	X	X
фраза BEFORE/AFTER (ДО/ПОСЛЕ)	X	X
несколько фраз BEFORE/AFTER (ДО/ПОСЛЕ)	—	X
несколько фраз TALLYING (СЧИТАЯ)	—	X
фраза REPLACING (ЗАМЕНЯЯ)	X	X
фраза BEFORE/AFTER (ДО/ПОСЛЕ)	X	X
несколько фраз BEFORE/AFTER (ДО/ПОСЛЕ)	—	X
несколько фраз REPLACING (ЗАМЕНЯЯ)	—	X
фразы TALLYING (СЧИТАЯ) и REPLACING (ЗАМЕНЯЯ)	X	X
фраза CONVERTING (ПРЕВРАЩАЯ)	—	X
Оператор MOVE (ПОМЕСТИТЬ)	X	X
ТО идентификатор (В идентификатор)	X	X
ТО несколько идентификаторов (В несколько идентификаторов)	X	X
дередактирование цифровых редактируемых данных	—	X
фраза CORRESPONDING (СООТВЕТСТВЕННО)	—	X
Оператор MULTIPLY (УМНОЖИТЬ)	X	X
BY идентификатор (НА идентификатор)	X	X
BY несколько идентификаторов (НА несколько идентификаторов)	X	X
GIVING идентификатор (ПОЛУЧАЯ идентификатор)	X	X
GIVING несколько идентификаторов (ПОЛУЧАЯ несколько идентификаторов)	X	X
фраза ROUNDED (ОКРУГЛЯЯ)	X	X
фраза ON SIZE ERROR (ПРИ ПЕРЕПОЛНЕНИИ)	X	X
фраза NOT ON SIZE ERROR (БЕЗ ПЕРЕПОЛНЕНИЯ)	X	X
фраза END-MULTIPLY (КОНЕЦ-УМНОЖИТЬ)	X	X
Оператор PERFORM (ВЫПОЛНИТЬ)	X	X
имя-процедуры необязательно	X	X
фраза THROUGH имя-процедуры (ПО имя-процедуры)	X	X
вариант повелительный-оператор	X	X
фраза END-PERFORM (КОНЕЦ-ВЫПОЛНИТЬ)	X	X
фраза TIMES (РАЗ)	X	X
фраза UNTIL (ДО)	X	X
фраза TEST BEFORE/AFTER (С ПРОВЕРКОЙ В НАЧАЛЕ/В КОНЦЕ)	—	X

GOST EXPERT ГОСТ 22558-89,

Элемент	Уровень 1	Уровень 2
фраза VARYING (МЕНЯЯ)	—	X
фраза TEST BEFORE/AFTER (С ПРОВЕРКОЙ В НАЧАЛЕ/В КОНЦЕ)	—	X
фраза AFTER (ЗАТЕМ)	—	X
допускаются по крайней мере 6 фраз AFTER (ЗАТЕМ)	—	X
Оператор SEARCH (ИСКАТЬ)	—	X
фраза VARYING (МЕНЯЯ)	—	X
фраза AT END (В КОНЦЕ)	—	X
фраза WHEN (КОГДА)	—	X
несколько фраз WHEN (КОГДА)	—	X
фраза END-SEARCH (КОНЕЦ-ИСКАТЬ)	—	X
Оператор SEARCH ALL (ИСКАТЬ ОСОБО)	—	X
фраза AT END (В КОНЦЕ)	—	X
фраза WHEN (КОГДА)	—	X
фраза END-SEARCH (КОНЕЦ-ИСКАТЬ)	—	X
Оператор SET (УСТАНОВИТЬ)	X	X
имя-индекса/идентификатор TO (НА)	X	X
имя-индекса UP BY/DOWN BY (имя-индекса ПРИБАВЛЯЯ/ВЫЧИТАЯ)	X	X
мнемоническое-имя TO ON/OFF (мнемоническое-имя НА ВКЛЮЧЕНО/ВЫКЛЮЧЕНО)	X	X
имя-условия TO TRUE (имя-условия НА ИСТИНА)	—	X
Оператор STOP (ОСТАНОВИТЬ)	X	X
RUN (РАБОТУ)	X	X
литерал	+	+
Оператор STRING (СОБРАТЬ)	—	X
несколько DELIMITED BY (ОГРАНИЧИВАЯСЬ)	—	X
фраза WITH POINTER (УКАЗАТЕЛЬ)	—	X
фраза ON OVERFLOW (ПРИ ПЕРЕПОЛНЕНИИ)	—	X
фраза NOT ON OVERFLOW (БЕЗ ПЕРЕПОЛНЕНИЯ)	—	X
фраза END-STRING (КОНЕЦ-СОБРАТЬ)	—	X
Оператор SUBTRACT (ОТНЯТЬ)	X	X
идентификатор/литерал	X	X
несколько идентификаторов/литералов	X	X
FROM идентификатор (ОТ идентификатор)	X	X
FROM несколько идентификаторов (ОТ несколько идентификаторов)	X	X
GIVING идентификатор (ПОЛУЧАЯ идентификатор)	X	X
GIVING несколько идентификаторов (ПОЛУЧАЯ несколько идентификаторов)	X	X
фраза ROUNDED (ОКРУГЛЯЯ)	X	X
фраза ON SIZE ERROR (ПРИ ПЕРЕПОЛНЕНИИ)	X	X
фраза NOT ON SIZE ERROR (БЕЗ ПЕРЕПОЛНЕНИЯ)	X	X
фраза END-SUBTRACT (КОНЕЦ-ОТНЯТЬ)	X	X

GOST EXPERT ГОСТ 22558-89,

Элемент	Уровень 1	Уровень 2
фраза CORRESPONDING (СООТВЕТСТВЕННО)	—	X
Оператор UNSTRING (РАЗОБРАТЬ)	—	X
фраза DELIMITED BY (ОГРАНИЧИВАЯСЬ)	—	X
фраза DELIMITER IN (ОГРАНИЧИТЕЛЬ В)	—	X
фраза COUNT IN (СЧЕТ В)	—	X
фраза WITH POINTER (УКАЗАТЕЛЬ)	—	X
фраза TALLYING (СЧИТАЯ В)	—	X
фраза ON OVERFLOW (ПРИ ПЕРЕПОЛНЕНИИ)	—	X
фраза NOT ON OVERFLOW (БЕЗ ПЕРЕПОЛНЕНИЯ)	—	X
фраза END-UNSTRING (КОНЕЦ-РАЗОБРАТЬ)	—	X

2.3. Список элементов в модуле последовательного ввода-вывода

Элемент	Уровень 1	Уровень 2
ПОНЯТИЯ ЯЗЫКА		
Слова, определенные пользователем		
имя-файла	X	X
имя-записи	X	X
Зарезервированные слова		
Специальный регистр LINAGE-COUNTER (СЧЕТЧИК-ВЕРСТКИ)	—	X
Состояние ввода-вывода	X	X
РАЗДЕЛ ОБОРУДОВАНИЯ		
Секция ввода-вывода		
Параграф FILE-CONTROL (УПРАВЛЕНИЕ-ФАЙЛАМИ)	X	X
Статья управления файлом	X	X
фраза SELECT (ДЛЯ)	X	X
фраза OPTIONAL (НЕОБЯЗАТЕЛЬНОГО)	—	X
только входной, входной-выходной и дополняемый	—	X
фраза ACCESS MODE IS SEQUENTIAL (ДОСТУП ПОСЛЕДОВАТЕЛЬНЫЙ)	X	X
фраза ASSIGN (НАЗНАЧИТЬ)	X	X
имя-реализации	X	X
литерал	X	X
фраза FILE STATUS (СОСТОЯНИЕ ФАЙЛА)	X	X
фраза ORGANIZATION IS SEQUENTIAL (ОРГАНИЗАЦИЯ ПОСЛЕДОВАТЕЛЬНАЯ)	X	X
фраза PADDING CHARACTER (ЛИТЕРА ЗАПОЛНИТЕЛЬ)	—	X

GOST EXPERT ГОСТ 22558-89,

Элемент	Уровень 1	Уровень 2
фраза RECORD DELIMITER (ОГРАНИЧИТЕЛЬ ЗАПИСИ)	—	X
фраза RESERVE (РЕЗЕРВИРОВАТЬ)	—	X
Параграф I-O-CONTROL (УПРАВЛЕНИЕ-ВВОДОМ-ВЫВОДОМ)	X	X
фраза MULTIPLE FILE TAPE (НА ОДНОЙ КАТУШКЕ)	—	+
фраза RERUN (ПЕРЕПРОГОН)	+	+
фраза SAME AREA (ОБЩАЯ ОБЛАСТЬ)	X	X
фраза SAME RECORD AREA (ОБЩАЯ ОБЛАСТЬ ЗАПИСИ)	—	X
РАЗДЕЛ ДАННЫХ		
Секция файлов		
Статья описания файла	X	X
индикатор уровня FD (ОФ)	X	X
фраза BLOCK CONTAINS (В БЛОКЕ)	X	X
целое-1 RECORDS/CHARACTERS (целое-1 ЗАПИСЕЙ/ЛИТЕР)	X	X
целое-1 TO целое-2 RECORDS/CHARACTERS (целое-1 ДО целое-2 ЗАПИСЕЙ/ЛИТЕР)	—	X
фраза CODE-SET (АЛФАВИТ)	X	X
фраза DATA RECORDS (ЗАПИСИ ДАННЫХ)	+	+
фраза LABEL RECORDS (МЕТКИ)	+	+
фраза LINAGE (ВЕРСТКА)	—	X
фраза FOOTING (КОНЦОВКА)	—	X
фраза TOP (ВЕРХНЕЕ ПОЛЕ)	—	X
фраза BOTTOM (НИЖНЕЕ ПОЛЕ)	—	X
фраза RECORD (В ЗАПИСИ)	X	X
целое-1 CHARACTERS (целое-1 ЛИТЕР)	X	X
фраза VARYING IN SIZE (ПЕРЕМЕННОЕ ЧИСЛО)	—	X
FROM целое-2 TO целое-3 CHARACTERS (ОТ целое-2 ДО целое-3 ЛИТЕР)	—	X
фраза DEPENDING ON (В ЗАВИСИМОСТИ ОТ)	—	X
целое-4 TO целое-5 CHARACTERS (целое-4 ДО целое-5 ЛИТЕР)	X	X
фраза VALUE (ЗНАЧЕНИЕ)	+	+
имя-реализации литерал	+	+
имя-реализации несколько литералов	+	+
имя-реализации имя-данного	—	+
имя-реализации несколько имен-данных	—	+
Статья описания записи	X	X
РАЗДЕЛ ПРОЦЕДУР		
Декларативные процедуры	X	X
DECLARATIVES (ДЕКЛАРАТИВЫ)	X	X
END DECLARATIVES (КОНЕЦ ДЕКЛАРАТИВ)	X	X
Оператор CLOSE (ЗАКРЫТЬ)	X	X

GOST EXPERT ГОСТ 22558-89,

Элемент	Уровень 1	Уровень 2
имя-файла	X	X
несколько имен-файлов	X	X
фраза REEL/UNIT (КАТУШКУ/ТОМ)	X	X
фраза FOR REMOVAL (С УДАЛЕНИЕМ)	—	X
фраза WITH NO REWIND/LOCK (БЕЗ ПЕРЕМОТКИ/С ЗАМКОМ)	—	X
Оператор OPEN (ОТКРЫТЬ)	X	X
имя-файла	X	X
несколько имен-файлов	X	X
фраза INPUT (ВХОДНОЙ)	X	X
фраза WITH NO REWIND (БЕЗ ПЕРЕМОТКИ)	—	X
фраза REVERSED (РЕВЕРСНО)	—	+
фраза OUTPUT (ВЫХОДНОЙ)	X	X
фраза WITH NO REWIND (БЕЗ ПЕРЕМОТКИ)	—	X
фраза I-O (ВХОДНОЙ-ВЫХОДНОЙ)	X	X
фраза EXTEND (ДОПОЛНЯЕМЫЙ)	—	X
несколько INPUT (ВХОДНОЙ), OUTPUT (ВЫХОДНОЙ) и I-O (ВХОДНОЙ-ВЫХОДНОЙ)	X	X
несколько EXTEND (ДОПОЛНЯЕМЫЙ)	—	X
Оператор READ (ЧИТАТЬ)	X	X
фраза NEXT (СЛЕДУЮЩУЮ)	—	X
фраза INTO (В)	X	X
фраза AT END (В КОНЦЕ)	X	X
фраза NOT AT END (НЕ В КОНЦЕ)	X	X
фраза END-READ (КОНЕЦ-ЧИТАТЬ)	X	X
Оператор REWRITE (ОБНОВИТЬ)	X	X
фраза FROM (ИЗ ПОЛЯ)	X	X
Оператор USE (ИСПОЛЬЗОВАТЬ)		
фраза EXCEPTION/ERROR PROCEDURE (ПРОЦЕДУРЫ ОШИБКИ)	X	X
ON имя-файла (ДЛЯ имя-файла)	X	X
ON несколько-имен-файлов (ДЛЯ несколько имен-файлов	—	X
ON INPUT (ДЛЯ ВХОДНЫХ)	X	X
ON OUTPUT (ДЛЯ ВЫХОДНЫХ)	X	X
ON I-O (ДЛЯ ВХОДНЫХ-ВЫХОДНЫХ)	X	X
ON EXTEND (ДЛЯ ДОПОЛНЯЕМЫХ)	—	X
Оператор WRITE (ПИСАТЬ)	X	X
фраза FROM (ИЗ ПОЛЯ)	X	X
фраза BEFORE/AFTER ADVANCING (ДО/ПОСЛЕ ПРОДВИЖЕНИЯ)	X	X
целое LINE/LINES (целое СТРОК)	X	X
идентификатор LINE/LINES (идентификатор СТРОК)	X	X
мнемоническое-имя	—	X
PAGE (СТРАНИЦЫ)	X	X
фраза AT END-OF-PAGE (В КОНЦЕ СТРАНИЦЫ)	—	X
фраза NOT AT END-OF-PAGE (НЕ В КОНЦЕ СТРАНИЦЫ)	—	X
фраза END-WRITE (КОНЕЦ-ПИСАТЬ)	—	X

GOST EXPERT ГОСТ 22558-89,

2.4. Список элементов в модуле относительного ввода-вывода

Элемент	Уровень 1	Уровень 2
ПОНЯТИЯ ЯЗЫКА		
Слова, определенные пользователем		
имя-файла	X	X
имя-записи	X	X
Состояние ввода-вывода	X	X
РАЗДЕЛ ОБОРУДОВАНИЯ		
Секция ввода-вывода		
Параграф FILE-CONTROL (УПРАВЛЕНИЕ-ФАЙЛАМИ)	X	X
Статья управления файлом	X	X
фраза SELECT (ДЛЯ)	X	X
фраза OPTIONAL (НЕОБЯЗАТЕЛЬНОГО)	—	X
только входной, входной-выходной и дополняемый файлы	—	X
фраза ACCESS MODE (ДОСТУП)	X	X
SEQUENTIAL (ПОСЛЕДОВАТЕЛЬНЫЙ)	X	X
RANDOM (ПРОИЗВОЛЬНЫЙ)	X	X
DYNAMIC (ДИНАМИЧЕСКИЙ)	X	X
RELATIVE KEY (ОТНОСИТЕЛЬНЫЙ КЛЮЧ)	X	X
фраза ASSIGN (НАЗНАЧИТЬ)	X	X
имя-реализации	X	X
литерал	X	X
фраза FILE STATUS (СОСТОЯНИЕ ФАЙЛА)	X	X
фраза ORGANIZATION RELATIVE (ОРГАНИЗАЦИЯ ОТНОСИТЕЛЬНАЯ)	X	X
фраза RESERVE (РЕЗЕРВИРОВАТЬ)	—	X
Параграф I-O-CONTROL (УПРАВЛЕНИЕ-ВВОДОМ-ВЫВОДОМ)	X	X
фраза RERUN (ПЕРЕПРОГОН)	X+	X+
фраза SAME AREA (ОБЩАЯ ОБЛАСТЬ)	X	X
фраза SAME RECORD AREA (ОБЩАЯ ОБЛАСТЬ ЗАПИСИ)	—	X
РАЗДЕЛ ДАННЫХ		
Секция файлов		
Статья описания файла	X	X
индикатор уровня FD (ОФ)	X	X
фраза BLOCK CONTAINS (В БЛОКЕ)	X	X
целое-1 RECORD/CHARACTERS (целое-1 ЗАПИСЕЙ/ЛИТЕР)	X	X
целое-1 TO целое-2 RECORDS/CHARACTERS (целое-1 ДО целое-2 ЗАПИСЕЙ/ЛИТЕР)	—	X
фраза DATA RECORDS (ЗАПИСИ ДАННЫХ)	+	+
фраза LABEL RECORDS (МЕТКИ)	+	+
фраза RECORD (В ЗАПИСИ)	X	X

GOST EXPERT ГОСТ 22558-89,

Элемент	Уровень 1	Уровень 2
целое-1 CHARACTERS (целое-1 ЛИТЕР)	X	X
фраза VARYING IN SIZE (ПЕРЕМЕННОЕ ЧИСЛО)	X	X
FROM целое-2 TO целое-3 CHARACTERS (ОТ целое-2 ДО целое-3 ЛИТЕР)	—	X
фраза DEPENDING ON (В ЗАВИСИМОСТИ ОТ)	—	X
целое-4 TO целое-5 CHARACTERS (целое-4 ДО целое-5 ЛИТЕР)	X	X
фраза VALUE (ЗНАЧЕНИЕ)	+	+
имя-реализации литерал	+	+
имя-реализации несколько литералов	+	+
имя-реализиции имя-данного	—	+
имя-реализации несколько имен-данных	—	+
Статья описания записи	X	X
РАЗДЕЛ ПРОЦЕДУР		
Декларативные процедуры	X	X
DECLARATIVES (ДЕКЛАРАТИВЫ)	X	X
END DECLARATIVES (КОНЕЦ ДЕКЛАРАТИВ)	X	X
Оператор CLOSE (ЗАКРЫТЬ)	X	X
имя-файла	X	X
несколько имен-файлов	X	X
фраза WITH LOCK (С ЗАМКОМ)	—	X
Оператор DELETE (УДАЛИТЬ)	X	X
фраза INVALID KEY (ПРИ ОШИБКЕ КЛЮЧА)	X	X
фраза NOT INVALID KEY (БЕЗ ОШИБКИ КЛЮЧА)	X	X
фраза END-DELETE (КОНЕЦ-УДАЛИТЬ)	X	X
Оператор OPEN (ОТКРЫТЬ)	X	X
имя-файла	X	X
несколько имен-файлов	X	X
фраза INPUT (ВХОДНОЙ)	X	X
фраза OUTPUT (ВЫХОДНОЙ)	X	X
фраза I-O (ВХОДНОЙ-ВЫХОДНОЙ)	X	X
фраза EXTEND (ДОПОЛНЯЕМЫЙ)	—	X
несколько INPUT (ВХОДНОЙ), OUTPUT (ВЫХОДНОЙ) и I-O (ВХОДНОЙ-ВЫХОДНОЙ)	X	X
несколько EXTEND (ДОПОЛНЯЕМЫЙ)	—	X
Оператор READ (ЧИТАТЬ)	X	X
фраза NEXT (СЛЕДУЮЩУЮ)	—	X
фраза INTO (В)	X	X
фраза AT END (В КОНЦЕ)	X	X
фраза NOT AT END (НЕ В КОНЦЕ)	X	X
фраза INVALID KEY (ПРИ ОШИБКЕ КЛЮЧА)	X	X
фраза NOT INVALID KEY (БЕЗ ОШИБКИ КЛЮЧА)	X	X
фраза END-READ (КОНЕЦ-ЧИТАТЬ)	X	X
Оператор REWRITE (ОБНОВИТЬ)	X	X
фраза FROM (ИЗ ПОЛЯ)	X	X
фраза INVALID KEY (ПРИ ОШИБКЕ КЛЮЧА)	X	X

GOST EXPERT ГОСТ 22558-89,

Элемент	Уровень 1	Уровень 2
фраза NOT INVALID KEY (БЕЗ ОШИБКИ КЛЮЧА)	X	X
фраза END-REWRITE (КОНЕЦ-ОБНОВИТЬ)	X	X
Оператор START (ПОДВЕСТИ)	—	X
фраза KEY (КЛЮЧ)	—	X
EQUAL TO (РАВНО)	—	X
=	—	X
GREATER THAN (БОЛЬШЕ)	—	X
>	—	X
NOT LESS THAN (НЕ МЕНЬШЕ)	—	X
NOT < (НЕ <)	—	X
GREATER THAN OR EQUAL TO (БОЛЬШЕ ИЛИ РАВНО)	—	X
>=	—	X
фраза INVALID KEY (ПРИ ОШИБКЕ КЛЮЧА)	—	X
фраза NOT INVALID KEY (БЕЗ ОШИБКИ КЛЮЧА)	—	X
фраза END-START (КОНЕЦ-ПОДВЕСТИ)	—	X
Оператор USE (ИСПОЛЬЗОВАТЬ)	X	X
фраза EXCEPTION/ERROR PROCEDURE (ПРОЦЕДУРЫ ОШИБКИ)	X	X
ON имя-файла (ДЛЯ имя-файла)	X	X
ON несколько имен-файлов (ДЛЯ несколько имен-файлов)	—	X
ON INPUT (ДЛЯ ВХОДНЫХ)	X	X
ON OUTPUT (ДЛЯ ВЫХОДНЫХ)	X	X
ON I-O (ДЛЯ ВХОДНЫХ-ВЫХОДНЫХ)	X	X
ON EXTEND (ДЛЯ ДОПОЛНЯЕМЫХ)	—	X
Оператор WRITE (ПИСАТЬ)	X	X
фраза FROM (ИЗ ПОЛЯ)	X	X
фраза INVALID KEY (ПРИ ОШИБКЕ КЛЮЧА)	X	X
фраза NOT INVALID KEY (БЕЗ ОШИБКИ КЛЮЧА)	X	X
фраза END-WRITE (КОНЕЦ-ПИСАТЬ)	X	X

2.5. Список элементов в модуле индексного ввода-вывода

Элемент	Уровень 1	Уровень 2
ПОНЯТИЯ ЯЗЫКА		
Слова, определенные пользователем		
имя-файла	X	X
имя-записи	X	X
Состояние ввода-вывода	X	X

GOST EXPERT ГОСТ 22558-89,

Элемент	Уровень 1	Уровень 2
РАЗДЕЛ ОБОРУДОВАНИЯ		
Секция ввода-вывода		
Параграф FILE-CONTROL (УПРАВЛЕНИЕ-ФАЙЛАМИ)	X	X
Статья управления файлом	X	X
фраза SELECT (ДЛЯ)	X	X
фраза OPTIONAL (НЕОБЯЗАТЕЛЬНОГО)	—	X
только входные, входные-выходные и дополняемые файлы	—	X
фраза ACCESS MODE (ДОСТУП)	X	X
SEQUENTIAL (ПОСЛЕДОВАТЕЛЬНЫЙ)	X	X
RANDOM (ПРОИЗВОЛЬНЫЙ)	X	X
DYNAMIC (ДИНАМИЧЕСКИЙ)	—	X
фраза ALTERNATE RECORD KEY (ДОПОЛНИТЕЛЬНЫЙ КЛЮЧ ЗАПИСИ)	—	X
фраза WITH DUPLICATES (С ДУБЛИРОВАНИЕМ)	—	X
фраза ASSIGN (НАЗНАЧИТЬ)	X	X
имя-реализации	X	X
литерал	X	X
фраза FILE STATUS (СОСТОЯНИЕ ФАЙЛА)	X	X
фраза ORGANIZATION IS INDEXED (ОРГАНИЗАЦИЯ ИНДЕКСНАЯ)	X	X
фраза RECORD KEY (КЛЮЧ ЗАПИСИ)	X	X
фраза RESERVE (РЕЗЕРВИРОВАТЬ)	—	X
Параграф I-O-CONTROL (УПРАВЛЕНИЕ-ВВОДОМ-ВЫВОДОМ)	X	X
фраза RERUN (ПЕРЕПРОГОН)	+	+
фраза SAME AREA (ОБЩАЯ ОБЛАСТЬ)	X	X
фраза SAME RECORD AREA (ОБЩАЯ ОБЛАСТЬ ЗАПИСИ)	—	X
РАЗДЕЛ ДАННЫХ		
Секция файлов		
Статья описания файла	X	X
индикатор уровня FD (ОФ)	X	X
фраза BLOCK CONTAINS (В БЛОКЕ)	X	X
целое-1 RECORDS/CHARACTERS (целое-1 ЗАПИСЕЙ/ЛИТЕР)	X	X
целое-1 TO целое-2 RECORDS/CHARACTERS (целое-1 ДО целое-2 ЗАПИСЕЙ/ЛИТЕР)	—	X
фраза DATA RECORDS (ЗАПИСИ ДАННЫХ)	+	+
фраза LABEL RECORDS (МЕТКИ)	+	+
фраза RECORD (В ЗАПИСИ)	X	X
целое-1 CHARACTERS (целое-1 ЛИТЕР)	X	X
фраза VARYING IN SIZE (ПЕРЕМЕННОЕ ЧИСЛО)	—	X
FROM целое-2 TO целое-3 CHARACTERS (ОТ целое-2 ДО целое-3 ЛИТЕР)	—	X

GOST EXPERT ГОСТ 22558-89,

Элемент	Уровень 1	Уровень 2
фраза DEPENDING ON (В ЗАВИСИМОСТИ ОТ)	—	X
целое-4 TO целое-5 CHARACTERS (целое-4 ДО целое-5 ЛИТЕР)	X	X
фраза VALUE (ЗНАЧЕНИЕ)	+	+
имя-реализации литерал	+	+
имя-реализации несколько литералов	+	+
имя-реализации имя-данного	—	+
имя-реализации несколько имен-данных	—	+
Статья описания записи	X	X
РАЗДЕЛ ПРОЦЕДУР		
Декларативные процедуры	X	X
DECLARATIVES (ДЕКЛАРАТИВЫ)	X	X
END DECLARATIVES (КОНЕЦ ДЕКЛАРАТИВ)	X	X
Оператор CLOSE (ЗАКРЫТЬ)	X	X
имя-файла	X	X
несколько имен-файлов	X	X
фраза WITH LOCK (С ЗАМКОМ)	—	X
Оператор DELETE (УДАЛИТЬ)	X	X
фраза INVALID KEY (ПРИ ОШИБКЕ КЛЮЧА)	X	X
фраза NOT INVALID KEY (БЕЗ ОШИБКИ КЛЮЧА)	X	X
фраза END-DELETE (КОНЕЦ-УДАЛИТЬ)	X	X
Оператор OPEN (ОТКРЫТЬ)	X	X
имя-файла	X	X
несколько имен-файлов	X	X
фраза INPUT (ВХОДНОЙ)	X	X
фраза OUTPUT (ВЫХОДНОЙ)	X	X
фраза I-O (ВХОДНОЙ-ВЫХОДНОЙ)	X	X
фраза EXTEND (ДОПОЛНЯЕМЫЙ)	X	X
несколько INPUT (ВХОДНОЙ), OUTPUT (ВЫХОДНОЙ) и I-O (ВХОДНОЙ-ВЫХОДНОЙ)	X	X
несколько EXTEND (ДОПОЛНЯЕМЫЙ)	—	X
Оператор READ (ЧИТАТЬ)	X	X
фраза NEXT (СЛЕДУЮЩУЮ)	—	X
фраза INTO (В)	X	X
фраза AT END (В КОНЦЕ)	X	X
фраза NOT AT END (НЕ В КОНЦЕ)	X	X
фраза KEY (КЛЮЧ)	—	X
фраза INVALID KEY (ПРИ ОШИБКЕ КЛЮЧА)	X	X
фраза NOT INVALID KEY (БЕЗ ОШИБКИ КЛЮЧА)	X	X
фраза END-READ (КОНЕЦ-ЧИТАТЬ)	X	X
Оператор REWRITE (ОБНОВИТЬ)	X	X
фраза FROM (ИЗ ПОЛЯ)	X	X
фраза INVALID KEY (ПРИ ОШИБКЕ КЛЮЧА)	X	X
фраза NOT INVALID KEY (БЕЗ ОШИБКИ КЛЮЧА)	X	X
фраза END-REWRITE (КОНЕЦ-ОБНОВИТЬ)	X	X
Оператор START (ПОДВЕСТИ)	—	X

GOST EXPERT ГОСТ 22558-89,

Элемент	Уровень 1	Уровень 2
фраза KEY (КЛЮЧ)	X	X
EQUAL TO (РАВНО)	—	X
=	—	X
GREATER THAN (БОЛЬШЕ)	—	X
>	—	X
NOT LESS THAN (НЕ МЕНЬШЕ)	—	X
NOT < (НЕ <)	—	X
GREATER THAN OR EQUAL TO (БОЛЬШЕ ИЛИ РАВНО)	—	X
>=	—	X
фраза INVALID KEY (ПРИ ОШИБКЕ КЛЮЧА)	—	X
фраза NOT INVALID KEY (БЕЗ ОШИБКИ КЛЮЧА)	—	X
фраза END-START (КОНЕЦ-ПОДВЕСТИ)	—	X
Оператор USE (ИСПОЛЬЗОВАТЬ)	X	X
фраза EXCEPTION/ERROR PROCEDURE (ПРОЦЕДУРЫ ОШИБКИ)	X	X
ON имя-файла (ДЛЯ имя-файла)	X	X
ON несколько имен-файлов (ДЛЯ несколько имен-файлов)	—	X
ON INPUT (ДЛЯ ВХОДНЫХ)	X	X
ON OUTPUT (ДЛЯ ВЫХОДНЫХ)	X	X
ON I-O (ДЛЯ ВХОДНЫХ-ВЫХОДНЫХ)	X	X
ON EXTEND (ДЛЯ ДОПОЛНЯЕМЫХ)	X	X
Оператор WRITE (ПИСАТЬ)	X	X
фраза FROM (ИЗ ПОЛЯ)	X	X
фраза INVALID KEY (ПРИ ОШИБКЕ КЛЮЧА)	X	X
фраза NOT INVALID KEY (БЕЗ ОШИБКИ КЛЮЧА)	X	X
фраза END-WRITE (КОНЕЦ-ПИСАТЬ)	X	X

2.6. Список элементов в модуле межпрограммных связей

Элемент	Уровень 1	Уровень 2
ПОНЯТИЯ ЯЗЫКА		
Структура исходной программы		
Вложенные исходные программы	—	X
РАЗДЕЛ ИДЕНТИФИКАЦИИ		
Параграф PROGRAM-ID (ПРОГРАММА)		
фраза COMMON (ОБЩАЯ)	—	X
фраза INITIAL (НАЧАЛЬНАЯ)	—	X
РАЗДЕЛ ДАННЫХ		
Секция файлов		
Статья описания файла (индикатор уровня FD (ОФ))		
фраза EXTERNAL (ВНЕШНЕЕ)	—	X

GOST EXPERT ГОСТ 22558-89,

Элемент	Уровень 1	Уровень 2
фраза GLOBAL (ГЛОБАЛЬНОЕ)	—	X
Статья описания данного (номер уровня 01)		
фраза GLOBAL (ГЛОБАЛЬНОЕ)	—	X
Секция рабочей памяти		
Статья описания данного (номер уровня 01)		
фраза EXTERNAL (ВНЕШНЕЕ)	—	X
фраза GLOBAL (ГЛОБАЛЬНОЕ)	—	X
Секция связи	X	X
Статья описания записи	X	X
Статья описания данного с уровнем 77	X	X
Секция отчетов		
Статья описания отчета (индикатор уровня RD (00))		
фраза GLOBAL (ГЛОБАЛЬНОЕ)	—	X
РАЗДЕЛ ПРОЦЕДУР		
Заголовок раздела процедур		
фраза USING (ИСПОЛЬЗУЯ)	X	X
разрешается по крайней мере 5 операндов	X	—
нет ограничения на число операндов	—	X
Оператор CALL (ВЫЗВАТЬ)	X	X
литерал	X	X
идентификатор	—	X
фраза USING (ИСПОЛЬЗУЯ)	X	X
идентификатор	X	X
разрешается по крайней мере 5 операндов	X	—
нет ограничения на число операндов	—	X
фраза BY REFERENCE (ССЫЛКУ)	—	X
фраза BY CONTENT (ЗНАЧЕНИЕ)	—	X
фраза ON OVERFLOW (ПРИ ПЕРЕПОЛНЕНИИ)	—	X
фраза ON EXCEPTION (ПРИ ОШИБКЕ)	—	X
фраза NOT ON EXCEPTION (БЕЗ ОШИБКИ)	—	X
фраза END-CALL (КОНЕЦ-ВЫЗВАТЬ) (формат 1)	X	X
фраза END-CALL (КОНЕЦ-ВЫЗВАТЬ) (формат 2)	—	X
Оператор CANCEL (ОСВОБОДИТЬ)	—	X
литерал	—	X
идентификатор	—	X
Оператор EXIT PROGRAM (ВЫЙТИ ИЗ ПРОГРАММЫ)	X	X
Оператор USE (ИСПОЛЬЗОВАТЬ)		
фраза EXCEPTION/ERROR PROCEDURE (ПРОЦЕДУРЫ ОШИБКИ)		
фраза GLOBAL (ГЛОБАЛЬНО)	—	X
Оператор USE BEFORE REPORTING (ИСПОЛЬЗОВАТЬ ДО ВЫДАЧИ)		
фраза GLOBAL (ГЛОБАЛЬНО)	—	X

GOST EXPERT ГОСТ 22558-89,

2.7. Список элементов в модуле сортировки-слияния

Элемент	Уровень 1
ПОНЯТИЯ ЯЗЫКА	
Слова, определенные пользователем	
имя-файла	X
имя-записи	X
РАЗДЕЛ ОБОРУДОВАНИЯ	
Секция ввода-вывода	
Параграф FILE-CONTROL (УПРАВЛЕНИЕ-ФАЙЛАМИ)	X
Статья управления файлом	XX
фраза SELECT (ДЛЯ)	XX
фраза ASSIGN (НАЗНАЧИТЬ)	XX
имя-реализация	XX
литерал	X
Параграф I-O-CONTROL (УПРАВЛЕНИЕ ВВОДОМ-ВЫВОДОМ)	X
фраза SAME SORT/SORT-MERGE AREA (ОБЩАЯ ОБЛАСТЬ СОРТИРОВКИ/СОРТИРОВКИ-СЛИЯНИЯ)	X
фраза SAME RECORD AREA (ОБЩАЯ ОБЛАСТЬ ЗАПИСИ)	X
РАЗДЕЛ ДАННЫХ	
Секция файлов	
Статья описания сортируемого-сливаемого файла	X
индикатор уровня SD (00)	XX
фраза DATA RECORDS (ЗАПИСИ ДАННЫХ)	+XX
фраза RECORD (В ЗАПИСИ)	XX
целое-1 CHARACTERS (целое-1 ЛИТЕР)	X
фраза VARYING IN SIZE (ПЕРЕМЕННОЕ ЧИСЛО) FROM целое-2 TO целое-3 CHARACTERS (ОТ целое-2 ДО целое-3 ЛИТЕР)	X
фраза DEPENDING ON (В ЗАВИСИМОСТИ ОТ)	X
целое-4 TO целое-5 CHARACTERS (целое-4 ДО целое-5 ЛИТЕР)	X
Статья описания записи	X
РАЗДЕЛ ПРОЦЕДУР	
Оператор MERGE (СЛИТЬ)	X
фраза ASCENDING/DESCENDING KEY (ПО ВОЗРАСТАНИЮ/УБЫВАНИЮ КЛЮЧА)	X
фраза COLLATING SEQUENCE (АЛФАВИТ)	X
фраза USING (ИСПОЛЬЗУЯ)	X
фраза OUTPUT PROCEDURE (ПРОЦЕДУРА ВЫВОДА)	X
имя-процедуры	X
фраза GIVING (ПОЛУЧАЯ)	X
Оператор RELEASE (ПЕРЕДАТЬ)	X
фраза FROM (ИЗ ПОЛЯ)	X
Оператор RETURN (ВЕРНУТЬ)	X
фраза INTO (В)	X
фраза AT END (В КОНЦЕ)	X

GOST EXPERT ГОСТ 22558-89,

Элемент	Уровень 1
фраза NOT AT END (НЕ В КОНЦЕ)	X
фраза END-RETURN (КОНЕЦ-ВЕРНУТЬ)	X
Оператор SORT (СОРТИРОВАТЬ)	X
фраза ASCENDING/DESCENDING KEY (ПО ВОЗРАСТАНИЮ/УБЫВАНИЮ КЛЮЧА)	X
фраза WITH DUPLICATES (С ДУБЛИРОВАНИЕМ)	X
фраза COLLATING SEQUENCE (АЛФАВИТ)	X
фраза INPUT PROCEDURE (ПРОЦЕДУРА ВВОДА)	X
имя-процедуры	X
фраза USING (ИСПОЛЬЗУЯ)	X
фраза OUTPUT PROCEDURE (ПРОЦЕДУРА ВЫВОДА)	X
имя-процедуры	X
фраза GIVING (ПОЛУЧАЯ)	X

2.8. Список элементов в модуле обработки исходных текстов

Элемент	Уровень 1	Уровень 2
ПОНЯТИЯ ЯЗЫКА		
Набор литер		
Литеры, используемые в пунктуации =	—	X
Слова, определенные пользователем		
имя-библиотеки	—	X
имя-текста	X	X
ВСЕ РАЗДЕЛЫ		
Оператор COPY (КОПИРОВАТЬ)	X	X
фраза OF/IN имя-библиотеки (ИЗ имя-библиотеки)	—	X
фраза REPLACING (ЗАМЕНЯЯ)	—	X
псевдотекст	—	X
идентификатор	—	X
литерал	—	X
слово	—	X
Оператор REPLACE (ЗАМЕНИТЬ)	—	X
псевдотекст BY псевдотекст (псевдотекст НА псевдотекст)	—	X
OFF (ОТКЛЮЧИТЬ)	—	X

2.9. Список элементов в модуле генератора отчетов

Элемент	Уровень 1
ПОНЯТИЯ ЯЗЫКА	
Слова, определенные пользователем	

GOST EXPERT ГОСТ 22558-89,

Элемент	Уровень 1
имя-файла	X
имя-отчета	X
Зарезервированные слова	
Специальные регистры	
LINE-COUNTER (СЧЕТЧИК-СТРОК)	X
PAGE-COUNTER (СЧЕТЧИК-СТРАНИЦ)	X
РАЗДЕЛ ОБОРУДОВАНИЯ	
Секция ввода-вывода	
Параграф FILE-CONTROL (УПРАВЛЕНИЕ-ФАЙЛАМИ)	X
Статья управления файлом	X
фраза SELECT (ДЛЯ)	X
фраза OPTIONAL (НЕОБЯЗАТЕЛЬНОГО)	X
только дополняемые файлы	X
фраза ACCESS MODE IS SEQUENTIAL (ДОСТУП ПОСЛЕДОВАТЕЛЬНЫЙ)	X
фраза ASSIGN (НАЗНАЧИТЬ)	X
имя-реализации	X
литерал	X
фраза FILE STATUS (СОСТОЯНИЕ ФАЙЛА)	X
фраза ORGANIZATION IS SEQUENTIAL (ОРГАНИЗАЦИЯ ПОСЛЕДОВАТЕЛЬНАЯ)	X
фраза PADDING CHARACTER (ЛИТЕРА ЗАПОЛНИТЕЛЬ)	X
фраза RECORD DELIMITER (ОГРАНИЧИТЕЛЬ ЗАПИСИ)	X
фраза RESERVE (РЕЗЕРВИРОВАТЬ)	X
Параграф I-O-CONTROL (УПРАВЛЕНИЕ-ВВОДОМ-ВЫВОДОМ)	X
фраза MULTIPLE FILE TAPE (НА ОДНОЙ КАТУШКЕ)	+
фраза SAME AREA (ОБЩАЯ ОБЛАСТЬ)	X
РАЗДЕЛ ДАННЫХ	
Секция файлов	
Статья описания файла	X
индикатор уровня FD (ОФ)	X
фраза BLOCK CONTAINS (В БЛОКЕ)	X
целое RECORDS/CHARECTERS (целое ЗАПИСЕЙ/ЛИТЕР)	X
целое-1 TO целое-2 RECORDS/CHARACTERS (целое-1 ДО целое-2 ЗАПИСЕЙ/ЛИТЕР)	X
фраза CODE-SET (АЛФАВИТ)	X
фраза LABEL RECORDS (МЕТКИ)	X
фраза RECORD (В ЗАПИСИ)	X
целое-1 CHARACTERS (целое-1 ЛИТЕР)	X
целое-4 TO целое-5 CHARACTERS (целое-4 ДО целое-5 ЛИТЕР)	X
фраза REPORT (ОТЧЕТ)	X
фраза VALUE OF (ЗНАЧЕНИЕ)	X
имя-реализации литерал	+
имя-реализации несколько литералов	+
имя-реализации имя-данного	+
имя-реализации несколько имен-данных	+

GOST EXPERT ГОСТ 22558-89,

Элемент	Уровень 1
Секция отчетов	
Статья описания отчета	X
индикатор уровня RD (00)	X
фраза CODE (С КОДОМ)	X
фраза CONTROL (УПРАВЛЕНИЕ)	X
фраза PAGE (РАЗМЕР СТРАНИЦЫ)	X
Статья описания группы отчета	X
фраза BLANK WHEN ZERO (ПРОБЕЛ КОГДА НУЛЬ)	X
фраза COLUMN NUMBER (НОМЕР СТОЛБЦА)	X
фраза имя-данного	X
фраза GROUP INDICATE (ОПРЕДЕЛЯЕТ ГРУППУ)	X
фраза JUSTIFIED (СДВИНУТО)	X
фраза номер-уровня	X
от 01 до 49; представление одной или двумя цифрами	X
фраза LINE NUMBER (НОМЕР СТРОКИ)	X
фраза NEXT GROUP (СЛЕДУЮЩАЯ ГРУППА)	X
фраза PICTURE (ШАБЛОН)	X
фраза SIGN (ЗНАК)	X
фраза SOURCE (ИСТОЧНИК)	X
фраза SUM (СУММА)	X
фраза TYPE (ТИП)	X
фраза USAGE (об использовании)	X
DISPLAY (ДЛЯ ВЫДАЧИ)	X
фраза VALUE (ЗНАЧЕНИЕ)	X
литерал	X
РАЗДЕЛ ПРОЦЕДУР	
Декларативные процедуры	X
DECLARATIVES (ДЕКЛАРАТИВЫ)	X
END DECLARATIVES (КОНЕЦ ДЕКЛАРАТИВ)	X
Оператор CLOSE (ЗАКРЫТЬ)	X
фраза REEL/UNIT (КАТУШКУ/ТОМ)	X
фраза FOR REMOVAL (С УДАЛЕНИЕМ)	X
фраза WITH NO REWIND (БЕЗ ПЕРЕМОТКИ)	X
Оператор GENERATE (ГЕНЕРИРОВАТЬ)	X
имя-данного	X
имя-отчета	X
Оператор INITIATE (НАЧАТЬ)	X
Оператор OPEN (ОТКРЫТЬ)	X
фраза OUTPUT (ВЫХОДНОЙ)	X
фраза WITH NO REWIND (БЕЗ ПЕРЕМОТКИ)	X
фраза EXTEND (ДОПОЛНЯЕМЫЙ)	X
Оператор SUPPRESS (ПОДАВИТЬ)	X
Оператор TERMINATE (ЗАКОНЧИТЬ)	X
Оператор USE (ИСПОЛЬЗОВАТЬ)	X
фраза EXCEPTION/ERROR PROCEDURE (ПРОЦЕДУРЫ ОШИБКИ)	X
ON имя-файла (ДЛЯ имя-файла)	X
ON несколько имен-файлов (ДЛЯ несколько имен-файлов)	X
ON OUTPUT (ДЛЯ ВЫХОДНЫХ)	X
ON EXTEND (ДЛЯ ДОПОЛНЯЕМЫХ)	X
фраза BEFORE REPORTING (ДО ВЫДАЧИ)	X

GOST EXPERT ГОСТ 22558-89,

2.10. Список элементов в модуле коммуникаций

Элемент	Уровень 1	Уровень 2
ПОНЯТИЯ ЯЗЫКА		
Слова, определенные пользователем		
имя-коммуникации	X	X
РАЗДЕЛ ДАННЫХ		
Секция коммуникации		
Статья описания коммуникации	X	X
индикатор уровня CD (OK)	X	X
фраза FOR INPUT (ДЛЯ ВВОДА)	X	X
фраза INITIAL (НАЧАЛЬНОГО)	—	X
фраза END KEY (КЛЮЧ КОНЦА)	X	X
фраза MESSAGE COUNT (ЧИСЛО СООБЩЕНИЙ)	X	X
фраза MESSAGE DATA (ДАТА СООБЩЕНИЯ)	X	X
фраза MESSAGE TIME (ВРЕМЯ СООБЩЕНИЯ)	X	X
фраза SYMBOLIC QUEUE (СИМВОЛИЧЕСКАЯ ОЧЕРЕДЬ)	X	X
фраза SYMBOLIC SOURCE (СИМВОЛИЧЕСКИЙ ИСТОЧНИК)	X	X
фраза SYMBOLIC SUB-QUEUE-1 (СИМВОЛИЧЕСКАЯ ПОДОЧЕРЕДЬ-1)	—	X
фраза SYMBOLIC SUB-QUEUE-2 (СИМВОЛИЧЕСКАЯ ПОДОЧЕРЕДЬ-2)	—	X
фраза SYMBOLIC-SUB-QUEUE-3 (СИМВОЛИЧЕСКАЯ ПОДОЧЕРЕДЬ-3)	—	X
фраза STATUS KEY (КЛЮЧ СОСТОЯНИЯ)	X	X
фраза TEXT LENGTH (ДЛИНА ТЕКСТА)	X	X
несколько имен-данных	—	X
фраза FOR OUTPUT (ДЛЯ ВЫВОДА)	X	X
фраза DESTINATION COUNT (ЧИСЛО АДРЕСАТОВ)	X	X
должен быть один	X	—
может быть один или несколько	—	X
фраза DESTINATION TABLE (ТАБЛИЦА АДРЕСАТОВ)	—	X
фраза INDEXED BY (ИНДЕКСИРУЕТСЯ)	—	X
фраза ERROR KEY (КЛЮЧ ОШИБКИ)	X	X
фраза SYMBOLIC DESTINATION (СИМВОЛИЧЕСКИЙ АДРЕСАТ)	X	X
фраза STATUS KEY (КЛЮЧ СОСТОЯНИЯ)	X	X
фраза TEXT LENGTH (ДЛИНА ТЕКСТА)	X	X
фраза I-O (ДЛЯ ВВОДА-ВЫВОДА)	X	X
фраза INITIAL (НАЧАЛЬНОГО)	—	X
фраза END KEY (КЛЮЧ КОНЦА)	X	X
фраза MESSAGE DATE (ДАТА СООБЩЕНИЯ)	X	X
фраза MESSAGE TIME (ВРЕМЯ СООБЩЕНИЯ)	X	X
фраза STATUS KEY (КЛЮЧ СОСТОЯНИЯ)	X	X

GOST EXPERT ГОСТ 22558-89,

Элемент	Уровень 1	Уровень 2
фраза SYMBOLIC TERMINAL (СИМВОЛИЧЕСКИЙ ТЕРМИНАЛ)	X	X
фраза TEXT LENGTH (ДЛИНА ТЕКСТА)	X	X
несколько имен-данных	—	X
Статья описания записи	X	X
РАЗДЕЛ ПРОЦЕДУР		
Оператор ACCEPT MESSAGE COUNT (ПРИНЯТЬ ЧИСЛО СООБЩЕНИЙ)	X	X
Оператор DISABLE (ЗАПРЕТИТЬ)	—	X
фраза INPUT (ВВОД)	—	X
фраза TERMINAL (С ТЕРМИНАЛА)	—	X
фраза I-O TERMINAL (ВВОД-ВЫВОД С ТЕРМИНАЛА)	—	X
фраза OUTPUT (ВЫВОД)	—	X
фраза WITH KEY (КЛЮЧ)	—	+
Оператор ENABLE (РАЗРЕШИТЬ)	—	X
фраза INPUT (ВВОД)	—	X
фраза TERMINAL (С ТЕРМИНАЛА)	—	X
фраза I-O TERMINAL (ВВОД-ВЫВОД С ТЕРМИНАЛА)	—	X
фраза OUTPUT (ВЫВОД)	—	X
фраза WITH KEY (КЛЮЧ)	—	X
Оператор PURGE (ОЧИСТИТЬ)	—	X
Оператор RECEIVE (ПОЛУЧИТЬ)	X	X
фраза MESSAGE (СООБЩЕНИЕ)	X	X
фраза SEGMENT (СЕГМЕНТ)	—	X
фраза INTO идентификатор (В идентификатор)	X	X
фраза NO DATA (НЕТ ДАННЫХ)	X	X
фраза WITH DATA (ЕСТЬ ДАННЫЕ)	X	X
фраза END-RECEIVE (КОНЕЦ-ПОЛУЧИТЬ)		
Оператор SEND (ПОСЛАТЬ)	X	X
FROM идентификатор (ИЗ ПОЛЯ идентификатор) (часть сообщения)	—	X
FROM идентификатор (ИЗ ПОЛЯ идентификатор) (полное сообщение)	X	X
фраза WITH идентификатор (С идентификатор)	—	X
фраза WITH ESI (С ИКС)	—	X
фраза WITH EMI (С ИКЩ)	X	X
фраза WITH EGI (С ИКГ)	X	X
фраза BEFORE/AFTER ADVANCING (ДО/ПОСЛЕ ПРОДВИЖЕНИЯ)	X	X
целое-1 LINE/LINES (целое-1 СТРОК)	X	X
идентификатор LINE/LINES (идентификатор СТРОК)	X	X
мнемоническое-имя	—	X
PAGE (СТРАНИЦЫ)	X	X
REPLACING LINE (ЗАМЕНЯЯ СТРОКУ)	—	X

GOST EXPERT ГОСТ 22558-89,

2.11. Список элементов в модуле отладки

Элемент	Уровень 1	Уровень 2
ПОНЯТИЯ ЯЗЫКА		
Зарезервированные слова		
Специальный регистр DEBUG-ITEM (ДАННЫЕ-ОТЛАДКИ)	+	+
РАЗДЕЛ ОБОРУДОВАНИЯ		
Секция конфигурации		
Параграф SOURCE-COMPUTER (ИСХОДНАЯ МАШИНА)		
фраза WITH DEBUGGING MODE (В РЕЖИМЕ ОТЛАДКИ)	+	+
РАЗДЕЛ ПРОЦЕДУР		
Декларативные процедуры		
DECLARATIVES (ДЕКЛАРАТИВЫ)	+	+
END DECLARATIVES (КОНЕЦ ДЕКЛАРАТИВ)	+	+
Оператор USE FOR DEBUGGING (ИСПОЛЬЗОВАТЬ ДЛЯ ОТЛАДКИ)	+	+
имя-процедуры	+	+
ALL PROCEDURES (ПРИ ВСЕХ ПРОЦЕДУРАХ)	+	+
ALL REFERENCES идентификатор-1 (ПРИ ВСЕХ ССЫЛКАХ идентификатор-1)	—	+
имя-коммуникации	—	+
имя-файла	—	+

2.12. Список элементов в модуле сегментации

Элемент	Уровень 1	Уровень 2
ПОНЯТИЯ ЯЗЫКА		
Слова, определенные пользователем		
Номер сегмента	+	+
РАЗДЕЛ ОБОРУДОВАНИЯ		
Параграф OBJECT-COMPUTER (РАБОЧАЯ-МАШИНА)		
фраза SEGMENT-LIMIT (ГРАНИЦА СЕГМЕНТОВ)	—	+
РАЗДЕЛ ПРОЦЕДУР		
Номера-сегментов от 0 до 49 для постоянных сегментов	+	+
Номера-сегментов от 50 до 99 для независимых сегментов	+	+

GOST EXPERT ГОСТ 22558-89,

Элемент	Уровень 1	Уровень 2
Все секции, имеющие одинаковый номер сегмента, должны быть смежными в исходной программе	+	—
Секции, имеющие одинаковый номер сегмента, могут быть несмежными в исходной программе	—	+

3. СПИСОК ЭЛЕМЕНТОВ ПО РАЗДЕЛАМ КОБОЛА

3.1. Общее описание

Ниже приводится список элементов стандарта Кобола соответственно разделам Кобола.

В столбце, озаглавленном «Модуль», указывается модуль и уровень этого модуля для элемента. Для указания модулей используются следующие обозначения:

Ядро	ЯДР
Последовательный ввод-вывод	ПОД
Относительный ввод-вывод	ОТД
Индексный ввод-вывод	ИПД
Межпрограммные связи	МПС
Сортировка-слияние	СРТ
Обработка исходных текстов	ОИТ
Генератор отчетов	ГОТ
Коммуникации	КОМ
Отладка	ОТЛ
Сегментация	СЕГ

Для каждого элемента указан модуль, к которому он принадлежит, и минимальный уровень сложности этого модуля, на котором допустим элемент. Например, 2 ЯДР указывает, что элемент допустим на уровне 2 ядра, а 1 ИПД указывает, что элемент допустим на уровне 1 модуля индексного ввода-вывода. Литера +, следующая за сокращенным обозначением модуля, обозначает, что элемент является устаревшим элементом языка и будет удален в следующей редакции стандарта.

Элемент	Модуль
3.2. Список элементов понятий языка	
ПОНЯТИЯ ЯЗЫКА	
Набор литер	
Литеры, используемые для слов в английской нотации 0—9, A—Z, -(дефис) в русской нотации А—Я, D, F, G, I, J, L, N, Q, R, S, U, V, W, Y, Z, 0—9, -(дефис)	1 ЯДР

GOST EXPERT ГОСТ 22558-89,

Элемент	Модуль
Литеры, используемые для пунктуации » ( ) . , ; пробел	1 ЯДР
Литеры, используемые для пунктуации : (двоеточие)	2 ЯДР
Литеры, используемые для пунктуации =	1 ОИТ
Литеры, используемые в арифметических операциях + — * / **	2 ЯДР
Литеры, используемые в отношениях = > < >= <=	1 ЯДР
Литеры, используемые в редактировании B + — . , Z(П) * $ ( ) 0 CR(КР) DB(ДБ) /	1 ЯДР
Литеры, используемые при индексировании + —	1 ЯДР
Разрешена замена одной литерой	1 ЯДР
Разрешена замена двумя литерами	1 ЯДР +
Разделители	
» ( ) . , ; пробел	1 ЯДР
: (двоеточие)	2 ЯДР
Строка-литер	
Слова Кобола	
Максимум 30 литер	1 ЯДР
Слова, определенные пользователем	
имя-алфавита	1 ЯДР
имя-библиотеки	2 ОИТ
имя данного	1 ЯДР
имя записи	1 ПОД
	1 ОТД
	1 ИПД
	1 СРТ
имя индекса	1 ЯДР
имя класса	1 ЯДР
имя коммуникации	1 КОМ
имя отчета	1 ГОТ
имя параграфа	1 ЯДР
имя программного модуля	1 ЯДР +
имя программы	1 ЯДР
имя секции	1 ЯДР
имя текста	1 ОИТ
имя условия	2 ЯДР
имя файла	1 ПОД
	1 ОТД
	1 ИПД
	1 СРТ
	1 ГОТ
мнемоническое имя	1 ЯДР
номер сегмента	1 СЕГ +
номер уровня	1 ЯДР
символическая литера	2 ЯДР
Системные имена	
имя машины	1 ЯДР
имя реализации	1 ЯДР
имя языка	1 ЯДР +

GOST EXPERT ГОСТ 22558-89,

Элемент	Модуль
Зарезервированные слова	
Обязательные слова	1 ЯДР
Ключевые слова	1 ЯДР
Слова специальные литеры	
знаки арифметических операций + — */**	2 ЯДР
знаки арифметических операций при индексировании + —	1 ЯДР
литеры отношения = > < >= <=	1 ЯДР
Необязательные слова	1 ЯДР
Слова специального назначения	
Стандартные константы: ZERO (НУЛЬ), ZEROS, ZEROES (НУЛИ), SPACE (ПРОБЕЛ), SPACES (ПРОБЕЛЫ), HIGH-VALUE (НАИБОЛЬШЕЕ-ЗНАЧЕНИЕ), HIGH-VALUES (НАИБОЛЬШИЕ-ЗНАЧЕНИЯ), LOW-VALUE (НАИМЕНЬШЕЕ-ЗНАЧЕНИЕ), LOW-VALUES (НАИМЕНЬШИЕ-ЗНАЧЕНИЯ), QUOTE (КАВЫЧКА), QUOTES (КАВЫЧКИ)	1 ЯДР
Стандартные константы: символическая-литера, ALL литерал (ВСЕ литерал), ALL стандартная-константа (ВСЕ стандартная-константа), ALL символическая-литера (ВСЕ символическая-литера)	2 ЯДР
Специальные регистры	
LINAGE-COUNTER (СЧЕТЧИК-ВЕРСТКИ)	2 ПОД
LINE-COUNTER (СЧЕТЧИК-СТРОК)	1 ГОТ
PAGE-COUNTER (СЧЕТЧИК-СТРАНИЦ)	1 ГОТ
DEBUG-ITEM (ДАННЫЕ-ОТЛАДКИ)	1 ОТЛ +
Литералы	
Числовые литералы: от 1 до 18 цифр	1 ЯДР
Нечисловые литералы: от 1 до 160 литер	1 ЯДР
строка-литер (ШАБЛОН строка-литер)	1 ЯДР
Статья-комментарий	1 ЯДР +
Однозначность ссылки	
Уточнение	
уточнение недопустимо, имена должны быть однозначны при ссылке	1 ЯДР
50 уточнителей	2 ЯДР
Индексирование	
3 уровня индексов	1 ЯДР
7 уровней индексов	2 ЯДР
индексирование литералом	1 ЯДР
индексирование именем-данного	1 ЯДР
индексирование именем-индекса	1 ЯДР
относительное индексирование	1 ЯДР
Модификация ссылки	2 ЯДР
Формат представления	
Порядковый номер	1 ЯДР
Продолжение строк	
продолжение нечисловых литералов	1 ЯДР
продолжение слов Кобола, числовых литералов, строк-литер шаблона	2 ЯДР
Строки пробелов	1 ЯДР

GOST EXPERT ГОСТ 22558-89,

Элемент	Модуль
Строки комментария	
строки комментария со звездочкой (*)	1 ЯДР
строки комментария с дробной чертой (/)	1 ЯДР
Отладочная строка с литерой D (T) в поле индикатора	1 ЯДР
Структура исходной программы	
Раздел идентификации обязателен	1 ЯДР
Раздел оборудования необязателен	1 ЯДР
Раздел данных необязателен	1 ЯДР
Раздел процедур необязателен	1 ЯДР
Заголовок конца программы	2 ЯДР
Вложенные исходные программы	2 МПС

3.3. Список элементов раздела идентификации

РАЗДЕЛ ИДЕНТИФИКАЦИИ	
Параграф PROGRAM-ID (ПРОГРАММА)	1 ЯДР
имя-параграфа	1 ЯДР
фраза COMMON (ОБЩАЯ)	2 МПС
фраза INITIAL (НАЧАЛЬНАЯ)	2 МПС
Параграф AUTHOR (АВТОР)	1 ЯДР +
Параграф INSTALLATION (ПРЕДПРИЯТИЕ)	1 ЯДР +
Параграф DATE-WRITTEN (ДАТА-НАПИСАНИЯ)	1 ЯДР +
Параграф DATE-COMPILED (ДАТА-ТРАНСЛЯЦИИ)	2 ЯДР +
Параграф SECURITY (ПОЛНОМОЧИЯ)	1 ЯДР +
Обработка исходных текстов в разделе идентификации	
Оператор COPY (КОПИРОВАТЬ)	1 ОИТ
OF/IN имя-библиотеки (ИЗ имя-библиотеки)	2 ОИТ
фраза REPLACING (ЗАМЕНЯЯ)	2 ОИТ
псевдотекст	2 ОИТ
идентификатор	2 ОИТ
литерал	2 ОИТ
слово	2 ОИТ
Оператор REPLACE (ЗАМЕНИТЬ)	2 ОИТ
псевдотекст BY псевдотекст (псевдотекст НА псевдотекст)	2 ОИТ
OFF (ОТКЛЮЧИТЬ)	2 ОИТ

3.4. Список элементов раздела оборудования

РАЗДЕЛ ОБОРУДОВАНИЯ	
Секция конфигурации	1 ЯДР
Параграф SOURCE-COMPUTER (ИСХОДНАЯ-МАШИНА)	1 ЯДР
имя-машины	1 ЯДР
фраза WITH DEBUGGING MODE (В РЕЖИМЕ ОТЛАДКИ)	1 ЯДР
	1 ОТЛ +
Параграф OBJECT-COMPUTER (РАБОЧАЯ-МАШИНА)	1 ЯДР
имя-машины	1 ЯДР
фраза MEMORY SIZE (РАЗМЕР ПАМЯТИ)	1 ЯДР +
фраза PROGRAM COLLATING SEQUENCE (ПРОГРАММНЫЙ-АЛФАВИТ)	1 ЯДР
фраза SEGMENT-LIMIT (ГРАНИЦА СЕГМЕНТОВ)	1 СЕГ +

GOST EXPERT ГОСТ 22558-89,

Элемент	Модуль
Параграф SPECIAL-NAMES (СПЕЦИАЛЬНЫЕ-ИМЕНА)	1 ЯДР
фраза ALPHABET (АЛФАВИТ)	1 ЯДР
вариант STANDARD-1 (СТАНДАРТ-А)	1 ЯДР
вариант STANDARD-2 (СТАНДАРТ-М)	1 ЯДР
вариант NATIVE (ВНУТРЕННИЙ)	1 ЯДР
вариант имя-реализации	1 ЯДР
вариант литерал	2 ЯДР
фраза CLASS (КЛАСС)	1 ЯДР
фраза CURRENCY SIGN (ВАЛЮТНЫЙ ЗНАК)	1 ЯДР
фраза DECIMAL-POINT (ДЕСЯТИЧНАЯ ТОЧКА)	1 ЯДР
фраза имя-реализации	1 ЯДР
вариант IS мнемоническое имя	1 ЯДР
вариант ON STATUS IS имя-условия (ВКЛЮЧЕНО имя-условия)	1 ЯДР
вариант OFF STATUS IS имя-условия (ВЫКЛЮЧЕНО имя-условия)	1 ЯДР
фраза SYMBOLIC CHARACTERS (СИМВОЛИЧЕСКАЯ ЛИТЕРА)	2 ЯДР
Секция ввода-вывода	1 ПОД
	1 ОТД
	1 ИПД
	1 СРТ
	1 ГОТ
Параграф FILE-CONTROL (УПРАВЛЕНИЕ-ФАЙЛАМИ)	1 ПОД
	1 ОТД
	1 ИПД
	1 СРТ
	1 ГОТ
Статья управления файлом	1 ПОД
	1 ОТД
	1 ИПД
	1 СРТ
	1 ГОТ
фраза SELECT (ДЛЯ)	1 ПОД
	1 ОТД
	1 ИПД
	1 СРТ
	1 ГОТ
вариант OPTIONAL (НЕОБЯЗАТЕЛЬНО)	2 ПОД
	2 ОТД
	2 ИПД
	1 ГОТ
только входные, входные-выходные и дополняемые	2 ПОД
	2 ОТД
	2 ИПД
только дополняемые	1 ГОТ
фраза ACCESS MODE (ДОСТУП)	
SEQUENTIAL (ПОСЛЕДОВАТЕЛЬНЫЙ)	1 ПОД
	1 ОТД
	1 ИПД
	1 ГОТ

GOST EXPERT ГОСТ 22558-89,

Элемент	Модуль
RANDOM (ПРОИЗВОЛЬНЫЙ)	1 ОТД 1 ИПД
DYNAMIC (ДИНАМИЧЕСКИЙ)	2 ОТД 2 ИПД
фраза RELATIVE KEY (ОТНОСИТЕЛЬНЫЙ КЛЮЧ)	1 ОТД
фраза ALTERNATE RECORD KEY (ДОПОЛНИТЕЛЬНЫЙ КЛЮЧ ЗАПИСИ)	2 ИПД
фраза WITH DUPLICATES (С ДУБЛИРОВАНИЕМ)	2 ИПД
фраза ASSIGN (НАЗНАЧИТЬ)	1 ПОД 1 ОТД 1 ИПД 1 СРТ 1 ГОТ
имя-реализации	1 ПОД 1 ОТД 1 ИПД 1 СРТ
литерал	1 ГОТ 1 ПОД 1 ОТД 1 ИПД 1 СРТ 1 ГОТ
фраза FILE STATUS (СОСТОЯНИЕ ФАЙЛА)	1 ПОД 1 ОТД 1 ИПД
фраза ORGANIZATION (ОРГАНИЗАЦИЯ)	1 ГОТ
SEQUENTIAL (ПОСЛЕДОВАТЕЛЬНАЯ)	1 ПОД 1 ГОТ
RELATIVE (ОТНОСИТЕЛЬНАЯ) INDEXED (ИНДЕКСНАЯ)	1 ОТД 1 ИПД
фраза PADDING CHARACTER (ЛИТЕРА ЗАПОЛНИТЕЛЬ)	2 ПОД 1 ГОТ
фраза RECORD DELIMITER (ОГРАНИЧИТЕЛЬ ЗАПИСИ)	2 ПОД 1 ГОТ
фраза RECORD KEY (КЛЮЧ ЗАПИСИ)	1 ИПД
фраза RESERVE AREA (РЕЗЕРВИРОВАТЬ)	2 ПОД 2 ОТД 2 ИПД 1 ГОТ
Параграф I-O-CONTROL (УПРАВЛЕНИЕ-ВВОДОМ-ВЫВОДОМ)	1 ПОД 1 ОТД 1 ИПД 1 СРТ 1 ГОТ
фраза MULTIPLE FILE TAPE (НА ОДНОЙ КАТУШКЕ)	2 ПОД + 1 ГОТ +
фраза RERUN (ПЕРЕПРОГОН)	1 ПОД + 1 ГОТ +

GOST EXPERT ГОСТ 22558-89,

Элемент	Модуль
фраза SAME AREA (ОБЩАЯ ОБЛАСТЬ)	1 ИПД + 1 ПОД 1 ОТД 1 ИПД 1 ГОТ
фраза SAME RECORD AREA (ОБЩАЯ ОБЛАСТЬ ЗАПИСИ)	2 ПОД 2 ОТД 2 ИПД 1 СРТ
фраза SAME SORT/SORT-MERGE AREA (ОБЩАЯ ОБЛАСТЬ СОРТИРОВКИ/СОРТИРОВКИ-СЛИЯНИЯ)	1 СРТ
Обработка исходных текстов в разделе оборудования	
Оператор COPY (КОПИРОВАТЬ)	1 ОИТ
OF/IN имя-библиотеки (ИЗ имя-библиотеки)	2 ОИТ
фраза REPLACING (ЗАМЕНЯЯ)	2 ОИТ
псевдотекст	2 ОИТ
идентификатор	2 ОИТ
литерал	2 ОИТ
слово	2 ОИТ
Оператор REPLACE (ЗАМЕНИТЬ)	2 ОИТ
псевдотекст BY псевдотекст (псевдотекст НА псевдотекст)	2 ОИТ
OFF (ОТКЛЮЧИТЬ)	2 ОИТ

3.4. Список элементов раздела данных

РАЗДЕЛ ДАННЫХ	1 ПОД
Секция файлов	1 ОТД 1 ИПД 1 СРТ 1 ГОТ
Статья описания файла	1 ПОД 1 ОТД 1 ИПД 1 ГОТ
индикатор уровня FD (ОФ)	1 ПОД 1 ОТД 1 ИПД 1 ГОТ
фраза BLOCK CONTAINS (В БЛОКЕ) целое RECORDS/CHARACTERS (целое ЗАПИСЕЙ/ЛИТЕР)	1 ПОД 1 ОТД 1 ИПД 1 ГОТ
целое-1 TO целое-2 RECORDS/CHARACTERS (целое-1 ДО целое-2 ЗАПИСЕЙ/ЛИТЕР)	2 ПОД 2 ОТД 2 ИПД 1 ГОТ
фраза CODE-SET (АЛФАВИТ)	1 ПОД 1 ГОТ

GOST EXPERT ГОСТ 22558-89,

Элемент	Модуль
фраза DATA RECORDS (ЗАПИСИ ДАННЫХ)	1 ПОД + 1 ОТД + 1 ИПД +
фраза EXTERNAL (ВНЕШНЕЕ)	2 МПС
фраза GLOBAL (ГЛОБАЛЬНОЕ)	2 МПС
фраза LABEL RECORDS (МЕТКИ)	1 ПОД + 1 ОТД + 1 ИПД + 1 ГОТ+
фраза LINAGE (ВЕРСТКА)	2 ПОД
фраза FOOTING (КОНЦОВКА)	2 ПОД
фраза TOP (ВЕРХНЕЕ ПОЛЕ)	2 ПОД
фраза BOTTOM (НИЖНЕЕ ПОЛЕ)	2 ПОД
фраза RECORD (В ЗАПИСИ)	
целое-1 CHARACTERS (целое-1 ЛИТЕР)	1 ПОД 1 ОТД 1 ИПД 1 ГОТ
фраза VARYING IN SIZE (ПЕРЕМЕННОЕ ЧИСЛО)	2 ПОД 2 ОТД 2 ИПД
целое-4 TO целое-5 CHARACTERS (целое-4 ДО целое-5 ЛИТЕР)	1 ПОД 1 ОТД 1 ИПД 1 ГОТ
фраза REPORT (ОТЧЕТ)	1 ГОТ
фраза VALUE OF (ЗНАЧЕНИЕ)	
имя-реализации литерал	1 ПОД + 1 ОТД + 1 ИПД + 1 ГОТ +
имя-реализации несколько литералов	1 ПОД + 1 ОТД + 1 ИПД + 1 ГОТ +
имя-реализации имя-данного	2 ПОД + 2 ОТД + 2 ИПД + 1 ГОТ +
имя-реализации несколько имен-данных	2 ПОД + 2 ОТД + 2 ИПД + 1 ГОТ +
Статья описания сортируемого-сливаемого файла	1 СРТ
индикатор уровня SD (ОС)	1 СРТ
фраза DATA RECORDS (ЗАПИСИ ДАННЫХ)	1 СРТ +
фраза RECORD (В ЗАПИСИ)	
целое-1 CHARACTERS (целое-1 ЛИТЕР)	1 СРТ
фраза VARYING IN SIZE (ПЕРЕМЕННОЕ ЧИСЛО)	1 СРТ
целое-4 TO целое-5 CHARACTERS (целое-4 ДО целое-5 ЛИТЕР)	1 СРТ

GOST EXPERT ГОСТ 22558-89,

Элемент	Модуль
Статья описания записи в секции файлов	1 ПОД
	1 ОТД
	1 ИПД
	1 СРТ
Секция рабочей памяти	1 ЯДР
Статья описания записи	1 ЯДР
Статья описания уровня 77	1 ЯДР
Секция связи	1 МПС
Статья описания записи	1 МПС
Статья описания уровня 77	1 МПС
Секция коммуникаций	1 КОМ
Статья описания коммуникации	1 КОМ
индикатор уровня CD (OK)	1 КОМ
фраза FOR INPUT (ДЛЯ ВВОДА)	1 КОМ
фраза INITIAL (НАЧАЛЬНОГО)	2 КОМ
фраза END KEY (КЛЮЧ КОНЦА)	1 КОМ
фраза MESSAGE COUNT (ЧИСЛО СООБЩЕНИЙ)	1 КОМ
фраза MESSAGE DATE (ДАТА СООБЩЕНИЯ)	1 КОМ
фраза MESSAGE TIME (ВРЕМЯ СООБЩЕНИЯ)	1 КОМ
фраза SYMBOLIC QUEUE (СИМВОЛИЧЕСКАЯ ОЧЕРЕДЬ)	1 КОМ
фраза SYMBOLIC SOURCE (СИМВОЛИЧЕСКИЙ ИСТОЧНИК)	1 КОМ
фраза SYMBOLIC SUB-QUEUE-1 (СИМВОЛИЧЕСКАЯ ПОДОЧЕРЕДЬ-1)	2 КОМ
фраза SYMBOLIC SUB-QUEUE-2 (СИМВОЛИЧЕСКАЯ ПОДОЧЕРЕДЬ-2)	2 КОМ
фраза SYMBOLIC SUB-QUEUE-3 (СИМВОЛИЧЕСКАЯ ПОДОЧЕРЕДЬ-3)	2 КОМ
фраза STATUS KEY (КЛЮЧ СОСТОЯНИЯ)	1 КОМ
фраза TEXT LENGTH (ДЛИНА ТЕКСТА)	1 КОМ
несколько имен-данных	2 КОМ
фраза FOR OUTPUT (ДЛЯ ВЫВОДА)	1 КОМ
фраза DESTINATION COUNT (ЧИСЛО АДРЕСАТОВ)	1 КОМ
должен быть один	1 КОМ
должен быть один или несколько	2 КОМ
фраза DESTINATION TABLE (ТАБЛИЦА АДРЕСАТОВ)	2 КОМ
фраза INDEXED BY (ИНДЕКСИРУЕТСЯ)	2 КОМ
фраза ERROR KEY (КЛЮЧ ОШИБКИ)	1 КОМ
фраза SYMBOLIC DESTINATION (СИМВОЛИЧЕСКИЙ АДРЕСАТ)	1 КОМ
фраза STATUS KEY (КЛЮЧ СОСТОЯНИЯ)	1 КОМ
фраза TEXT LENGTH (ДЛИНА ТЕКСТА)	1 КОМ
фраза FOR I-O (ДЛЯ ВВОДА-ВЫВОДА)	1 КОМ
фраза INITIAL (НАЧАЛЬНОГО)	2 КОМ
фраза END KEY (КЛЮЧ КОНЦА)	1 КОМ
фраза MESSAGE DATA (ДАТА СООБЩЕНИЯ)	1 КОМ
фраза MESSAGE TIME (ВРЕМЯ СООБЩЕНИЯ)	1 КОМ
фраза STATUS KEY (КЛЮЧ СОСТОЯНИЯ)	1 КОМ
фраза SYMBOLIC TERMINAL (СИМВОЛИЧЕСКИЙ ТЕРМИНАЛ)	1 КОМ

GOST EXPERT ГОСТ 22558-89,

Элемент	Модуль
фраза TEXT LENGTH (ДЛИНА ТЕКСТА)	1 КОМ
несколько имен-данных	2 КОМ
Статья описания записи	1 КОМ
Секция отчетов	1 ГОТ
Статья описания отчета	1 ГОТ
индикатор уровня RD (00)	1 ГОТ
фраза CODE (С КОДОМ)	1 ГОТ
фраза CONTROL (УПРАВЛЕНИЕ)	1 ГОТ
фраза GLOBAL (ГЛОБАЛЬНОЕ)	2 МПС
фраза PAGE (РАЗМЕР СТРАНИЦЫ)	1 ГОТ
Статья описания группы отчета	1 ГОТ
Следующие фразы появляются в статье описания записи, статье описания данного, статье описания уровня 77 и в статье описания группы отчета:	
Фраза BLANK WHEN ZERO (ПРОБЕЛ КОГДА НУЛЬ)	1 ЯДР
	1 ГОТ
Фраза COLUMN NUMBER (НОМЕР СТОЛБЦА)	1 ГОТ
Фраза имя-данного	1 ЯДР
	1 ГОТ
Фраза EXTERNAL (ВНЕШНЕЕ)	2 МПС
Фраза FILLER (ЗАПОЛНИТЕЛЬ)	1 ЯДР
Фраза GLOBAL (ГЛОБАЛЬНОЕ)	2 МПС
Фраза GROUP INDICATE (ОПРЕДЕЛЯЕТ ГРУППУ)	1 ГОТ
Фраза JUSTIFIED (СДВИНУТО)	1 ЯДР
	1 ГОТ
Фраза номер-уровня	1 ЯДР
от 01 до 49; одна или две цифры	1 ЯДР
	1 ГОТ
66	2 ЯДР
77	1 ЯДР
88	2 ЯДР
Фраза LINE NUMBER (НОМЕР СТРОКИ)	1 ГОТ
Фраза NEXT GROUP (СЛЕДУЮЩАЯ ГРУППА)	1 ГОТ
Фраза OCCURS (ПОВТОРЯЕТСЯ)	1 ЯДР
целое TIMES (целое РАЗ)	1 ЯДР
фраза ASCENDING/DESCENDING KEY (ПО ВОЗРАСТАНИЮ/УБЫВАНИЮ КЛЮЧА)	2 ЯДР
фраза INDEXED BY (ИНДЕКСИРУЕТСЯ)	1 ЯДР
фраза целое-1 TO целое-2 DEPENDING ON (целое-1 ДО целое-2 В ЗАВИСИМОСТИ ОТ)	2 ЯДР
Фраза PICTURE (ШАБЛОН)	1 ЯДР
	1 ГОТ
строка литер содержит не более 30 литер	1 ЯДР
	1 ГОТ
литеры данных: X 9 A	1 ЯДР
	1 ГОТ
операционные символы: S(3) V(T) P(M)	1 ЯДР
	1 ГОТ
литеры фиксированной вставки B + — . , [illegible] CR КР) DB (ДВ) /	1 ЯДР

GOST EXPERT ГОСТ 22558-89,

Элемент	Модуль
	1 ГОТ
литеры замещения или плавающей вставки $(¤)	+ —
Z(П) *	1 ЯДР
	1 ГОТ
замена валютного знака	1 ЯДР
	1 ГОТ
замена десятичной точки	1 ЯДР
	1 ГОТ
Фраза REDEFINES (ПЕРЕОПРЕДЕЛЯЕТ)	1 ЯДР
не может быть вложенной	1 ЯДР
может быть вложенной	2 ЯДР
Фраза RENAMES (ПЕРЕИМЕНОВЫВАЕТ)	2 ЯДР
Фраза SIGN (ЗНАК)	1 ЯДР
	1 ГОТ
Фраза SOURCE (ИСТОЧНИК)	1 ГОТ
Фраза SUM (СУММА)	1 ГОТ
Фраза SYNCHRONIZED (ВЫДЕЛЕНО)	1 ЯДР
Фраза TYPE (ТИП)	1 ГОТ
Фраза USAGE (*об использовании*)	1 ЯДР
	1 ГОТ
BINARY (ДВОИЧНОЕ)	1 ЯДР
COMPUTATIONAL (ДЛЯ ВЫЧИСЛЕНИЙ)	1 ЯДР
DISPLAY (ДЛЯ ВЫДАЧИ)	1 ЯДР
	1 ГОТ
INDEX (ДЛЯ ИНДЕКСА)	1 ЯДР
PACKED-DECIMAL (ДЕСЯТИЧНОЕ)	1 ЯДР
Фраза VALUE (ЗНАЧЕНИЕ)	1 ЯДР
	1 ГОТ
литерал	1 ЯДР
	1 ГОТ
несколько литералов	2 ЯДР
литерал-1 THROUGH литерал-2 (литерал-1 ПО литерал-2)	2 ЯДР
несколько диапазонов литералов	2 ЯДР
Обработка исходных текстов в разделе данных	
Оператор COPY (КОПИРОВАТЬ)	1 ОИТ
OF/IN имя-библиотеки (ИЗ имя-библиотеки)	2 ОИТ
фраза REPLACING (ЗАМЕНЯЯ)	2 ОИТ
псевдотекст	2 ОИТ
идентификатор	2 ОИТ
литерал	2 ОИТ
слово	2 ОИТ
Оператор REPLACE (ЗАМЕНИТЬ)	2 ОИТ
псевдотекст BY псевдотекст (псевдотекст НА псевдотекст)	2 ОИТ
OFF (ОТКЛЮЧИТЬ)	2 ОИТ

3.5 Список элементов раздела процедур

РАЗДЕЛ ПРОЦЕДУР	
Заголовок раздела процедур	1 ЯДР

GOST EXPERT ГОСТ 22558-89,

Элемент	Модуль
Фраза USING (ИСПОЛЬЗУЯ)	1 МПС
разрешается по крайней мере 5 операндов	1 МПС
нет ограничения на число операндов	2 МПС
Декларативные процедуры	1 ПОД
	1 ОТД
	1 ИПД
	1 ГОТ
	1 ОТЛ +
DECLARATIVES (ДЕКЛАРАТИВЫ)	1 ПОД
	1 ОТД
	1 ИПД
	1 ГОТ
	1 ОТЛ +
END DECLARATIVES (КОНЕЦ ДЕКЛАРАТИВ)	1 ПОД
	1 ОТД
	1 ИПД
	1 ГОТ
	1 ОТЛ +
Арифметические выражения	2 ЯДР
Знаки бинарных арифметических операций + — * / **	2 ЯДР
Знаки унарных арифметических операций + —	2 ЯДР
Условные выражения	1 ЯДР
Простое условие	1 ЯДР
Условие отношения	1 ЯДР
Знаки операций отношения	
[NOT] GREATER THAN ([НЕ] БОЛЬШЕ)	1 ЯДР
[NOT] > ([НЕ] >)	1 ЯДР
[NOT] LESS THAN ([НЕ] МЕНЬШЕ)	1 ЯДР
[NOT] < ([НЕ] <)	1 ЯДР
[NOT] EQUAL TO ([НЕ] РАВНО)	1 ЯДР
[NOT] = ([НЕ] =)	1 ЯДР
GREATER THAN OR EQUAL TO (БОЛЬШЕ ИЛИ РАВНО)	1 ЯДР
>=	1 ЯДР
LESS THAN OR EQUAL TO (МЕНЬШЕ ИЛИ РАВНО)	1 ЯДР
<=	1 ЯДР
Сравнение числовых операндов	1 ЯДР
Сравнение нечисловых операндов	1 ЯДР
Сравнение имен-индексов и(или) индексных данных	1 ЯДР
Условие класса	1 ЯДР
NUMERIC (ЧИСЛОВОЕ)	1 ЯДР
ALPHABETIC (БУКВЕННОЕ)	1 ЯДР
ALPHABETIC-LOWER (СТРОЧНЫЕ)	1 ЯДР
ALPHABETIC-UPPER (ПРОПИСНЫЕ)	1 ЯДР
имя-класса	1 ЯДР
Условие имени-условия	2 ЯДР
Условие знака	2 ЯДР
Условие состояния-переключателя	1 ЯДР
Сложное условие	2 ЯДР
Знаки логических операций AND (И) OR (ИЛИ) NOT (НЕ)	2 ЯДР
Отрицание условия	2 ЯДР
Комбинированное условие	2 ЯДР

GOST EXPERT ГОСТ 22558-89,

Элемент	Модуль
Условие в скобках	1 ЯДР
Сокращенные комбинированные условия отношения	2 ЯДР
Арифметические операторы	1 ЯДР
Арифметические операнды имеют длину до 18 цифр	1 ЯДР
Композиция операндов содержит не более 18 цифр	1 ЯДР
Оператор ACCEPT (ПРИНЯТЬ)	1 ЯДР
идентификатор	1 ЯДР
только одна передача данных	1 ЯДР
число передач данных не ограничивается	2 ЯДР
фраза FROM мнемоническое-имя (С мнемоническое-имя)	2 ЯДР
фраза FROM DATE/DAY/DAY-OF-WEEK/TIME (ДАТУ/ДЕНЬ/ДЕНЬ-НЕДЕЛИ/ВРЕМЯ)	2 ЯДР
Оператор ACCEPT MESSAGE COUNT (ПРИНЯТЬ ЧИСЛО СООБЩЕНИЙ)	1 КОМ
Оператор ADD (СЛОЖИТЬ)	1 ЯДР
идентификатор/литерал	1 ЯДР
несколько идентификаторов/литералов	1 ЯДР
ТО идентификатор (С идентификатор)	1 ЯДР
ТО несколько-идентификаторов (С несколько-идентификаторов)	1 ЯДР
ТО идентификатор/литерал GIVING идентификатор (С идентификатор/литерал ПОЛУЧАЯ идентификатор)	1 ЯДР
ТО идентификатор/литерал GIVING несколько-идентификаторов (С идентификатор/литерал ПОЛУЧАЯ несколько идентификаторов)	1 ЯДР
фраза ROUNDED (ОКРУГЛЯЯ)	1 ЯДР
фраза ON SIZE ERROR (ПРИ ПЕРЕПОЛНЕНИИ)	1 ЯДР
фраза NOT ON SIZE ERROR (БЕЗ ПЕРЕПОЛНЕНИЯ)	1 ЯДР
фраза END-ADD (КОНЕЦ-СЛОЖИТЬ)	1 ЯДР
фраза CORRESPONDING (СООТВЕТСТВЕННО)	2 ЯДР
Оператор ALTER (ИЗМЕНИТЬ)	1 ЯДР +
только одно имя-процедуры	1 ЯДР +
несколько имен-процедур	2 ЯДР +
Оператор CALL (ВЫЗВАТЬ)	1 МПС
литерал	1 МПС
идентификатор	2 МПС
фраза USING (ИСПОЛЬЗУЯ)	1 МПС
идентификатор	1 МПС
разрешается по крайней мере 5 операндов	1 МПС
нет ограничения на число операндов	2 МПС
фраза BY REFERENCE (ССЫЛКУ)	2 МПС
фраза BY CONTENT (ЗНАЧЕНИЕ)	2 МПС
фраза ON OVERFLOW (ПРИ ПЕРЕПОЛНЕНИИ)	2 МПС
фраза ON EXCEPTION (ПРИ ОШИБКЕ)	2 МПС
фраза NOT ON EXCEPTION (БЕЗ ОШИБКИ)	2 МПС
фраза END-CALL (КОНЕЦ-ВЫЗВАТЬ) (формат 1)	1 МПС
фраза END-CALL (КОНЕЦ-ВЫЗВАТЬ) (формат 2)	2 МПС
Оператор CANCEL (ОСВОБОДИТЬ)	2 МПС
литерал	2 МПС
идентификатор	2 МПС

GOST EXPERT ГОСТ 22558-89,

Элемент	Модуль
Оператор CLOSE (ЗАКРЫТЬ)	1 ПОД
	1 ОТД
	1 ИПД
	1 ГОТ
имя-файла	1 ПОД
	1 ОТД
	1 ИПД
	1 ГОТ
несколько имен-файлов	1 ПОД
	1 ОТД
	1 ИПД
	1 ГОТ
фраза REEL/UNIT (КАТУШКУ/ТОМ)	1 ПОД
	1 ГОТ
фраза FOR REMOVAL (С УДАЛЕНИЕМ)	2 ПОД
фраза WITH NO REWIND (БЕЗ ПЕРЕМОТКИ)	2 ПОД
	1 ГОТ
фраза WITH LOCK (С ЗАМКОМ)	2 ПОД
	2 ОТД
	2 ИПД
	1 ГОТ
Оператор COMPUTE (ВЫЧИСЛИТЬ)	2 ЯДР
арифметическое выражение	2 ЯДР
несколько идентификаторов	2 ЯДР
фраза ROUNDED (ОКРУГЛЯЯ)	2 ЯДР
фраза ON SIZE ERROR (ПРИ ПЕРЕПОЛНЕНИИ)	2 ЯДР
фраза NOT ON SIZE ERROR (БЕЗ ПЕРЕПОЛНЕНИЯ)	2 ЯДР
фраза END-COMPUTE (КОНЕЦ-ВЫЧИСЛИТЬ)	2 ЯДР
Оператор CONTINUE (ПРОДОЛЖИТЬ)	1 ЯДР
Оператор DELETE (УДАЛИТЬ)	1 ОТД
	1 ИПД
фраза INVALID KEY (ПРИ ОШИБКЕ КЛЮЧА)	1 ОТД
	1 ИПД
фраза NOT INVALID KEY (БЕЗ ОШИБКИ КЛЮЧА)	1 ОТД
	1 ИПД
фраза END-DELETE (КОНЕЦ-УДАЛИТЬ)	1 ОТД
	1 ИПД
Оператор DISABLE (ЗАПРЕТИТЬ)	2 КОМ
фраза INPUT (ВВОД)	2 КОМ
фраза TERMINAL (С ТЕРМИНАЛА)	2 КОМ
фраза I-O-TERMINAL (ВВОД-ВЫВОД С ТЕРМИНАЛА)	2 КОМ
фраза OUTPUT (ВЫВОД)	2 КОМ
фраза WITH KEY (КЛЮЧ)	2 КОМ +
Оператор DISPLAY (ВЫДАТЬ)	1 ЯДР
только одна передача данных	1 ЯДР
ограничений на число передач нет	2 ЯДР
идентификатор/литерал	1 ЯДР
несколько идентификаторов/литералов	1 ЯДР
фраза UPON мнемоническое-имя (НА мнемоническое-имя)	2 ЯДР
фраза WITH NO ADVANCING (БЕЗ ПРОДВИЖЕНИЯ)	2 ЯДР
Оператор DIVIDE (РАЗДЕЛИТЬ)	1 ЯДР
BY идентификатор/литерал (НА идентификатор/литерал)	1 ЯДР
GIVING идентификатор (ПОЛУЧАЯ идентификатор)	1 ЯДР

GOST EXPERT ГОСТ 22558-89,

Элемент	Модуль	
GIVING несколько-идентификаторов (ПОЛУЧАЯ несколько-идентификаторов)	1 ЯДР	
фраза ROUNDED (ОКРУГЛЯЯ)	1 ЯДР	
фраза REMAINDER (ОСТАТОК)	2 ЯДР	
фраза ON SIZE ERROR (ПРИ ПЕРЕПОЛНЕНИИ)	1 ЯДР	
фраза NOT ON SIZE ERROR (БЕЗ ПЕРЕПОЛНЕНИЯ)	1 ЯДР	
фраза END-DIVIDE (КОНЕЦ-РАЗДЕЛИТЬ)	1 ЯДР	
Оператор ENABLE (РАЗРЕШИТЬ)	2 КОМ	
фраза INPUT (ВВОД)	2 КОМ	
фраза TERMINAL (С ТЕРМИНАЛА)	2 КОМ	
фраза I-O-TERMINAL (ВВОД-ВЫВОД С ТЕРМИНАЛА)	2 КОМ	
фраза OUTPUT (ВЫВОД)	2 КОМ	
фраза WITH KEY (КЛЮЧ)	2 КОМ	+
Оператор ENTER (ВОЙТИ)	1 ЯДР	+
Оператор EVALUATE (ОЦЕНИТЬ)	2 ЯДР	
идентификатор/литерал	2 ЯДР	
арифметическое выражение	2 ЯДР	
условное выражение	2 ЯДР	
TRUE/FALSE (ИСТИНА/ЛОЖЬ)	2 ЯДР	
фраза ALSO (ТАКЖЕ)	2 ЯДР	
фраза WHEN (КОГДА)	2 ЯДР	
фраза ALSO (ТАКЖЕ)	2 ЯДР	
фраза WHEN OTHER (ИНАЧЕ)	2 ЯДР	
фраза END-EVALUATE (КОНЕЦ-ОЦЕНИТЬ)	2 ЯДР	
Оператор EXIT (ВЫЙТИ)	1 ЯДР	
Оператор EXIT PROGRAM (ВЫЙТИ ИЗ ПРОГРАММЫ)	1 МПС	
Оператор GENERATE (ГЕНЕРИРОВАТЬ)	1 ГОТ	
имя-данного	1 ГОТ	
имя-отчета	1 ГОТ	
Оператор GO TO (ПЕРЕЙТИ)	1 ЯДР	
имя-процедуры обязательно	1 ЯДР	
имя-процедуры необязательно	2 ЯДР	+
фраза DEPENDING ON (В ЗАВИСИМОСТИ ОТ)	1 ЯДР	
Оператор IF (ЕСЛИ)	1 ЯДР	
только повелительные операторы	1 ЯДР	
повелительные и (или) условные операторы	2 ЯДР	
вложенные операторы IF (ЕСЛИ)	1 ЯДР	
необязательное слово THEN (ТО)	1 ЯДР	
фраза NEXT SENTENCE (СЛЕДУЮЩЕЕ ПРЕДЛОЖЕНИЕ)	1 ЯДР	
фраза ELSE (ИНАЧЕ)	1 ЯДР	
фраза END-IF (КОНЕЦ-ЕСЛИ)	1 ЯДР	
Оператор INITIALIZE (ИНИЦИИРОВАТЬ)	2 ЯДР	
несколько идентификаторов	2 ЯДР	
фраза REPLACING (ЗАМЕНЯЯ)	2 ЯДР	
несколько фраз REPLACING (ЗАМЕНЯЯ)	2 ЯДР	
Оператор INITIATE (НАЧАТЬ)	1 ГОТ	
Оператор INSPECT (ПРОСМОТРЕТЬ)	1 ЯДР	
на вхождение одной литеры	1 ЯДР	
на вхождение нескольких литер	2 ЯДР	
фраза TALLYING (СЧИТАЯ)	1 ЯДР	
фраза BEFORE/AFTER (ДО/ПОСЛЕ)	1 ЯДР	

GOST EXPERT ГОСТ 22558-89,

Элемент	Модуль
несколько фраз BEFORE/AFTER (ДО/ПОСЛЕ)	2 ЯДР
несколько фраз TALLYING (СЧИТАЯ)	2 ЯДР
фраза REPLACING (ЗАМЕНЯЯ)	1 ЯДР
фраза BEFORE/AFTER (ДО/ПОСЛЕ)	1 ЯДР
несколько фраз BEFORE/AFTER (ДО/ПОСЛЕ)	2 ЯДР
несколько фраз REPLACING (ЗАМЕНЯЯ)	2 ЯДР
фразы TALLYING (СЧИТАЯ) и REPLACING (ЗАМЕНЯЯ)	1 ЯДР
фраза CONVERTING (ПРЕВРАЩАЯ)	2 ЯДР
Оператор MERGE (СЛИТЬ)	1 СРТ
фраза ASCENDING/DESCENDING KEY (ПО ВОЗРАСТАНИЮ/УБЫВАНИЮ КЛЮЧА)	1 СРТ
фраза COLLATING SEQUENCE (АЛФАВИТ)	1 СРТ
фраза USING (ИСПОЛЬЗУЯ)	1 СРТ
фраза OUTPUT PROCEDURE (ПРОЦЕДУРА ВЫВОДА)	1 СРТ
имя-процедуры	1 СРТ
фраза GIVING (ПОЛУЧАЯ)	1 СРТ
Оператор MOVE (ПОМЕСТИТЬ)	1 ЯДР
ТО идентификатор (В идентификатор)	1 ЯДР
ТО несколько идентификаторов (В несколько идентификаторов)	1 ЯДР
фраза CORRESPONDING (СООТВЕТСТВЕННО)	2 ЯДР
дередактирование цифровых редактируемых данных	2 ЯДР
Оператор MULTIPLY (УМНОЖИТЬ)	1 ЯДР
BY идентификатор (НА идентификатор)	1 ЯДР
BY несколько идентификаторов (НА несколько идентификаторов)	1 ЯДР
GIVING идентификатор (ПОЛУЧАЯ идентификатор)	1 ЯДР
GIVING несколько идентификаторов (ПОЛУЧАЯ несколько идентификаторов)	1 ЯДР
фраза ROUNDED (ОКРУГЛЯЯ)	1 ЯДР
фраза ON SIZE ERROR (ПРИ ПЕРЕПОЛНЕНИИ)	1 ЯДР
фраза NOT ON SIZE ERROR (БЕЗ ПЕРЕПОЛНЕНИЯ)	1 ЯДР
фраза END-MULTIPLY (КОНЕЦ-УМНОЖИТЬ)	1 ЯДР
Оператор OPEN (ОТКРЫТЬ)	1 ПОД
	1 ОТД
	1 ИПД
	1 ГОТ
имя-файла	1 ПОД
	1 ОТД
	1 ИПД
	1 ГОТ
несколько имен-файлов	1 ПОД
	1 ОТД
	1 ИПД
	1 ГОТ
фраза INPUT (ВХОДНОЙ)	1 ПОД
	1 ОТД
	1 ИПД
фраза WITH NO REWIND (БЕЗ ПЕРЕМОТКИ)	2 ПОД
фраза REVERSED (РЕВЕРСНО)	2 ПОД +

GOST EXPERT ГОСТ 22558-89,

фраза OUTPUT (ВЫХОДНОЙ)	1 ПОД 1 ОТД 1 ИПД 1 ГОТ
WITH NO REWIND (БЕЗ ПЕРЕМОТКИ)	2 ПОД 1 ГОТ
фраза I-O (ВХОДНОЙ-ВЫХОДНОЙ)	1 ПОД 1 ОТД 1 ИПД
фраза EXTEND (ДОПОЛНЯЕМЫЙ)	2 ПОД 2 ОТД 2 ИПД 1 ГОТ
несколько фраз INPUT (ВХОДНОЙ), OUTPUT (ВЫХОДНОЙ), I-O (ВХОДНОЙ-ВЫХОДНОЙ)	1 ПОД 1 ОТД 1 ИПД
несколько фраз EXTEND (ДОПОЛНЯЕМЫЙ)	2 ПОД 2 ОТД 2 ИПД
Оператор PERFORM (ВЫПОЛНИТЬ)	1 ЯДР
имя-процедуры необязательно	1 ЯДР
фраза THROUGH имя-процедуры (ПО имя-процедуры)	1 ЯДР
вариант повелительный-оператор	1 ЯДР
фраза END-PERFORM (КОНЕЦ-ВЫПОЛНИТЬ)	1 ЯДР
фраза TIMES (РАЗ)	1 ЯДР
фраза UNTIL (ДО)	1 ЯДР
фраза TEST BEFORE/AFTER (С ПРОВЕРКОЙ В НАЧАЛЕ/ В КОНЦЕ)	2 ЯДР
фраза VARYING (МЕНЯЯ)	2 ЯДР
фраза TEST BEFORE/AFTER (С ПРОВЕРКОЙ В НАЧАЛЕ/ В КОНЦЕ)	2 ЯДР
фраза AFTER (ЗАТЕМ)	2 ЯДР
допускаются по крайней мере 6 фраз AFTER (ЗАТЕМ)	2 ЯДР
Оператор PURGE (ОЧИСТИТЬ)	2 КОМ
Оператор READ (ЧИТАТЬ)	1 ПОД 1 ОТД 1 ИПД
фраза NEXT (СЛЕДУЮЩУЮ)	2 ПОД 2 ОТД 2 ИПД
фраза INTO (В)	1 ПОД 1 ОТД 1 ИПД
фраза AT END (В КОНЦЕ)	1 ПОД 1 ОТД 1 ИПД
фраза NOT AT END (НЕ В КОНЦЕ)	1 ПОД 1 ОТД 1 ИПД
фраза KEY (КЛЮЧ)	2 ИПД
фраза INVALID KEY (ПРИ ОШИБКЕ КЛЮЧА)	1 ОТД 1 ИПД
фраза NOT INVALID KEY (БЕЗ ОШИБКИ КЛЮЧА)	1 ОТД 1 ИПД

GOST EXPERT ГОСТ 22558-89,

Элемент	Модуль
фраза END-READ (КОНЕЦ-ЧИТАТЬ)	1 ПОД
	1 ОТД
	1 ИПД
Оператор RECEIVE (ПОЛУЧИТЬ)	1 КОМ
фраза MESSAGE (СООБЩЕНИЕ)	1 КОМ
фраза SEGMENT (СЕГМЕНТ)	1 КОМ
фраза INTO идентификатор (В идентификатор)	1 КОМ
фраза NO DATA (НЕТ ДАННЫХ)	1 КОМ
фраза WITH DATA (ЕСТЬ ДАННЫЕ)	1 КОМ
фраза END-RECEIVE (КОНЕЦ-ПОЛУЧИТЬ)	1 КОМ
Оператор RELEASE (ПЕРЕДАТЬ)	1 СРТ
фраза FROM (ИЗ ПОЛЯ)	1 СРТ
Оператор RETURN (ВЕРНУТЬ)	1 СРТ
фраза INTO (В)	1 СРТ
фраза AT END (В КОНЦЕ)	1 СРТ
фраза NOT AT END (НЕ В КОНЦЕ)	1 СРТ
фраза END-RETURN (КОНЕЦ-ВЕРНУТЬ)	1 СРТ
Оператор REWRITE (ОБНОВИТЬ)	1 ПОД
	1 ОТД
	1 ИПД
фраза FROM (ИЗ ПОЛЯ)	1 ПОД
	1 ОТД
	1 ИПД
фраза INVALID KEY (ПРИ ОШИБКЕ КЛЮЧА)	1 ОТД
	1 ИПД
фраза NOT INVALID KEY (БЕЗ ОШИБКИ КЛЮЧА)	1 ОТД
	1 ИПД
фраза END-REWRITE (КОНЕЦ-ОБНОВИТЬ)	1 ОТД
	1 ИПД
Оператор SEARCH (ИСКАТЬ)	2 ЯДР
фраза VARYING (МЕНЯЯ)	2 ЯДР
фраза AT END (В КОНЦЕ)	2 ЯДР
фраза WHEN (КОГДА)	2 ЯДР
несколько фраз WHEN (КОГДА)	2 ЯДР
фраза END-SEARCH (КОНЕЦ-ИСКАТЬ)	2 ЯДР
Оператор SEARCH ALL (ИСКАТЬ ОСОБО)	2 ЯДР
фраза AT END (В КОНЦЕ)	2 ЯДР
фраза WHEN (КОГДА)	2 ЯДР
фраза END-SEARCH (КОНЕЦ-ИСКАТЬ)	2 ЯДР
Оператор SEND (ПОСЛАТЬ)	1 КОМ
фраза FROM идентификатор (ИЗ ПОЛЯ идентификатор) (часть сообщения)	2 КОМ
фраза FROM идентификатор (ИЗ ПОЛЯ идентификатор) (полное сообщение)	1 КОМ
фраза WITH идентификатор (С идентификатор)	2 КОМ
фраза WITH ESI (С ИКС)	2 КОМ
фраза WITH EMI (С ИКЩ)	2 КОМ
фраза WITH EGI (С ИКГ)	2 КОМ
фраза BEFORE/AFTER ADVANCING (ДО/ПОСЛЕ ПРОДВИЖЕНИЯ)	1 КОМ
целое LINE/LINES (целое СТРОК)	1 КОМ
идентификатор LINE/LINES (идентификатор СТРОК)	1 КОМ

GOST EXPERT ГОСТ 22558-89,

Элемент	Модуль
мнемоническое-имя	2 КОМ
PAGE (СТРАНИЦЫ)	1 КОМ
фраза REPLACING LINE (ЗАМЕНЯЯ СТРОКУ)	2 КОМ
Оператор SET (УСТАНОВИТЬ)	1 ЯДР
имя-индекса/идентификатор TO (НА)	1 ЯДР
имя-индекса UP BY/DOWN BY (имя-индекса ПРИБАВЛЯЯ/ВЫЧИТАЯ)	1 ЯДР
мнемоническое-имя TO ON/OFF (мнемоническое-имя НА ВКЛЮЧЕНО/ВЫКЛЮЧЕНО)	1 ЯДР
имя-условия TO TRUE (имя-условия НА ИСТИНА)	2 ЯДР
Оператор SORT (СОРТИРОВАТЬ)	1 СРТ
фраза ASCENDING/DESCENDING KEY (ПО ВОЗРАСТАНИЮ/УБЫВАНИЮ КЛЮЧА)	1 СРТ
фраза DUPLICATES (С ДУБЛИРОВАНИЕМ)	1 СРТ
фраза COLLATING SEQUENCE (АЛФАВИТ)	1 СРТ
фраза INPUT PROCEDURE (ПРОЦЕДУРА ВВОДА)	1 СРТ
имя-процедуры	1 СРТ
фраза USING (ИСПОЛЬЗУЯ)	1 СРТ
фраза OUTPUT PROCEDURE (ПРОЦЕДУРА ВЫВОДА)	1 СРТ
имя-процедуры	1 СРТ
фраза GIVING (ПОЛУЧАЯ)	1 СРТ
Оператор START (ПОДВЕСТИ)	2 ОТД 2 ИПД
фраза KEY (КЛЮЧ)	2 ОТД 2 ИПД
EQUAL TO (РАВНО)	2 ОТД 2 ИПД
=	2 ОТД 2 ИПД
GREATER THAN (БОЛЬШЕ)	2 ОТД 2 ИПД
>	2 ОТД 2 ИПД
NOT LESS THAN (НЕ МЕНЬШЕ)	2 ОТД 2 ИПД
NOT < (НЕ <)	2 ОТД 2 ИПД
GREATER THAN OR EQUAL TO (БОЛЬШЕ ИЛИ РАВНО)	2 ОТД 2 ИПД
>=	2 ОТД 2 ИПД
фраза INVALID KEY (ПРИ ОШИБКЕ КЛЮЧА)	2 ОТД 2 ИПД
фраза NOT INVALID KEY (БЕЗ ОШИБКИ КЛЮЧА)	2 ОТД 2 ИПД
фраза END-START (КОНЕЦ-ПОДВЕСТИ)	2 ОТД 2 ИПД
Оператор STOP (ОСТАНОВИТЬ)	1 ЯДР
STOP (РАБОТУ)	1 ЯДР
литерал	1 ЯДР +
Оператор STRING (СОБРАТЬ)	2 ЯДР
несколько DELIMITED BY (ОГРАНИЧИВАЯСЬ)	2 ЯДР

GOST EXPERT ГОСТ 22558-89,

Элемент	Модуль
фраза WITH POINTER (УКАЗАТЕЛЬ)	2 ЯДР
фраза ON OVERFLOW (ПРИ ПЕРЕПОЛНЕНИИ)	2 ЯДР
фраза NOT ON OVERFLOW (БЕЗ ПЕРЕПОЛНЕНИЯ)	2 ЯДР
фраза END-STRING (КОНЕЦ-СОБРАТЬ)	2 ЯДР
Оператор SUBTRACT (ОТНЯТЬ)	1 ЯДР
идентификатор/литерал	1 ЯДР
несколько идентификаторов/литералов	1 ЯДР
FROM идентификатор (ОТ идентификатор)	1 ЯДР
FROM несколько идентификаторов (ОТ несколько идентификаторов)	1 ЯДР
GIVING идентификатор (ПОЛУЧАЯ идентификатор)	1 ЯДР
GIVING несколько идентификаторов (ПОЛУЧАЯ несколько идентификаторов)	1 ЯДР
фраза ROUNDED (ОКРУГЛЯЯ)	1 ЯДР
фраза ON SIZE ERROR (ПРИ ПЕРЕПОЛНЕНИИ)	1 ЯДР
фраза NOT ON SIZE ERROR (БЕЗ ПЕРЕПОЛНЕНИЯ)	1 ЯДР
фраза END-SUBTRACT (КОНЕЦ-ОТНЯТЬ)	1 ЯДР
фраза CORRESPONDING (СООТВЕТСТВЕННО)	2 ЯДР
Оператор SUPPRESS (ПОДАВИТЬ)	1 ГОТ
Оператор TERMINATE (ЗАКОНЧИТЬ)	1 ГОТ
Оператор UNSTRING (РАЗОБРАТЬ)	2 ЯДР
фраза DELIMITED BY (ОГРАНИЧИВАЯСЬ)	2 ЯДР
фраза DELIMITER IN (ОГРАНИЧИТЕЛЬ В)	2 ЯДР
фраза COUNT IN (СЧЕТ В)	2 ЯДР
фраза WITH POINTER (УКАЗАТЕЛЬ)	2 ЯДР
фраза TALLYING (СЧИТАЯ)	2 ЯДР
фраза ON OVERFLOW (ПРИ ПЕРЕПОЛНЕНИИ)	2 ЯДР
фраза NOT ON OVERFLOW (БЕЗ ПЕРЕПОЛНЕНИЯ)	2 ЯДР
фраза END-UNSTRING (КОНЕЦ-РАЗОБРАТЬ)	2 ЯДР
Оператор USE (ИСПОЛЬЗОВАТЬ)	1 ПОД
	1 ОТД
	1 ИПД
	1 ГОТ
	1 ОТЛ +
фраза EXCEPTION/ERROR PROCEDURE (ПРОЦЕДУРЫ ОШИБКИ)	1 ПОД
	1 ОТД
	1 ИПД
	1 ГОТ
фраза GLOBAL (ГЛОБАЛЬНО)	2 МПС
ON имя-файла (ДЛЯ имя-файла)	1 ПОД
	1 ОТД
	1 ИПД
	1 ГОТ
ON несколько имен-файлов (ДЛЯ несколько имен-файлов)	2 ПОД
	2 ОТД
	2 ИПД
	1 ГОТ
ON INPUT (ДЛЯ ВХОДНЫХ)	1 ПОД
	1 ОТД
	1 ИПД

GOST EXPERT ГОСТ 22558-89,

Элемент	Модуль
ON OUTPUT (ДЛЯ ВЫХОДНЫХ)	1 ПОД 1 ОТД 1 ИПД 1 ГОТ
ON I-O (ДЛЯ ВХОДНЫХ-ВЫХОДНЫХ)	1 ПОД 1 ОТД 1 ИПД
ON EXTEND (ДЛЯ ДОПОЛНЯЕМЫХ)	2 ПОД 2 ОТД 2 ИПД 1 ГОТ
фраза BEFORE REPORTING (ДО ВЫДАЧИ)	1 ГОТ
фраза GLOBAL (ГЛОБАЛЬНО)	2 МНС
фраза FOR DEBUGGING (ДЛЯ ОТЛАДКИ)	1 ОТЛ +
имя-процедуры	1 ОТЛ +
ALL PROCEDURES (ПРИ ВСЕХ ПРОЦЕДУРАХ)	1 ОТЛ +
ALL REFERENCES OF идентификатор-1 (ПРИ ВСЕХ ССЫЛКАХ НА идентификатор-1)	2 ОТЛ +
имя-коммуникации	2 ОТЛ +
имя-файла	2 ОТЛ +
Оператор WRITE (ПИСАТЬ)	1 ПОД 1 ОТД 1 ИПД
фраза FROM (ИЗ ПОЛЯ)	1 ПОД 1 ОТД 1 ИПД
фраза BEFORE/AFTER ADVANCING (ДО/ПОСЛЕ ПРОДВИЖЕНИЯ)	1 ПОД
целое LINE/LINES (целое СТРОК)	1 ПОД
идентификатор LINE/LINES (идентификатор СТРОК)	1 ПОД
мнемоническое-имя	2 ПОД
PAGE (СТРАНИЦЫ)	1 ПОД
фраза AT END-OF-PAGE/EOP (В КОНЦЕ СТРАНИЦЫ)	2 ПОД
фраза NOT AT END-OF-PAGE/EOP (НЕ В КОНЦЕ СТРАНИЦЫ)	2 ПОД
фраза INVALID KEY (ПРИ ОШИБКЕ КЛЮЧА)	1 ОТД 1 ИПД
фраза NOT INVALID KEY (БЕЗ ОШИБКИ КЛЮЧА)	1 ОТД 1 ИПД
фраза END-WRITE (КОНЕЦ-ПИСАТЬ)	2 ПОД 2 ОТД 2 ИПД
Сегментация	
Номера-сегментов от 0 до 49 для постоянных сегментов	1 СЕГ +
Номера-сегментов от 50 до 99 для независимых сегментов	1 СЕГ +
Все секции, имеющие одинаковый номер-сегмента, должны быть смежными в исходной программе	1 СЕГ +
Секции, имеющие одинаковый номер-сегмента, могут быть несмежными в исходной программе	2 СЕГ +
Обработка исходных текстов в разделе процедур	
Оператор COPY (КОПИРОВАТЬ)	1 ОИТ

GOST EXPERT ГОСТ 22558-89,

Элемент	Модуль
фраза OF/IN имя-библиотеки (ИЗ имя-библиотеки)	2 ОИТ
фраза REPLACING (ЗАМЕНЯЯ)	2 ОИТ
псевдотекст	2 ОИТ
идентификатор	2 ОИТ
литерал	2 ОИТ
слово	2 ОИТ
Оператор REPLACE (ЗАМЕНИТЬ)	2 ОИТ
псевдотекст BY псевдотекст (псевдотекст НА псевдотекст)	2 ОИТ
OFF (ОТКЛЮЧИТЬ)	2 ОИТ

Часть 2. КОНЦЕПЦИИ И СРЕДСТВА ЯЗЫКА

1. ВВЕДЕНИЕ

Кобол предлагает множество средств, позволяющих пользователю применять необходимую функцию, не программируя ее в деталях. Ниже обсуждается каждое из этих средств, рассматриваются доводы для включения его в язык и смысл его использования и организации.

2. ФАЙЛЫ

Файл — это совокупность записей, которые могут быть помещены в запоминающую среду или извлечены из нее. Пользователь выбирает не только организацию файла, но выбирает также метод и последовательность обработки. Для последовательной запоминающей среды организация файла и метод обработки ограничены, для массовой памяти таких ограничений нет.

Используются следующие соглашения при описании возможностей Кобол-программ для манипулирования файлами. Термин «имя-файла» обозначает определенное пользователем слово, используемое в исходной Кобол-программе для ссылки на файл. Термины «файл, на который ссылается имя-файла» и «файл» обозначают физический файл независимо от имени-файла, используемого в Кобол-программе. Термин «определитель файла» обозначает объект, содержащий информацию относительно файла. Все доступы к физическим файлам осуществляются через определитель файла. В различных реализациях определитель файла может быть таблицей информации о файле, блоком управления файлом и т. п.

2.1. Свойства файла

Файл имеет несколько свойств, которые придаются файлу при его создании и не могут изменяться за время существования файла. Основным свойством является организация файла, описыва-

GOST EXPERT ГОСТ 22558-89,

ющая его логическую структуру. Имеются три организации: последовательная, относительная и индексная. Другими фиксированными свойствами файла, обеспечиваемыми Кобол-программой, являются основной ключ записи, дополнительные ключи записи, набор кодов, минимальный и максимальный размер логической записи, тип записи (фиксированной или переменной длины), основная последовательность для ключей индексных файлов, фактор блокирования, литера заполнитель и ограничитель записи.

2.1.1. Последовательная организация

Последовательные файлы организованы так, что каждая запись, за исключением последней, имеет единственную последующую запись; каждая запись, за исключением первой, имеет единственную предшествующую запись. Отношение «предшественник-преемник» устанавливается порядком выполнения оператора WRITE (ПИСАТЬ) при создании файла. Установленный порядок предшествования не изменяется за исключением случая, когда записи добавляются в конец файла.

Последовательно организованный файл массовой памяти имеет такую же логическую структуру, как и файл любой последовательной запоминающей среды; тем не менее последовательный файл массовой памяти может быть обновлен на месте. При использовании этой техники новые записи не могут быть добавлены в файл и каждая обновленная запись должна быть одного размера с записью-оригиналом.

2.1.2. Относительная организация

Файл с относительной организацией — это файл массовой памяти, доступ к записям которого осуществляется по значению относительного номера записи.

Концептуально файл с относительной организацией состоит из последовательной цепочки областей, каждая из которых может содержать логическую запись. Каждая из этих областей обозначена относительным номером записи. Каждая логическая запись в относительном файле идентифицируется относительным номером записи для ее области памяти. Например, десятая запись — это запись, адресованная относительным номером записи 10 и находящаяся в десятой области записи независимо от того, занесены или не занесены записи в области записи от первой до девятой.

С целью более эффективного доступа к записям в относительном файле количество позиций литер, резервируемых в запоминающей среде для хранения отдельной логической записи, может отличаться от количества позиций литер, указанного в описании этой записи в программе.

2.1.3. Индексная организация

Файл с индексной организацией — это файл массовой памяти, доступ к записям которого осуществляется заданием значения оп-

GOST EXPERT ГОСТ 22558-89,

ределенного ключа в этой записи. Для каждого ключа, определенного для записей файла, поддерживается индекс. Каждый такой индекс представляет набор значений соответствующего ключа в каждой записи. Кроме того, каждый индекс является механизмом, обеспечивающим доступ к любой записи файла.

Каждый индексный файл имеет основной индекс, представляющий основной ключ записи каждой записи в файле. Каждая запись помещается в файл, изменяется или удаляется из файла только на основании значения ее основного ключа записи. Основной ключ каждой записи в файле должен быть однозначным (уникальным) и не должен изменяться при обновлении записи. Основной ключ записи объявляется в фразе RECORD KEY (КЛЮЧ ЗАПИСИ) статьи управления файлом для данного файла.

Дополнительные ключи записи обеспечивают дополнительные способы извлечения записей файла. Такие ключи называются в фразах ALTERNATE RECORD KEY (ДОПОЛНИТЕЛЬНЫЙ КЛЮЧ ЗАПИСИ) статьи управления файлом. Значение отдельного дополнительного ключа в каждой записи не обязательно должно быть однозначным. В случае, когда значения могут быть неоднозначными, в фразе ALTERNATE RECORD KEY (ДОПОЛНИТЕЛЬНЫЙ КЛЮЧ ЗАПИСИ) должен быть указан вариант DUPLICATES (С ДУБЛИРОВАНИЕМ).

2.1.4. Логические записи

Логическая запись — это порция данных, которую можно поместить в файл или извлечь из него. Количество записей, содержащихся в файле, ограничивается только возможностью запоминающей среды. Имеются два типа записей: запись фиксированной длины и запись переменной длины. При создании файла объявляется, какой тип записей он будет содержать. В любом случае содержимое области записи не отражает никакой информации, которую реализация может добавлять к записи в физической памяти (например, заголовок длины записи); длина записи, используемая программистом в Коболе, также не отражает таких добавлений.

2.1.4.1. *Записи фиксированной длины*

Записи фиксированной длины должны содержать одно и то же количество позиций литер во всех записях файла. Все операции ввода-вывода могут обрабатывать записи только такого размера. Записи фиксированной длины можно определить явно форматом 1 фразы RECORD (В ЗАПИСИ) статьи описания файла независимо от индивидуальных описаний записи.

2.1.4.2. *Записи переменной длины*

Записи переменной длины могут содержать различное количество позиций литер в записях файла. Для явного определения записей переменной длины можно указать вариант VARYING (ПЕРЕМЕННОЕ ЧИСЛО ЛИТЕР) фразы RECORD (В ЗАПИСИ)

GOST EXPERT ГОСТ 22558-89,

статьи описания файла. Длина записи зависит от данного, на которое имеется ссылка в варианте DEPENDING ON (В ЗАВИСИМОСТИ ОТ) фразы RECORD (В ЗАПИСИ) или в варианте DEPENDING ON (В ЗАВИСИМОСТИ ОТ) фразы OCCURS (ПОВТОРЯЕТСЯ), или от длины записи статьи описания записи в файле.

2.1.4.3. *Типы записей, определяемые реализацией*

Если в статье описания файла не указана фраза RECORD (В ЗАПИСИ) для файла или если фраза RECORD (В ЗАПИСИ) определяет диапазон позиций литер, реализация определяет, какой получается тип записей — переменной длины или фиксированной длины.

2.2. **Обработка файлов**

Файл может обрабатываться выполнением операций как над отдельными записями, так и над файлом как целым. Особые условия, которые встречаются во время обработки, сообщаются программе.

2.2.1. Операции над записями

Фраза ACCESS (ДОСТУП) статьи описания файла указывает способ, которым объектная программа оперирует записями в файле. Доступ может быть последовательным, произвольным и динамическим.

К файлам с относительной и индексной организацией возможен любой из методов доступа к файлу независимо от метода доступа, используемого при создании файла. К файлу с последовательной организацией возможен только последовательный доступ.

Для выходного отчета его содержимое, организация и формат могут быть определены при использовании средств генератора отчетов (ч. 2, п. 3.1).

2.2.1.1. *Последовательный доступ*

Последовательный доступ возможен для любых файлов, безотносительно к организации файла.

При последовательной организации порядок последовательного доступа является порядком, в котором записи заносились в файл.

При относительной организации порядок последовательного доступа возрастающий по значению относительных номеров записи. Доступны только записи, существующие в текущий момент в файле. Для установления начальной точки следующих друг за другом последовательных извлечений может быть использован оператор START (ПОДВЕСТИ).

При индексной организации порядок последовательного доступа возрастающий по значению ключа ссылки в соответствии с основной последовательностью, связанной с внутренним набором литер. Каждый из ключей, связанных с файлом, может быть уста-

GOST EXPERT ГОСТ 22558-89,

новлен в качестве ключа ссылки во время обработки файла. Порядок извлечения из набора записей, имеющих дублирующиеся значения ключа ссылки, является первоначальным порядком занесения этих записей в набор. Для установления начальной точки в индексном файле для ряда следующих друг за другом последовательных извлечений может быть использован оператор START (ПОДВЕСТИ).

2.2.1.2. *Произвольный доступ*

При произвольном доступе к файлу операторы ввода-вывода используются для доступа к записям в порядке, определенном пользователем. Произвольный доступ может применяться только к файлам с относительной или индексной организацией.

Для файла с относительной организацией программист указывает требуемую запись, помещая ее относительный номер записи в поле данного относительного ключа. При индексной организации программист указывает требуемую запись, помещая значение одного из ключей записи в поле данного основного ключа записи или дополнительного ключа записи.

2.2.1.3. *Динамический доступ*

При динамическом доступе программист может переходить от последовательного доступа к произвольному, используя соответствующие формы операторов ввода-вывода. Динамический доступ может использоваться только для файлов с относительной и индексной организацией.

2.2.1.4. *Режим открытия*

Режим открытия файла связан с действиями, которые должны быть выполнены для записей файла. Режимы открытия и их назначение следующие:

для ввода — для извлечения записей;

для вывода — для занесения записей в файл;

для дополнения — для добавления записей к существующему файлу;

для ввода-вывода — для извлечения и обновления записей.

Режим открытия определяется оператором OPEN (ОТКРЫТЬ).

Если файл открыт для ввода, возможен доступ к файлу по оператору READ (ЧИТАТЬ). Для файлов с индексной или относительной организацией, для которых указан последовательный или динамический доступ, может быть также использован оператор START (ПОДВЕСТИ).

Если файл открыт для вывода, записи помещаются в файл выполнением операторов GENERATE (ГЕНЕРИРОВАТЬ), TERMINATE (ЗАКОНЧИТЬ) или WRITE (ПИСАТЬ).

Если файл открыт для дополнения, новые записи добавляются в логический конец файла выполнением операторов GENERATE

GOST EXPERT ГОСТ 22558-89,

(ГЕНЕРИРОВАТЬ), TERMINATE (ЗАКОНЧИТЬ) или WRITE (ПИСАТЬ).

В режиме открытия для ввода-вывода можно обращаться только к файлам массовой памяти. Дополнительные возможности устройств массовой памяти позволяют обновление записей на месте, таким образом, всегда могут быть использованы операторы READ (ЧИТАТЬ) и REWRITE (ОБНОВИТЬ). Файлы массовой памяти могут быть обновлены таким же образом, как и файлы в последовательной запоминающей среде, перезаписыванием файла полностью в другой файл (возможно в другой участок массовой памяти), используя операторы READ (ЧИТАТЬ) и WRITE (ПИСАТЬ). Тем не менее, иногда более эффективно обновлять файлы массовой памяти на месте. При такой методике обработки файла массовой памяти оператор REWRITE (ОБНОВИТЬ) используется для возврата на то же место запоминающей среды только тех записей, которые были изменены. Операторы READ (ЧИТАТЬ) и REWRITE (ОБНОВИТЬ) являются единственными операторами, использование которых допускается во время обновления на месте последовательных файлов. Для индексных и относительных файлов применяются следующие дополнительные функции: оператор START (ПОДВЕСТИ) может быть использован при последовательном или динамическом доступе для изменения последовательности извлечения записей; оператор DELETE (УДАЛИТЬ) может быть использован при любом доступе для логического удаления записи из файла; оператор WRITE (ПИСАТЬ) может быть использован при произвольном или динамическом доступе для вставки новой записи в файл.

2.2.1.5. *Указатель текущего тома*

Указатель текущего тома — логическое понятие, используемое в этом документе для облегчения точного определения текущего физического тома последовательного файла. На состояние указателя текущего тома влияют операторы CLOSE (ЗАКРЫТЬ), OPEN (ОТКРЫТЬ), WRITE (ПИСАТЬ) и READ (ЧИТАТЬ).

2.2.1.6. *Индикатор позиции файла*

Индикатор позиции файла — логическое понятие, используемое в этом документе для облегчения точного определения следующей записи в данном файле, которая станет доступной в конкретной последовательности операций ввода-вывода. На установку индикатора позиции файла влияют только операторы START (ПОДВЕСТИ), OPEN (ОТКРЫТЬ) и READ (ЧИТАТЬ). Понятие индикатора позиции файла не имеет смысла для файлов, открытых для вывода или дополнения.

2.2.1.7. *Понятие верстки*

При определении выходного отчета может быть использована фраза LINAGE (ВЕРСТКА). Она облегчает определение логиче-

GOST EXPERT ГОСТ 22558-89,

ской страницы и размещение на логической странице верхнего и нижнего полей и области концовки. Использование фразы LINAGE (ВЕРСТКА) неявно определяет соответствующий специальный регистр LINAGE-COUNTER (СЧЕТЧИК-ВЕРСТКИ), который действует как указатель строки в теле страницы.

2.2.2. Операции над файлами

Несколько операторов Кобола обрабатывают файлы как единое целое или как набор записей. Это операторы CLOSE (ЗАКРЫТЬ), OPEN (ОТКРЫТЬ), MERGE (СЛИТЬ) и SORT (СОРТИРОВАТЬ).

2.2.2.1. *Сортировка и слияние*

2.2.2.1.1. Сортировка

При использовании сортировки бывает необходимо применить некоторую специальную обработку к содержимому сортируемого файла. Специальная обработка может заключаться в добавлении, удалении, создании, изменении, редактировании либо в других модификациях отдельных записей в файле. Применение специальной обработки может понадобиться до или после переупорядочения записей в результате сортировки, или же такая обработка может потребоваться и до, и после сортировки. Средства сортировки в Коболе позволяют пользователю записать эти процедуры и указать, в каком месте, до или после сортировки, они должны выполняться. Кобол-программа может содержать любое количество сортировок и каждая из них может иметь свои процедуры ввода и вывода. Средства сортировки автоматически приводят к выполнению этих процедур в указанном месте.

В процедуре ввода для создания сортируемого файла используется оператор RELEASE (ПЕРЕДАТЬ). Это значит, что по завершении выполнения процедуры ввода те записи, которые обработаны посредством оператора RELEASE (ПЕРЕДАТЬ) (а не оператором WRITE (ПИСАТЬ)), составляют сортируемый файл, доступ к которому возможен только по оператору SORT (СОРТИРОВАТЬ). Выполнение оператора SORT (СОРТИРОВАТЬ) упорядочивает весь набор записей в сортируемом файле соответственно ключам, указанным в операторе SORT (СОРТИРОВАТЬ).

К отсортированным записям сортируемого файла возможен доступ посредством оператора RETURN (ВЕРНУТЬ) во время выполнения процедуры вывода.

Для сортируемого файла нет процедур обработки меток, которыми мог бы управлять программист, и правила блокирования и распределения внутренней памяти определяются оператором SORT (СОРТИРОВАТЬ). Операторы RELEASE (ПЕРЕДАТЬ) и RETURN (ВЕРНУТЬ) ничего не предполагают относительно буферных полей, блоков или катушек. Следовательно, сортируемый файл можно рассматривать как внутренний файл, созданный (опе-

GOST EXPERT ГОСТ 22558-89,

ратором RELEASE (ПЕРЕДАТЬ)) из входного файла, обработанный (оператором SORT (СОРТИРОВАТЬ)) и затем доступный (оператором RETURN (ВЕРНУТЬ)) выходному файлу. Обращение и доступ к самому сортируемому файлу возможны только в операторе SORT (СОРТИРОВАТЬ). Описание сортируемого-сливаемого файла можно рассматривать как особый тип описания файла. То есть, сортируемый файл, как и любой файл, является набором записей.

2.2.2.1.2. Слияние

При использовании слияния бывает необходимо применить специальную обработку содержимого сливаемого файла. Специальная обработка может заключаться в добавлении, удалении, изменении, редактировании либо каких-либо других модификациях отдельных записей в файле. Средства слияния в Коболе позволяют пользователю задать процедуру вывода, которая должна выполняться при создании выходного результата слияния. Сливаемые записи становятся доступными из сливаемого файла при использовании оператора RETURN (ВЕРНУТЬ) в процедуре вывода.

Для сливаемого файла нет процедур обработки меток, которыми мог бы управлять программист, и правила блокирования и распределения внутренней памяти определяются оператором MERGE (СЛИТЬ). Оператор RETURN (ВЕРНУТЬ) ничего не предполагает относительно буферных полей, блоков или катушек.

Сливаемый файл, таким образом, можно рассматривать как внутренний файл, созданный из входных файлов их комбинацией (оператором MERGE (СЛИТЬ)) и доступный затем (по оператору RETURN (ВЕРНУТЬ)) выходному файлу. Обращение и доступ к самому сливаемому файлу возможны только в операторе MERGE (СЛИТЬ). Описание сортируемого-сливаемого файла можно рассматривать как особый тип описания файла. То есть, сливаемый файл, как и любой файл, является набором записей.

2.2.3. Обработка особых ситуаций

Во время выполнения любой операции ввода или вывода могут возникнуть особые условия, препятствующие нормальному завершению операции. Имеются три метода сообщения объектной программе об этих условиях: ключи состояния, декларативы ошибок и необязательные фразы, связанные с повелительным оператором.

2.2.3.1. *Состояние ввода-вывода*

Состояние ввода-вывода — это логическое понятие, используемое в этом документе для облегчения точного определения состояния выполнения операции ввода-вывода.

На установку состояния ввода-вывода влияют только операторы CLOSE (ЗАКРЫТЬ), DELETE (УДАЛИТЬ), OPEN (ОТКРЫТЬ), READ (ЧИТАТЬ), REWRITE (ОБНОВИТЬ), START (ПОДВЕСТИ) и WRITE (ПИСАТЬ).

GOST EXPERT ГОСТ 22558-89,

Значение состояния ввода-вывода для данного файла доступно программе только посредством имени-данного, указанного в фразе FILE STATUS (СОСТОЯНИЕ ФАЙЛА) статьи управления файлом для этого файла. Значение состояния ввода-вывода помещается в это данное во время выполнения оператора ввода-вывода и до выполнения любого повелительного оператора, связанного с этим оператором ввода-вывода, или до выполнения декларативы ошибки.

2.2.3.2. *Декларативы ошибки*

Если для файла указана процедура USE AFTER EXCEPTION (ИСПОЛЬЗОВАТЬ ПОСЛЕ ОШИБКИ), процедура выполняется каждый раз, когда возникают условия ввода или вывода, приводящие к неуспешной операции ввода-вывода. Однако декларатива ошибки не выполняется, если имеется условие неверного ключа и указана фраза INVALID KEY (ПРИ ОШИБКЕ КЛЮЧА), или при наличии условия конца и указанной фразе AT END (В КОНЦЕ).

2.2.3.3. *Необязательные фразы*

Фразы INVALID KEY (ПРИ ОШИБКЕ КЛЮЧА) могут быть указаны в операторах DELETE (УДАЛИТЬ), READ (ЧИТАТЬ), REWRITE (ОБНОВИТЬ), START (ПОДВЕСТИ) или WRITE (ПИСАТЬ). Некоторые из условий, приводящие к условию ошибки ключа, возникают, когда запрашиваемый ключ не существует в файле (операторы DELETE (УДАЛИТЬ), READ (ЧИТАТЬ) или START (ПОДВЕСТИ)); когда ключ уже есть в файле и дублирования не разрешаются (оператор WRITE (ПИСАТЬ)); когда ключа в файле нет или когда ключ не является последним прочитанным ключом (оператор REWRITE (ОБНОВИТЬ)).

Если условие ошибки ключа возникает во время выполнения оператора, для которого указана фраза INVALID KEY (ПРИ ОШИБКЕ КЛЮЧА), выполняется оператор, определяемый этой фразой INVALID KEY (ПРИ ОШИБКЕ КЛЮЧА).

В операторе READ (ЧИТАТЬ) может быть указана фраза AT END (В КОНЦЕ). Условие «в конце» возникает:

в файле с последовательным доступом, когда в файле не существует следующей логической записи;

когда число значащих цифр в относительном номере записи больше размера данного — относительного ключа;

когда необязательный файл отсутствует;

при попытке выполнения оператора READ (ЧИТАТЬ) при уже существующем условии «в конце». Если во время выполнения оператора, для которого указана фраза AT END (В КОНЦЕ), возникает условие «в конце», выполняется оператор, определенный фразой AT END (В КОНЦЕ).

GOST EXPERT ГОСТ 22558-89,

3. ГЕНЕРАТОР ОТЧЕТОВ

Назначением генератора отчетов является организация, форматирование и представление содержания выдаваемых отчетов. Хотя отчет может быть составлен без использования этих средств, генератор отчетов обеспечивает более удобные возможности для построения и составления отчетов. Программирование процедур, которое обычно производит программист, автоматически обеспечивается системой управления генератором отчетов. Таким образом, программист избавлен от написания процедур перемещения данных, конструирования печатаемых строк, подсчета строк на странице, нумерации страниц, составления строк заголовка и строк концовки, распознавания конца логических подразделений данных, обновления счетчиков сумм и т. п. Все эти действия выполняются автоматически на основании спецификаций, главным образом, секции отчетов раздела данных исходной программы.

3.1. **Секция отчетов**

Секция отчетов в разделе данных содержит одну или несколько статей описания отчета RD (ОО), каждая из которых представляет полное описание отчета.

Отчет, имя которого указано в статье описания отчета, не относится прямо к выходному файлу. Вместо этого он связан с именем-файла в секции файлов и это имя-файла связывается с файлом, когда выполняется оператор OPEN (ОТКРЫТЬ), указывающий имя-файла. С одним и тем же именем файла может быть связано несколько отчетов и чтобы отличить отчеты друг от друга, используется фраза CODE (С КОДОМ). Для внешнего определителя файла, соотнесенного имени-файла, отдельно компилируемые программы могут указывать различные отчеты для одного и того же имени-файла. Статья описания файла, к которому относится отчет, не может содержать статей описания записей, описывающих записи данных. Эта статья описания файла должна указывать имя статьи описания отчета для каждого отчета, связанного с этим именем-файла в данной программе.

Статья описания отчета содержит набор фраз, именующих отчет и снабжающих специфической информацией о формате печатаемой страницы и организации подразделений отчета. В статье описания отчета может быть задан код идентификации так, чтобы каждый отчет можно было идентифицировать отдельно в промежуточном выходном файле.

За каждой статьей описания отчета следуют одна или несколько статей с номером уровня 01, за каждой из которых следует иерархическая структура, подобная описаниям записи в Коболе. Каждая статья с номером уровня 01 и ее подчиненные статьи описывают группу отчета. Каждая группа отчета состоит из нуля, одной или нескольких печатаемых строк, рассматриваемых как одно

GOST EXPERT ГОСТ 22558-89,

целое. Группа отчета, которую нужно напечатать, печатается целиком на одной логической странице; группа никогда не разбивается при переходе на следующую страницу.

3.2. **Структура отчета**

При определении структуры отчета главное внимание должно быть уделено требованиям вертикального и горизонтального позиционирования, манипулированию данными, физическим и логическим подразделениям отчета.

3.2.1. Вертикальное позиционирование

Средства генератора отчетов позволяют пользователю описывать группы отчета, содержащие множество строк. Вертикальное позиционирование строк на странице определяется фразой LINE NUMBER (НОМЕР СТРОКИ), связанной с каждой строкой. Фраза NEXT GROUP (СЛЕДУЮЩАЯ ГРУППА) указывает, сколько строк следует пропустить после представления последней строки группы. Первая фраза LINE NUMBER (НОМЕР СТРОКИ) следующей группы определяет дополнительную информацию о пропуске строк, которую нужно использовать при позиционировании этой группы.

3.2.2. Горизонтальное позиционирование

Генератор отчетов позволяет пользователю позиционировать поля данных в строке отчета посредством фразы COLUMN NUMBER (НОМЕР СТОЛБЦА). Система управления генератором отчетов обеспечивает заполнение пробелами промежутка между всеми определяемыми полями.

3.2.3. Манипулирование данными

При использовании средств генератора отчетов перемещением данных в группу отчета управляют фразы секции отчетов, а не операторы раздела процедур. Фразы секции отчетов, осуществляющие манипулирование данными, следующие: SOURCE (ИСТОЧНИК), SUM (СУММА) и VALUE (ЗНАЧЕНИЕ).

Фраза SOURCE (ИСТОЧНИК) определяет посылаемое данное неявного оператора MOVE (ПОМЕСТИТЬ). Принимающее печатаемое данное определяется описанием данного группы отчета, в котором появляется фраза SOURCE (ИСТОЧНИК).

Фраза SUM (СУММА) приводит к автоматическому учреждению счетчика суммы. Объект фразы SUM (СУММА) называет данное (данные), которое прибавляется к счетчику суммы при выполнении оператора GENERATE (ГЕНЕРИРОВАТЬ). Пересылка содержимого счетчика суммы в принимающее печатаемое данное, определенное описание данного группы отчета, в которой имеется фраза SUM (СУММА), совершается автоматически при представлении этой группы отчета.

Фраза VALUE (ЗНАЧЕНИЕ) указывает литерал, который появляется в печатаемом данном группы отчета каждый раз при представлении группы отчета.

GOST EXPERT ГОСТ 22558-89,

Итак, данное в группе отчета представляется, если оно сопровождается фразой COLUMN NUMBER (НОМЕР СТОЛБЦА), указывающей, где оно должно быть представлено. Значение, помещаемое в печатаемое данное, определяется фразами VALUE (ЗНАЧЕНИЕ), SOURCE (ИСТОЧНИК) или SUM (СУММА), находящимися в описании группы отчета. Ни при каких обстоятельствах печатаемое данное группы отчета не может получить значение непосредственно от оператора раздела процедур.

3.2.4. Подразделения отчета

При определении того, что представляется на странице, учитываются физическая и логическая организация отчета.

3.2.4.1. *Физическое подразделение отчета*

Длина страницы, размер полей заголовка и концовки, размер поля, в котором будут располагаться строки фрагментов, определяется фразой PAGE (РАЗМЕР СТРАНИЦЫ). Система управления генератором отчетов использует фразы LINE NUMBER (НОМЕР СТРОКИ) и NEXT GROUP (СЛЕДУЮЩАЯ ГРУППА) для размещения этих групп отчета, и, если необходимо, для перехода на новую страницу с автоматическим воспроизведением групп отчета PAGE HEADING (ЗАГОЛОВОК СТРАНИЦЫ) и PAGE FOOTING (КОНЦОВКА СТРАНИЦЫ).

3.2.4.2. *Логическое подразделение отчета*

Группы отчета типа фрагмент могут быть выстроены во вложенный набор управляемых групп. Каждая управляемая группа может начинаться группой отчета управляемый заголовок и заканчиваться группой отчета управляемая концовка.

Если определены вложенные управляемые группы, распознавание изменения значения управляющего данного в иерархии управления называется прерыванием управления, а строки заголовков и концовок, связанных с управляющим именем-данного, называются группами отчета управляемый заголовок и управляемая концовка.

Во время выполнения оператора GENERATE (ГЕНЕРИРОВАТЬ) система управления генератором отчетов использует иерархию управления для автоматической проверки прерывания управления. Если имеет место прерывание управления, считается, что для всех младших управляющих данных в иерархии управления также имеет место прерывание управления, даже если в действительности их значения не изменились. Распознавание прерывания управления вызывает следующую последовательность действий:

(1) представляются все группы управляемой концовки от младшей до уровня, на котором произошло прерывание управления, включая его;

(2) представляются все группы управляемого заголовка от

GOST EXPERT ГОСТ 22558-89,

уровня, соответствующего прерыванию управления, до самого младшего;

(3) представляется группа фрагмента отчета, имя которого указано в операторе GENERATE (ГЕНЕРИРОВАТЬ).

3.3. **Операторы раздела процедур генератора отчетов**

Операторы раздела процедур генератора отчетов следующие: INITIATE (НАЧАТЬ), GENERATE (ГЕНЕРИРОВАТЬ), TERMINATE (ЗАКОНЧИТЬ), SUPPRESS (ПОДАВИТЬ) и USE BEFORE REPORTING (ИСПОЛЬЗОВАТЬ ДО ВЫДАЧИ).

Оператор INITIATE (НАЧАТЬ) приводит к автоматическому выполнению ряда инициирующих действий системой управления генератором отчетов. Отчет должен быть инициирован прежде, чем можно будет выполнять для него детальную обработку.

Оператор GENERATE (ГЕНЕРИРОВАТЬ) с операндом имя-данного приводит к форматизации и записи на устройство вывода указанной группы отчета типа DETAIL (ФРАГМЕНТ). Кроме того, система управления генератором отчетов настраивается на выполнение ряда неявных действий, описанных в п. 3.2 настоящей части.

Оператор GENERATE (ГЕНЕРИРОВАТЬ) с операндом имя-отчета обеспечивает средства генерации итогов. В отчете, производимом оператором данного типа, автоматически подавляется печать всех строк фрагментов и отчет состоит только из суммарных итогов, накопленных во время обработки групп отчета типа DETAIL (ФРАГМЕНТ). Действия, производимые системой управления генератором отчетов по оператору GENERATE имя-отчета (ГЕНЕРИРОВАТЬ имя-отчета), аналогичны действиям, производимым по оператору GENERATE имя-данного (ГЕНЕРИРОВАТЬ имя-данного), за исключением того, что в первом случае подавляется печать строк фрагмента.

Оператор TERMINATE (ЗАКОНЧИТЬ) вызывает выполнение системой управления генератором отчетов всех автоматических функций, связанных с завершением отчета. Оператор TERMINATE (ЗАКОНЧИТЬ) должен быть выполнен до закрытия файла, содержащего отчет.

Оператор SUPPRESS (ПОДАВИТЬ) обеспечивает средства подавления печати целой группы отчета во время выполнения программы.

Вариант BEFORE REPORTING (ДО ВЫДАЧИ) оператора USE (ИСПОЛЬЗОВАТЬ) обеспечивает механизм, при помощи которого в автоматических процедурах, выполняемых системой управления генератором отчетов, в отдельных случаях могут быть выполнены операторы раздела процедур. Операторы процедуры USE BEFORE REPORTING (ИСПОЛЬЗОВАТЬ ДО ВЫДАЧИ) могут изменять содержимое данных, имена которых указаны во

GOST EXPERT ГОСТ 22558-89,

фразах SOURCE (ИСТОЧНИК). Таким образом, имеется возможность управления содержимым данных, которые производятся автоматически в группах отчета.

4. ОБРАБОТКА ТАБЛИЦ

Таблицы данных являются наиболее общим компонентом задач обработки данных. Хотя повторяющиеся данные, составляющие таблицу, могут быть описаны рядом статей описания данных, имеющих один и тот же номер уровня и подчиненных одному и тому же групповому данному, существуют две причины, почему такой подход не может быть удовлетворительным. Во-первых, с точки зрения документации однородность таких данных не очевидна и, во-вторых, доступ во время выполнения к отдельным элементам при таком подходе затрудняется.

Таблицы данных определяются в Коболе включением фразы OCCURS (ПОВТОРЯЕТСЯ) в статьи описания данных. Эта фраза указывает, что данное должно быть повторено установленное число раз. Данное рассматривается как элемент таблицы, и его имя и описание применяются к каждому повторению или вхождению. Так как каждое вхождение табличного элемента не имеет приписанного ему особого имени данного, ссылка на конкретное вхождение может быть сделана только указанием имени данного табличного элемента вместе с номером вхождения этого элемента. Номер вхождения называется индексом.

Число вхождений табличного элемента может быть фиксированным или переменным.

4.1. Определение таблиц

Чтобы определить одномерную таблицу, программист использует фразу OCCURS (ПОВТОРЯЕТСЯ) как часть описания табличного элемента, но фраза OCCURS (ПОВТОРЯЕТСЯ) не должна появляться в описании групповых данных, которые содержат табличный элемент. Пример 1 показывает одномерную таблицу, определяемую данным TABLE-ELEMENT (ТАБЛИЧНЫЙ-ЭЛЕМЕНТ).

Пример 1

```
        01 TABLE-1.
           02 TABLE-ELEMENT OCCURS 20 TIMES.
             03 FAM...
             03 NAME...
01 ТАБЛИЦА-1.
  02 ТАБЛИЧНЫЙ-ЭЛЕМЕНТ ПОВТОРЯЕТСЯ 20 РАЗ.
    03 ФАМИЛИЯ...
    03 ИМЯ...
```


GOST EXPERT ГОСТ 22558-89,

В примере 2 TABLE-ELEMENT (ТАБЛИЧНЫЙ-ЭЛЕМЕНТ) определяет одномерную таблицу, но FIO (ФИО) не определяет одномерную таблицу, поскольку в описании группового данного TABLE-ELEMENT (ТАБЛИЧНЫЙ-ЭЛЕМЕНТ), содержащего FIO (ФИО), имеется фраза OCCURS (ПОВТОРЯЕТСЯ).

Пример 2

```
02 TABLE-1.
  03 TABLE-ELEMENT OCCURS 20 TIMES.
    04 FIO OCCURS 5 TIMES.
      05 FAM . . .
      05 NAME . . .
02 ТАБЛИЦА-1.
  03 ТАБЛИЧНЫЙ-ЭЛЕМЕНТ ПОВТОРЯЕТСЯ 20 РАЗ.
    04 ФИО ПОВТОРЯЕТСЯ 5 РАЗ.
      05 ФАМИЛИЯ . . .
      05 ИМЯ . . .
```

В обоих случаях полному набору вхождений данного TABLE-ELEMENT (ТАБЛИЧНЫЙ-ЭЛЕМЕНТ) присвоено имя TABLE-1 (ТАБЛИЦА-1). Тем не менее, нет необходимости давать таблице имя группы, если не требуется ссылка на всю таблицу как на групповое данное.

Ни одна из трех одномерных таблиц, приведенных в двух следующих примерах, не имеет имени группы.

Пример 3

```
01 INFORMATION.
  02 OTDEL . . .
  02 FIO OCCURS 20 TIMES . . .
  02 N-ROOM . . .
01 СВЕДЕНИЯ.
  02 ОТДЕЛ . . .
  02 ФИО ПОВТОРЯЕТСЯ 20 РАЗ . . .
  02 N-КОМНАТЫ . . .
```

Пример 4

```
01 INFORMATION.
  02 DEPARTMENT OCCURS 20 TIMES . . .
  02 FIO . . .
  02 N-ROOM OCCURS 5 TIMES . . .
01 СВЕДЕНИЯ.
  02 ЦЕХ ПОВТОРЯЕТСЯ 20 РАЗ . . .
  02 ФИО . . .
  02 N-КОМНАТЫ ПОВТОРЯЕТСЯ 5 РАЗ . . .
```

Определение одномерной таблицы внутри каждого вхождения элемента другой одномерной таблицы приводит к образованию двумерной таблицы. Для определения двумерной таблицы фраза OCCURS (ПОВТОРЯЕТСЯ) должна появиться в описании данных

GOST EXPERT ГОСТ 22558-89,

табличного элемента и в описании только одного группового элемента данных, который содержит этот табличный элемент. Так, в примере 5 N-ROOM (N-КОМНАТЫ) является элементом двумерной таблицы — он входит 5 раз в каждый элемент данного DEPARTMENT (ЦЕХ), который сам повторяется 20 раз.

FIO (ФИО) является элементом одномерной таблицы.

Пример 5

```
02 DEPARTMENT OCCURS 20 TIMES ...
   03 FIO ...
   03 N-ROOM OCCURS 5 TIMES ...
02 ЦЕХ ПОВТОРЯЕТСЯ 20 РАЗ ...
   03 ФИО ...
   03 N-КОМНАТЫ ПОВТОРЯЕТСЯ 5 РАЗ ...
```

В общем случае для определения n-мерной таблицы фраза OCCURS (ПОВТОРЯЕТСЯ) должна появиться в описании данного элемента таблицы и в описаниях (n—1) групповых данных, содержащих этот элемент.

4.2. **Начальные значения таблиц**

Начальные значения элементов таблиц определяются в секции рабочей памяти одним из следующих способов:

(1) таблица может быть описана как ряд отдельных статей описания данных, подчиненных одному и тому же групповому данному, каждая из которых определяет значение элемента или части элемента таблицы. При определении записи и ее элементов можно использовать любую фразу описания данного (фраза USAGE (об использовании), PICTURE (ШАБЛОН), и т. п.), если это необходимо для полноты определения. Иерархическая структура таблицы затем показывается использованием статьи REDEFINES (ПЕРЕОПРЕДЕЛЯЕТ) и связанных с ней подчиненных статей. Подчиненные статьи, следующие за статьей REDEFINES (ПЕРЕОПРЕДЕЛЯЕТ) и повторяющиеся соответственно фразам OCCURS (ПОВТОРЯЕТСЯ), не могут содержать фразу VALUE (ЗНАЧЕНИЕ);

(2) величины всех размерностей таблицы могут быть инициированы фразой VALUE (ЗНАЧЕНИЕ), относящейся к описанию статьи, определяющей таблицу в целом. Статьи самого низкого уровня покажут иерархическую структуру таблицы; статьи самого низкого уровня не могут содержать фразы VALUE (ЗНАЧЕНИЕ);

(3) значения отдельных элементов таблицы могут быть определены использованием фразы VALUE (ЗНАЧЕНИЕ).

4.3. **Ссылки на табличные элементы**

Всякий раз, когда пользователь ссылается на табличный элемент или имя-условия, связанное с табличным элементом, за исключением операторов USE FOR DEBUGGING (ИСПОЛЬЗОВАТЬ

GOST EXPERT ГОСТ 22558-89,

ДЛЯ ОТЛАДКИ) и SEARCH (ИСКАТЬ), ссылка должна указывать, какое именно вхождение предполагается. Для доступа к одномерной таблице достаточно одного номера вхождения требуемого элемента. Для таблиц размерности больше единицы номер вхождения должен быть указан для каждой размерности таблицы. Таким образом, в примере 5 ссылка на четвертый DEPARTMENT (ЦЕХ) или четвертый FIO (ФИО) будет полной, в то время как ссылка на четвертый N-ROOM (N-КОМНАТЫ) полной не будет. Для ссылки на N-ROOM (N-КОМНАТЫ), который является элементом двумерной таблицы, пользователь должен указать, например, четвертый элемент N-ROOM (N-КОМНАТЫ) в пятом элементе DEPARTMENT (ЦЕХ).

4.4. **Индексирование**

Номера вхождений указываются добавлением одного или нескольких индексов к имени-данного.

Индексом может быть либо целое, либо имя-данного, относящееся к целому числовому элементарному данному, либо имя-индекса, связанное с таблицей. За именем-данного или именем-индекса может следовать знак операции + или — и целое, используемое соответственно как приращение или уменьшение. Допустимо одновременное использование целых, имен-данных и имен-индексов.

Индексы, заключенные в круглые скобки, записываются непосредственно за уточнением имени элемента таблицы. Количество индексов в такой ссылке должно равняться размерности таблицы, в которую входит данный элемент. Таким образом, каждой фразе OCCURS (ПОВТОРЯЕТСЯ) в иерархии, содержащей имя-данного, включая и само имя-данного, должен быть соотнесен индекс.

Когда требуется более одного индекса, они записываются в порядке последовательного снижения уровня вложенности размерностей организации данных. Если многомерную таблицу рассматривать как последовательность вложенных таблиц и наиболее объемлющую или внешнюю таблицу рассматривать как старший уровень, а самую внутреннюю или наименее объемлющую таблицу — как младший уровень, индексы записываются слева направо в порядке: старший уровень, промежуточный, младший.

В ссылке на данное нельзя использовать индексирование, если данное не является табличным элементом или данным или именем-условия в табличном элементе.

Наименьший допустимый номер вхождения равняется единице. Наибольший допустимый номер вхождения в каждом отдельном случае равняется максимальному числу вхождений данного, определяемому фразой OCCURS (ПОВТОРЯЕТСЯ).

4.4.1. Индексирование с помощью целого или имени-данного

GOST EXPERT ГОСТ 22558-89,

Если для представления индекса используется целое или имя-данного, они могут использоваться для ссылки на данные в разных таблицах. Элементы таблиц не обязательно должны быть одного и того же размера для всех таблиц. Одно и то же целое или имя-данного может появляться как единственный индекс одного данного и как один из двух и более индексов другого данного.

4.4.2. Индексирование с помощью имени-индекса

Для облегчения операций поиска в таблице и манипулирования отдельными данными допускается индексирование с помощью имени-индекса. Чтобы использовать это средство, программист соотносит одно или несколько имен-индексов данному, в статье описания которого имеется фраза OCCURS (ПОВТОРЯЕТСЯ). Значение имени-индекса соответствует номеру вхождения для данного, с которым связано имя-индекса.

Фраза INDEXED BY (ИНДЕКСИРУЕТСЯ), посредством которой идентифицируется имя-индекса и соотносится определенной таблице, является необязательной частью фразы OCCURS (ПОВТОРЯЕТСЯ). В Коболе нет отдельной статьи для описания имени-индекса, поскольку его определение полностью зависит от оборудования. Во время выполнения значение имени-индекса будет соответствовать номеру вхождения для той конкретной размерности таблицы, с которой было связано имя-индекса; способ соответствия определяется реализацией. Начальное значение имени-индекса во время выполнения не определено, то есть начальное значение нужно установить перед использованием.

Начальное значение имени-индекса присваивается оператором PERFORM (ВЫПОЛНИТЬ) с фразой VARYING (МЕНЯЯ), оператором SEARCH ALL (ИСКАТЬ ОСОБО) или оператором SET (УСТАНОВИТЬ).

Использование целого или имени-данного в качестве индекса для ссылки на табличный элемент или данное в табличном элементе не вызывает изменений значения имени-индекса, связанного с этой таблицей.

Имя-индекса может быть использовано для ссылки только на ту таблицу, которой оно соотнесено фразой INDEXED BY (ИНДЕКСИРУЕТСЯ).

Операции поиска данных, организованных в таблицы, обеспечиваются в Коболе оператором SEARCH (ИСКАТЬ), позволяющим осуществлять последовательный или непоследовательный (например, бинарный) поиск. Оператор используется для поиска в таблице элемента, удовлетворяющего определенному условию, и для установки значения соответствующего имени-индекса для указания этого табличного элемента.

GOST EXPERT ГОСТ 22558-89,

Относительное индексирование является дополнительным вариантом для ссылок на табличные элементы или на данное в табличном элементе. Когда за именем табличного элемента следует индекс в виде (имя-индекса + или — целое), номер вхождения, требуемый для завершенной (полной) ссылки, такой же, как если бы перед ссылкой имя-индекса было увеличено или уменьшено на целое посредством оператора SET (УСТАНОВИТЬ). Использование относительного индексирования не изменяет значения имени-индекса.

Значение имени-индекса может быть сделано доступным объектной программе путем его запоминания в индексном данном. Индексные данные описываются в программе статьей описания данного, содержащей фразу USAGE IS INDEX (ДЛЯ ИНДЕКСА). Значение имени-индекса помещается в индексное данное выполнением оператора SET (УСТАНОВИТЬ).

4.4.3. Примеры индексирования

Предполагая следующее описание данных:

```
02 ANK...
02 A1 OCCURS 20 TIMES INDEXED BY A1-INDEX...
  03 B1...
  03 B2 OCCURS 5 TIMES...
    04 C1...
    88 MAX VALUE IS...
    04 C2...
      05 D1 OCCURS 10 TIMES...
        06 E1...
        06 E2...
02 АНК...
02 A1 ПОВТОРЯЕТСЯ 20 РАЗ ИНДЕКСИРУЕТСЯ
   A1-ИНДЕКС...
  03 B1...
  03 B2 ПОВТОРЯЕТСЯ 5 РАЗ...
    04 C1...
    88 MAX ЗНАЧЕНИЕ...
    04 C2...
      05 D1 ПОВТОРЯЕТСЯ 10 РАЗ...
        06 E1...
        06 E2...
```

В ссылках на A1 и B1 требуется только один индекс, в ссылках на B2, C1, MAX и C2 требуются два индекса, и в ссылках на D1, E1 и E2 требуются три индекса.

Проиллюстрируем требования указания индексов в порядке от старшего к младшим:

E1 (18, 2, 7) значит E1 в седьмом D1, во втором B2 и в восемнадцатом A1.

GOST EXPERT ГОСТ 22558-89,

Возможность одновременного использования целых, имен-данных и имен-индексов проиллюстрирована на следующем примере:

EI (A1-INDEX, 4, ANK +5).

(EI (AI-ИНДЕКС, 4, АНК+5)).

5. ОБЩАЯ ОБЛАСТЬ ПАМЯТИ

Это средство в основном ориентировано на экономию памяти в объектной программе, так как позволяет нескольким файлам иметь одну и ту же область файла и области ввода-вывода.

Если использована фраза SAME RECORD (ОБЩАЯ ОБЛАСТЬ ЗАПИСИ), общей является только область записи, а области ввода-вывода для каждого файла остаются независимыми. В этом случае любое количество файлов, имеющих общей одну и ту же область записи, могут быть активными одновременно. Этот фактор может дать экономию времени при выполнении объектной программы.

Проиллюстрируем это. Если программист назначил одну и ту же область записи и старому, и новому файлам, он не только сэкономил память в объектной программе, но поскольку эта техника исключает пересылку каждой записи из области ввода в область вывода, получит в результате экономию и во времени. Кроме того, при использовании этой техники программисту нет необходимости определять в деталях запись и старого, и нового файлов. Вместо этого он полностью определяет запись в одном случае, и просто включает статью с номером уровня 01 в другом. Поскольку эти области записей фактически являются одной и той же областью, один набор имен достаточен для всех потребностей обработки без необходимости уточнения.

Если используется фраза SAME (ОБЩАЯ ОБЛАСТЬ) без варианта RECORD (ЗАПИСИ), общими являются не только области файла, но и области ввода-вывода.

В результате только один из файлов, разделяющих один и тот же набор общих областей, может быть активным в определенное время. Эта форма фразы предназначена для применения, когда ряд файлов используется на разных фазах объектной программы. В этих случаях фраза SAME (ОБЩАЯ ОБЛАСТЬ) позволяет программисту экономить память.

6. ОРГАНИЗАЦИЯ ПРОГРАММ И ПРОГРАММНЫХ СВЯЗЕЙ

Довольно часто полные проблемы обработки данных решаются разработкой отдельно компилируемых, но логически скоординированных программ, которые на некотором этапе до выполнения можно компилировать отдельно и затем собрать для полного ре-

GOST EXPERT ГОСТ 22558-89,

шения проблемы. Организация программ на Коболе и единицы исполнения поддерживает такой подход разделения решения больших проблем на малые, более управляемые порции, которые можно программировать и тестировать независимо.

6.1. **Понятие программы и единицы исполнения**

Имеются два уровня компьютерных программ в Коболе: исходный уровень и объектный уровень.

На исходном уровне наиболее объемлющей единицей компьютерной программы является исходная программа. Исходная программа может содержать другие исходные программы. Исходная программа — синтаксически правильный набор операторов Кобола, по определению настоящего документа состоящий из раздела идентификации, за которым необязательно следуют раздел оборудования и (или) раздел данных, и (или) раздел процедур. Исходная программа, не содержащаяся в другой исходной программе, может быть преобразована компилятором в объектную программу, которая может быть выполнена либо одна, либо вместе с другими объектными программами.

В общем случае программа, содержащаяся в другой программе, не может быть преобразована компилятором в объектную программу, так как по определениям этого документа в содержащейся программе явно разрешается ссылаться на данные, описанные в содержащей ее исходной программе.

Раздел процедур исходной программы организован как последовательность процедур двух типов. Декларативные процедуры, обычно именуемые декларативами, — это процедуры, которые будут выполняться только при появлении специальных условий во время выполнения программы. Недекларативные процедуры — процедуры, выполняемые соответственно нормальному ходу управления в программе. Декларативы могут содержать недекларативные процедуры, но последние будут выполняться только во время выполнения деклараатив, в которых они содержатся. Недекларативные процедуры могут содержать другие недекларативные процедуры, но не должны содержать декларативы. Ни декларативы, ни недекларативные процедуры не могут содержать в себе программы. Другими словами, в Коболе термины «процедура» и «программа» не являются синонимами.

На объектном уровне наиболее объемлющей единицей организации компьютерных программ является единица исполнения. Единица исполнения — это полное решение проблемы, состоящее из объектной программы или из нескольких взаимосвязанных объектных программ. Единица исполнения — это независимая единица, которая может выполняться без связей или координирования с другой единицей исполнения, за исключением того, что может обрабатывать файлы данных и сообщения и устанавливать

GOST EXPERT ГОСТ 22558-89,

или проверять переключатели, которые записаны или будут считываться другими единицами исполнения.

При вызове программы параметры, которыми она должна оперировать, могут быть переданы ей вызывающей ее программой. Поскольку любая компилируемая программа может быть первой программой, выполняемой в единице исполнения, первая выполняемая программа единицы исполнения может получать параметры.

Единица исполнения может также содержать объектный код и области памяти для данных, получаемые в результате компиляции программы, записанной на языке, отличном от Кобола; в этом случае требования к организации связей между Кобол-программой и программой, записанной не на Коболе, определяются реализацией.

6.2. **Доступные данные и файлы**

Некоторые данные имеют связанную с ними концепцию памяти, определяющую, где представляются значения и другие свойства данных относительно программ единицы исполнения. Подобно этому, определители файла тоже имеют связанную с ними концепцию памяти, определяющую, где представляется информация о позиционировании и состоянии файла, а также о других свойствах обработки файла, относительно программ единицы исполнения.

6.2.1. Имена

Имя-данного именует данное. Имя-файла именует определитель файла. Эти имена классифицируются как глобальные или локальные.

Глобальное имя используется для ссылки на объект, которому оно соотнесено, либо из программы, в которой глобальное имя объявлено, либо из любой другой программы, содержащейся в программе, объявляющей это глобальное имя.

Локальное имя можно использовать для ссылки на объект, которому оно соотнесено, только из программы, в которой локальное имя объявлено. Некоторые имена всегда глобальны, некоторые — всегда локальны, а некоторые являются либо локальными, либо глобальными в зависимости от спецификаций программы, в которой имена объявлены.

Имя-записи глобально, если указана фраза GLOBAL (ГЛОБАЛЬНОЕ) в статье описания записи, объявляющей это имя, или в случае статьи описания записи в секции файлов, если фраза GLOBAL (ГЛОБАЛЬНОЕ) указана в описании файла для имени-файла, связанного с описанием записи. Имя-данного глобально, если фраза GLOBAL (ГЛОБАЛЬНОЕ) указана либо в статье описания данного, объявляющей это имя-данного, либо в другой статье, которой подчинена упомянутая статья описания данного. Имя-условия, объявленное в статье описания данного, глобально, если эта статья подчинена другой статье, в которой указана фраза

GOST EXPERT ГОСТ 22558-89,

GLOBAL (ГЛОБАЛЬНОЕ). Однако иногда особые правила запрещают указание фразы GLOBAL (ГЛОБАЛЬНОЕ) для определенных статей описания данных, описания файла или описания записи.

Имя-файла глобально, если фраза GLOBAL (ГЛОБАЛЬНОЕ) указана в соответствующей ему статье описания файла.

Если имя-данного, имя-файла или имя-условия, объявленное в описании данного, не глобально, имя является локальным.

Глобальные имена транзитивны в программах, содержащихся в других программах.

6.2.2. Объекты

Для обеспечения доступности данных обычно требуется сохранение в памяти определенных представлений данных. Определители файлов также требуют сохранения в памяти определенной информации, относящейся к файлу. Память, соотнесенная данному или определителю файла, может быть внешней или внутренней по отношению к программе, в которой объект объявлен.

6.2.2.1. *Типы объектов*

6.2.2.1.1. Записи рабочей памяти.

Записи рабочей памяти — это участки памяти, удовлетворяющие статьям описания записей в секции рабочей памяти.

Каждая статья описания записи в программе объявляет отдельный объект. Переименование и переопределение не объявляет новых объектов; они обеспечивают альтернативную группировку или описание объектов, ранее объявленных.

6.2.2.1.2. Определители файлов

Определитель файла — это область памяти, содержащая информацию о файле и используемая для связывания имени-файла с физическим файлом и имени-файла с соответствующей ему областью записи.

6.2.2.1.3. Области записи для файлов

Считается, что никакая отдельная статья описания записи в секции файлов не объявляет область памяти для записи. Область памяти — это скорее максимально требуемая память для удовлетворения соответствующих статей описания записей. Статьи могут описывать записи фиксированной или переменной длины. В этом представлении статьи описания записи считаются соответствующими в двух случаях. Во-первых, когда описания записи подчинены одной и той же статье описания файла, они всегда соответствующие. Во-вторых, когда описания записи подчинены различным статьям описания файлов и на эти статьи описания файлов имеется ссылка во фразе SAME RECORD (ОБЩАЯ ОБЛАСТЬ ЗАПИСИ), описания записей соответствующие.

Все соответствующие статьи описания записи являются переопределениями одной и той же области памяти.

GOST EXPERT ГОСТ 22558-89,

6.2.2.1.4. Другие объекты

Примерами других объектов, объявляемых в Кобол-программе, являются: описания коммуникации, описания отчетов и управляющая информация, связанная с секциями коммуникаций, отчетов и связей.

6.2.2.2. *Свойства объектов*

Данное или определитель файла является внешним, если память, связанная с этим объектом, связана с единицей исполнения в целом, а не с отдельной программой единицы исполнения. На внешний объект может ссылаться любая программа, описывающая этот объект. Ссылки на внешний объект из различных программ, использующих разные описания объекта, являются ссылками на один и тот же объект.

Объект является внутренним, если память, связанная с этим объектом, связана только с программой, описывающей этот объект.

Внешние и внутренние объекты могут иметь либо глобальные, либо локальные имена.

6.2.2.2.1. Записи рабочей памяти

Записи данных, описанной в секции рабочей памяти, присваивается свойство внешней посредством фразы EXTERNAL (ВНЕШНЕЕ) в статье описания записи. Каждое данное, описанное статьей описания данного, подчиненной статье, описывающей внешнюю запись, также получает свойство внешнего.

Если запись или данное не обладают свойством внешних, они являются частью внутренних данных программы, в которой они описаны.

6.2.2.2.2. Определители файла

Определителю файла придается свойство внешнего посредством фразы EXTERNAL (ВНЕШНЕЕ) в соответствующей статье описания файла. Если определитель файла не обладает свойством внешнего, он является внутренним для программы, в которой описано соответствующее имя-файла.

6.2.2.2.3. Области записи для файлов

Записи данных, описанные как подчиненные статье описания файла, не содержащей фразу EXTERNAL (ВНЕШНЕЕ), или статье описания сортируемого-сливаемого файла, так же как и любые данные, описанные как подчиненные статьям описания данных для таких записей, всегда внутренние для программы, описывающей имя-файла. Если фраза EXTERNAL (ВНЕШНЕЕ) включена в статью описания файла, записи данных и данные получают свойство внешних.

6.2.2.2.4. Другие объекты

Записи данных, подчиненные данные и разнообразная соответствующая управляющая информация, описанная в секциях свя-

GOST EXPERT ГОСТ 22558-89,

зи, коммуникаций и отчетов программы, всегда рассматриваются как внутренние для программы, описывающей эти данные. Специальные соглашения применяются к данным, описанным в секции связи, при установлении связи между описанными записями данных и другими данными, доступными другим программам (п. 6.4.2 настоящей части).

6.2.3. Разрешение имени

Если программы, содержащиеся в других программах, присваивают одни и те же имена данным, условиям и определителям файла, применяются определенные соглашения. Рассмотрим ситуацию, когда программа А содержит программу В1, которая в свою очередь содержит программу С1; кроме того, программы А и В1 (но не С1) содержат статьи раздела данных для имени условия, имени данного или имени файла с одинаковым именем DUPLICATE-NAME (ИМЯ-ДУБЛИКАТ):

(1) если каждое из имен DUPLICATE-NAME (ИМЯ-ДУБЛИКАТ) относится к внутреннему объекту, существуют два разных, хотя и с идентичными именами, объекта; если оба имени относятся к внешнему объекту, существует один объект;

(2) ссылки программы А на DUPLICATE-NAME (ИМЯ-ДУБЛИКАТ) — всегда ссылки на объект, объявляемый ею. Ссылки программы В1 на DUPLICATE -NAME (ИМЯ-ДУБЛИКАТ) — всегда ссылки на объект, объявляемый программой В1;

(3) если DUPLICATE-NAME (ИМЯ-ДУБЛИКАТ) — локальное имя в обеих программах А и В1, программа С1 не может ссылаться на это имя;

(4) если DUPLICATE-NAME (ИМЯ-ДУБЛИКАТ) в программе В1 — глобальное имя, объект, именованный им, доступен программе С1 посредством ссылки на имя в программе В1, независимо от того, является ли DUPLICATE-NAME (ИМЯ-ДУБЛИКАТ) глобальным именем в программе А;

(5) если DUPLICATE-NAME (ИМЯ-ДУБЛИКАТ) в программе А глобальное имя, а в программе В1 — локальное, ссылка программы С1 на DUPLICATE-NAME (ИМЯ-ДУБЛИКАТ) является ссылкой на объект с именем, объявленным в программе А.

6.3. Классы программ

Все программы, образующие часть единицы исполнения, могут не обладать никаким, обладать одним или обоими из свойств: общая и начальная.

6.3.1. Общие программы

Общая программа — это программа, которая, несмотря на то, что непосредственно содержится в другой объемлющей программе, может быть вызвана любой программой, прямо или косвенно содержащейся в этой объемлющей программе. Свойство «общая» присваивается указанием фразы COMMON (ОБЩАЯ) в разделе

GOST EXPERT ГОСТ 22558-89,

идентификации программы. Фраза COMMON (ОБЩАЯ) облегчает написание подпрограмм, которые предназначены для использования всеми программами, содержащимися в программе.

6.3.2. Начальные программы

Начальная программа — это программа, состояние которой инициируется при вызове программы. Таким образом, когда бы начальная программа ни вызывалась, ее состояние всегда такое же, каким оно было, когда программа была вызвана первый раз в этой единице исполнения. Во время инициации начальной программы инициируются все внутренние данные этой программы; следовательно, каждому из внутренних данных программы, имеющему в своем описании фразу VALUE (ЗНАЧЕНИЕ), присваивается указанное для него значение, для данных, в статье описания которых нет фразы VALUE (ЗНАЧЕНИЕ), устанавливаются неопределенные значения. Файлы с внутренними определителями файла, связанные с программой, не находятся в режиме открытия. Механизмы управления для всех содержащихся в программе операторов PERFORM (ВЫПОЛНИТЬ) устанавливаются в начальные состояния. Свойство «начальная» устанавливается указанием фразы INITIAL (НАЧАЛЬНАЯ) в разделе идентификации программы.

6.4. **Межпрограммные связи**

Когда полное решение проблемы обработки данных разделено на несколько программ, составляющие программы должны иметь возможность взаимной связи. Эта взаимосвязь может быть четырех видов: передача управления, передача параметров, ссылка на общие данные и ссылка на общие файлы. Указанные четыре вида межпрограммных связей обеспечиваются и когда взаимосвязанные программы компилируются отдельно, и когда одна из взаимосвязанных программ содержится в другой программе. Механизмы, обеспечиваемые в последних двух случаях, отличаются от механизмов в первых двух случаях; например, программа, содержащаяся в другой программе, может ссылаться на любое имя-данного или имя-файла, являющееся глобальным именем в содержащей программе (см. п. 6.2.1 настоящей части).

6.4.1. Передача управления

Оператор CALL (ВЫЗВАТЬ) обеспечивает возможность передачи управления от одной программы другой программе в единице исполнения. Вызываемая программа в свою очередь может содержать операторы CALL (ВЫЗВАТЬ). Когда управление передается вызываемой программе, выполнение продолжается от оператора к оператору, начиная с первого недекларативного оператора. Если управление достигает оператора STOP RUN (ОСТАНОВИТЬ РАБОТУ), это означает логический конец единицы исполнения. Если управление достигает оператора EXIT PROGRAM

GOST EXPERT ГОСТ 22558-89,

(ВЫЙТИ ИЗ ПРОГРАММЫ), это означает логический конец только вызываемой программы, затем управление возвращается следующему в вызывающей программе после оператора CALL (ВЫЗВАТЬ) выполнимому оператору. Таким образом, оператор EXIT PROGRAM (ВЫЙТИ ИЗ ПРОГРАММЫ) завершает выполнение только той программы, в которой он имеется, в то время как оператор STOP RUN (ОСТАНОВИТЬ РАБОТУ) завершает выполнение единицы исполнения.

Оператор CALL (ВЫЗВАТЬ) может быть использован для вызова программы , записанной на языке, отличном от Кобола, но механизм возврата и передачи данных между программами в настоящем документе не определяется. Кобол-программа также может быть вызвана из программы, записанной на языке, отличном от Кобола, но механизм вызова и передачи данных между программами в настоящем документе не определяется. В обоих вышеизложенных случаях в настоящем документе определены только те части механизма передачи параметров, которые относятся к Кобол-программе.

6.4.1.1. *Имена программ*

Чтобы вызвать программу, оператор CALL (ВЫЗВАТЬ) идентифицирует имя программы. Имена, присваиваемые программам, непосредственно или косвенно содержащимся в других программах, должны быть однозначными (уникальными).

Имена, присваиваемые каждой из отдельно компилируемых программ, образующих единицу исполнения, должны быть также однозначны (уникальны).

6.4.1.2. *Область действия оператора CALL (ВЫЗВАТЬ)*

Вызывающая программа может обладать или не обладать любым свойством программы; она может компилироваться отдельно или нет; она может или содержаться в другой программе, или содержать другую программу:

(1) любая вызывающая программа может вызвать любую отдельно компилируемую программу в единице исполнения;

(2) вызывающая программа может вызвать любую программу, непосредственно содержащуюся в вызывающей программе;

(3) любая вызывающая программа может вызвать любую программу, обладающую свойством «общая» и содержащуюся непосредственно в программе, которая косвенно или непосредственно содержит вызывающую программу, если только вызывающая программа сама не содержится в программе, обладающей свойством «общая»;

(4) вызывающая программа может вызвать программу, не обладающую свойством «общая» и не компилируемую отдельно, тогда и только тогда, когда эта программа непосредственно содержится в вызывающей программе.

GOST EXPERT ГОСТ 22558-89,

6.4.1.3. *Область действия имен программ*

Определенные соглашения применяются, если в единице исполнения имя содержащейся программы в отдельно компилируемой программе идентично имени другой отдельно компилируемой программы.

Рассмотрим случай, когда программа А содержит программу В1 и программу DUPLICATE-NAME (ИМЯ-ДУБЛИКАТ), программа В1 содержит программу ВВ и программа DUPLICATE-NAME (ИМЯ-ДУБЛИКАТ) содержит программу DD.

Имя DUPLICATE-NAME (ИМЯ-ДУБЛИКАТ) также указано для отдельно компилируемой программы.

(1) Если программа А, но ни одна из программ, содержащихся в ней, вызывает программу DUPLICATE-NAME (ИМЯ-ДУБЛИКАТ), активируется программа, содержащаяся в программе А.

(2) Если программу DUPLICATE-NAME (ИМЯ-ДУБЛИКАТ) вызывает программа В1 или ВВ, то:

а) если программа DUPLICATE-NAME (ИМЯ-ДУБЛИКАТ), содержащаяся в программе А, обладает свойством «общая», то вызывается она;

б) если программа DUPLICATE-NAME (ИМЯ-ДУБЛИКАТ), содержащаяся в программе А, не обладает свойством «общая», вызывается отдельно компилируемая программа.

(3) Если программа DD или программа DUPLICATE-NAME (ИМЯ-ДУБЛИКАТ), содержащаяся в программе А, вызывает программу DUPLICATE-NAME (ИМЯ-ДУБЛИКАТ), вызываемой программой является отдельно компилируемая программа.

(4) Если любая отдельно компилируемая программа в единице исполнения или любая другая программа, содержащаяся в такой программе, вызывает программу DUPLICATE-NAME (ИМЯ-ДУБЛИКАТ), вызываемой программой является отдельно компилируемая программа, именуемая DUPLICATE-NAME (ИМЯ-ДУБЛИКАТ).

6.4.2. Передача параметров программам

Программа вызывает другую программу, чтобы вызванная программа выполнила от имени вызывающей программы некоторую часть решения проблемы обработки данных. В большинстве случаев необходимо, чтобы вызывающая программа определила вызываемой программе точную часть решения проблемы, задавая определенные значения данных, требующихся вызываемой программе и доступных ей. Одним из методов обеспечения доступности значений данных является передача параметров программе, описываемая в данном пункте. Другой метод — использование общих данных (п. 6.4.3 настоящей части). Значения данных, передаваемых в качестве параметров, также идентифицируют некото-

GOST EXPERT ГОСТ 22558-89,

рые данные для совместного использования; следовательно, два метода не являются взаимно независимыми.

6.4.2.1. *Идентификация параметров*

Данные, передаваемые вызывающей программой в качестве параметров другой программе, должны быть доступны для вызывающей программы, и данные, получающие передаваемые данные, должны быть объявлены в разделе данных вызываемой программы. В вызываемой программе требуемые параметры идентифицируются списком ссылок на параметры заголовка раздела процедур этой программы с помощью имен, присвоенных параметрам в статьях описания данных этой программы. В вызывающей программе значения параметров, передаваемых вызываемой программе, идентифицируются списком ссылок в операторе CALL (ВЫЗВАТЬ), используемом для вызова этой программы. Для этих списков во время выполнения устанавливается позиционное соответствие между значениями их элементов, как они известны каждой программе, то есть, первому параметру в одном списке соответствует первый параметр в другом списке, второму — второй и т. д. Так, например, программа, которую может вызвать другая программа, может включать строки:

PROGRAM-ID. EXAMPLE.
PROCEDURE DIVISION USING NUM, PCODE, COST.
ПРОГРАММА. ПРИМЕР.
РАЗДЕЛ ПРОЦЕДУР ИСПОЛЬЗУЯ НОМ, РКОД, СТОИМ.

и может быть вызвана выполнением оператора

CALL "EXAMPLE" USING NBR, PTYPE, PRICE.
ВЫЗВАТЬ "ПРИМЕР" ИСПОЛЬЗУЯ ЧИС, РТИП, ЦЕНА.

При этом устанавливается следующее соответствие:

Вызываемая программа	Вызывающая программа
NUM (НОМ)	NBR (ЧИС)
PCODE (РКОД)	PTYPE (РТИП)
COST (СТОИМ)	PRICE (ЦЕНА)

Существенны только позиции имен-данных, но не сами имена.

6.4.2.2. *Значения параметров*

Вызывающая программа управляет методами, по которым вызываемая программа вычисляет значения передаваемых ей параметров и возвращает результаты как измененные значения параметров.

Отдельные параметры, на которые имеются ссылки в фразе USING (ИСПОЛЬЗУЯ) оператора CALL (ВЫЗВАТЬ), можно передавать как значение или ссылку.

Вызываемая программа может использовать и изменять значение данного, упомянутого в операторе CALL (ВЫЗВАТЬ) вызывающей программы в качестве параметра, передаваемого как

GOST EXPERT ГОСТ 22558-89,

ссылка. Вызываемой программе запрещено обращаться и изменять данное вызывающей программы, если оно указано в операторе CALL (ВЫЗВАТЬ) в качестве параметра, передаваемого как значение. Значение параметра вычисляется при выполнении оператора CALL (ВЫЗВАТЬ) и предоставляется вызываемой программе. Это значение может быть изменено вызываемой программой во время ее выполнения, но значение соответствующего данного в вызывающей программе не изменяется. Таким образом, параметр, передаваемый как ссылка, может использоваться вызываемой программой для возврата значения вызывающей программе, в то время как параметр, передаваемый как значение, не может быть так использован.

Параметры, ссылки на которые указаны в заголовке раздела процедур вызываемой программы, должны быть описаны в секции связи раздела данных этой программы.

6.4.3. Общие данные

Две программы в единице исполнения могут ссылаться на общие данные в следующих случаях:

(1) на содержимое данного внешней записи данных можно ссылаться из любой программы при условии, что в программе описана эта запись данных (см. п. 6.2.2 настоящей части);

(2) если программа содержится в другой программе, обе программы могут ссылаться на данные, обладающие свойством «глобальное» либо в содержащей программе, либо в любой программе, которая непосредственно или косвенно содержит содержащую программу (см. п. 6.2.1 настоящей части);

(3) механизм, при помощи которого значение параметра передается как ссылка из вызывающей программы в вызываемую программу, устанавливает общее данное; вызываемая программа, используя, возможно, другой идентификатор, может ссылаться на данное в вызывающей программе.

6.4.4. Общие файлы

Две программы в единице исполнения могут ссылаться на общие определители файла при следующих обстоятельствах:

(1) на внешний определитель файла можно ссылаться из любой программы, описывающей этот определитель файла (см. п. 6.2.2 настоящей части);

(2) если программа содержится в другой программе, обе программы могут ссылаться на общий определитель файла, используя соответствующее глобальное имя-файла либо в содержащей программе, либо в любой программе, которая непосредственно или косвенно содержит содержащую программу (см. п. 6.2.1 настоящей части).

6.5. **Внутрипрограммные связи**

Процедуры, составляющие раздел процедур программы, со-

GOST EXPERT ГОСТ 22558-89,

общаются друг с другом посредством передачи управления или обращения к общим данным.

6.5.1. Передача управления

Имеются четыре способа передачи управления в программах:

(1) Оператор GO TO (ПЕРЕЙТИ К).

(2) Оператор PERFORM (ВЫПОЛНИТЬ).

(3) Процедура ввода, связанная с оператором SORT (СОРТИРОВАТЬ), или процедура вывода, связанная с оператором SORT (СОРТИРОВАТЬ) или MERGE (СЛИТЬ).

(4) Декларативная процедура, которая активируется при возникновении определенных условий, включая условия ошибки.

Процедуру ввода-вывода можно рассматривать как неявный оператор PERFORM (ВЫПОЛНИТЬ), который выполняется совместно с оператором MERGE (СЛИТЬ) или SORT (СОРТИРОВАТЬ); по этой причине ограничения на оператор PERFORM (ВЫПОЛНИТЬ) относятся в равной степени к процедурам ввода-вывода.

К декларативным процедурам применяются более строгие ограничения, чем ограничения для оператора PERFORM (ВЫПОЛНИТЬ).

6.5.2. Общие данные

На все данные, объявленные в разделе данных программы, можно ссылаться в операторах процедур, процедур ввода-вывода и деклараатив, составляющих эту программу. При определенных условиях программа может ссылаться на данные, не объявленные в ее разделе данных (см. п. 6.2 настоящей части).

6.6. **Сегментация**

Средства сегментации дают возможность пользователю физически подразделять на более мелкие части раздел процедур объектной Кобол-программы. Все исходные параграфы, которые содержат одинаковые номера сегментов в заголовках их секций, будут рассматриваться во время выполнения как единый сегмент. Так как номера сегментов могут принимать значения от 00 до 99, то объектная программа может быть разделена максимум на 100 сегментов.

Сегменты программы могут быть трех типов: фиксированные постоянные (неперекрываемые), фиксированные перекрываемые и независимые. Тип задается номерами сегментов.

Фиксированные неперекрываемые сегменты находятся всегда в памяти во время выполнения всей программы; они не могут перекрываться в памяти, за исключением того случая, когда система выполняет другую программу; в таком случае фиксированные сегменты могут быть временно «свернуты».

Фиксированные перекрываемые сегменты могут перекрываться в памяти во время выполнения программы, но такое перекры-

GOST EXPERT ГОСТ 22558-89,

тие не должно касаться пользователя, то есть они логически идентичны фиксированным сегментам, но физически отличны от них.

Независимые сегменты могут налагаться один на другой в памяти, но такое перекрытие приведет к инициализации сегмента. Поэтому независимые сегменты логически отличны от фиксированных перекрываемых или фиксированных неперекрываемых сегментов и физически отличны от фиксированных сегментов.

7. СРЕДСТВА КОММУНИКАЦИЙ

Средства коммуникаций дают возможность получения, обработки и создания сообщений или их частей. Посредством системы управления сообщениями они позволяют связываться с местными и дистанционными коммуникационными устройствами.

7.1. **Система управления сообщениями**

Реализация средств коммуникации в программах на Коболе требует наличия системы управления сообщениями в операционной среде объектной Кобол-программы.

Система управления сообщениями логически связана с операционной системой, под управлением которой выполняется объектная Кобол-программа. Система управления сообщениями выполняет следующие основные функции:

(1) осуществляет связь между объектной Кобол-программой и сетью коммуникационных устройств, подобно тому как операционная система осуществляет связь между объектной программой Кобола и такими устройствами, как устройства ввода перфокарт, устройства печати, магнитные ленты и устройства массовой памяти;

(2) поддерживает порядок на линии, выполняя вызов адресата, упорядочение передач сообщений по каналам связи и синхронизацию;

(3) выполняет такие зависящие от устройства работы, как перекодировка символов и вставка управляющих символов, что дает возможность пользователю Кобола создавать программы, независимые от устройств.

Первая функция — связь между объектной программой Кобола и коммуникационными устройствами — наиболее очевидна для пользователя. О наличии двух других функций пользователь может и не знать. Сообщения от коммуникационных устройств помещаются системой управления сообщениями во входные очереди в ожидании передачи объектной Кобол-программе. Выходные сообщения объектной Кобол-программы помещаются системой управления сообщениями в выходные очереди в ожидании передачи на коммуникационные устройства. Структура, форматы и символические имена очередей определяются для системы управления

GOST EXPERT ГОСТ 22558-89,

сообщениями пользователем до выполнения объектной программы. До выполнения программы определяются также символические имена источников сообщений и адресатов. В программе на Коболе могут использоваться только символические имена, известные системе управления сообщениями.

Во время выполнения объектной программы система управления сообщениями выполняет все необходимые действия по обновлению различных очередей.

7.2. **Объектная программа Кобола**

Объектная программа взаимодействует с системой управления сообщениями, когда необходимо переслать и получить данные или опросить состояние различных очередей, которые создаются и поддерживаются системой управления сообщениями. Кроме того, объектная программа может потребовать от системы управления сообщениями установить или прервать логическую связь между коммуникационным устройством и заданной частью структуры очереди. Способ реализации физической связи является функцией системы управления сообщениями.

7.3. **Связь программы на Коболе с системой управления сообщениями и коммуникационными устройствами**

Взаимосвязь с коммуникационными устройствами устанавливается в Коболе посредством статьи описания коммуникации CD (ОК) в секции коммуникаций раздела данных.

Различают два вида связи:

(1) между объектной программой и системой управления сообщениями;

(2) между системой управления сообщениями и коммуникационными устройствами.

Для управления связью с системой управления сообщениями в исходной программе на Коболе используют три оператора:

(1) RECEIVE (ПОЛУЧИТЬ), который вызывает передачу объектной программе данных из очереди;

(2) SEND (ПОСЛАТЬ), который вызывает передачу данных из объектной программы в одну или более очередей;

(3) ACCEPT MESSAGE COUNT (ПРИНЯТЬ ЧИСЛО СООБЩЕНИЙ), который указывает системе управления сообщениями, что она должна сообщить объектной программе число полных сообщений в данной структуре очереди.

Для управления связью между системой управления сообщениями и коммуникационными устройствами в исходной программе используются два оператора:

(1) ENABLE (РАЗРЕШИТЬ), который устанавливает логическую связь между системой управления сообщениями и одним или несколькими заданными коммуникационными устройствами;

GOST EXPERT ГОСТ 22558-89,

(2) DISABLE (ЗАПРЕТИТЬ), который разрывает логическую связь между системой управления сообщениями и одним или несколькими заданными коммуникационными устройствами.

На рис. 1 показана взаимосвязь программы на Коболе с коммуникационными устройствами (п. 7.5.2 настоящей части).

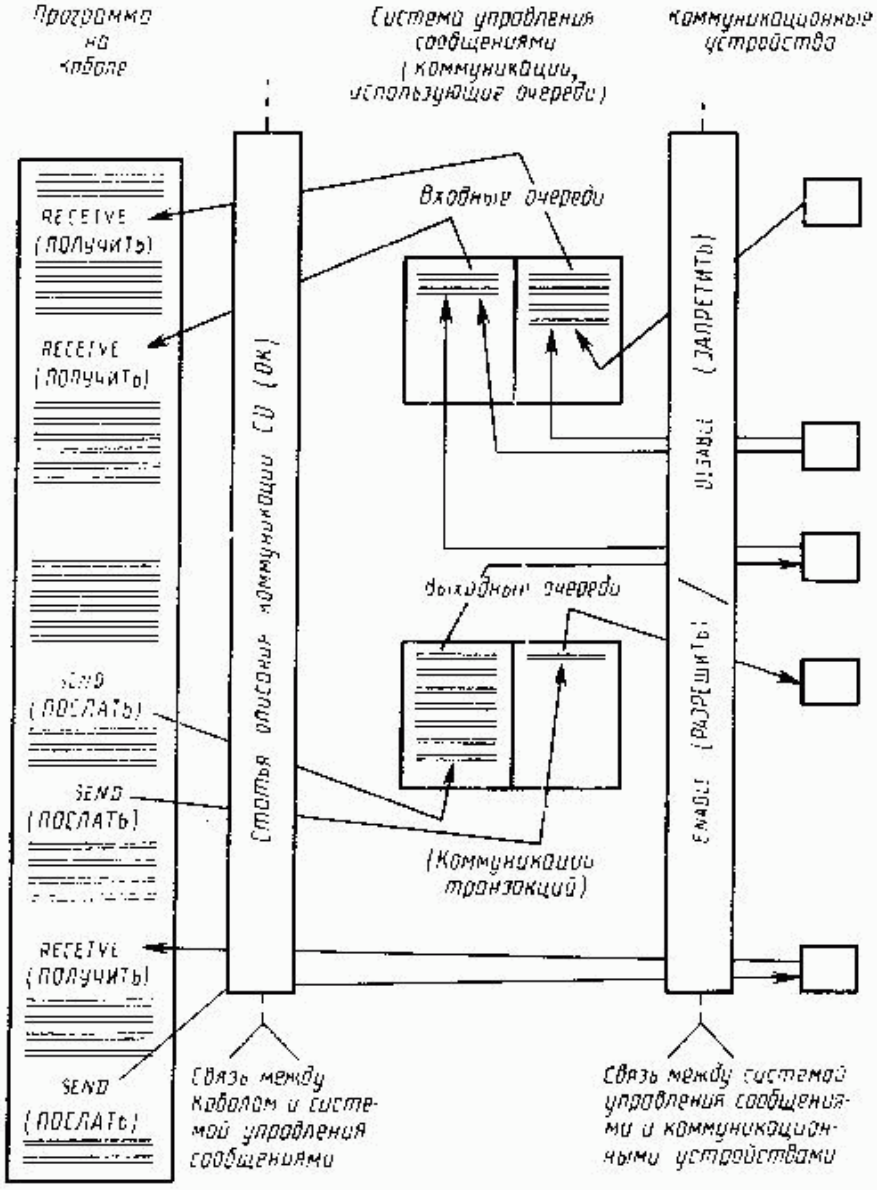


Рис. 1

7.3.1. Вызов объектной программы Кобола

Существуют два метода вызова объектной программы Кобола, использующей средства коммуникаций: запланированный запуск и вызов системой управления сообщениями.

Разница между этими методами заключается только в том, как производится заполнение некоторых областей в указанной статье CD (ОК).

GOST EXPERT ГОСТ 22558-89,

7.3.1.1. *Запланированный запуск объектной программы Кобола*

Объектная программа Кобола, использующая средства коммуникаций, может быть вызвана для выполнения через обычные операционные средства, например, язык управления заданиями. В этом случае программа может использовать три метода для определения того, какие сообщения, если они имеются, доступны во входных очередях:

(1) оператор ACCEPT MESSAGE COUNT (ПРИНЯТЬ ЧИСЛО СООБЩЕНИЙ);

(2) оператор ACCEPT (ПРИНЯТЬ) с вариантом NO DATA (НЕТ ДАННЫХ);

(3) оператор ACCEPT (ПРИНЯТЬ) без варианта NO DATA (НЕТ ДАННЫХ). В этом случае подразумевается, что программа переходит в состояние ожидания, если нет доступных данных.

7.3.1.2. *Вызов объектной программы Кобола системой управления сообщениями*

Иногда желательно вызвать для выполнения объектную программу, использующую средства коммуникаций, только тогда, когда для нее имеется требуемая информация. Такой вызов осуществляется системой управления сообщениями, когда она определяет, что требуется объектная программа для обработки имеющегося сообщения. Каждая объектная программа, вызываемая системой управления сообщениями, создает единицу исполнения. До начала выполнения объектной программы система управления сообщениями помещает имена символических очередей и подочередей в области данных, определенные статьей описания коммуникаций с фразой FOR INITIAL INPUT (ДЛЯ НАЧАЛЬНОГО ВВОДА), или система управления сообщениями помещает символическое имя терминала в область данных, определенную статьей описания коммуникации с фразой FOR INITIAL I-O (ДЛЯ НАЧАЛЬНОГО ВВОДА-ВЫВОДА).

В результате выполнения последующего оператора RECEIVE (ПОЛУЧИТЬ), связанного с данной статьей описания коммуникации, имеющееся сообщение будет передаваться объектной программе.

7.3.1.3. *Определение метода вызови*

Исходную программу можно написать так, что объектная программа может быть вызвана для выполнения любым из двух методов. Для определения метода вызова объектной программы необходимо следующее:

(1) исходная программа должна содержать одну статью описания коммуникации, содержащую фразу FOR INITIAL INPUT (ДЛЯ НАЧАЛЬНОГО ВВОДА) или FOR INITIAL I-O (ДЛЯ НАЧАЛЬНОГО ВВОДА-ВЫВОДА);

GOST EXPERT ГОСТ 22558-89,

(2) если программа содержит статью описания коммуникации с фразой FOR INITIAL INPUT (ДЛЯ НАЧАЛЬНОГО ВВОДА), в разделе процедур могут содержаться операторы для проверки начального значения имени символической очереди в этой статье описания коммуникации. Если оно заполнено пробелами, использованы операторы управления заданием для вызова объектной программы. Если не заполнено пробелами, объектную программу вызвала система управления сообщениями и инициировала данное с символическим именем очереди, содержащей сообщение для обработки;

(3) когда программа содержит статью описания коммуникации с фразой FOR INITIAL I-O (ДЛЯ НАЧАЛЬНОГО ВВОДА-ВЫВОДА), в разделе процедур могут содержаться операторы для проверки начального значения символического имени терминала в данной статье CD (ОК). Если оно заполнено пробелами, использованы операторы управления заданием для вызова объектной программы. Если оно не заполнено пробелами, система управления сообщениями вызвала объектную программу и инициировала данное с символическим именем коммуникационного терминала, который является источником сообщения, подлежащего обработке.

7.4. **Понятие сообщений и сегментов сообщения**

Сообщение состоит из некоторого произвольного количества информации (обычно последовательности литер), начало и конец которой определены или подразумеваются. Сообщения являются основной, но необязательно самой элементарной единицей данных, которую можно обработать средствами коммуникаций Кобола.

Сообщения могут логически делиться на меньшие единицы данных, называемые сегментами сообщений. Внутри сообщения сегменты разграничиваются посредством индикаторов конца сегмента ESI (ИКС). Сообщение, состоящее из одного или более сегментов, отделяется от следующего сообщения посредством индикатора конца сообщения EMI (ИКЩ). Аналогично, группа из нескольких сообщений может быть логически отделена от следующих сообщений посредством индикатора конца группы EGI (ИКГ). Когда программа на Коболе получает сообщение или его сегмент, область описания коммуникации обновляется системой управления сообщениями так, что она указывает, какой индикатор, если он имеется, связан с текстом, переданным при выполнении оператора RECEIVE (ПОЛУЧИТЬ). При выводе индикатор, связанный с текстом, передаваемым системе управления сообщениями через оператор SEND (ПОСЛАТЬ), определяется в операторе SEND (ПОСЛАТЬ). Таким образом, существование логических индикаторов распознается и задается как системой управления сообщениями, так и объектной программой Кобола. Од-

GOST EXPERT ГОСТ 22558-89,

нако индикаторы не включаются в тексты сообщений, обрабатываемые программами Кобола.

Между индикаторами EGI, EMI, ESI (ИКГ, ИКЩ и ИКС) существует отношение предшествования. EGI (ИКГ) является наиболее объемлющим индикатором, а ESI (ИКС) — наименее объемлющим. Наличие некоторого индикатора, связанного с текстом сообщения, предполагает существование всех менее объемлющих индикаторов, связанных с этим текстом. Например, наличие индикатора EGI (ИКГ) предполагает наличие EMI (ИКЩ) и ESI (ИКС).

7.5. **Понятие очередей**

Следующие рассуждения применимы только в том случае, когда коммуникационное оборудование устанавливается посредством описания коммуникации без фразы FOR I-O (ДЛЯ ВВОДА-ВЫВОДА).

Очереди состоят из одного или нескольких сообщений, получаемых от одного или более коммуникационных устройств, и как таковые образуют буферы данных между объектной программой и системой управления сообщениями. Входные очереди логически отделены от выходных.

Система управления сообщениями логически помещает в очереди или извлекает из них только полные сообщения. Части сообщений логически не помещаются в очередь, пока системе не будет доступно полное сообщение, то есть система управления сообщениями не передает сегмент сообщения объектной программе, пока все сегменты сообщения не будут во входной очереди, даже если используется оператор RECEIVE (ПОЛУЧИТЬ) с вариантом SEGMENT (СЕГМЕНТ). Для выходных сообщений система не посылает ни одного сегмента сообщения, пока все сегменты не будут в выходной очереди. Запрос о глубине очереди или числе сообщений данной очереди отражает только число полных сообщений, существующих в ней.

Процесс помещения сообщений в очередь называется постановкой в очередь, а процесс получения из очереди — извлечением из очереди.

7.5.1. Независимые постановка в очередь и извлечение из очереди

Некоторые сообщения могут быть получены системой управления сообщениями с коммуникационного устройства до выполнения объектной программы Кобола. В этом случае система помещает сообщение в подходящую входную очередь (при условии, что входная очередь разрешена) до тех пор, пока объектная программа не потребует извлечения его из очереди с помощью оператора RECEIVE (ПОЛУЧИТЬ). В свою очередь, объектная программа Кобола имеет возможность постановки в выходные очереди сооб-

GOST EXPERT ГОСТ 22558-89,

щений, которые не передаются на коммуникационное устройство вплоть до завершения выполнения программы. Это может произойти в следующих случаях:

а) когда запрещен обмен данными между заданной выходной очередью и ее адресатом;

б) когда объектная программа создает выходные очереди быстрее, чем адресат может их принять.

7.5.2. Разрешение и запрещение очередей

Обычно система управления сообщениями разрешает или запрещает очереди, основываясь на времени дня, активности сообщений или других факторах, не связанных с программой Кобола. Однако программа Кобола может выполнить эти функции сама с помощью операторов ENABLE (РАЗРЕШИТЬ) и DISABLE (ЗАПРЕТИТЬ).

7.5.3. Методы постановки в очередь и извлечения из очереди

В системах, которые разрешают пользователю задавать определенные функции системы управления сообщениями, может возникнуть необходимость еще до выполнения программ, использующих средства коммуникации, задать для системы управления сообщениями алгоритм постановки в очередь и извлечения из очереди. Типичный алгоритм постановки в очередь будет, например, указывать, что все сообщения из одного источника надо помещать в данную входную очередь, или все сообщения, посылаемые данному адресату, должны быть помещены в данную выходную очередь.

Выборка сообщений из очереди часто производится так, что первым из очереди выбирается то сообщение, которое в нее было первым поставлено. Однако пользователь может задать другой алгоритм выборки, например, воспользоваться приоритетной очередностью.

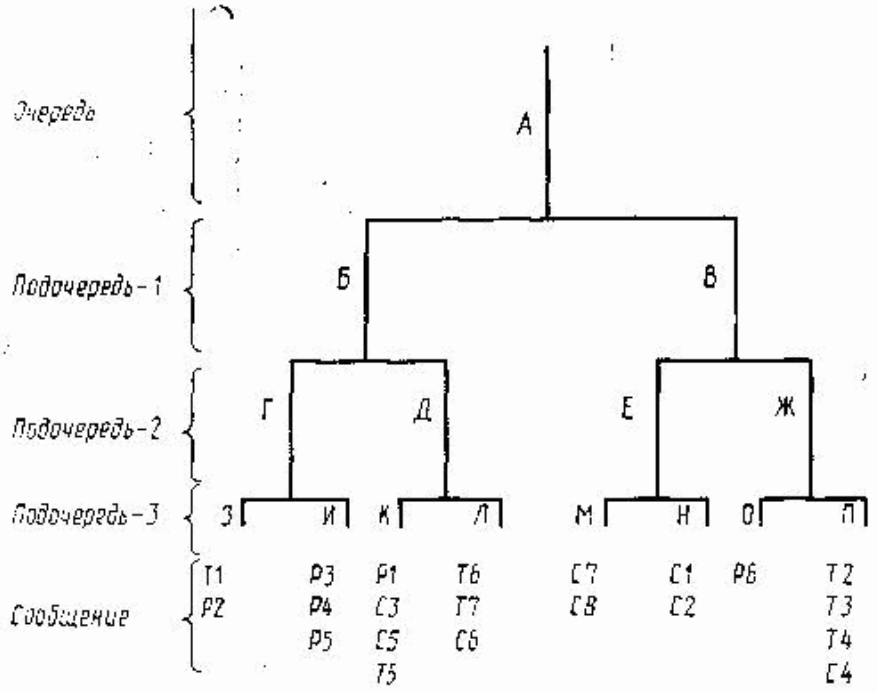
7.5.4. Иерархия очередей

Для более гибкого управления сообщениями, помещаемыми в очередь и извлекаемыми из очереди, можно определить в системе управления сообщениями иерархию входных очередей, то есть очереди, состоящие из очередей. Пользователю Кобола доступны 4 уровня очередей. В порядке уменьшения значимости уровни очередей названы так: очередь, подочередь-1, подочередь-2 и подочередь-3. Иерархия очередей показана на рис. 2.

По рис. 2 рассмотрим действия системы управления сообщениями. Предположим, что система управления сообщениями помещает сообщения в очередь и извлекает сообщения из очереди по следующему алгоритму:

а) сообщение помещается в очередь в соответствии со значением некоторого данного в каждом сообщении;

GOST EXPERT ГОСТ 22558-89,



А, Б, ..., П — очереди и подочереди; С1, ..., Р1, ..., Т1, ... — сообщения, обозначенные в соответствии с их источником и порядковым номером

Рис. 2

б) если на некотором уровне подочередь не задана, то при выполнении оператора RECEIVE (ПОЛУЧИТЬ) система управления сообщениями выберет подочередь этого уровня в алфавитном порядке. Например, если пользователем не задана подочередь-1, система управления сообщениями выберет сообщение из подочереди-1, обозначенной буквой Б.

Нижеследующие примеры иллюстрируют действия алгоритма для очереди, имеющей структуру, показанную на рис. 2.

Пример 1

В статье описания коммуникации задана очередь А. При выполнении оператора RECEIVE (ПОЛУЧИТЬ) для данного имени-коммуникации система управления сообщениями передает программе сообщение Т1.

Пример 2

В статье описания коммуникации заданы очередь А и подочередь-1 В. При выполнении оператора RECEIVE (ПОЛУЧИТЬ) для данного имени-коммуникации система управления сообщениями передает программе сообщение С7.

Пример 3

В статье описания коммуникации заданы: очередь А, подочередь-1 Б, подочередь-2 Д. При выполнении оператора RECEIVE (ПОЛУЧИТЬ) для данного имени-коммуникации система управления сообщениями передает программе сообщение Р1.

GOST EXPERT ГОСТ 22558-89,

Пример 4

В статье описания коммуникации заданы: очередь А, подочередь-1 В, подочередь-2 Ж и подочередь-3 О. При выполнении оператора RECEIVE (ПОЛУЧИТЬ) для данного имени-коммуникации система управления сообщениями передает программе сообщение Р6.

Если требуется получить следующее сообщение очереди, независимо от того, в какой подочереди оно находится, в статье описания коммуникации необходимо указать только имя очереди. При передаче сообщения система управления сообщениями сообщит объектной программе Кобола имя соответствующей подочереди через данное в статье описания коммуникации. Если требуется получить следующее сообщение в данной подочереди, в статье описания коммуникации необходимо указать имя очереди и имена подочередей.

Для вывода пользователь Кобола указывает только адресат сообщения, а система управления сообщениями помещает сообщение в подходящую выходную очередь.

Не существует однозначного соответствия между коммуникационным устройством и источником (адресатом). Источник или адресат может состоять из одного или более физических устройств. Устройство или устройства, являющиеся источником (адресатом), должны быть определены для системы управления сообщениями.

7.6. **Понятие коммуникации транзакций**

В противоположность вышеизложенному механизму очередей, некоторые применения требуют прямого диалога между коммуникационным устройством и объектной программой. В этом случае нет необходимости создания очередей сообщений, поскольку сообщения обрабатываются безотлагательно. В Коболе возможно определить такого рода обработку использованием в статье CD (ОК) фразы FOR I-O (ДЛЯ ВВОДА-ВЫВОДА). Описание коммуникации, содержащее фразу FOR I-O (ДЛЯ ВВОДА-ВЫВОДА), может поддерживать связь только с одним терминалом; тем не менее единица исполнения может содержать более одного описания коммуникации с фразой FOR I-O (ДЛЯ ВВОДА-ВЫВОДА) и эти описания коммуникаций могут поддерживать связь с одним и тем же терминалом или с различными терминалами. Если указана фраза FOR INITIAL I-O (ДЛЯ НАЧАЛЬНОГО ВВОДА-ВЫВОДА), программа должна быть вызвана системой управления сообщениями.

1 Зак. 254

ГОСТ 22558-89,

Часть 3. ГЛОССАРИЙ

1. ВВЕДЕНИЕ

Приведенные ниже определения терминов соответствуют смыслу, который приписывается им в Коболе, и могут иметь иное значение в других языках.

Определения служат справочным или вводным материалом, который следует просмотреть прежде, чем читать следующее далее детальное описание языка. Определения в большинстве случаев краткие и не содержат подробных синтаксических правил. Полные спецификации элементов, определяемых ниже, находятся в частях данного документа.

2. ОПРЕДЕЛЕНИЯ

Адресат — символическое обозначение получателя передач из очереди.

Арифметическая операция — действие, вызываемое выполнением арифметического оператора или вычислением арифметического выражения, имеющее результатом математически правильное значение.

Арифметический оператор — оператор, вызывающий выполнение арифметической операции. Арифметическими операторами являются ADD (СЛОЖИТЬ), COMPUTE (ВЫЧИСЛИТЬ), DIVIDE (РАЗДЕЛИТЬ), MULTIPLY (УМНОЖИТЬ) и SUBTRACT (ОТНЯТЬ).

Арифметическое выражение — идентификатор элементарного числового данного, числовой литерал или идентификаторы и литералы, разделенные знаками арифметических операций, или два арифметических выражения, разделенные знаком арифметической операции, или арифметическое выражение, заключенное в скобки.

Библиотечный текст — последовательность слов текста, строк комментария, разделителя пробел или разделителя ограничитель псевдотекста в библиотеке Кобола.

Блок — физическая порция данных, которая обычно состоит из одной или нескольких логических записей. Для файлов массовой памяти блок может содержать часть логической записи. Размер блока не зависит ни от размера файла, внутри которого он содержится, ни от размера логической записи (записей), которая либо содержится в блоке, либо перекрывает блок (ч. 4, п. 4.3.1.2). Термин является синонимом термина физическая запись.

Буква — в русской нотации литера, принадлежащая одному из следующих двух множеств:

1) прописные буквы: Б, Г, Д, Ж, З, И, Й, Л, П, У, Ф, Ц, Ч, Ш, Щ, Ы, Ь, Э, Ю, Я, A, B, C, D, E, F, G, H,

GOST EXPERT ГОСТ 22558-89,

I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z;

2) строчные буквы: б, в, г, д, ж, з, и, й, к, л, м, н, п, т, ф, ц, ч, ш, щ, ы, ь, э, ю, я, a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z.

В английской нотации литеры русского алфавита не являются буквами.

Буквенная литера — буква или пробел.

Буквенно-цифровая литера — любая литера из набора литер машины.

Валютный знак — литера $ (¤) из набора литер Кобола, представляющая обозначение валютной единицы, принятое в Коболе.

Валютный символ — литера, определенная фразой CURRENCY SIGN (ВАЛЮТНЫЙ ЗНАК) в параграфе SPECIAL-NAMES (СПЕЦИАЛЬНЫЕ-ИМЕНА) для представления символа валютной единицы. Если в исходной Кобол-программе нет фразы CURRENCY SIGN (ВАЛЮТНЫЙ ЗНАК), валютный символ идентичен валютному знаку.

Вариант — упорядоченный набор из одной или более строк литер Кобола, образующий часть оператора Кобола или часть фразы Кобола.

Верхнее поле — пустое поле, предшествующее телу страницы.

Внешнее данное — данное, описанное как часть внешней записи в одной или нескольких программах единицы исполнения и на которое можно ссылаться из любой программы, в которой оно описано.

Внешние данные — данные, описанные в программе как внешние данные, и внешние определители файлов.

Внешний определитель файла — определитель файла, доступный одной или нескольким объектным программам единицы исполнения.

Внешний переключатель — устройство оборудования или программное средство, определяемое и именуемое реализацией, которое используется для указания одного из двух альтернативных состояний.

Внешняя запись данных — логическая запись, описанная в одной или нескольких программах единицы исполнения, на составляющие данные которой можно ссылаться из любой программы, в которой они описаны.

Внутреннее данное — данное, описанное только в одной программе единицы исполнения. Внутреннее данное может иметь глобальное имя.

GOST EXPERT ГОСТ 22558-89,

Внутренние данные — данные, описанные в программе, за исключением всех внешних данных и внешних определителей файла. Данные, описанные в секции связи программы, рассматриваются как внутренние данные.

Внутренний набор литер — определенный реализацией набор литер, допустимых для машины, указанной в параграфе OBJECT-COMPUTER (РАБОЧАЯ-МАШИНА).

Внутренний определитель файла — определитель файла, доступный только одной объектной программе единицы исполнения.

Внутренняя основная последовательность — определенная реализацией основная последовательность, принятая для машины, указанной в параграфе OBJECT-COMPUTER (РАБОЧАЯ-МАШИНА).

Возрастающий ключ — ключ, по значениям которого данные упорядочены в соответствии с правилами сравнения данных от наименьшего до наибольшего значения ключа.

Время выполнения — время, в которое происходит выполнение объектной программы.

Время компиляции — время, в которое происходит компиляция исходной Кобол-программы в объектную программу.

Входной файл — файл, открытый в режиме ввода.

Входной-выходной файл — файл, открытый в режиме ввода-вывода.

Вызываемая программа — программа, являющаяся объектом оператора CALL (ВЫЗВАТЬ), объединяемая во время выполнения с вызывающей программой для образования единицы исполнения.

Вызывающая программа — программа, выполняющая оператор CALL (ВЫЗВАТЬ) по отношению к другой программе.

Выражение — арифметическое или условное выражение.

Выходной файл — файл, открытый в режиме вывода или в режиме дополнения.

Глагол — слово, обозначающее действие, которое нужно произвести компилятору Кобола или объектной программе.

Глобальное имя — имя, объявленное только в одной программе, но на которое можно ссылаться из этой программы и из любой программы, содержащейся в ней. Глобальными именами могут быть имена-условий, имена-данных, имена-файлов, имена-записей, имена-отчетов и некоторые специальные регистры (ч. 10, пп. 1.3.8.2, 4.2.4, 4.4.4).

Группа отчета — порция отчета, описанная в секции отчетов раздела данных статьей с номером уровня 01 и подчиненными ей статьями.

Группа тела отчета — общее имя для группы отчета типа DETAIL (ФРАГМЕНТ), CONTROL FOOTING (УПРАВЛЯЕМАЯ

GOST EXPERT ГОСТ 22558-89,

КОНЦОВКА) или CONTROL HEADING (УПРАВЛЯЕМЫЙ ЗАГОЛОВОК).

Групповое данное — данное, состоящее из подчиненных данных.

Данное — единица данных (за исключением литералов), определенная Кобол-программой.

Данное-источник — данное, идентификатор которого указан во фразе SOURCE (ИСТОЧНИК) и которое поставляет значение печатаемого данного.

Декларативное предложение — управляющее компиляцией предложение, состоящее из единственного оператора USE (ИСПОЛЬЗОВАТЬ), который заканчивается точкой с последующим пробелом.

Декларативы — набор из одной или более секций специального назначения, записанных в начале раздела процедур; первой из этих секций предшествует ключевое слово DECLARATIVES (ДЕКЛАРАТИВЫ), а за последней из них следуют ключевые слова END DECLARATIVES (КОНЕЦ ДЕКЛАРАТИВ). Каждая из этих секций определяется заголовком секции, за которым следует управляющий компиляцией оператор USE (ИСПОЛЬЗОВАТЬ) и далее нуль, один или несколько параграфов.

Дередактирование — логическое удаление всех литер редактирования из числового редактируемого данного с тем, чтобы получить это значение как числовое нередактируемое.

Динамический доступ — метод доступа, при котором отдельные логические записи могут быть получены из файла или помещены в файл массовой памяти непоследовательным образом и могут быть получены из файла последовательным образом в области действия одного и того же оператора OPEN (ОТКРЫТЬ) (см. произвольный доступ; последовательный доступ).

Дополнительный ключ записи — ключ, отличный от основного ключа записи, значение которого идентифицирует запись в индексном файле.

Единица исполнения — множество из одной или нескольких программ, которые функционируют во время выполнения как одно целое для обеспечения решения проблемы.

Заголовок конца программы — комбинация слов, заканчивающаяся разделителем точка и указывающая конец исходной Кобол-программы. Заголовок конца программы имеет вид:

END PROGRAM имя-программы.

КОНЕЦ ПРОГРАММЫ имя-программы.

Заголовок отчета — группа отчета, которая представляется только в начале отчета.

Заголовок параграфа — зарезервированное слово, за которым непосредственно следует разделитель точка и которое указывает

начало параграфа в разделах идентификации и оборудования. Допустимыми заголовками параграфов являются:

а) в разделе идентификации

PROGRAM-ID. (ПРОГРАММА.)
AUTHOR. (АВТОР.)
INSTALLATION. (ПРЕДПРИЯТИЕ.)
DATE-WRITTEN. (ДАТА-НАПИСАНИЯ.)
ДАТЕ-COMPILED. (ДАТА-ТРАНСЛЯЦИИ.)
SECURITY. (ПОЛНОМОЧИЯ.)

б) в разделе оборудования

SOURCE-COMPUTER. (ИСХОДНАЯ-МАШИНА.)
OBJECT-COMPUTER. (РАБОЧАЯ-МАШИНА.)
SPECIAL-NAMES. (СПЕЦИАЛЬНЫЕ-ИМЕНА.)
FILE-CONTROL. (УПРАВЛЕНИЕ-ФАЙЛАМИ.)
I-O-CONTROL. (УПРАВЛЕНИЕ-ВВОДОМ-ВЫВОДОМ.)

Заголовок раздела — комбинация слов, оканчивающаяся разделителем точка и указывающая начало раздела. Заголовки разделов Кобол-программы следующие:

IDENTIFICATION DIVISION.
ENVIRONMENT DIVISION.
DATA DIVISION.
PROCEDURE DIVISION [USING {имя-данного-1} . . .].
РАЗДЕЛ ИДЕНТИФИКАЦИИ.
РАЗДЕЛ ОБОРУДОВАНИЯ.
РАЗДЕЛ ДАННЫХ.
РАЗДЕЛ ПРОЦЕДУР [ИСПОЛЬЗУЯ {имя-данного-1} . . .].

Заголовок секции — комбинация слов, за которыми следует разделитель точка; эта комбинация указывает начало секции в разделах оборудования, данных и процедур. Допустимые заголовки секций следующие:

а) в разделе оборудования

CONFIGURATION SECTION.
(СЕКЦИЯ КОНФИГУРАЦИИ.)
INPUT-OUTPUT SECTION.
(СЕКЦИЯ ВВОДА-ВЫВОДА.)

б) в разделе данных:

FILE SECTION.
(СЕКЦИЯ ФАЙЛОВ.)
WORKING-STORAGE SECTION.
(СЕКЦИЯ РАБОЧЕЙ-ПАМЯТИ.)
LINKAGE SECTION.
(СЕКЦИЯ СВЯЗИ.)
COMMUNICATION SECTION.
(СЕКЦИЯ КОММУНИКАЦИЙ.)

ГОСТ 22558-89,

REPORT SECTION.
(СЕКЦИЯ ОТЧЕТОВ.)

в) в разделе процедур заголовок секции состоит из зарезервированного слова SECTION (СЕКЦИЯ), перед (за) которым следует имя секции, далее номер сегмента (необязательно) и, наконец, разделитель точка.

Заголовок страницы — группа отчета, которая представляется в начале страницы отчета, как это определено системой управления генератором отчетов.

Запись — наиболее объемлющее данное. Номер уровня записи равен 01. Запись может быть элементарным или групповым данным. Термин является синонимом термина логическая запись.

Запись переменной длины — запись файла, статья описания которого позволяет записям содержать переменное число позиций литер.

Запись фиксированной длины — запись, соответствующая файлу, статья описания файла или статья описания сортируемого-сливаемого файла которого требует, чтобы все записи содержали одно и то же количество литер.

Зарезервированное слово — слова Кобола из фиксированного списка слов, которые могут быть использованы в исходных Кобол-программах только в определенном смысле и которые не могут использоваться как слова, определенные пользователем, или как системные имена.

Знак арифметической операции — одна литера или фиксированная двухлитерная комбинация, которая принадлежит следующему множеству:

Знак	Операция
+	сложение
—	вычитание
*	умножение
/	деление
**	возведение в степень

Знак логической операции — одно из зарезервированных слов: AND (И), OR (ИЛИ) или NOT (НЕ). При формировании условий оба или любое из AND (И) или OR (ИЛИ) могут использоваться как логические связки. NOT (НЕ) может использоваться для логического отрицания.

Знак операции отношения — зарезервированное слово, литера отношения, группа последовательных зарезервированных слов или группа последовательных зарезервированных слов и литер отношения, используемые при построении условий отношения. Допустимыми знаками операций отношения являются:

GOST EXPERT ГОСТ 22558-89,

Знак	Операция
IS [NOT] GREATER THAN ([НЕ] БОЛЬШЕ) IS [NOT] > ([НЕ] >)	Больше чем или не больше чем
IS NOT LESS THAN ([НЕ] МЕНЬШЕ) IS [NOT] < ([НЕ] <)	Меньше чем или не меньше чем
IS NOT EQUAL TO ([НЕ] РАВНО) IS NOT = [НЕ] =)	Равно или не равно
IS GREATER THAN OR EQUAL TO (БОЛЬШЕ ИЛИ РАВНО) IS >= (>=)	Больше чем или равно
IS LESS THAN OR EQUAL TO (МЕНЬШЕ ИЛИ РАВНО) IS <= (<=)	Меньше чем или равно

Знак унарной операции — плюс (+) или минус (—), находящийся перед переменной или левой скобкой арифметического выражения и который равносилен умножению выражения на +1 или —1 соответственно.

Знак числа — алгебраический знак, связанный с числовым данным или числовым литералом для указания положительного или отрицательного его значения.

Значение истинности — представление результата вычисления условного выражения в терминах одного из двух значений: «истина» или «ложь».

Идентификатор — синтаксически правильная комбинация имени-данного и его уточнителей, индексов и модификаторов ссылки, если они требуются для однозначности ссылки, именующая данное. Тем не менее правила для «идентификатора» в общих форматах в особых случаях могут запрещать уточнение, индексирование или модификацию ссылок.

Идентификатор результата — определенное пользователем данное для хранения результата арифметической операции.

GOST EXPERT ГОСТ 22558-89,

Иерархия управления — определенная последовательность подразделов отчета, указанная порядком перечисления варианта FINAL (ПО КОНЦУ) и имен данных во фразе CONTROL (УПРАВЛЕНИЕ).

Имя алфавита — определенное пользователем слово в параграфе SPECIAL-NAMES (СПЕЦИАЛЬНЫЕ-ИМЕНА) раздела оборудования, которое именует определенный набор литер и (или) основную последовательность (ч. 6, п. 4.5).

Имя библиотеки — определенное пользователем слово, которое именует библиотеку Кобола, используемую компилятором при компиляции данной исходной программы.

Имя данного — определенное пользователем слово, которое именует данное, описанное в статье описания данного. В общих форматах «имя-данного» представляет слово, которое не может ни уточняться, ни индексироваться, ни быть модифицированной ссылкой, если в правилах для этого формата нет специального разрешения.

Имя записи — определенное пользователем слово, которое именует запись, описанную статьей описания записи в разделе данных Кобол-программы.

Имя индекса — определенное пользователем слово, именующее позицию или регистр памяти, связанные с конкретной таблицей.

Имя класса — слово, определенное пользователем в параграфе SPECIAL-NAMES (СПЕЦИАЛЬНЫЕ-ИМЕНА) раздела оборудования, предоставляющее имя выражению, для которого может быть определено значение истинности того, что данное состоит исключительно из тех литер, которые указаны в определении имени-класса.

Имя коммуникации — определенное пользователем слово, именующее область взаимодействия с системой управления сообщениями и описанное в статье описания коммуникации в секции коммуникаций раздела данных.

Имя машины — системное имя, идентифицирующее машину, на которой должна компилироваться или исполняться программа.

Имя отчета — определенное пользователем слово, которое именует отчет, описываемый статьей описания отчета в секции отчетов раздела данных.

Имя очереди — символическое имя, указывающее системе управления сообщениями логический путь, по которому может быть доступно сообщение или часть законченного сообщения в очереди.

Имя параграфа — определенное пользователем слово, которое идентифицирует и начинает параграф в разделе процедур.

Имя программы — определенное пользователем слово, которое идентифицирует исходную Кобол-программу в разделе идентификации и заголовке конца программы.

GOST EXPERT ГОСТ 22558-89,

Имя программного модуля или имя модуля — определенное пользователем слово, которое идентифицирует процедуру, записанную на языке, отличном от Кобола.

Имя процедуры — определенное пользователем слово, используемое для именования параграфа или секции в разделе процедур. Имя процедуры может быть именем параграфа, возможно уточненным, или именем секции.

Имя реализации — системное имя, относящееся к определенным особенностям, имеющимся в распоряжении вычислительной системы данной реализации.

Имя секции — определенное пользователем слово, которое именует секцию в разделе процедур.

Имя текста — определенное пользователем слово, которое именует библиотечный текст.

Имя условия — определенное пользователем слово, поставленное в соответствие специальному значению, набору значений или некоторому диапазону значений внутри полного множества значений, которые может принимать условная переменная, или определенное пользователем слово, присвоенное состоянию определенного реализацией переключателя или устройства.

Когда используется «имя-условия» в общих форматах, оно представляет уникальную ссылку на данное, состоящую из синтаксически правильной комбинации имени-условия вместе с уточнителями или индексами, как того требует однозначность ссылки.

Имя файла — определенное пользователем слово, которое именует определитель файла, описываемый статьей описания файла или сортируемого-сливаемого файла в секции файлов раздела данных.

Имя языка — системное имя, указывающее определенный язык программирования.

Индекс — номер вхождения, идентифицирующий отдельный элемент таблицы. Индекс может быть представлен целым, именем-данного, за которым может следовать целое со знаком + или —, или именем-индекса, за которым может следовать целое со знаком + или —.

Индексируемое имя данного — идентификатор, состоящий из имени данного, за которым следует один или несколько индексов, заключенных в скобки.

Индексная организация — постоянная логическая структура файла, при которой каждая запись идентифицируется значением одного или более ключей внутри этой записи.

Индексное данное — данное, в котором в определенной реализацией форме может запоминаться значение, связанное с именем индекса.

Индексный файл — файл с индексной организацией.

GOST EXPERT ГОСТ 22558-89,

Индикатор уровня — две буквенные литеры, определяющие специальный тип файла или позицию в иерархии. Индикаторами уровня в разделе данных являются FD (ОФ), CD (ОК), RD (ОО), SD (ОС).

Индикаторы сообщения. EGI (ИКГ) — индикатор конца группы, EMI (ИКЩ) — индикатор конца сообщения и ESI (ИКС) — индикатор конца сегмента являются логическими индикаторами, указывающими системе управления сообщениями на определенное условие (конца группы, конца сообщения, конца сегмента).

В иерархии EGI (ИКГ), EMI (ИКЩ) и ESI (ИКС) логически EGI (ИКГ) является эквивалентом ESI (ИКС), EMI (ИКЩ) и EGI (ИКГ). Логически EMI (ИКЩ) эквивалентно ESI (ИКС) и EMI (ИКЩ). Таким образом, сегмент может заканчиваться ESI (ИКС), EMI (ИКЩ) или EGI (ИКГ). Сообщение может заканчиваться EMI (ИКЩ) или EGI (ИКГ).

Источник — символическое обозначение инициатора передачи сообщения в очередь.

ИСХОДНАЯ-МАШИНА — имя параграфа в разделе оборудования, описывающего компьютерную среду, в которой компилируется исходная программа.

Исходная программа — синтаксически правильный набор операторов Кобола, начинающийся с раздела идентификации, оператора COPY (КОПИРОВАТЬ) или оператора REPLACE (ЗАМЕНИТЬ). Кобол-программа заканчивается заголовком конца программы, если он определен, или отсутствием дальнейших строк исходной программы.

Катушка — отдельная часть запоминающей среды, размеры которой определяются реализацией, содержащая часть файла, файл или несколько файлов. Термин является синонимом тома и устройства.

Ключ — данное, идентифицирующее местоположение записи, или набор данных, идентифицируюших упорядоченность данных.

Ключ записи — ключ, содержимое которого идентифицирует запись в индексном файле. В индексном файле ключ может быть основным или дополнительным.

Ключ ссылки — ключ, основной или дополнительный, используемый для доступа к записям индексного файла в текущий момент времени выполнения.

Ключевое слово — зарезервированное слово, вхождение которого в формат, содержащий это слово, обязательно при использовании этого формата в исходной программе.

Комбинированное условие — условие, полученное посредством связывания двух или более условий при помощи знаков логических операций AND (И) или OR (ИЛИ).

GOST EXPERT ГОСТ 22558-89,

Коммуникационное устройство — механизм (оборудование или оборудование и системные программы), обеспечивающий отправку данных в очередь сообщений и (или) получение данных из этой очереди. Эти функции может выполнять как вычислительная машина, так и периферийное устройство. Одна или несколько программ некоторой вычислительной машины, содержащих статьи описания коммуникаций, определяют один или несколько таких механизмов.

Конец младшего порядка — самая правая литера строки литер.

Конец раздела процедур — физическая позиция в исходной Кобол-программе, за которой не следуют никакие процедуры.

Конец старшего порядка — самая левая литера в строке литер.

Концовка отчета — группа отчета, которая представляется только в конце отчета.

Концовка страницы — группа отчета, которая представляется в конце страницы отчета, как это определено системой управления генератором отчетов.

Литера — основная неделимая единица языка.

Литера заполнитель — литера, используемая для заполнения неиспользуемых позиций литер в физической записи.

Литера отношения — литера, которая принадлежит следующему множеству:

Литера	Значение
>	больше
<	меньше
=	равно

Литера пунктуации — литера, которая принадлежит следующему набору литер:

Литера	Значение
,	запятая
:	двоеточие
;	точка с запятой
.	точка
"	кавычки
(	левая круглая скобка
)	правая круглая скобка
	пробел
=	равно

Литера редактирования — отдельная литера или двухлитерная комбинация в строке литер шаблона, принадлежащая следующему набору литер:

GOST EXPERT ГОСТ 22558-89,

Литера	Значение
	пробел
0	нуль
+	плюс
—	минус
CR (КР)	кредит
DB (ДБ)	дебет
Z (П)	подавление нулей
*	защита чека
,	запятая (десятичная точка)
.	точка (десятичная точка)
$ (¤)	валютный знак
/	дробная черта

Литерал — строка литер, значение которой определяется упорядоченным набором литер, составляющих эту строку.

Логическая запись — наиболее объемлющее данное. Номер уровня записи равен 01. Запись может быть элементарным или групповым данным. Термин является синонимом записи.

Логическая запись генератора отчетов — запись, состоящая из печатаемой генератором отчета строки и соответствующей управляющей информации, необходимой для ее отбора и вертикального позиционирования.

Логическая страница — логическое понятие, состоящее из верхнего поля, тела страницы и нижнего поля.

Массовая память — среда памяти, в которой данные могут быть организованы и сохраняемы как в последовательной, так и в непоследовательной форме.

Метод доступа — метод обращения к записям в файле.

Мнемоническое имя — определенное пользователем слово, соотнесенное в разделе оборудования указанному имени реализации.

Модификатор ссылки — самая левая позиция литеры и длина, используемые для образования данного и ссылки на него (ч. 4. п. 4.3.8.3).

Набор литер Кобола — полный набор литер Кобола состоит из перечисленных ниже литер:

для английской нотации:

Литера	Значение
0, 1, 2, 3, 4, 5, 6, 7, 8, 9	цифра
A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T U, V, W, X, Y, Z	буква прописная
a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z	буква строчная

GOST EXPERT ГОСТ 22558-89,

	пробел
+	плюс
—	минус
*	звездочка
/	дробная черта
=	равно
,	запятая (десятичная точка)
;	точка с запятой
.	точка (десятичная точка)
"	кавычки
(	левая круглая скобка
)	правая круглая скобка
>	больше
<	меньше
$(¤)	валютный знак
:	двоеточие

Для русской нотации набор литер Кобола включает перечисленный выше набор литер Кобола для английской нотации, а также следующие литеры русского алфавита, не совпадающие по начертанию с буквами латинского алфавита:

Б, Г, Д, Ж, З, И, Й, Л, П, У, Ф, Ц, Ч, Ш, Щ, Ы, Ь, Э, Ю, Я	буква прописная
б, в, г, д, ж, з, и, й, к, л, м, н, п, т, ф, ц, ч, ш, щ, ы, ь, э, ю, я	буква строчная

Начальная программа — программа, которая приводится в наначальное состояние каждый раз при вызове ее в единице исполнения.

Начальное состояние — состояние программы при первом вызове ее в единице исполнения (ч. 10, п. 2.4).

Необязательное слово — зарезервированное слово, включаемое в конкретный формат по желанию пользователя для придания тексту удобочитаемости. Присутствие такого слова в формате необязательно.

Необязательный файл — файл, который не обязательно должен иметься в наличии во время выполнения объектной программы. Объектная программа производит запрос о наличии или отсутствии файла.

Несвязанные данные — элементарные данные в секции рабочей памяти и в секции связи, не имеющие иерархических взаимосвязей с какими-либо другими данными.

Неуспешное завершение — попытка выполнить оператор, в результате которой выполняются не все действия, определенные этим

GOST EXPERT ГОСТ 22558-89,

оператором. Неуспешное завершение оператора не изменяет данные, на которые есть ссылки в операторе, но может изменить индикаторы состояния.

Нечисловое данное — данное, описанное таким образом, что его значение может быть любой комбинацией литер из набора литер вычислительной машины. Определенные категории нечисловых данных могут формироваться из более ограниченного набора литер.

Нечисловой литерал — строка литер, ограниченная слева и справа кавычками. Строка литер может включать любую литеру из набора литер машины.

Неявный ограничитель области действия — разделитель точка, заканчивающая область действия предшествующего незавершенного оператора, или фраза оператора, указывающая на конец области действия любого оператора, содержащегося в предыдущей фразе.

Нижнее поле — пустое поле, следующее за телом страницы.

Номер записи — порядковый номер записи в файле с последовательной организацией.

Номер сегмента — определенное пользователем слово, которое классифицирует секции в разделе процедур для целей сегментации. Номера сегмента могут состоять только из цифр 0, 1, ..., 9 и могут быть выражены одной или двумя цифрами.

Номер строки — целое, обозначающее вертикальную позицию строки отчета на странице.

Номер уровня — определенное пользователем слово, состоящее из одной или двух цифр, которое указывает позицию данного в иерархической структуре логической записи или специальные свойства статьи описания данного. Номера уровней в диапазоне от 1 до 49 указывают позицию данного в иерархической структуре логической записи. Номера уровней от 1 до 9 могут быть представлены одной цифрой или цифрой с предшествующим нулем. Номера уровней 66, 77 и 88 идентифицируют специальные свойства статьи описания данного.

Область записи — область памяти, отведенная для целей обработки записи, описанной в статье описания записи секции файлов раздела данных. Текущее число позиций литер в области записи определяется явной или неявной фразой RECORD (В ЗАПИСИ) в секции файлов.

Общая программа — программа, которую, несмотря на то, что она непосредственно содержится в другой программе, можно вызвать из любой программы, прямо или косвенно содержащейся в этой другой программе.

Объект статьи — набор операндов и зарезервированных слов в

статье раздела данных Кобол-программы, следующий непосредственно за субъектом статьи.

Объектная программа — совокупность выполнимых машинных команд и других средств, предназначенных для взаимодействия с данными, обеспечивающая решение проблемы. Объектная программа обычно является результатом обработки исходной программы компилятором Кобола. Там, где нет опасности неоднозначности, слово «объектная программа» может заменяться словом «программа».

Ограничитель — литера или последовательность смежных литер, которая идентифицирует конец строки литер и отделяет эту строку литер от следующей строки литер. Ограничитель не является частью строки литер, которую он ограничивает.

Ограничитель псевдотекста — две последовательные литеры = (равно), используемые для ограничения псевдотекста.

Операнд — компонента, над которой производится действие. В настоящем документе любое слово (или слова), записанное строчными буквами в формате оператора или статьи, рассматривается как операнд и как таковой представляет ссылку на данное, указанное операндом.

Оператор — синтаксически правильная комбинация слов, литералов и разделителей, начинающаяся глаголом и записанная в исходной Кобол-программе.

Оператор ввода-вывода — оператор, побуждающий к обработке файлов выполнением операций над отдельными записями или над всем файлом. Операторы ввода-вывода: ACCEPT (ПРИНЯТЬ) — с вариантом идентификатор, CLOSE (ЗАКРЫТЬ), DELETE (УДАЛИТЬ), DISABLE (ЗАПРЕТИТЬ), DISPLAY (ВЫДАТЬ), ENABLE (РАЗРЕШИТЬ), OPEN (ОТКРЫТЬ), PURGE (ОЧИСТИТЬ), READ (ЧИТАТЬ), RECEIVE (ПОЛУЧИТЬ), REWRITE (ОБНОВИТЬ), SEND (ПОСЛАТЬ), SET (УСТАНОВИТЬ) — с вариантом TO ON (ВКЛЮЧЕНО) или TO OFF (ВЫКЛЮЧЕНО), START (ПОДВЕСТИ) и WRITE (ПИСАТЬ).

Оператор ветвления процедуры — оператор, совершающий явную передачу управления не следующему выполнимому оператору в последовательности операторов исходной программы. Операторы ветвления процедур: ALTER (ИЗМЕНИТЬ), CALL (ВЫЗВАТЬ), EXIT (ВЫЙТИ), EXIT PROGRAM (ВЫЙТИ ИЗ ПРОГРАММЫ), GO TO (ПЕРЕЙТИ К), MERGE (СЛИТЬ) — с вариантом OUTPUT PROCEDURE (ПРОЦЕДУРА ВЫВОДА), PERFORM (ВЫПОЛНИТЬ) и SORT (СОРТИРОВАТЬ) — с вариантом INPUT PROCEDURE (ПРОЦЕДУРА ВВОДА) или OUTPUT PROCEDURE (ПРОЦЕДУРА ВЫВОДА).

GOST EXPERT ГОСТ 22558-89,

Оператор с ограничителем области действия — любой оператор, включающий явный ограничитель области действия (ч. 4, п. 4.4.3).

Оператор, управляющий компиляцией — оператор, вызывающий выполнение компилятором специальных действий во время компиляции.

Описание записи — совокупность статей описания данных, относящихся к отдельной записи. Термин является синонимом термина статья описания записи.

Определенное пользователем слово — слово Кобола, которое должно быть задано пользователем в соответствии с форматом фразы или оператора.

Определитель файла — область памяти, содержащая информацию о файле и используемая для связи между именем-файла и физическим файлом и между именем-файла и соответствующей областью записи.

Организация файла — постоянная логическая структура файла, устанавливаемая при создании файла.

Основная последовательность — последовательность, в которой воспринимаемые машиной литеры упорядочены для целей сортировки, слияния или сравнения и для последовательной обработки индексных файлов.

Основной ключ записи — ключ, содержимое которого однозначно идентифицирует запись в индексном файле.

Отдельно компилируемая программа — программа, компилируемая со всеми содержащимися в ней программами отдельно от всех остальных программ.

Отладочная секция — секция Кобол-программы, содержащая оператор USE FOR DEBUGGING (ИСПОЛЬЗОВАТЬ ДЛЯ ОТЛАДКИ).

Отладочная строка — строка Кобол-программы, содержащая литеру D (T) в поле индикатора строки.

Относительная организация — постоянная логическая структура файла, в которой каждая запись единственным образом идентифицируется целой положительной величиной, указывающей порядковую позицию записи в файле.

Относительный ключ — ключ, значение которого идентифицирует логическую запись в относительном файле.

Относительный номер записи — порядковый номер записи в файле с относительной организацией. Этот номер рассматривается как числовой литерал, являющийся целым.

Относительный файл — файл с относительной организацией.

Отношение. Термин является синонимом знака операции отношения.

Отрицание комбинированного условия — знак логической операции NOT (НЕ), непосредственно за которым следует в скобках комбинированное условие.

Отрицание простого условия — знак логической операции NOT (НЕ), непосредственно за которым следует простое условие.

Очередь — логический набор сообщений, ожидающих передачи или обработки.

Параграф — в разделе процедур это имя параграфа, за которым следует разделитель точка и нуль, одно или несколько предложений. В разделе идентификации и разделе оборудования — это заголовок параграфа, за которым следует нуль, одна или несколько статей.

Переменная — данное, значение которого может изменяться при выполнении объектной программы. Переменные, используемые в арифметическом выражении, должны быть числовыми элементарными данными.

Переменно повторяющееся данное — переменно повторяющееся данное является табличным элементом, повторяющимся переменное число раз. В статье описания такого данного должна содержаться фраза OCCURS DEPENDING ON (ПОВТОРЯЕТСЯ В ЗАВИСИМОСТИ ОТ) или оно должно входить в другое данное с такой статьей описания.

Печатаемая группа — группа отчета, содержащая по крайней мере одну печатаемую строку.

Печатаемое данное — данное, размер и содержимое которого указаны элементарной статьей отчета, содержащей фразы COLUMN NUMBER (НОМЕР СТОЛБЦА), PICTURE (ШАБЛОН), а также SOURCE (ИСТОЧНИК), SUM (СУММА) или VALUE (ЗНАЧЕНИЕ).

Повелительный оператор — оператор, который начинается с повелительного глагола, указывающий безусловное действие, которое нужно выполнить, или условный оператор с явным ограничителем области действия. Повелительный оператор может состоять из последовательности повелительных операторов.

Подочередь — логическое иерархическое подразделение очереди.

Подпрограмма — программа, являющаяся объектом оператора CALL (ВЫЗВАТЬ), объединяемая в рабочее время с вызывающей программой для образования единицы исполнения. Термин является синонимом термина вызываемая программа.

Подразумеваемая десятичная точка — позиция десятичной точки в данном, которой не соответствует явное физическое представление литеры точки.

Позиция литеры — объем физической памяти, необходимый для хранения одной литеры стандартного формата данных, описанных

GOST EXPERT ГОСТ 22558-89,

DISPLAY (ДЛЯ ВЫДАЧИ). Более подробные характеристики физической памяти определяются реализацией.

Позиция цифры — объем физической памяти, необходимый для хранения одной цифры. Этот объем может изменяться в зависимости от фразы об использовании в статье описания данного, определяющей данное. Если в статье описания данного указана фраза DISPLAY (ДЛЯ ВЫДАЧИ), позиция цифры является синонимом позиции литеры. Более подробные характеристики физической памяти определяются реализацией.

Поле концовки — позиция тела страницы, примыкающая к нижней границе.

Последовательная организация — постоянная логическая структура файла, при которой записи идентифицированы отношением «предшественник-преемник», установленным при занесении записи в файл.

Последовательный доступ — метод доступа, при котором логические записи извлекаются из файла или размещаются в нем в последовательности, упорядоченной отношением «предшественник-преемник», устанавливаемым при создании файла.

Последовательный файл — файл с последовательной организацией.

Предложение — последовательность из одного или нескольких операторов, последний из которых заканчивается разделителем точкой.

Прерывание управления — изменение значения данного, указанного во фразе CONTROL (УПРАВЛЕНИЕ), используемое для управления иерархической структурой отчета.

Произвольный доступ — метод доступа, при котором указанное программой значение ключа идентифицирует логическую запись, получаемую или удаляемую из относительного или индексного файла или помещаемую в него.

Простое условие — любое одиночное условие, принадлежащее следующему множеству:

условие отношения;
условие класса;
условие имени условия;
условие состояния переключателя;
условие знака;
(простое условие).

Процедура — параграф или группа логически последовательных параграфов либо секция или группа логически последовательных секций в разделе процедур.

Процедура ввода — набор операторов, которому передается управление во время выполнения оператора SORT (СОРТИРОВАТЬ) для управления передачей сортируемых записей.

GOST EXPERT ГОСТ 22558-89,

Процедура вывода — набор операторов, которым передается управление во время выполнения оператора SORT (СОРТИРОВАТЬ) после завершения функции сортировки или во время выполнения оператора MERGE (СЛИТЬ) после завершения функции слияния.

Псевдотекст — последовательность слов текста, строк комментариев или разделителя пробел, заключенная в ограничители псевдотекста, причем последние не принадлежат псевдотексту. Псевдотекст представляет фрагмент исходной программы или фрагмент библиотечного текста.

OBJECT-COMPUTER (РАБОЧАЯ-МАШИНА) — имя параграфа в разделе оборудования, описывающего конфигурацию машины, на которой следует выполнять объектную программу.

Раздел—набор из нуля, одной или нескольких секций или параграфов, образующих тело раздела, который формируется и составляется в соответствии со специальными правилами. Каждый раздел состоит из заголовка раздела и соответствующего тела раздела. Имеется четыре раздела Кобол-программы: раздел идентификации, раздел оборудования, раздел данных и раздел процедур.

Разделитель — литера или две последовательные литеры, используемые для ограничения строк литер.

Режим ввода — состояние файла после выполнения оператора OPEN (ОТКРЫТЬ), указанного с вариантом INPUT (ВХОДНОЙ), для этого файла и до выполнения оператора CLOSE (ЗАКРЫТЬ) без варианта REEL (КАТУШКУ) или UNIT (ТОМ) для этого файла.

Режим ввода-вывода — состояние файла после выполнения оператора OPEN (ОТКРЫТЬ), указанного с вариантом I-O (ВХОДНОЙ-ВЫХОДНОЙ), для этого файла и до выполнения оператора CLOSE (ЗАКРЫТЬ) без варианта REEL (КАТУШКУ) или UNIT (ТОМ) для этого файла.

Режим вывода — состояние файла после выполнения оператора OPEN (ОТКРЫТЬ) для этого файла с вариантом OUTPUT (ВЫХОДНОЙ) или EXTEND (ДОПОЛНЯЕМЫЙ) и до выполнения оператора CLOSE (ЗАКРЫТЬ) без варианта REEL (КАТУШКУ) или UNIT (ТОМ) для этого файла.

Режим дополнения — состояние файла после выполнения для него оператора OPEN (ОТКРЫТЬ), указанного с вариантом EXTEND (ДОПОЛНЯЕМЫЙ) и до выполнения оператора CLOSE (ЗАКРЫТЬ) для этого же файла.

Режим открытия — состояние файла после выполнения оператора OPEN (ОТКРЫТЬ) для этого файла и до выполнения оператора CLOSE (ЗАКРЫТЬ) без варианта REEL (КАТУШКУ) или UNIT (ТОМ) для этого файла. Определенный режим открытия определяется в операторе OPEN (ОТКРЫТЬ) вариантом INPUT

GOST EXPERT ГОСТ 22558-89,

(ВХОДНОЙ), OUTPUT (ВЫХОДНОЙ), I-O (ВХОДНОЙ-ВЫХОДНОЙ) или EXTEND (ДОПОЛНЯЕМЫЙ) соответственно как режим открытия для ввода, вывода, ввода-вывода или дополнения.

Ресурс — возможность или средство, управляемые операционной системой, которые могут быть использованы выполняющейся программой.

Связанные данные — данные, описанные посредством последовательных статей раздела данных и имеющие определенную иерархическую взаимосвязь друг с другом.

Сегмент сообщения — логическое подразделение сообщения, обычно связанное с индикатором конца сегмента (см. индикаторы сообщения).

Секция — набор из нуля, одного или нескольких параграфов или статей, называемый телом секции, которому предшествует заголовок секции. Каждая секция состоит из заголовка секции и относящегося к нему тела секции.

Секция ввода-вывода — секция раздела оборудования, которая называет файлы и внешнюю среду, требуемые для объектной программы, и которая задает информацию, необходимую для передачи и обработки данных в процессе выполнения объектной программы.

Секция коммуникаций — секция раздела данных, состоящая из одной или нескольких статей описания коммуникаций CD (ОК) и описывающая области взаимодействия между системой управления сообщениями и программой.

Секция конфигурации — секция раздела оборудования, которая описывает общие спецификации исходной и рабочей программ.

Секция отчетов — секция раздела данных, содержащая статьи описания составляемых Кобол-программой отчетов и связанные с ними статьи описания групп отчетов.

Секция рабочей памяти — секция раздела данных, которая описывает данные рабочей памяти, состоящие из несвязанных данных, либо из записей рабочей памяти, либо из тех и других.

Секция связи — секция в разделе данных вызываемой программы, описывающая данные, доступные из вызывающей программы. К этим данным можно обращаться как вызывающей программе, так и вызываемой.

Секция файлов — секция раздела данных, которая содержит статьи описания файлов и статьи описания сортируемых-сливаемых файлов вместе с описаниями соответствующих им записей.

Символическая литера — определенное пользователем слово, указывающее определенную пользователем стандартную константу.

GOST EXPERT ГОСТ 22558-89,

Система управления генератором отчетов (СУГО) — обеспечиваемая реализацией система управления, которая осуществляет построение отчетов во время выполнения.

Система управления массовой памятью (СУМП) — система управления вводом-выводом, которая управляет обработкой файлов в массовой памяти.

Система управления сообщениями — система управления коммуникациями, которая поддерживает обработку сообщений.

Системное имя — слово Кобола, используемое для связи с операционной средой.

Следующая запись — запись, которая логически следует за текущей записью файла.

Следующее выполнимое предложение — следующее предложение, которому будет передано управление после завершения выполнения текущего оператора (ч. 4, п. 4.4.2).

Следующий выполнимый оператор — следующий оператор, которому будет передано управление после завершения выполнения текущего оператора (ч. 4, п. 4.4.2).

Сливаемый файл — совокупность записей, подлежащих слиянию посредством выполнения оператора MERGE (СЛИТЬ). Сливаемый файл создается и может использоваться только при выполнении функции слияния.

Слово — строка литер, состоящая не более чем из 30 литер и образующая определенное пользователем слово, системное имя или зарезервированное слово (ч. 4, п. 4.2.2.1).

Слово Кобола — строка литер, состоящая не более чем из 30 литер и образующая определенное пользователем слово, системное имя или зарезервированное слово (ч. 4, п. 4.2.2.1).

Слово-специальная литера — зарезервированное слово, являющееся знаком арифметической операции или литерой отношения.

Слово текста — литера или непрерывная последовательность литер между полем A и полем R в библиотеке Кобола, исходной программе или в псевдотексте.

Словом текста являются:

(1) разделитель за исключением пробела, ограничителя псевдотекста, открывающего и закрывающего ограничителя нечислового литерала. Правая и левая скобки, независимо от контекста в библиотеке, исходной программе или псевдотексте, всегда рассматриваются как слова текста;

(2) нечисловой литерал вместе с ограничивающими его открывающими и закрывающими кавычками или числовой литерал;

(3) любая другая непрерывная последовательность литер Кобола, за исключением строк комментария и слова COPY (КОПИРОВАТЬ), ограниченная разделителями и не являющаяся разделителем или литералом.

GOST EXPERT ГОСТ 22558-89,

Сложное условие — условие, в котором одно или несколько условий связаны одной или несколькими логическими операциями (см. отрицание простого условия; комбинированное условие; отрицание комбинированного условия).

Сокращенное комбинированное условие отношения — комбинированное условие, в котором опущены общие для последовательности образующих его условий субъекты, и, возможно, знаки операции отношения.

Сообщение — данные, с которыми связан индикатор конца сообщения или индикатор конца группы (см. индикаторы сообщения).

Сортируемый файл — совокупность записей, подлежащих сортировке посредством оператора SORT (СОРТИРОВАТЬ). Сортируемый файл создается и может быть использован только функцией сортировки.

Состояние ввода-вывода. Логический объект, содержащий двухлитерное значение, указывающее заключительное состояние операции ввода-вывода. Это значение, однако, доступно программе при использовании фразы FILE STATUS (СОСТОЯНИЕ ФАЙЛА) в статье управления файлами.

Специальная литера — литера, которая принадлежит следующему набору литер:

Литера	Значение
+	плюс
—	минус
*	звездочка
/	дробная черта
=	равно
,	запятая (десятичная точка)
;	точка с запятой
.	точка (десятичная точка)
"	кавычки
(	левая круглая скобка
)	правая круглая скобка
>	больше
<	меньше
$(¤)	валютный знак
:	двоеточие

SPECIAL-NAMES (СПЕЦИАЛЬНЫЕ-ИМЕНА) — имя параграфа раздела оборудования, в котором имена реализации соотнесены с мнемоническими именами, определенными пользователем.

Специальные регистры — области памяти вычислительной машины, учреждаемые компилятором Кобола и используемые для

GOST EXPERT ГОСТ 22558-89,

запоминания служебной информации при выполнении специфических функций Кобола.

Стандартная константа — генерируемое компилятором значение, на которое ссылаются посредством определенных зарезервированных слов (ч. 4, п. 4.2.2.2.3).

Стандартный формат данного — концепция, используемая при описании данных в разделе данных Кобола, согласно которой характеристики или свойства данных выражены в форме, ориентированной на появление данного на печатаемой странице неопределенной длины и ширины, а не в форме, ориентированной на способ размещения данного в памяти вычислительной машины или на некотором внешнем носителе.

Статья — набор последовательных фраз описания в разделе идентификации, разделе оборудования или разделе данных исходной программы, заканчивающийся разделителем точкой.

Статья имени программы — статья параграфа PROGRAM-ID (ПРОГРАММА) раздела идентификации, содержащая фразы, которые определяют имя программы и назначают программе избранные свойства.

Статья исходной машины — статья параграфа SOURCE-COMPUTER (ИСХОДНАЯ-МАШИНА) в разделе оборудования, содержащая фразы с описанием компьютерной среды, в которой будет компилироваться исходная программа.

Статья-комментарий — статья в разделе идентификации, которая может представлять собой любую комбинацию литер из набора литер машины.

Статья описания группы отчета — статья в секции отчетов раздела данных, состоящая из номера уровня 01, необязательного имени данного, фразы TYPE (ТИП) и необязательного набора фраз отчета.

Статья описания данного — статья в разделе данных, состоящая из номера уровня, за которым следует, если требуется, имя данного, а затем, если требуется, набор фраз данных определенного вида.

Статья описания записи — совокупность статей описания данных, содержащихся в записи. Термин является синонимом термина описание записи.

Статья описания коммуникации — статья секции коммуникаций раздела данных, состоящая из индикатора уровня CD (ОК), за которым следует имя коммуникации и, если требуется, ряд фраз описания. Эта статья описывает интерфейс между системой управления сообщениями и Кобол-программой.

Статья описания отчета — статья в секции отчетов раздела данных, состоящая из индикатора уровня RD (00), за которым следует имя отчета и затем, если это требуется, набор фраз отчета.

GOST EXPERT ГОСТ 22558-89,

Статья описания сортируемого-сливаемого файла — статья в секции файлов раздела данных, которая состоит из индикатора уровня SD (ОС), за которым следует имя файла, а затем набор фраз файла.

Статья описания уровня 77 — статья описания данного в секции рабочей памяти, описывающая несвязанное данное с номером уровня 77.

Статья описания файла — статья в секции файлов раздела данных, состоящая из индикатора уровня FD (ОФ), за которым следует имя файла и затем набор требуемых фраз.

Статья объектной машины — статья параграфа OBJECT-COMPUTER (РАБОЧАЯ-МАШИНА) раздела оборудования, содержащая фразы, которые описывают компьютерную среду, в которой будет выполняться объектная программа.

Статья специальных имен — статья параграфа SPECIAL-NAMES (СПЕЦИАЛЬНЫЕ-ИМЕНА) в разделе оборудования, обеспечивающая средства определения валютного знака; выбора десятичной точки; определения символических литер; установления соответствия между именами реализации и мнемоническими именами, определяемыми пользователем; установления соответствия между именами алфавитов и наборами литер или основными последовательностями и установления соответствия между именами классов и наборами литер.

Статья управления вводом-выводом — статья параграфа I-O—CONTROL (УПРАВЛЕНИЕ-ВВОДОМ-ВЫВОДОМ) в разделе оборудования, содержащая фразы с информацией, необходимой для передачи и обработки данных названных файлов во время выполнения программы.

Статья управления файлом — фраза SELECT (ДЛЯ) в разделе оборудования и все подчиненные ей фразы, которые объявляют соответствующие физические свойства файла.

Столбец — позиция литеры в печатаемой строке. Номера столбцов начинаются с единицы, определяющей самую левую позицию литеры в печатаемой строке, и увеличиваются на единицу при продвижении к самой правой позиции в этой строке.

Страница — вертикальное подразделение данных отчета, базирующееся на внутренних требованиях выдачи и (или) внешних характеристиках носителя отчета.

Строка — часть страницы, представляющая один горизонтальный ряд позиций литер. Каждая позиция литеры строки отчета выровнена по вертикали с соответствующей позицией литеры строки отчета, находящейся над ней. Строки отчета нумеруются от 1, изменяясь на 1, начиная сверху. Термин является синонимом строки отчета.

GOST EXPERT ГОСТ 22558-89

Строка комментария — строка исходной программы, содержащая литеру * (звездочка) в поле индикатора строки и любые литеры из набора литер машины в полях А и В. Строка комментария служит только для документации в программе. Специальная форма строки комментария, содержащая литеру / (дробная черта) в поле индикатора строки и любые литеры из набора литер машины в полях А и В, вызывают переход на следующую страницу перед печатью комментария.

Строка литер — последовательность смежных литер, образующая слово Кобола, литерал, строку литер шаблона или статью-комментарий (ч. 4, п. 4.2.2.1).

Строка отчета — часть страницы, представляющая один горизонтальный ряд позиций литер. Каждая позиция литеры строки отчета выровнена по вертикали с соответствующей позицией литеры строки отчета, находящейся над ней. Строки отчета нумеруются от 1, изменяясь на 1, начиная сверху. Термин является синонимом строки.

Структура данных внутри записи — полный набор групповых и элементарных данных записи, определяемый последовательным подмножеством статей описания данного, описывающих эту запись. Эти статьи описания данного включают все статьи с номером уровня, большим номера уровня первой статьи, описывающей структуру данных внутри записи.

Субъект статьи — операнд или зарезервированное слово, указанное непосредственно после индикатора уровня или номера уровня в статье раздела данных.

СУГО — система управления генератором отчетов; обеспечиваемая реализацией система управления времени выполнения, которая осуществляет построение отчетов.

СУМП — система управления массовой памятью; система управления вводом-выводом, которая управляет обработкой файлов в массовой памяти.

Счетчик — данное, используемое для хранения чисел или представлений чисел в виде, позволяющем эти числа увеличивать или уменьшать на значение другого числа, или изменять, или устанавливать в нуль или в произвольное положительное или отрицательное значение.

LINAGE-COUNTER (СЧЕТЧИК-ВЕРСТКИ) — специальный регистр, значение которого указывает текущую позицию в теле страницы.

Счетчик суммы — числовое данное со знаком, учреждаемое фразой SUM (СУММА) в секции отчетов раздела данных. Счетчик суммы используется системой управления генератором отчетов для хранения результатов операций суммирования, производимых при генерации отчета.

GOST EXPERT ГОСТ 22558-89,

Таблица — набор логически следующих друг за другом данных, которые определены в разделе данных Кобол-программы с помощью фразы OCCURS (ПОВТОРЯЕТСЯ).

Текущая запись — запись, доступная в области записи, соответствующей файлу.

Тело страницы — часть логической страницы, в которой могут быть представлены печатаемые строки и (или) оставлены пустые строки (ч. 7, п. 3.7).

Терминал — источник передачи в очередь или получатель передачи из очереди.

Том — отдельная часть запоминающей среды, размеры которой определяются реализацией и которая может содержать часть файла, весь файл или несколько файлов. Термин является синонимом термина катушка.

Убывающий ключ — ключ, по значениям которого данное упорядочено в соответствии с правилами для сравнения данных от наибольшего до наименьшего значения ключа.

Указатель позиции файла — логическое понятие, содержащее значение текущего ключа в ключе ссылки для индексного файла, или номер текущей записи для последовательного файла, или относительный номер текущей записи для относительного файла, или указывающее, что следующей логической записи не существует, или что число значащих цифр в относительном номере записи больше размера относительного ключа, или что необязательный входной файл отсутствует, или что уже существует условие AT END (В КОНЦЕ), или что не установлена существующая следующая запись.

Указатель текущего тома — логическое понятие, указывающее текущий том последовательного файла.

I-O-CONTROL (УПРАВЛЕНИЕ-ВВОДОМ-ВЫВОДОМ) — имя параграфа раздела оборудования, в котором указаны требования рабочей программы к точкам перепрогона, совместному использованию общих областей несколькими файлами и размещению нескольких файлов на одном устройстве ввода-вывода.

FILE-CONTROL (УПРАВЛЕНИЕ-ФАЙЛАМИ) — имя параграфа раздела оборудования, в котором объявляются имена файлов данных для конкретной исходной программы.

Управляемая группа — набор групп тела отчета, представляемый для данного значения управляющего данного или в конце отчета FINAL (ПО КОНЦУ). Каждая управляемая группа может начинаться группой управляемый заголовок, заканчиваться группой управляемая концовка и содержать в себе группы фрагмент.

Управляемая концовка — группа отчета, представляемая в конце управляемой группы, в которую она входит.

GOST EXPERT ГОСТ 22558-89,

Управляемый заголовок — группа отчета, представляемая в начале управляемой группы, в которую она входит.

Управляющее данное — данное, изменение значения которого вызывает прерывание управления.

Управляющее имя данного — имя данного, указанное во фразе CONTROL (УПРАВЛЕНИЕ) и относящееся к управляющему данному.

Уровень прерывания управления — относительная позиция в иерархии управления, на которой встретилось самое старшее из обнаруженных прерываний управления.

Условие — состояние программы во время ее выполнения, для которого можно определить значение истинности. Термин «условие» (условие-1, условие-2, ...), появляющийся в спецификациях языка или в ссылках общего формата, соответствует условному выражению, для которого может быть определено значение истинности и которое состоит из простого условия или комбинированных условий, представляющих синтаксически правильные комбинации простых условий, знаков логических операций и скобок.

Условие «в конце» — условие, имеющее место:

(1) если при выполнении оператора READ (ЧИТАТЬ) для файла с последовательным доступом не существует следующей логической записи в файле или если количество значащих цифр в относительном номере записи больше размера относительного ключа, или отсутствует входной необязательный файл;

(2) если при выполнении оператора RETURN (ВЕРНУТЬ) для соответствующего сортируемого или сливаемого файла не существует следующей логической записи;

(3) если при выполнении оператора SEARCH (ИСКАТЬ) операция поиска заканчивается и ни одно из условий, указанных в соответствующих вариантах WHEN (КОГДА), не удовлетворяется.

Условие знака — выражение, для которого может быть определено значение истинности того, что алгебраическое значение данного или арифметического выражения меньше нуля, больше нуля или равно нулю.

Условие имени-условия — выражение, для которого может быть определено значение истинности того, что значение условной переменной является одним из значений, соотнесенных имени-условия, соответствующему этой условной переменной.

Условие класса — выражение, для которого может быть определено значение истинности того, что значение данного полностью буквенное или полностью числовое или состоит исключительно из литер, указанных в определении имени-класса.

Условие конфликтных свойств файла — имела место неуспешная попытка выполнить операцию ввода-вывода для файла и ука-

GOST EXPERT ГОСТ 22558-89,

занные в программе свойства этого файла не совпадают с фиксированными свойствами файла.

Условие отношения — выражение, для которого может быть определено значение истинности того, что значение арифметического выражения, данного нечислового литерала или имени-индекса находится в определенном отношении со значением другого арифметического выражения, данного, нечислового литерала или имени-индекса (см. знак операции отношения).

Условие INVALID KEY (ПРИ ОШИБКЕ КЛЮЧА) — условие, возникающее во время выполнения, если значение ключа, соотнесенного индексному или относительному файлу, определяется как ошибочное.

Условие состояния переключателя — выражение, для которого может быть определено значение истинности того, что определенный реализацией переключатель, который может находиться в одном из состояний «включено» или «выключено», установлен в определенное состояние.

Условная переменная — данное, одному или нескольким значениям которого соотнесено имя условия.

Условная фраза — условная фраза; определяет действие, которое должно быть выполнено при определении значения истинности условия, получаемого в результате выполнения условного оператора.

Условное выражение — простое или сложное условие, указанное в операторах EVALUATE (ОЦЕНИТЬ), IF (ЕСЛИ), PERFORM (ВЫПОЛНИТЬ) или SEARCH (ИСКАТЬ) (см. простое условие; сложное условие).

Условный оператор — оператор, указывающий, что должно быть определено значение истинности условия и что последующие действия рабочей программы зависят от этого значения истинности. Условные операторы приведены в ч. 4, п. 6.4.2.

Устаревший элемент — элемент языка Кобол, который предполагается удалить из языка в следующей редакции стандарта языка Кобол.

Уточненное имя данного — идентификатор, состоящий из имени данного, за которым следует одна или более пар из связки OF или IN (ИЗ) и последующего уточнителя имени данного.

Уточнитель:

(1) Имя-данного или имя, связанное с индикатором уровня, которое используется в ссылках либо вместе с другим именем-данного, подчиненным уточнителю, либо вместе с именем-условия.

(2) Имя секции, которое указывается в ссылках на имя параграфа, содержащееся в этой секции.

(3) Имя-библиотеки, используемое в ссылках на имя текста, связанного с этой библиотекой (ч. 4, п. 4.3.8.1).

GOST EXPERT ГОСТ 22558-89,

Файл — совокупность логических записей.

Файл массовой памяти — совокупность записей, соотнесенных к среде массовой памяти.

Файл отчетов — выходной файл, статья описания которого содержит фразу REPORT (ОТЧЕТ). Файл отчетов состоит из записей, записанных под управлением системы управления генератором отчетов.

Физическая запись — термин является синонимом блока.

Физическая страница — понятие, зависящее от устройства оборудования и определяемое реализацией.

Фиксированные свойства файла — информация о файле, формируемая во время создания файла и которая не может в дальнейшем изменяться во время существования файла. Эти свойства включают организацию файла (последовательную, относительную или индексную), основной ключ записи, дополнительные ключи записи, набор кодов, минимальный или максимальный размер записи, тип записи (фиксированная или переменная), основную последовательность ключей для индексных файлов, блокирующий фактор, литеру заполнитель и ограничитель записи.

Формат — определенное упорядочение набора данных.

Формат представления — формат, который обеспечивает стандартный способ описания исходных Кобол-программ.

Фраза — упорядоченный набор последовательных строк литер Кобола, описывающих свойства в статье или часть оператора Кобола.

Фраза данных — фраза статьи описания данного раздела данных, описывающая определенное свойство данного.

Фраза оборудования — фраза, которая указана как часть статьи раздела оборудования.

Фраза отчета — фраза в секции отчетов раздела данных, которая указана в статье описания отчета или статье описания группы отчета.

Фраза файла — фраза, указанная в разделе данных как часть статьи описания файла и статьи описания сортируемого-сливаемого файла.

Целое — числовой литерал или числовое данное, которое не содержит ни одной позиции литеры справа от подразумеваемой десятичной точки. Термин «целое», указанный в общих форматах, не должен представлять числовое данное или значение, имеющее знак или равное нулю, если это не оговорено явно правилами данного формата.

Цифровая литера — литера, принадлежащая следующему набору цифр: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

Число сообщений — счетчик числа завершенных сообщений, имеющихся в указанной очереди сообщений.

GOST EXPERT ГОСТ 22558-89,

Числовое данное — данное, описание которого ограничивает представление его значения литерами цифр; данное со знаком может содержать также +, — или другое представление знака числа.

Числовой литерал — литерал, состоящий из одной или более цифровых литер, который может также содержать десятичную точку или знак числа либо и то и другое. Десятичная точка не должна быть самой правой литерой. Если указан знак числа, он должен быть самой левой литерой.

Элемент таблицы — одно из набора повторяющихся данных, составляющих таблицу.

Элементарное данное — данное, которое рассматривается как логически неделимое.

Явная десятичная точка — физическое представление позиции десятичной точки в данном, использующее одну из литер, обозначающих десятичную точку в Коболе: «.» (точку) или «,» (запятую).

Явный ограничитель области действия — зарезервированное слово, которое заканчивает область действия определенного оператора раздела процедур.

Часть 4. ОСНОВНЫЕ ПОНЯТИЯ

1. ВВЕДЕНИЕ

Понятия языка и правила, описанные ниже, относятся к полному языку Кобол. Если отдельный уровень модуля не допускает какие-либо из этих понятий, то в части, описывающей этот модуль, указываются соответствующие ограничения. Всюду в настоящем документе определения, относящиеся к высшему уровню, выделены рамкой. Следует также отметить, что ограничения, содержащиеся в одном модуле, могут, вообще говоря, затрагивать другие модули. Например, на уровне 1 ядра недопустимы уточнения, поэтому любой модуль, используемый с уровнем 1 ядра, должен иметь такое же ограничение. Приведенные для отдельных операторов в настоящем документе схемы алгоритмов иллюстрируют логику соответствующих операторов и не являются указаниями о способах их реализации.

2. ОБОЗНАЧЕНИЯ, ИСПОЛЬЗУЕМЫЕ В ФОРМАТАХ И ПРАВИЛАХ

2.1. Определение общего формата

Общий формат — это определенное упорядочение элементов фразы или оператора. Элементы фразы или оператора определяются ниже. В настоящем документе формат рассматривается в связи с информацией, определяющей фразы или операторы. Если

GOST EXPERT ГОСТ 22558-89,

допускается более одного определенного упорядочения, общий формат разделяется на перенумерованные форматы. Фразы должны записываться в последовательности, указываемой общими форматами. Если используются необязательные фразы, они должны подчиняться указанной последовательности. В определенных случаях, явно устанавливаемых правилами, фразы могут записываться в последовательности, отличной от указанной соответствующим форматом. Применение, требования или ограничения формулируются в виде правил.

2.1.1. Элементы

Элементы, составляющие фразу или оператор, представляются: словами, составленными из прописных букв; словами, составленными из строчных букв; номерами уровней; квадратными и фигурными скобками, связками и специальными литерами.

2.1.2. Слова

Все подчеркнутые слова из прописных букв называются ключевыми словами и обязательны при использовании форматов, частями которых они являются. Неподчеркнутыми словами из прописных букв пользоваться не обязательно; их можно не включать в исходную программу. Слова из прописных букв, независимо от подчеркивания, должны быть написаны правильно.

Слова из строчных букв в общем формате являются общими терминами, используемыми для представления слов Кобола, литералов, строк литер шаблона, статей-комментариев или цельных синтаксических конструкций, задаваемых пользователем. При повторении общих терминов в общем формате к каждому из них добавляется номер или буква для идентификации этих терминов при объяснении или обсуждении.

2.1.3. Номера уровней

Если определенные номера уровней входят в форматы статей раздела данных, то именно они должны быть использованы в соответствующих статьях в Кобол-программе. В этом документе пары цифр 01, 02, ..., 09 используют для задания номеров уровня от 1 до 9.

2.1.4. Квадратные, фигурные скобки и указатели выбора

Если часть общего формата заключена в квадратные скобки [ ], то один из вариантов, заключенных в скобки, может быть явно указан или эта часть общего формата может быть опущена.

Если часть общего формата заключена в фигурные скобки { }, то один из вариантов, заключенных в скобки, должен быть указан явно или неявно выбран.

Если выражение в фигурных скобках содержит только зарезервированные слова, не являющиеся ключевыми, то такой вариант

GOST EXPERT ГОСТ 22558-89,

является вариантом по умолчанию, если только один из остальных вариантов не будет явно указан.

Указатели выбора {| |}, заключающие часть общего формата, означают, что должен быть сделан выбор одного или нескольких из заключенных в них вариантов, но каждый из вариантов может быть указан только один раз.

Варианты в общем формате или части общего формата указываются вертикально расположенными альтернативными возможностями, квадратными и фигурными скобками или указателями выбора, или комбинацией скобок и указателей выбора. Вариант выбирается определением одной из указанных альтернативных возможностей или определением однозначной комбинации возможностей из ряда квадратных, фигурных скобок или указателей выбора.

2.1.5. Многоточие

В тексте, за исключением форматов, многоточие (...) указывает на пропуск слова или слов, если такой пропуск не нарушает понимание текста. Это общепринятое значение многоточия и его назначение становится очевидным из контекста.

В общем формате многоточие представляет позицию, в которой допускается повторение. Часть формата, которая может повторяться, определяется следующим образом:

если в формате встречается многоточие, берется непосредственно слева от многоточия скобка], } или |} и при продолжении просмотра справа налево определяется логически соответствующая [, { или {|; многоточие относится к части формата между определенной таким образом парой ограничителей.

2.1.6. Пунктуация в формате

Разделительная запятая и точка с запятой могут использоваться везде, где используется разделительный пробел в форматах (п. 4.2.1 настоящей части). В исходной программе эти разделители взаимозаменяемы.

Разделительная точка, используемая в форматах, имеет статус зарезервированного слова.

2.1.7. Использование слов-специальных литер в форматах

Слова-специальные литеры +, --, >, <, =, >=, <= в форматах не подчеркиваются, однако они обязательны.

3. ПРАВИЛА

3.1. Синтаксические правила

Синтаксическими правилами являются такие правила, которые определяют или объясняют порядок расположения слов или элементов при образовании более крупных языковых конструкций,

GOST EXPERT ГОСТ 22558-89,

таких как фразы или операторы. Синтаксические правила также накладывают или снимают ограничения на отдельные слова или элементы.

Эти правила используются для определения или объяснения того, как должен быть записан оператор, то есть каков порядок элементов в нем и каковы ограничения или расширения на представление каждого элемента.

3.2. **Общие правила**

Общими правилами являются такие правила, которые определяют или объясняют смысл элементов или их взаимосвязь; они используются для определения или объяснения семантики операторов и действий, оказываемых ими на выполнение или компиляцию.

4. ПОНЯТИЯ ЯЗЫКА

4.1. **Набор литер**

Основной и неделимой единицей языка является литера. Набор литер, используемых при образовании строк литер Кобола и разделителей, определен в глоссарии (см. ч. 3). Для нечисловых литералов, статей-комментариев и строк комментария набор литер расширяется и включает весь набор литер машины.

Литеры, допустимые для каждого типа строк литер и в качестве разделителей, определены в п. 4.2 настоящей части.

Некоторые литеры, составляющие набор литер Кобола, не могут быть представлены графически литерами национального или интернационального стандарта на набор литер. В таком случае допускается замена непредставимых литер.

Если набор литер содержит меньше 72 литер, недостающие литеры могут быть заменены парами литер. Такая замена рассматривается как устаревшее средство в этом варианте стандарта и будет исключена при следующем пересмотре стандарта.

4.2. **Структура языка**

Отдельные литеры языка связываются в строки литер и разделители. Разделители определяют границы строк литер. Допускается несколько идущих подряд разделителей. Последовательность строк литер и разделителей образует текст исходной программы.

4.2.1. Разделители

Разделителями является цепочка из одной или нескольких литер. Правила образования разделителей следующие:

(1) литера пунктуации пробел является разделителем. Всюду, где пробел используется в качестве разделителя или части разделителя, может стоять несколько пробелов. Все пробелы, следующие непосредственно за разделителями: запятой, точкой с запятой

GOST EXPERT ГОСТ 22558-89,

или точкой, являются частью этих разделителей и не рассматриваются как отдельный разделительный пробел;

(2) литеры пунктуации запятая (если она не используется в строке литер шаблона) и точка с запятой, за которыми следует пробел, являются разделителями и могут использоваться везде, где используется разделитель пробел. Они могут использоваться для удобочитаемости программы;

(3) литера пунктуации точка, за которой следует пробел, является разделителем (разделитель точка). Она должна использоваться только для указания конца предложения или там, где это указано в форматах;

(4) литеры пунктуации правая и левая скобки являются разделителями. Скобки могут проставляться (при соблюдении баланса левых и правых скобок) как ограничители в арифметических выражениях, для условий, индексов и модификаторов ссылки;

(5) литера пунктуации «(кавычки) является разделителем. Непосредственно перед открывающими кавычками должен стоять пробел или левая скобка; непосредственно за закрывающими кавычками, соответствующими открывающим кавычкам, должен стоять один из разделителей: пробел, запятая, точка с запятой, точка или правая скобка;

(6) ограничители псевдотекста являются разделителями. Непосредственно перед открывающим ограничителем псевдотекста должен стоять пробел; за закрывающим ограничителем псевдотекста должен стоять один из разделителей: пробел, запятая, точка с запятой или точка.

Ограничители псевдотекста могут использоваться только в паре, ограничивая псевдотекст;

(7) литера пунктуации двоеточие (:) является разделителем и должна использоваться только там, где это указано в общих форматах;

(8) разделитель пробел может быть проставлен по желанию непосредственно перед любым разделителем, однако следует учитывать:

а) правила, определенные форматом представления (п. 7 настоящей части);

б) в случае разделителя «закрывающая кавычка» предшествующий пробел рассматривается как часть нечислового литерала, а не как разделитель;

в) в случае открывающего ограничителя псевдотекста предшествующий пробел необходим;

(9) разделитель пробел может по желанию проставляться непосредственно за любым разделителем, за исключением открывающих кавычек, для которых пробел рассматривается как часть нечислового литерала, а не как разделитель.

GOST EXPERT ГОСТ 22558-89,

Любая литера пунктуации, которая используется как часть спецификации строки литер шаблона или числового литерала, рассматривается не как литера пунктуации, а как символ, используемый в спецификации строки литер шаблона или числового литерала. Строки литер шаблона ограничиваются только разделителями: пробелом, запятой, точкой с запятой или точкой.

Правила, установленные для образования разделителей, не применяются к литерам, которые составляют нечисловые литералы, статьи-комментарии или строки комментария.

4.2.2. Строки литер

Строка литер есть литера или последовательность смежных литер, которые образуют литерал, слово Кобола, строку литер шаблона или статью-комментарий. Строка литер ограничивается разделителями.

4.2.2.1. *Слова Кобола*

Словом Кобола является строка литер, состоящая не более чем из 30 литер и образующая слово, определенное пользователем, системное имя или зарезервированное слово. Каждая литера слова Кобола принадлежит набору букв, цифр и дефиса. Дефис не может быть первой или последней литерой слова. Каждая строчная буква рассматривается как эквивалент соответствующей ей прописной буквы. В исходной программе зарезервированные слова и слова, определенные пользователем, образуют непересекающиеся множества; системные имена и слова, определенные пользователем, образуют пересекающиеся множества. Одно и то же слово Кобола может использоваться как системное имя и как слово, определенное пользователем, внутри исходной программы. Отдельное появление такого слова Кобола классифицируется контекстом статьи или фразы, в котором оно появляется.

4.2.2.1.1. *Слова, определенные пользователем*

Словом, определенным пользователем, является слово Кобола, задаваемое пользователем в соответствии с форматом фразы или оператора. Каждая литера такого слова выбирается для английской нотации из набора литер A, ..., Z, 0, 1, ..., 9, - (для русской нотации из набора литер А, . . . , Я, D, F, G, I, J, L, N, Q, R, S, U, V, W, Y, Z, 0, 1, 2, . . . , 9, -). Слово не может начинаться или заканчиваться дефисом. Имеются следующие типы слов, определенных пользователем:

1. Имя алфавита;
2. Имя библиотеки;
3. Имя данного;
4. Имя записи;
5. Имя индекса;
6. Имя класса;
7. Имя коммуникации;

GOST EXPERT ГОСТ 22558-89,

8. Имя отчета;
9. Имя параграфа;
10. Имя программного модуля;
11. Имя программы;
12. Имя секции;
13. Имя текста;
14. Имя условия;
15. Имя файла;
16. Мнемоническое имя;
17. Номер сегмента;
18. Номер уровня;
19. Символическая литера.

Внутри заданной исходной программы, за исключением любой содержащейся в ней программы, определенные пользователем слова группируются в следующие непересекающиеся множества:

1. Имена алфавитов;
2. Имена библиотек;
3. Имена индексов;
4. Имена классов;
5. Имена коммуникаций;
6. Имена отчетов;
7. Имена параграфов;
8. Имена программных модулей;
9. Имена программ;
10. Имена секций;
11. Имена текстов;
12. Имена условий, имена данных, имена записей;
13. Имена файлов;
14. Мнемонические имена;
15. Символические литеры.

Все слова, определенные пользователем, за исключением номеров сегментов и номеров уровней, могут принадлежать только одному из этих непересекающихся множеств. Кроме того, все слова, определенные пользователем, в данном непересекающемся множестве либо не должны дублироваться, либо должны допускать установление однозначности, определяемое правилами однозначности ссылок (п. 4.3.8 настоящей части).

За исключением имени параграфа, имени секции, номера уровня и номера сегмента, все слова, определенные пользователем, должны содержать по крайней мере одну буквенную литеру. Номера сегментов и номера уровней могут повторяться: представление номера сегмента или номера уровня может совпадать с любым другим номером сегмента или номером уровня.

4.2.2.1.1.1. Имя условия

Имя условия — имя, поставленное в соответствие специально-

GOST EXPERT ГОСТ 22558-89,

му значению, множеству значений или области значений внутри полного множества значений, допустимых для некоторого имени данного. Само имя данного называется условной переменной.

Имена условий могут быть определены в разделе данных или в параграфе SPECIAL-NAMES (СПЕЦИАЛЬНЫЕ-ИМЕНА) раздела оборудования, где переключателям, определенным реализацией, должно быть приписано имя условия для состояния «включено» или «выключено», или для того и другого.

Имя условия используется для сокращения записи условия отношения, определяющего, что соответствующая условная переменная равна одному из множества значений, предписанных этому имени условия. Имя условия используется также в оператора SET (УСТАНОВИТЬ), указывая, что соответствующее значение должно быть помещено в условную переменную.

4.2.2.1.1.2. Мнемоническое имя

Мнемоническое имя — это слово, определенное пользователем, которое ставится в соответствие имени реализации. Это соответствие устанавливается в параграфе SPECIAL-NAMES (СПЕЦИАЛЬНЫЕ-ИМЕНА) раздела оборудования (ч. 6, п. 4.5).

4.2.2.1.1.3. Имя параграфа

Именем параграфа является слово, которое называет параграф в разделе процедур. Имена параграфов эквивалентны тогда и только тогда, когда они составлены из одной и той же последовательности одного и того же количества цифр и (или) других литер.

4.2.2.1.1.4. Имя секции

Имя секции — слово, которое называет секцию в разделе процедур. Имена секций эквивалентны только тогда, когда они составлены из одной и той же последовательности одного и того же количества цифр и (или) других литер.

4.2.2.1.1.5. Другие имена, определяемые пользователем

Спецификации всех других типов слов, определяемых пользователем (см. ч. 3).

4.2.2.1.2. *Системные имена*

Системным именем является слово Кобола, которое используется для связи с операционной средой. Правила для образования системных имен определяются реализацией с тем ограничением, что каждая литера, используемая при образовании системного имени, должна быть выбрана из набора литер для слов, определенных пользователем (см. п. 4.2.2.1.1 настоящей части), причем «-» не может быть первой или последней литерой системного имени.

Имеются следующие типы системных имен:

1. Имя машины;
2. Имя реализации;
3. Имя языка.

GOST EXPERT ГОСТ 22558-89,

В каждой реализации эти три типа системных имен образуют непересекающиеся множества; отдельное системное имя может принадлежать только одному из них. Для каждого из перечисленных типов системных имен имеется определение (см. ч. 3).

4.2.2.1.3. *Зарезервированные слова*

Зарезервированным словом является слово Кобола из определенного списка слов, которые могут быть использованы в исходной Кобол-программе, но которые не могут использоваться как слова, определяемые пользователем, или как системные имена. Зарезервированные слова могут использоваться только в соответствии с общими форматами (п. 8 настоящей части).

Имеются следующие типы зарезервированных слов:

1. Обязательные слова;
2. Необязательные слова;
3. Слова специального назначения.

4.2.2.1.3.1. Обязательные слова

Обязательное слово—это слово, вхождение которого обязательно при использовании формата, содержащего это слово. Имеются два типа обязательных слов:

(1) ключевые слова. В каждом формате такие слова записаны прописными буквами и подчеркнуты;

(2) слова-специальные литеры. Это знаки арифметических операций и литеры отношения.

4.2.2.1.3.2. Необязательные слова

Внутри каждого из форматов слова, записанные прописными буквами и не подчеркнутые, называются необязательными и могут включаться в текст по желанию пользователя для придания тексту удобочитаемости. Наличие или отсутствие необязательного слова внутри формата не изменяет семантики Кобол-программы, в которой оно используется.

4.2.2.1.3.3. Слова специального назначения

Имеются два типа слов специального назначения:

1. Специальные регистры;
2. Стандартные константы.

4.2.2.1.3.3.1. Специальные регистры

Для ссылок на специальные регистры используются определенные зарезервированные слова. Специальными регистрами являются фиксированные поля памяти вычислительной машины, которые главным образом используются для запоминания информации, получаемой при выполнении специфических функций Кобола. Если не оговорено противное, для каждой программы создается один специальный регистр каждого типа. Специальный регистр может использоваться в общих форматах всюду, где используются имена данных или идентификаторы, принадлежащие той же категории, что и специальный регистр, если не оговорено противное.

GOST EXPERT ГОСТ 22558-89,

Если разрешаются уточнения, специальные регистры при необходимости могут уточняться для обеспечения однозначности (п. 4.3.8.1 настоящей части).

Специальными регистрами являются следующие:

1. DEBUG-ITEM (ДАННЫЕ-ОТЛАДКИ) (ч. 15, п. 1.3.2);
2. LINAGE-COUNTER (СЧЕТЧИК-ВЕРСТКИ) (ч. 7, п. 1.3.8);
3. LINE-COUNTER (СЧЕТЧИК-СТРОК) (ч. 13, п. 1.2.3);
4. PAGE-COUNTER (СЧЕТЧИК-СТРАНИЦ) (ч. 13, п. 1.2.2).

4.2.2.1.3.3.2. Стандартные константы

Определенные зарезервированные слова используются для именования специальных констант и ссылок на них. Эти зарезервированные слова определены в п. 4.2.2.2.3 в настоящей части.

4.2.2.2. *Литералы*

Литерал — строка литер, значение которой определяется упорядоченным набором литер, из которых она составлена, или спецификацией зарезервированного слова, являющегося стандартной константой. Каждый литерал принадлежит к одному из двух видов: числовому или нечисловому.

4.2.2.2.1. *Нечисловые литералы*

Нечисловой литерал определяется как ограниченная кавычками слева и справа строка литер. Реализация должна допускать длину литерала от одной до 160 литер. Длина нечислового литерала относится к его представлению в объектной программе.

4.2.2.2.1.1. Общий формат

«{литера-1}...»

4.2.2.2.1.2. Синтаксические правила

(1) Литера-1 может быть любой литерой из набора литер машины.

(2) Если литера-1 должна представлять литеру кавычек, для представления одной литеры кавычек должны использоваться две следующие друг за другом литеры кавычек.

4.2.2.2.1.3. Общие правила

(1) Значением нечислового литерала в рабочей программе является значение, представляемое литерой-1.

(2) Ограничивающие нечисловой литерал кавычки не являются частью значения нечислового литерала.

(3) Все нечисловые литералы относятся к буквенно-цифровой категории.

4.2.2.2.2. Числовые литералы

Числовой литерал определяется как строка литер, состоящая из цифр от 0 до 9, знака плюс, знака минус и десятичной точки. Реализация должна разрешать длину числового литерала от 1 до 18 цифр. Правила образования числового литерала следующие:

(1) литерал должен содержать по меньшей мере одну цифру;

(2) литерал не должен содержать более одной литеры знака,

GOST EXPERT ГОСТ 22558-89,

Если указан знак, то он должен быть самой левой литерой в литерале. Если литерал не имеет знака, то он является положительным;

(3) литерал не должен содержать более одной десятичной точки. Десятичная точка рассматривается как подразумеваемая десятичная точка и может занимать в литерале любую позицию, кроме самой правой. Если литерал не содержит десятичной точки, то он является целым числом.

Литерал, составленный по правилам для образования числовых литералов, но заключенный в кавычки, является нечисловым и обрабатывается компилятором как нечисловой;

(4) значением числового литерала является алгебраическая величина, представленная литерами числового литерала. Каждый числовой литерал относится к числовой категории (ч. 6, п. 5.9.1).

Размер числового литерала в терминах литер стандартного формата данных равен количеству составляющих его цифр.

4.2.2.2.3. *Значения стандартных констант*

Значения стандартных констант генерируются компилятором. Пользователь может ссылаться на них посредством зарезервированных слов, приведенных ниже. Эти слова при использовании их в качестве стандартных констант не должны заключаться в кавычки. Единственная и множественная форма стандартных констант эквивалентны и могут при употреблении заменяться одна другой.

Значения стандартных констант и зарезервированные слова, используемые для обращения к ним, следующие:

(1) [ALL] ZERO, [ALL] ZEROES ([ВСЕ] НУЛЬ, [ВСЕ] НУЛИ) представляют числовое значение 0 или одну или более литер 0 из набора литер машины;

(2) [ALL] SPACE, [ALL] SPACES ([ВСЕ] ПРОБЕЛ, [ВСЕ] ПРОБЕЛЫ) представляют одну или несколько литер пробела из набора литер машины;

(3) [ALL] HIGH-VALUE, [ALL] HIGH-VALUES ([ВСЕ] НАИБОЛЬШЕЕ-ЗНАЧЕНИЕ, [ВСЕ] НАИБОЛЬШИЕ-ЗНАЧЕНИЯ) — везде, за исключением параграфа SPECIAL-NAMES (СПЕЦИАЛЬНЫЕ-ИМЕНА), представляют одну или несколько литер, имеющих наибольшую порядковую позицию в программной основной последовательности;

(4) [ALL] LOW-VALUE, [ALL] LOW-VALUES ([ВСЕ] НАИМЕНЬШЕЕ-ЗНАЧЕНИЕ, [ВСЕ] НАИМЕНЬШИЕ-ЗНАЧЕНИЯ) — везде, за исключением параграфа SPECIAL-NAMES (СПЕЦИАЛЬНЫЕ-ИМЕНА), представляют одну или несколько литер, имеющих наименьшую порядковую позицию в программной основной последовательности;

GOST EXPERT ГОСТ 22558-89,

(5) [ALL] QUOTE, [ALL] QUOTES ([ВСЕ] КАВЫЧКА, [ВСЕ] КАВЫЧКИ) представляет одну или более литер. Слова QUOTE, QUOTES (КАВЫЧКА, КАВЫЧКИ) не могут использоваться вместо литер кавычек в исходной программе для ограничения нечислового литерала. Например, QUOTE ABC QUOTE (КАВЫЧКА ABC КАВЫЧКА) не является правильным представлением нечислового литерала «ABC»;

(6) ALL литерал (ВСЕ литерал) представляет всю строку или часть строки, генерируемой последовательной конкатенацией литер, образующих литерал. Литерал должен быть нечисловым. Литерал не может быть стандартной константой;

(7) [ALL] символическая-литера ([ВСЕ] символическая-литера) представляет одну или несколько литер, определяемых как значение этой символической литеры в фразе SYMBOLIC CHARACTERS (СИМВОЛИЧЕСКАЯ ЛИТЕРА) параграфа SPECIAL NAMES (СПЕЦИАЛЬНЫЕ-ИМЕНА) (ч. 6, п. 4.5).

Когда стандартная константа представляет собой строку из одной или нескольких литер, длина такой строки определяется компилятором из контекста согласно следующим правилам:

(1) когда стандартная константа указана в фразе VALUE (ЗНАЧЕНИЕ) или когда стандартной константе сопоставляется другое данное, например, в случае перемещения или сравнения с некоторым данным, строка литер, представленная стандартной константой, повторяется литера за литерой вправо до тех пор, пока результирующая строка не станет равной или большей по размеру (в литерах) соответствующему данному.

Затем результирующая строка усекается справа до размера (в литерах) соответствующего данного. Такое повторение выполняется независимо от фразы JUSTIFIED (СДВИНУТО), относящейся к этому данному.

(2) когда стандартная константа, отличная от ALL литерал (ВСЕ литерал), не сопоставляется с другим данным, например, в случае, когда она появляется в операторах DISPLAY (ВЫДАТЬ), STOP (ОСТАНОВИТЬ), STRING (СОБРАТЬ) или UNSTRING (РАЗОБРАТЬ), ее длина равна длине строки в одну литеру;

(3) когда стандартная константа ALL литерал (ВСЕ литерал) не сопоставляется с другим данным, ее длина равна длине литерала.

Стандартная константа может использоваться всюду в форматах вместо литерала, за следующим исключением:

(1) вместо числового литерала разрешается использовать только стандартную константу ZERO, ZEROS, ZEROES (НУЛЬ, НУЛИ);

(2) сопоставление стандартной константы ALL литерал (ВСЕ литерал) при длине литерала больше одной литеры с числовым

GOST EXPERT ГОСТ 22558-89,

или числовым редактируемым данным рассматривается в настоящем стандарте как устаревшее средство. При следующем пересмотре стандарта оно будет исключено;

(3) когда стандартная константа используется вместо литерала во фразе ALL литерал (ВСЕ литерал), слово ALL (ВСЕ) избыточно и используется только для удобочитаемости.

За исключением параграфа SPECIAL-NAMES (СПЕЦИАЛЬНЫЕ-ИМЕНА), в исходной программе, где используются стандартные константы HIGH-VALUE (НАИБОЛЬШЕЕ-ЗНАЧЕНИЕ) и LOW-VALUE (НАИМЕНЬШЕЕ-ЗНАЧЕНИЕ), действительные литеры, соотнесенные с каждой стандартной константой, зависят от определенного программного алфавита. (ч. 6, пп. 4.4.1, 4.5).

Каждое зарезервированное слово, которое используется для обращения к значению стандартной константы, является отдельной строкой литер, за исключением конструкций, использующих слово ALL (ВСЕ), таких как ALL литерал (ВСЕ литерал), ALL SPACES (ВСЕ ПРОБЕЛЫ) и т. п., которые состоят из двух отдельных строк литер.

4.2.2.3. *Строка литер шаблона*

Строка литер шаблона представляется определенной комбинацией набора литер Кобола, используемых в качестве символов, и валютным символом. Объяснение строки литер шаблона и правила, управляющие шаблоном, приведены ниже (ч. 6, п. 5.9).

Любая литера пунктуации, используемая в строке литер шаблона, рассматривается не как знак пунктуации, а как символ спецификации строки литер шаблона.

4.2.2.4. *Статья-комментарий*

Статья-комментарий — это статья в разделе идентификации, которая может быть любой комбинацией литер из набора литер вычислительной машины. Статья-комментарий рассматривается в настоящем стандарте как устаревший элемент и будет исключена при следующем пересмотре стандарта.

4.3. Понятие машинно-независимого описания данного

Чтобы сделать данные максимально машинно-независимыми, особенности или свойства данных описываются по отношению к стандартному формату данного, а не к формату, ориентированному на оборудование. Этот стандартный формат данного ориентирован на общие применения в обработке данных и использует десятичную систему для представления чисел (независимо от основания системы счисления, используемой в машине) и остальные литеры набора литер Кобола для изображения нечисловых данных.

4.3.1. Понятие логической записи

Чтобы отделить логические характеристики данных от физических характеристик среды памяти данных, используются опреде-

GOST EXPERT ГОСТ 22558-89,

ленные специальные фразы или набор фраз, рассматриваемые ниже.

4.3.1.1. *Физические характеристики файла*

Физические характеристики файла относятся к представлению данных на вводе или выводе и включают такие особенности как:

(1) группирование логических записей в пределах физических границ среды файла;

(2) средства, с помощью которых файл может быть идентифицирован.

4.3.1.2. *Логические характеристики файла*

Логические характеристики файла представлены явным определением каждой логической единицы в самом файле. В Кобол-программе оператор ввода и вывода обращается к одной логической записи.

Важно отметить различие между физической и логической записями. Логическая запись в Коболе — это порция связанной информации, однозначно идентифицируемой и обрабатываемой как единое целое.

Физическая запись — это физическая единица информации, размер и способ записи которой удобен для запоминания данных на входном или выходном устройстве определенной машины. Размер физической записи зависит от оборудования и не находится в прямой связи с размером хранимого файла информации.

Логическая запись может содержаться внутри одной физической записи или несколько логических записей могут содержаться внутри одной физической записи; в случае файлов массовой памяти логическая запись может содержаться в нескольких физических записях. Во входном языке имеются несколько методов для описания взаимосвязи логических и физических записей. После установления взаимосвязи управление доступом к логической записи, связанной с физической записью, обеспечивается взаимодействием объектной программы с предоставляемыми реализацией оборудованием и системами программного обеспечения. В этом документе ссылка на запись означает ссылку на логическую запись, если специально не оговорен термин «физическая запись».

Понятие логической записи применимо не только к файлам данных, но распространяется также на рабочую память. Таким образом, рабочая память может быть сгруппирована в логические записи и определена последовательностью статей описания записи.

При занесении логической записи или извлечении ее из физической записи выполняются все преобразования, требуемые указанной фразой CODE-SET (АЛФАВИТ). При необходимости добавляются или удаляются литеры заполнители. Ни одна из фраз, используемых для описания данных в логической записи, не влияет на эти преобразования.

GOST EXPERT ГОСТ 22558-89,

4.3.1.3. *Понятие записи*

Описание записи состоит из ряда статей описания данных, которые описывают особенности отдельной записи. Каждая статья описания данных состоит из номера уровня, за которым следует имя, если оно требуется, и далее, при необходимости, следует ряд независимых фраз.

4.3.2. Понятие уровня

Понятие уровня неотъемлемо от понятия структуры логической записи. Это понятие вводится в целях описания подразделений записи для обращения к составляющим ее данным. Подразделение может быть продолжено для возможности обращения к более мелким данным.

Неделимые составляющие записи называются элементарными данными; соответственно запись состоит из последовательности элементарных данных, либо сама запись является элементарным данным.

Для обращения к ряду элементарных данных последние объединяются в группы или групповые данные. Каждая группа представляет названную последовательность, включающую одно или более элементарных данных. Группы, в свою очередь, могут быть объединены в группы из двух или нескольких групп и т. д. Таким образом, элементарное данное может принадлежать более чем одной группе.

4.3.2.1. *Номера уровней*

Система номеров уровней задает организацию элементарных данных или групп данных. Записи — наиболее объемлющие данные, им присвоен номер уровня 01. Менее объемлющим данным присваиваются численно большие (не обязательно последовательные) номера уровней, не превосходящие 49. Имеются специальные номера уровней 66, 77, 88, которые являются исключением из этого правила. Каждому используемому номеру уровня в исходной программе соответствует отдельная статья.

Группа включает в себя все группы и элементарные данные, следующие за ней, пока не встретится номер уровня, меньший или равный номеру уровня группы. Все данные, которые подчинены непосредственно некоторому групповому данному, должны описываться с одинаковым номером уровня и иметь номер уровня больший, чем номер уровня, используемый для описания этого группового данного.

Имеется три типа статей, для которых понятие уровня не имеет силы:

(1) статьи, описывающие элементарные данные или группы, вводимые фразой RENAMES (ПЕРЕИМЕНОВЫВАЕТ);

(2) статьи, описывающие несвязанные данные в секциях рабочей памяти и связи;

(3) статьи, которые описывают имена условий.

Данным, описанным фразой RENAMES (ПЕРЕИМЕНОВЫВАЕТ) с целью перегруппировки данных, присвоен специальный номер уровня 66.

Несвязанным данным, которые не являются подразделением других данных и сами не подразделяются, присвоен специальный номер уровня 77.

Статьям, которые описывают имена условий, соответствующие отдельным значениям условных переменных, присвоен специальный номер уровня 88.

4.3.3. Понятие класса данного

Пять категорий данных (ч. 6, п. 5.9) сгруппированы в три класса: буквенный, числовой и буквенно-цифровой. Для буквенных и числовых данных понятия класс и категория являются синонимами. Буквенно-цифровой класс включает категории буквенно-цифровую редактируемую, числовую редактируемую и буквенно-цифровую (без редактирования). Каждое элементарное данное принадлежит одному из классов и одной из категорий. Групповые данные обрабатываются как буквенно-цифровые данные, независимо от класса элементарных данных, на которые подразделяется это групповое данное. Таблица показывает отношение между классами и категориями данных.

4.3.4. Выбор представления литер и основания системы счисления

Значение числового данного может быть представлено в нескольких формах, таких как двоичная или десятичная, в зависимости от оборудования. Кроме того, имеется несколько способов представления десятичных чисел. Так как эти представления являются в действительности комбинациями битов, они обычно называются двоично-кодированной десятичной формой. Выбор основания системы счисления обычно зависит от арифметических возможностей машины. Если допустима более чем одна арифметическая форма, то выбранная форма указывается фразой об использовании. Двоично-кодированная десятичная форма используется также для представления литер и символов, которые являются буквенно-цифровыми данными. Выбор надлежащей двоично-кодированной буквенно-цифровой или двоично-кодированной десятичной формы зависит от возможностей машины и ее внешней среды.

Когда ЭВМ обеспечивает несколько значений представления данных, должен быть использован стандартный формат данных, если описание данного не оговаривает противного. Если как внешняя среда, так и машина допускают более чем одну форму представления или если внешняя среда, соответствующая данному, отсутствует, выбор оговаривается во фразе об использовании, фразе PICTURE (ШАБЛОН) и других фразах описания данных. Каж-

GOST EXPERT ГОСТ 22558-89,

дая реализация предоставляет полное объяснение возможных форм представления для ЭВМ, на которой она реализует Кобол. Метод выбора подходящей формы представления данного, используемый реализацией, позволяет программисту предвидеть и (или) управлять этим выбором.

Данное	Класс	Категория
Элементарное	Буквенный	Буквенная
	Числовой	Числовая
	Буквенно-цифровой	Числовая редактируемая, буквенно-цифровая редактируемая, буквенно-цифровая
Неэлементарное (групповое)	Буквенно-цифровой	Буквенная, числовая, числовая редактируемая, буквенно-цифровая редактируемая, буквенно-цифровая

Под размером элементарного данного или группы данных понимается число литер в представлении данного в стандартном формате данных. Выделение данных и особенности использования могут привести к различию между этим размером и действительным числом литер, необходимых для внутреннего представления.

4.3.5. Алгебраические знаки

Алгебраические знаки разделяются на две категории: знак числа, который относится к числовому данному и числовому литералу со знаком, и знак редактирования, который появляется в редактирующей фразе для установления знака данного.

Фраза SIGN (ЗНАК) позволяет программисту задать представление знака числа. Эта фраза необязательна; если она не используется, представление знака числа определяется реализацией.

Редактирующие знаки вставляются в данное посредством использования символов фразы PICTURE (ШАБЛОН), управляемых знаком.

4.3.6. Стандартные правила выравнивания

Стандартные правила для расположения значения в поле элементарного данного зависят от категории принимающего поля. Это следующие правила:

(1) если принимающее поле описано как числовое, то:

а) данное выравнивается по десятичной точке и помещается в принимающее поле с дополнением нулями или усечением при необходимости на каждом из его концов;

GOST EXPERT ГОСТ 22558-89,

б) когда подразумеваемая десятичная точка явно не задана, данное рассматривается так, как если бы оно имело подразумеваемую десятичную точку, расположенную непосредственно за ее самой правой литерой, и было выровнено так, как указано выше;

(2) если принимающее данное является числовым редактируемым данным, то пересылаемое данное выравнивается по десятичной точке и дополняется нулями или усекается на каждом из концов, как указано позициями литер принимающего данного; кроме того, если это требуется редактированием, замещаются ведущие нули;

(3) если принимающее данное буквенно-цифровое (отличное от числового редактируемого данного), буквенно-цифровое редактируемое или буквенное, пересылаемое данное помещается в принимающие позиции литер выровненным по самой левой позиции литер в данном и справа дополняется пробелами или усекается при необходимости.

Если задана фраза JUSTIFIED (СДВИНУТО) для принимающего данного, перечисленные стандартные правила изменяются в соответствии с описанием для фразы JUSTIFIED (СДВИНУТО) (ч. 6, п. 5.6).

4.3.7. Выравнивание данного для повышения эффективности объектного кода

В некоторых вычислительных машинах память организована таким образом, что существуют естественные адресуемые границы машинной памяти, например, границы слова, полуслова, байта. Способ хранения данных определяется объектной программой и нет необходимости учитывать эти естественные границы при описании данного.

Однако определенное использование данных (например, в арифметических операциях или при индексировании) может быть облегчено, если данные хранятся выровненными относительно естественных границ памяти. В частности, если части двух или нескольких данных расположены между соседними естественными границами или некоторая естественная граница дробит единое данное, для организации доступа к этим данным и их запоминания в рабочей программе могут потребоваться дополнительные машинные операции.

Данные, выровненные относительно естественных границ памяти с целью избежания дополнительных машинных операций при доступе к ним, объявляются выделенными.

Выделение может достигаться двумя путями:

(1) применением фразы SYNCHRONIZED (ВЫДЕЛЕНО);

(2) соответствующей естественным границам организацией данных без использования фразы SYNCHRONIZED (ВЫДЕЛЕНО).

GOST EXPERT ГОСТ 22558-89,

Каждая реализация, обеспечивающая специальные виды выравнивания, должна указывать точную интерпретацию действий, подлежащих выполнению. Использование выровненных данных в группе может быть эффективным при выполнении операторов, использующих группу в качестве операнда. Каждая реализация, обеспечивающая специальные типы выравнивания данного, должна описать воздействие неявного заполнителя и семантику оператора, использующего эту группу.

4.3.8. Однозначность ссылок

Каждое имя, определенное пользователем в Кобол-программе, присваивается им определенному ресурсу, который будет использоваться в решении проблемы обработки данных (см. п. 4.2.2 настоящей части). Что касается использования ресурсов, оператор в Кобол-программе должен содержать ссылку, которая однозначно идентифицирует эти ресурсы.

Для гарантии однозначности ссылки имя, определенное пользователем, можно уточнять, индексировать или модифицировать ссылку согласно правилам, описанным ниже.

Когда одно и то же имя используется в отдельных программах для двух или более появлений ресурсов одного и того же типа и когда само по себе уточнение не позволяет в одной из этих программ отличить два идентично названных ресурса, тогда применяются специальные соглашения, ограничивающие область действия имен. Эти соглашения гарантируют, что ресурсы идентифицированы так, как описано в программе, содержащей ссылку (ч. 10, п. 1.3.8).

Если это не оговорено правилами для оператора, любое индексирование и модификация ссылки производятся только один раз как первая операция при выполнении оператора.

4.3.8.1. *Уточнение*

Каждое определенное пользователем имя, явно используемое в Кобол-программе, должно быть однозначным в силу одного из перечисленных ниже обстоятельств:

(1) никакое другое имя не имеет идентичного написания;

(2) имя однозначно в контексте фразы REDEFINES (ПЕРЕОПРЕДЕЛЯЕТ) (ч. 6, п. 5.10);

(3) имя определяется внутри такой иерархии имен, что ссылка на него может быть сделана однозначной посредством упоминания одного или более старших уровней иерархии.

Старшие уровни называются уточнителями, а процесс, определяющий однозначность ссылок, называется уточнением. В исходной программе могут появляться идентичные определенные пользователем имена; однако затем должна быть установлена однозначность посредством уточнения для каждого определенного пользователем имени, на которое имеется явная ссылка, за исключени-

GOST EXPERT ГОСТ 22558-89,

ем случая переопределения. Нет необходимости указывать все имеющиеся в распоряжении уточнители для установления однозначности. Зарезервированные слова, именующие специальные регистры, требуют уточнения для обеспечения однозначности всякий раз, когда исходная программа приводит к появлению более одного экземпляра какого-либо из специальных регистров. На имя параграфа или имя секции, появляющиеся в программе, нельзя ссылаться из любой другой программы;

(4) программа содержится в другой программе либо содержит другую программу (ч. 10, п. 1.3.8).

Независимо от вышесказанного, одно и то же имя данных не может использоваться как имя внешней записи и как имя любого другого внешнего данного в программе, содержащей другую программу или содержащейся в некоторой программе и описывающей эту внешнюю запись данных. Одно и то же имя данных не может употребляться как имя данного, обладающего свойством глобального, и как имя любого другого данного, описанного в программе, описывающей это глобальное данное.

Общими форматами для уточнения являются:

Формат 1

{имя-данного-1 / имя-условия-1} [{{IN / OF} имя-данного-2}... [{IN / OF} {имя-файла-1 / имя-коммуникации-1}]]

[{OF / IN} {имя-файла-1 / имя-коммуникации-1}]

{имя-данного-1 / имя-условия-1} [{ИЗ имя-данного-2}.. [ИЗ {имя-файла-1 / имя-коммуникации-1}]]

[ИЗ {имя-файла-1 / имя-коммуникации-1}]

Формат 2

имя-параграфа-1 {OF / IN} имя-секции-1

имя-параграфа-1 ИЗ имя-секции-1

Формат 3

имя-текста-1 {OF / IN} имя-библиотеки-1

имя-текста-1 ИЗ имя-библиотеки-1

GOST EXPERT ГОСТ 22558-89,

Формат 4

LINAGE-COUNTER {IN / OF} имя-файла-2

СЧЕТЧИК-ВЕРСТКИ ИЗ имя-файла-2

Формат 5

{PAGE-COUNTER / LINE-COUNTER} {IN / OF} имя-отчета-1

{СЧЕТЧИК-СТРАНИЦ / СЧЕТЧИК-СТРОК} ИЗ имя-отчета-1

Формат 6

имя-данного-3 { {IN / OF} имя-данного-4 [{IN / OF} имя-отчета-2] | {IN / OF} имя-отчета-2 }

имя-данного-3 {ИЗ имя-данного-4 [ИЗ имя-отчета-2] / ИЗ имя-отчета-2}

Правила уточнения:

(1) Для каждого неоднозначного имени, определенного пользователем, на которое имеется явная ссылка, однозначность должна устанавливаться посредством последовательности уточнителей, которая устраняет неоднозначность ссылки.

(2) Имя может уточняться даже тогда, когда нет необходимости в уточнении; если имеется несколько комбинаций уточнителей, обеспечивающих однозначность, использоваться может любая из таких комбинаций.

(3) IN и OF являются логическими эквивалентами.

(4) В формате 1 уточнитель должен быть именем, относящимся к индикатору уровня, или именем группы, в которую входит уточняемое данное, или именем условной переменной, связанной с уточняемым именем-условия. Уточнители указываются в порядке последовательного нарастания уровней иерархии.

(5) В формате 1 имя-данного-1 или имя-данного-2 может быть именем-записи.

(6) Если имеется явная ссылка на имя параграфа, то имя параграфа не должно дублироваться внутри секции. Когда имя параграфа уточняется именем секции, слово SECTION (СЕКЦИЯ) опускается. Имя параграфа не нужно уточнять, если к нему обращаются внутри той же секции. На имя параграфа или имя сек-

GOST EXPERT ГОСТ 22558-89,

ции, появляющееся в программе, нельзя ссылаться из любой другой программы.

(7) Если во время компиляции компилятору доступна более чем одна библиотека Кобола, имя текста должно уточняться при каждом обращении к нему.

(8) Если в исходной программе имеется несколько статей описания файла, содержащих фразу LINAGE (ВЕРСТКА), LINAGE-COUNTER (СЧЕТЧИК-ВЕРСТКИ) должен уточняться каждый раз при обращении к нему.

(9) Если в исходной программе определено несколько статей описания отчета, LINE-COUNTER (СЧЕТЧИК-СТРОК) должен быть уточнен при каждом обращении к нему в разделе процедур. В секции отчетов неуточненная ссылка на LINE-COUNTER (СЧЕТЧИК-СТРОК) неявно уточнена именем отчета, в статье описания которого имеется эта ссылка. При обращении к счетчику строк разных отчетов, LINE-COUNTER (СЧЕТЧИК-СТРОК) должен явно уточняться соответствующим именем отчета.

(10) Если в исходной программе определено несколько статей описания отчета, PAGE-COUNTER (СЧЕТЧИК-СТРАНИЦ) должен быть уточнен при каждом обращении к нему в разделе процедур. В секции отчетов неуточненная ссылка на PAGE-COUNTER (СЧЕТЧИК-СТРАНИЦ) неявно уточнена именем отчета, в статье описания которого имеется эта ссылка. При обращении к счетчику страниц разных отчетов PAGE-COUNTER (СЧЕТЧИК-СТРАНИЦ) должен явно уточняться соответствующим именем отчета.

4.3.8.2. *Индексирование*

4.3.8.2.1. *Назначение*

Индексы используются при обращении к отдельному элементу из списка или таблицы однотипных элементов, которым не поставлены в соответствие индивидуальные имена данных (ч. 6, п. 5.8).

4.3.8.2.2. *Общий формат*

$$\left\{\begin{array}{l}\text{имя-данного-1}\\ \text{имя-условия-1}\end{array}\right\}\left(\left\{\begin{array}{ll}\text{целое-1}\\ \text{имя-данного-2} & [\{\pm\}\ \text{целое-2}]\\ \text{имя-индекса-1} & [\{\pm\}\ \text{целое-3}]\end{array}\right\}\ \ldots\right)$$

4.3.8.2.3. *Синтаксические правила*

(1) Статья описания данного, содержащая имя-данного-1 или имя-данного, соотнесенное имени-условия-1, должна содержать фразу OCCURS (ПОВТОРЯЕТСЯ) или должна подчиняться статье описания данного, содержащей фразу OCCURS (ПОВТОРЯЕТСЯ).

(2) За исключением случаев, указанных в синтаксическом правиле (4), при обращении к табличному элементу количество индексов должно равняться числу фраз OCCURS (ПОВТОРЯЕТСЯ) в описании данного табличного элемента. При использовании не-

GOST EXPERT ГОСТ 22558-89,

скольких индексов они записываются в порядке от старшего к младшим уровням иерархии таблицы.

(3) Имя-индекса-1 должно соответствовать статье описания данного в иерархии таблицы, содержащей фразу INDEXED (ИНДЕКСИРУЕТСЯ), определяющую это имя-индекса.

(4) При каждом обращении к табличному элементу должно использоваться индексирование, за исключением следующих случаев:

а) в операторе USE FOR DEBUGGING (ИСПОЛЬЗОВАТЬ ДЛЯ ОТЛАДКИ);

б) в качестве субъекта в операторе SEARCH (ИСКАТЬ);

в) во фразе REDEFINES (ПЕРЕОПРЕДЕЛЯЕТ);

г) в варианте ASCENDING/DESCENDING KEY IS (ПО ВОЗРАСТАНИЮ/УБЫВАНИЮ КЛЮЧА) фразы OCCURS (ПОВТОРЯЕТСЯ).

(5) Имя-данного-2 может уточняться и должно быть числовым элементарным данным, представляющим целое.

(6) Целое-1 может быть со знаком; знак должен быть только положительным.

4.3.8.2.4. *Общие правила*

(1) Значение индекса должно быть положительным целым. Наименьшее допустимое значение, представляемое индексом, равняется 1 и оно указывает на первый элемент таблицы. Следующие за ним элементы таблицы указываются с помощью номера вхождения 2, 3, ... Наибольший допустимый номер вхождения в каждом отдельном случае равен максимальному числу повторений данного, указанному фразой OCCURS (ПОВТОРЯЕТСЯ).

(2) Значение имени-индекса 1 соответствует номеру вхождения элемента таблицы, связанного с ним. Это соответствие определяется реализацией.

(3) Значение имени-индекса-1 должно устанавливаться до его использования в качестве индекса. Начальное значение имени-индекса может быть задано оператором PERFORM (ВЫПОЛНИТЬ) с фразой VARYING (МЕНЯЯ); оператором SEARCH (ИСКАТЬ) с фразой ALL (ОСОБО) или оператором SET (УСТАНОВИТЬ). Значение имени-индекса может изменяться только операторами PERFORM (ВЫПОЛНИТЬ), SEARCH (ИСКАТЬ) или SET (УСТАНОВИТЬ).

(4) Если указано целое-2 или целое-3, значение индекса определяется как номер вхождения, представленный значением имени-индекса-1, либо значением данного, указанного именем-данного-2, увеличенным на значение целого-2 или целого-3 (когда указан знак арифметической операции +) или уменьшенным на значение целого-2 или целого-3 (когда указан знак арифметической операции —).

GOST EXPERT ГОСТ 22558-89,

4.3.8.3. *Модификация ссылки*

4.3.8.3.1. *Назначение*

Модификация ссылки определяет данное указанием его самой левой позиции литеры и длины данного.

4.3.8.3.2. *Общий формат*

имя-данного-1 (позиция-самой-левой-литеры: [длина])

4.3.8.3.3. *Синтаксические правила*

(1) Имя-данного-1 должно быть именем данного, статья описания которого содержит фразу об использовании USAGE IS DISPLAY (ДЛЯ ВЫДАЧИ).

(2) Позиция самой левой литеры и длина должны быть арифметическими выражениями.

(3) Если особо не оговорено, модификация ссылки допустима везде, где разрешен идентификатор соотнесенный данному, принадлежащему к буквенно-цифровому классу.

(4) Имя-данного-1 может быть уточнено или индексировано.

4.3.8.3.4. *Общие правила*

(1) Каждой литере данного, соотнесенного имени-данного-1, приписывается порядковый номер, который равен 1 для самой левой позиции и увеличивается на единицу для каждой следующей позиции вплоть до самой правой позиции. Если статья описания данного для имени-данного-1 содержит фразу SIGN IS SEPARATE (ЗНАК ОТДЕЛЬНО), позиции знака приписывается порядковый номер в этом данном.

(2) Если данное, соотнесенное имени-данного-1, описано как числовое, числовое редактируемое, буквенное или буквенно-цифровое редактируемое, над ним производятся действия по модификации ссылок, как если бы оно было переопределено как буквенно-цифровое данное того же размера, что и данное, соотнесенное имени-данного-1.

(3) Модификация ссылки для операнда выполняется по следующим правилам:

а) если для операнда указано индексирование, модификация ссылки производится сразу же после вычисления индексов;

б) если для операнда индексирование не указано, модификация ссылки производится тогда, когда производилось бы вычисление индексов, если бы были указаны индексы.

(4) Модификация ссылки создает уникальное данное, являющееся подстрокой данного, соотнесенного имени-данного-1. Это данное определяется по следующим правилам:

а) вычисление позиции-самой-левой-литеры определяет порядковую позицию самой левой литеры единственного данного по отношению к самой левой литере данного, соотнесенного имени-данного-1. Результатом вычисления позиции самой левой литеры должно быть положительное целое число, не равное нулю, меньшее

GOST EXPERT ГОСТ 22558-89,

или равное количеству литер в данном, соотнесенном имени-данного-1;

б) вычисление длины определяет размер данного, которое будет использовано в операции. Результатом вычисления длины должно быть положительное целое число, не равное нулю. Сумма позиции-самой-левой-литеры и длины минус единица должна быть меньше или равна количеству литер в данном, соотнесенном имени-данного-1.

Если длина не указана, создаваемое модификацией ссылки данное распространяется от позиции-самой-левой-литеры (включая ее) и до самой правой литеры (тоже включая ее) данного, соотнесенного имени-данного-1.

(4) Создаваемое модификацией ссылки данное рассматривается как элементарное данное без фразы JUSTIFIED (СДВИНУТО). Оно относится к тому же классу и категории, что и данное, соотнесенное имени-данного-1, учитывая то обстоятельство, что класс числовой, числовой редактируемый и буквенно-цифровой редактируемый рассматриваются как класс и категория буквенно-цифровые.

4.3.8.4. *Идентификатор*

Термин идентификатор используется в случае, когда имя данного, дублируемое в программе, должно сопровождаться синтаксически правильной комбинацией уточнителей, индексов или модификаторов ссылки, необходимой для обеспечения однозначности ссылки.

Общий формат идентификатора:

имя-данного-1 [{IN / OF} имя-данного-2] ... [{IN / OF} { имя-коммуникации-1 / имя-файла-1 / имя-отчета-1 }]

[({индекс} ...)] [(позиция-самой-левой-литеры: [длина])]

имя-данного-1 [ИЗ имя-данного-2] ... [ИЗ { имя-коммуникации-1 / имя-файла-1 / имя-отчета-1 }]

[({индекс} ...)] [(позиция-самой-левой-литеры: [длина])]

4.3.8.5. *Имя условия*

При наличии явной ссылки каждое из имен условий не должно дублироваться или должно позволять однозначное определение посредством уточнения и (или) индексирования, за исключением случаев, когда соглашения об области действия имен сами обеспечивают однозначность ссылок (ч. 10, п. 1.3.8).

GOST EXPERT ГОСТ 22558-89,

Если для определения однозначности имени условия используется уточнение, соответствующая условная переменная может использоваться как первый уточнитель. При уточнении используется иерархия имен, относящихся к условной переменной.

Если ссылки на условную переменную требуют индексирования, то ссылки на любое из ее имен условия требуют той же комбинации индексов.

Формат и ограничения на совместное использование уточнения и индексирования имен-условий в точности совпадают с форматом и ограничениями для идентификатора, при этом имя-данного-1 нужно заменить на имя-условия-1.

В общем формате и дальнейшем тексте «имя-условия» обозначает уточненное или индексированное имя условия, если это необходимо.

4.4. **Явные и неявные спецификации**

Имеется четыре типа явных и неявных определений, встречающихся в исходной Кобол-программе:

(1) явные и неявные обращения (ссылки) к данным в разделе процедур;

(2) явные и неявные передачи управления;

(3) явные и неявные свойства;

(4) явные и неявные ограничители области действия.

4.4.1. Явные и неявные обращения к данным в разделе процедур

Исходная Кобол-программа может явно или неявно обращаться к данным в операторах раздела процедур. Явное обращение имеет место в том случае, когда имя данного, к которому происходит обращение, записывается в операторе раздела процедур или копируется в разделе процедур с помощью оператора COPY (КОПИРОВАТЬ). Неявное обращение имеет место в том случае, когда оператор раздела процедур обращается к данному без указания его имени. Неявное обращение имеет место также во время выполнения оператора PERFORM (ВЫПОЛНИТЬ), когда индекс или данное, представленное именем индекса или идентификатором во фразах VARYING, AFTER, UNTIL (МЕНЯЯ, ЗАТЕМ, ДО), устанавливаются в начало, изменяются или вычисляются механизмом управления, относящимся к этому оператору PERFORM (ВЫПОЛНИТЬ). Такое неявное обращение имеет место только в том случае, когда данное участвует в выполнении оператора.

4.4.2. Явные и неявные передачи управления

Механизм, управляющий программным потоком, передает управление от оператора к оператору в той последовательности, в которой они записаны в исходной программе, пока явная передача управления не прервет эту последовательность или не останется ни одного выполнимого оператора, которому можно передать уп-

GOST EXPERT ГОСТ 22558-89,

правление. Передача управления от оператора к оператору может происходить без явного указания в операторах раздела процедур; в этом случае она является неявной передачей управления.

Язык Кобол обеспечивает явные и неявные средства передачи управления.

В дополнение к неявной передаче управления в последовательности операторов, неявная передача управления имеет место и тогда, когда нормальный ход выполнения операторов меняется без выполнения оператора ветвления процедуры. Кобол обеспечивает следующие типы неявного изменения управления, прерывающие передачу управления от оператора к оператору:

(1) если параграф выполняется под управлением другого оператора (например, PERFORM (ВЫПОЛНИТЬ), USE (ИСПОЛЬЗОВАТЬ), SORT (СОРТИРОВАТЬ) и MERGE (СЛИТЬ)) и этот параграф является последним в области управляющего оператора, тогда неявная передача управления происходит от последнего оператора в параграфе к механизму управления последнего выполняемого управляющего оператора. Дальше, если параграф, выполняемый под управлением оператора PERFORM (ВЫПОЛНИТЬ), является первым параграфом в области действия этого оператора PERFORM (ВЫПОЛНИТЬ), то неявная передача управления имеет место между механизмом управления, относящимся к этому оператору, и первым оператором в этом параграфе при каждой итерации выполнения параграфа;

(2) при выполнении оператора SORT (СОРТИРОВАТЬ) и MERGE (СЛИТЬ) происходит неявная передача управления к соответствующей процедуре ввода и вывода;

(3) при выполнении оператора Кобола, приводящего к выполнению декларативной секции, происходит неявная передача управления к этой секции. Заметим, что другая неявная передача управления происходит после выполнения декларативной секции, как указано выше (пункт (1)).

Явная передача управления состоит в изменении механизма неявной передачи посредством оператора ветвления процедуры или условного оператора. Явная передача управления может быть вызвана только в результате выполнения оператора ветвления процедуры или условного оператора. Выполнение оператора ветвления процедуры ALTER (ИЗМЕНИТЬ) само по себе не осуществляет явной передачи управления, но влияет на нее при выполнении соответствующего оператора GO TO (ПЕРЕЙТИ). Оператор ветвления EXIT PROGRAM (ВЫЙТИ ИЗ ПРОГРАММЫ) приводит к явной передаче управления только при выполнении этого оператора в вызванной программе.

В этом документе термин «следующий выполнимый оператор» используется для обращения к следующему оператору Кобо-

GOST EXPERT ГОСТ 22558-89,

ла, которому передается управление согласно приведенным выше правилам и правилам, соответствующим каждому элементу языка.

Нет следующего выполнимого оператора, когда программа не содержит раздел процедур, а также для перечисленных ниже операторов:

(1) последний оператор в декларативной секции, когда содержащий его параграф не выполняется под управлением некоторого другого оператора Кобола;

(2) последний оператор в декларативной секции, когда оператор, находящийся в области активного оператора PERFORM (ВЫПОЛНИТЬ), выполняется в другой секции и этот последний оператор декларативной секции не является в то же время последним оператором процедуры, которая является выходом из активного оператора PERFORM (ВЫПОЛНИТЬ);

(3) последний оператор в программе, когда содержащий его параграф не выполняется под управлением некоторого другого оператора Кобола в этой программе;

(4) оператор STOP RUN (ОСТАНОВИТЬ РАБОТУ) или EXIT PROGRAM (ВЫЙТИ ИЗ ПРОГРАММЫ), который передает управление за пределы Кобол-программы;

(5) заголовок конца программы.

Когда нет следующего выполнимого оператора и управление не передается за пределы Кобол-программы, передача управления в программе не определена. Если выполнение программы активировано в недекларативной части процедур другой программы посредством оператора CALL (ВЫЗВАТЬ), выполняется неявный оператор EXIT PROGRAM (ВЫЙТИ ИЗ ПРОГРАММЫ).

4.4.3. Явные и неявные свойства

Свойства могут быть заданы явно и неявно. Любое свойство, которое задано явно, называется явным свойством. Если свойство не задано явно, оно определяется по умолчанию. Такое свойство определяется как неявное.

Например, использование данного нет необходимости задавать в случае, когда его использование DISPLAY (ДЛЯ ВЫДАЧИ).

4.4.4. Явные и неявные ограничители области действия

Ограничители области действия служат для указания границ области действия определенных операторов раздела процедур (п. 6.4.2.4 настоящей части). Ограничители области действия бывают двух типов: явные и неявные.

Явными ограничителями области действия являются следующие:

END-ADD (КОНЕЦ-СЛОЖИТЬ)
END-CALL (КОНЕЦ-ВЫЗВАТЬ)
END-COMPUTE (КОНЕЦ-ВЫЧИСЛИТЬ)

GOST EXPERT ГОСТ 22558-89,

END-DELETE (КОНЕЦ-УДАЛИТЬ)
END DIVIDE (КОНЕЦ-РАЗДЕЛИТЬ)
END-EVALUATE (КОНЕЦ-ОЦЕНИТЬ)
END-IF (КОНЕЦ-ЕСЛИ)
END-MULTIPLY (КОНЕЦ-УМНОЖИТЬ)
END-PERFORM (КОНЕЦ-ВЫПОЛНИТЬ)
END-READ (КОНЕЦ-ЧИТАТЬ)
END-RECEIVE (КОНЕЦ-ПОЛУЧИТЬ)
END-RETURN (КОНЕЦ-ВЕРНУТЬ)
END-REWRITE (КОНЕЦ-ОБНОВИТЬ)
END-SEARCH (КОНЕЦ-ИСКАТЬ)
END-START (КОНЕЦ-ПОДВЕСТИ)
END-STRING (КОНЕЦ-СОБРАТЬ)
END-SUBTRACT (КОНЕЦ-ОТНЯТЬ)
END-UNSTRING (КОНЕЦ-РАЗОБРАТЬ)
END-WRITE (КОНЕЦ-ПИСАТЬ)

Неявными ограничителями области действия являются:

(1) разделитель точка в конце любого предложения, которая заканчивает область действия всех предыдущих операторов, еще не завершенных;

(2) в любом операторе, содержащем другой оператор, следующая фраза внешнего оператора, находящаяся после внутреннего оператора, завершает область действия любого незавершенного внутреннего оператора. Примерами таких фраз являются ELSE (ИНАЧЕ), WHEN (КОГДА), NOT AT END (НЕ В КОНЦЕ) и т. п.

4.5. **Внешний переключатель**.

Внешний переключатель — это устройство оборудования или программное средство, определяемое и именуемое реализацией, и используемое для указания одного из двух альтернативных состояний. Альтернативные состояния относятся к состоянию «включено» или «выключено» соответствующего внешнего переключателя.

Состояние внешнего переключателя может быть опрошено проверкой имен-условий, связанных с этим переключателем. Соответствие имени-условия внешнему переключателю и соответствие указанного пользователем мнемонического-имени имени-реализации, именующему внешний переключатель, устанавливается в параграфе SPECIAL-NAMES (СПЕЦИАЛЬНЫЕ-ИМЕНА) раздела оборудования (ч. 6, п. 4.5.1).

Реализация определяет область действия (программа, единица исполнения и т. п.) каждого внешнего переключателя и возмож-

GOST EXPERT ГОСТ 22558-89,

ности (внешние по отношению к Коболу), используемые для изменения состояния внешнего переключателя. Например, если область действия внешнего переключателя есть единица исполнения, каждое имя-реализации, именующее такой внешний переключатель, соотнесено только одному переключателю, доступному каждой объектной программе, функционирующей в этой единице исполнения.

Состояние некоторых переключателей может изменяться оператором SET (УСТАНОВИТЬ) (ч. 6, п. 6.23).

5. ИСХОДНАЯ КОБОЛ-ПРОГРАММА

5.1. **Введение**

Исходная Кобол-программа — это синтаксически правильный набор операторов Кобола.

5.2. **Организация**

За исключением операторов COPY (КОПИРОВАТЬ) и REPLACE (ЗАМЕНИТЬ) и заголовка конца программы, операторы, статьи, параграфы и секции исходной Кобол-программы группируются в четыре раздела, расположенные друг за другом в следующем порядке:

1. Раздел идентификации
2. Раздел оборудования
3. Раздел данных
4. Раздел процедур.

Конец исходной Кобол-программы указывается либо заголовком конца программы, либо отсутствием дальнейших строк исходной программы.

5.3. **Структура**

Ниже приведен общий формат и порядок представления статей и операторов, в котором записывается исходная Кобол-программа.

5.3.1. Общий формат

раздел-идентификации
[раздел-оборудования]
[раздел-данных]
[раздел-процедур]
[заголовок-конца-программы]

6. РАЗДЕЛЫ

6.1. **Раздел идентификации**

6.1.1. Общее описание

Раздел идентификации определяет программу. Кроме этого, пользователь может включать в параграфы показанного ниже общего формата дату написания программы, дату компиляции исходной программы и другую необязательную информацию.

GOST EXPERT ГОСТ 22558-89,

6.1.2. Организация

Заголовки параграфов определяют тип информации, содержащейся в каждом параграфе. Имя программы должно быть дано в первом параграфе PROGRAM-ID (ПРОГРАММА). Остальные параграфы необязательны и могут быть включены в этот раздел по выбору пользователя в порядке, приведенном в формате ниже.

6.1.3. Структура

Ниже приведен формат параграфов раздела идентификации и определен порядок их следования в исходной программе.

6.1.3.1. *Общий формат*

IDENTIFICATION DIVISION.
PROGRAM-ID. имя-программы.
[AUTHOR. [статья-комментарий] . . .]
[INSTALLATION. [статья-комментарий] . . .]
[DATE-WRITTEN. [статья-комментарий] . . .]
[DATE-COMPILED. [статья-комментарий] . . .]
[SECURITY. [статья-комментарий] . . .]
РАЗДЕЛ ИДЕНТИФИКАЦИИ.
ПРОГРАММА. имя-программы.
[АВТОР. [статья-комментарий] . . .]
[ПРЕДПРИЯТИЕ. [статья-комментарий] . . .]
[ДАТА-НАПИСАНИЯ. [статья-комментарий] . . .]
[ДАТА-ТРАНСЛЯЦИИ. [статья-комментарий] . . .]
[ПОЛНОМОЧИЯ. [статья-комментарий] . . .]

6.2. **Раздел оборудования**

6.2.1. Общее описание

Раздел оборудования описывает стандартные аспекты обработки данных, которые зависят от физических особенностей конкретной машины. Этот раздел позволяет определить конфигурацию компилирующей и объектной машины, а также дать информацию, относящуюся к управлению вводом-выводом, специфическим особенностям оборудования машины и методам управления.

6.2.2. Организация

Раздел оборудования состоит из секции конфигурации и секции ввода-вывода.

Секция конфигурации характеризует исходную и рабочую машины. Эта секция разделена на три параграфа: параграф SOURCE-COMPUTER (ИСХОДНАЯ-МАШИНА), описывающий конфигурацию машины, на которой компилируется исходная программа; параграф OBJECT-COMPUTER (РАБОЧАЯ-МАШИНА), описывающий конфигурацию машины, на которой следует выполнять объектную программу, и параграф SPECIAL-NAMES (СПЕЦИАЛЬНЫЕ-ИМЕНА), который предусматривает средства

GOST EXPERT ГОСТ 22558-89,

определения валютного знака, выбора десятичной точки, определения символических литер, установления соответствия между: именами реализации и определенными пользователем мнемоническими именами; именами алфавитов и наборами литер машины или основными последовательностями; именами классов и наборами литер.

Секция ввода-вывода содержит информацию, необходимую для управления обработкой и передачей данных из внешней среды в объектную программу и обратно. Эта секция разделена на два параграфа. Параграф FILE-CONTROL (УПРАВЛЕНИЕ-ФАЙЛАМИ) называет файл и ставит его в соответствие внешней среде. Параграф I-O-CONTROL (УПРАВЛЕНИЕ-ВВОДОМ-ВЫВОДОМ) определяет специальные методы управления, которые следует использовать в объектной программе.

6.2.3. Структура

Ниже приводится общий формат секций и параграфов в разделе оборудования и определяется порядок представления их в исходной программе.

6.2.3.1. *Общий формат*

ENVIRONMENT DIVISION.
[CONFIGURATION SECTION.
[SOURCE-COMPUTER. [статья-исходной-машины]]
[OBJECT-COMPUTER. [статья-объектной-машины]]
[SPECIAL-NAMES. [статья-специальных-имен]]]
[INPUT-OUTPUT SECTION.
FILE-CONTROL. {статья-управления-файлом} . . .
[I-O-CONTROL. [статья-управления-вводом-выводом]]]

РАЗДЕЛ ОБОРУДОВАНИЯ.
[СЕКЦИЯ КОНФИГУРАЦИИ.
[ИСХОДНАЯ-МАШИНА. [статья-исходной-машины]]
[РАБОЧАЯ-МАШИНА. [статья-объектной-машины]]
[СПЕЦИАЛЬНЫЕ-ИМЕНА. [статья-специальных-имен]]]
[СЕКЦИЯ ВВОДА-ВЫВОДА.
УПРАВЛЕНИЕ-ФАЙЛАМИ. {статья-управления-файлом} . . .
[УПРАВЛЕНИЕ-ВВОДОМ-ВЫВОДОМ. [статья-управления-вводом-выводом]]]

6.3. Раздел данных

6.3.1. Общий подход

Раздел данных описывает данные, которые объектная программа должна воспринимать как входные, обрабатывать, создавать и выдавать как выходные.

GOST EXPERT ГОСТ 22558-89,

6.3.2. Физические и логические аспекты описания данных

6.3.2.1. *Организация раздела данных*

Раздел данных подразделяется на секции: файлов, рабочей памяти, связи, коммуникаций и отчетов.

Секция файлов определяет структуру файлов данных. Каждый файл определяется статьей описания файла и одним или несколькими описаниями записи, или статьей описания файла и одной или несколькими статьями описаний отчета. Описания записи указываются непосредственно за статьей описания файла. Когда описание файла определяет файл как используемый только в качестве выходного файла генератора отчетов, описание записи должно быть опущено. Статьи описания отчетов указываются в отдельной секции раздела данных — в секции отчетов.

Секция рабочей памяти описывает записи и подчиненные данные, которые не являются частями внешних файлов данных, а получаются и обрабатываются во внутренней памяти, а также данные, значения которых определены в исходной программе и не меняются в процессе выполнения рабочей программы.

Секция связи указывается в вызываемой программе и описывает данные, к которым должны обращаться вызывающая и вызываемая программы. Структура этой секции аналогична секции рабочей памяти.

Секция коммуникаций описывает данные в исходной программе, которые служат для взаимодействия между системой управления сообщениями и программой.

Секция отчетов описывает содержание и формат подлежащих выдаче отчетов.

6.3.2.2. *Структура раздела данных*

Ниже приведен общий формат секций раздела данных и определен порядок их представления в исходной программе.

6.3.2.2.1. *Общий формат*

```
DATA DIVISION.
[FILE SECTION.

 [ статья-описания-файла {статья-описания-записи} ...
   статья-описания-сортируемого-сливаемого-файла
                                {статья-описания-записи} ...
   статья-описания-файла-отчетов                            ] ... ]

[WORKING-STORAGE SECTION.

 [ статья-описания-уровня-77
   статья-описания-записи                                   ] ... ]
```


GOST EXPERT ГОСТ 22558-89,

[LINKAGE SECTION.

[статья-описания-уровня-77 | статья-описания-записи] ...]

[COMMUNICATION SECTION.

[статья-описания-коммуникации {статья-описания-записи}...]...]

[REPORT SECTION.

[статья-описания-отчета {статья-описания-группы-отчета}...]...]

РАЗДЕЛ ДАННЫХ.

[СЕКЦИЯ ФАЙЛОВ.

[статья-описания-файла {статья-описания-записи} ...
  статья-описания-сортируемого-сливаемого-файла {статья-описания-записи} ...
  статья описания-файла-отчетов] ...]

[СЕКЦИЯ РАБОЧЕЙ-ПАМЯТИ.

[статья-описания-уровня-77 | статья-описания-записи] ...]

[СЕКЦИЯ СВЯЗИ.

[статья-описания-уровня-77 | статья-описания-записи] ...]

[СЕКЦИЯ КОММУНИКАЦИЙ.

[статья-описания-коммуникации [статья-описания-записи]...]...]

[СЕКЦИЯ ОТЧЕТОВ.

[статья-описания-отчета {статья-описания-группы-отчета}...]...]

6.4. **Раздел процедур**

6.4.1. Общее описание

Раздел процедур содержит декларативы и процедуры.

6.4.1.1. *Декларативы*

Секции деклараатив должны быть сгруппированы в начале раздела процедур; им предшествует ключевое слово DECLARATIVES (ДЕКЛАРАТИВЫ) и их заключают ключевые слова END DECLARATIVES (КОНЕЦ ДЕКЛАРАТИВ) (ч. 7, п. 4.6; ч. 8, п. 4.8; ч. 9, п. 4.8; ч. 13, пп. 4.8, 4.9 и ч. 15, п. 3.2).

6.4.1.2. *Процедуры*

Процедура состоит из параграфа или группы последовательных параграфов, либо из секции или из группы последовательных секций внутри раздела процедур. Если один параграф включен в секцию, то и все параграфы должны быть включены в секции. Имя процедуры — это слово, используемое для обращения к парагра-

GOST EXPERT ГОСТ 22558-89,

фу или к секции в исходной программе. Имя процедуры может быть именем параграфа, которое может быть уточнено, или именем секции.

Секция состоит из заголовка секции, за которым следует нуль, один или несколько последовательно записанных параграфов. Секция заканчивается непосредственно перед следующей секцией или по окончании раздела процедур, или в декларативной части раздела процедур ключевыми словами END DECLARATIVES (КОНЕЦ ДЕКЛАРАТИВ).

Параграф состоит из имени параграфа, за которым следует точка с пробелом и нуль, одно или несколько последовательно записанных предложений. Параграф оканчивается непосредственно перед следующим именем параграфа или именем секции, или по окончании раздела процедур, или в декларативах в разделе процедур ключевыми словами END DECLARATIVES (КОНЕЦ ДЕКЛАРАТИВ).

Предложение состоит из одного или нескольких операторов и заканчивается разделителем точкой.

Оператор — синтаксически правильная комбинация слов, литералов и разделителей, начинающаяся глаголом языка Кобол.

Термин «идентификатор» определяется как слово или слова, необходимые для однозначности обращения к данному.

6.4.1.3. *Выполнение*

Выполнение начинается с первого оператора раздела процедур, исключая декларативы. Затем операторы выполняются в том порядке, в котором они представлены для компиляции, исключение из которого составляет порядок, описанный правилами этой части.

6.4.1.4. *Структура раздела процедур*

6.4.1.4.1. *Заголовок раздела процедур*

Раздел процедур должен начинаться следующим заголовком:

PROCEDURE DIVISION [USING {имя-данного-1}...].

РАЗДЕЛ ПРОЦЕДУР [ИСПОЛЬЗУЯ {имя-данного-1}...].

6.4.1.4.2. Структура тела раздела процедур

Тело раздела процедур должно соответствовать одному из следующих форматов:

Формат 1

```
DECLARATIVES.
   {имя-секции SECTION [номер сегмента].
   Оператор USE
    [имя-параграфа.
    [предложение] ...] ...}...
END DECLARATIVES.]
   {имя-секции SECTION [номер-сегмента].
```


GOST EXPERT ГОСТ 22558-89,

[имя-параграфа.
[предложение] ...] ...}...
[ДЕКЛАРАТИВЫ.
{СЕКЦИЯ имя-секции [номер-сегмента].
Оператор ИСПОЛЬЗОВАТЬ
[имя-параграфа.
[предложение] ...] ...}...
КОНЕЦ ДЕКЛАРАТИВ.]
{СЕКЦИЯ имя-секции [номер-сегмента].
[имя-параграфа.
[предложение] ...] ...}...
Формат 2
{имя-параграфа.
[предложение] ...}...

6.4.2. Предложения и операторы

Имеется четыре типа операторов: повелительные, условные, операторы, управляющие компиляцией, и операторы с ограничителем области действия.

Имеется три типа предложений: повелительные, условные и предложения, управляющие компиляцией.

6.4.2.1. *Условные операторы, предложения и фразы*

6.4.2.1.1. *Определение условного оператора*

Условный оператор указывает, что должно быть определено значение истинности условия и что последующие действия объектной программы зависят от этого значения истинности.

Условным оператором является один из следующих:

(1) оператор EVALUATE (ОЦЕНИТЬ), IF (ЕСЛИ), SEARCH (ИСКАТЬ) или RETURN (ВЕРНУТЬ);

(2) оператор READ (ЧИТАТЬ) с фразой AT END (В КОНЦЕ), NOT AT END (НЕ В КОНЦЕ), INVALID KEY (ПРИ ОШИБКЕ КЛЮЧА) или NOT INVALID KEY (БЕЗ ОШИБКИ КЛЮЧА);

(3) оператор WRITE (ПИСАТЬ) с фразой INVALID KEY (ПРИ ОШИБКЕ КЛЮЧА), NOT INVALID KEY (БЕЗ ОШИБКИ КЛЮЧА), END-OF-PAGE (В КОНЦЕ СТРАНИЦЫ) или NOT END-OF-PAGE (НЕ В КОНЦЕ СТРАНИЦЫ);

(4) операторы DELETE (УДАЛИТЬ), REWRITE (ОБНОВИТЬ) и START (ПОДВЕСТИ) с фразой INVALID KEY (ПРИ ОШИБКЕ КЛЮЧА) или NOT INVALID KEY (БЕЗ ОШИБКИ КЛЮЧА);

(5) арифметические операторы ADD (СЛОЖИТЬ), COMPUTE (ВЫЧИСЛИТЬ), DIVIDE (РАЗДЕЛИТЬ), MULTIPLY (УМНОЖИТЬ), SUBTRACT (ОТНЯТЬ) с фразой ON SIZE ERROR (ПРИ ПЕРЕПОЛНЕНИИ) или NOT ON SIZE ERROR (БЕЗ ПЕРЕПОЛНЕНИЯ);

GOST EXPERT ГОСТ 22558-89,

(6) оператор RECEIVE (ПОЛУЧИТЬ) с фразой NO DATA (НЕТ ДАННЫХ) или WITH DATA (ЕСТЬ ДАННЫЕ);

(7) оператор STRING (СОБРАТЬ) или UNSTRING (РАЗОБРАТЬ) с фразой ON OVERFLOW (ПРИ ПЕРЕПОЛНЕНИИ) или NOT ON OVERFLOW (БЕЗ ПЕРЕПОЛНЕНИЯ);

(8) оператор CALL (ВЫЗВАТЬ) с фразой ON OVERFLOW (ПРИ ПЕРЕПОЛНЕНИИ), ON EXCEPTION (ПРИ ОШИБКЕ) или NOT ON EXCEPTION (БЕЗ ОШИБКИ).

6.4.2.1.2. *Определение условной фразы*

Условная фраза указывает действие, которое должно быть выполнено при определении значения истинности условия в результате выполнения условного оператора.

Условной фразой является одна из следующих:

(1) фраза AT END (В КОНЦЕ) или NOT AT END (НЕ В КОНЦЕ) в операторе READ (ЧИТАТЬ);

(2) фраза INVALID KEY (ПРИ ОШИБКЕ КЛЮЧА) или NOT INVALID KEY (БЕЗ ОШИБКИ КЛЮЧА) в операторах DELETE (УДАЛИТЬ), READ (ЧИТАТЬ), REWRITE (ОБНОВИТЬ), START (ПОДВЕСТИ) или WRITE (ПИСАТЬ);

(3) фраза END-OF-PAGE (В КОНЦЕ СТРАНИЦЫ) или NOT END-OF-PAGE (НЕ В КОНЦЕ СТРАНИЦЫ) в операторе WRITE (ПИСАТЬ);

(4) фраза SIZE ERROR (ПРИ ПЕРЕПОЛНЕНИИ) или NOT ON SIZE ERROR (БЕЗ ПЕРЕПОЛНЕНИЯ) в операторах ADD (СЛОЖИТЬ), COMPUTE (ВЫЧИСЛИТЬ), DIVIDE (РАЗДЕЛИТЬ), MULTIPLY (УМНОЖИТЬ) или SUBTRACT (ОТНЯТЬ);

(5) фраза NO DATA (НЕТ ДАННЫХ) или WITH DATA (ЕСТЬ ДАННЫЕ) в операторе RECEIVE (ПОЛУЧИТЬ);

(6) фраза ON OVERFLOW (ПРИ ПЕРЕПОЛНЕНИИ) или NOT ON OVERFLOW (БЕЗ ПЕРЕПОЛНЕНИЯ) в операторе STRING (СОБРАТЬ) или UNSTRING (РАЗОБРАТЬ);

(7) фраза ON OVERFLOW (ПРИ ПЕРЕПОЛНЕНИИ), ON EXCEPTION (ПРИ ОШИБКЕ) или NOT ON EXCEPTION (БЕЗ ОШИБКИ) в операторе CALL (ВЫЗВАТЬ).

6.4.2.1.3. *Определение условного предложения*

Условное предложение является условным оператором, возможно, с предшествующим повелительным оператором, заканчивающимся разделителем точкой.

6.4.2.2. *Операторы и предложения, управляющие компиляцией*

6.4.2.2.1. *Определение оператора, управляющего компиляцией*

Оператор, управляющий компиляцией, состоит из глагола, управляющего компиляцией, и операндов. Глаголами, управляющими

GOST EXPERT ГОСТ 22558-89,

компиляцией, являются COPY (КОПИРОВАТЬ), REPLACE (ЗАМЕНИТЬ) и USE (ИСПОЛЬЗОВАТЬ) (ч. 12, пп. 2, 3; ч. 7, п. 4.6; ч. 8, п. 4.8; ч. 9, п. 4.8; ч. 13, п. 4.8; ч. 15, п. 3.2). Оператор, управляющий компиляцией, приводит к выполнению определенного действия во время компиляции.

6.4.2.2.2. *Определение предложения, управляющего компиляцией*

Предложение, управляющее компиляцией, состоит из единственного оператора, управляющего компиляцией, заканчивающегося разделителем точкой.

6.4.2.3. *Повелительные операторы и повелительные предложения*

6.4.2.3.1. *Определение повелительного оператора*

Повелительный оператор начинается с повелительного глагола и указывает безусловное действие, которое должно выполняться в объектной программе, или является условным оператором, ограниченным явным ограничителем области действия (оператором с ограничителем области действия).

Повелительный оператор может состоять из последовательности повелительных операторов, каждый из которых может быть отделен от следующего разделителем. Повелительные глаголы следующие:

	ACCEPT	(ПРИНЯТЬ)
(1)	ADD	(СЛОЖИТЬ)
	ALTER	(ИЗМЕНИТЬ)
(7)	CALL	(ВЫЗВАТЬ)
	CANCEL	(ОСВОБОДИТЬ)
	CLOSE	(ЗАКРЫТЬ)
(1)	COMPUTE	(ВЫЧИСЛИТЬ)
	CONTINUE	(ПРОДОЛЖИТЬ)
(2)	DELETE	(УДАЛИТЬ)
	DISABLE	(ЗАПРЕТИТЬ)
	DISPLAY	(ВЫДАТЬ)
(1)	DIVIDE	(РАЗДЕЛИТЬ)
	ENABLE	(РАЗРЕШИТЬ)
	EXIT	(ВЫЙТИ)
	GENERATE	(ГЕНЕРИРОВАТЬ)
	GO TO	(ПЕРЕЙТИ)
	INITIALIZE	(ИНИЦИИРОВАТЬ)
	INITIATE	(НАЧАТЬ)
	INSPECT	(ПРОСМОТРЕТЬ)
	MERGE	(СЛИТЬ)
	MOVE	(ПОМЕСТИТЬ)
(1)	MULTIPLY	(УМНОЖИТЬ)
	OPEN	(ОТКРЫТЬ)
	PERFORM	(ВЫПОЛНИТЬ)

GOST EXPERT ГОСТ 22558-89,

	PURGE	(ОЧИСТИТЬ)
(5)	READ	(ЧИТАТЬ)
(4)	RECEIVE	(ПОЛУЧИТЬ)
	RELEASE	(ПЕРЕДАТЬ)
(2)	REWRITE	(ОБНОВИТЬ)
	SEND	(ПОСЛАТЬ)
	SET	(УСТАНОВИТЬ)
	SORT	(СОРТИРОВАТЬ)
(2)	START	(ПОДВЕСТИ)
	STOP	(ОСТАНОВИТЬ)
(3)	STRING	(СОБРАТЬ)
(1)	SUBTRACT	(ОТНЯТЬ)
	SUPPRESS	(ПОДАВИТЬ)
	TERMINATE	(ЗАКОНЧИТЬ)
(3)	UNSTRING	(РАЗОБРАТЬ)
(6)	WRITE	(ПИСАТЬ)

Цифры в скобках обозначают следующие варианты форматов операторов:

(1) — без необязательных фраз ON SIZE ERROR (ПРИ ПЕРЕПОЛНЕНИИ) и NOT ON SIZE ERROR (БЕЗ ПЕРЕПОЛНЕНИЯ);

(2) — без необязательных фраз INVALID KEY (ПРИ ОШИБКЕ КЛЮЧА) и NOT INVALID KEY (БЕЗ ОШИБКИ КЛЮЧА);

(3) — без необязательных фраз ON OVERFLOW (ПРИ ПЕРЕПОЛНЕНИИ) и NOT ON OVERFLOW (БЕЗ ПЕРЕПОЛНЕНИЯ);

(4) — без необязательных фраз NO DATA (НЕТ ДАННЫХ) и WITH DATA (ЕСТЬ ДАННЫЕ);

(5) — без необязательных фраз AT END (В КОНЦЕ) и NOT AT END (НЕ В КОНЦЕ), INVALID KEY (ПРИ ОШИБКЕ КЛЮЧА), NOT INVALID KEY (БЕЗ ОШИБКИ КЛЮЧА);

(6) — без необязательных фраз INVALID KEY (ПРИ ОШИБКЕ КЛЮЧА), NOT INVALID KEY (БЕЗ ОШИБКИ КЛЮЧА), END-OF-PAGE (В КОНЦЕ СТРАНИЦЫ), NOT AT END-OF-PAGE (НЕ В КОНЦЕ СТРАНИЦЫ);

(7) — без необязательных фраз ON OVERFLOW (ПРИ ПЕРЕПОЛНЕНИИ), ON EXCEPTION (ПРИ ОШИБКЕ) и NOT ON EXCEPTION (БЕЗ ОШИБКИ).

В общем формате операторов «повелительный-оператор» нужно понимать как последовательность следующих друг за другом повелительных операторов, которая должна оканчиваться точкой или любой фразой оператора, содержащего этот «повелительный-оператор».

6.4.2.3.2. *Определение повелительного предложения*

Повелительное предложение — это повелительный оператор, заканчивающийся разделителем точка.

GOST EXPERT ГОСТ 22558-89,

6.4.2.4. *Операторы с ограничителем области действия*

Оператором с ограничителем области действия является оператор, включающий явный ограничитель области действия (см. п. 4.4.4 настоящей части).

6.4.3. Область действия операторов

Ограничители области действия определяют область действия конкретных операторов раздела процедур. Операторы, включающие явные ограничители области действия, называются операторами с ограничителем области действия (см. пп. 4.4.4, 6.4.2.4 настоящей части). Область действия операторов, содержащихся в других операторах (вложенные операторы), может быть также неявно ограниченной.

Когда операторы вложены в другие операторы, в которых могут быть необязательные условные фразы, каждая необязательная условная фраза рассматривается, как если бы она была следующей фразой ближайшего предыдущего незавершенного оператора, с которым эта фраза могла бы быть связана, но с которым ни одна такая фраза еще не связана. Незавершенный оператор — это оператор, не имеющий заведомо явного или неявного завершения (см. п. 4.4.4 настоящей части).

7. ФОРМАТ ПРЕДСТАВЛЕНИЯ

7.1. Общее описание

Формат представления, который обеспечивает стандартный способ представления исходной Кобол-программы и библиотечного текста Кобола, описывается в терминах позиций литер в строке входной-выходной среды. Каждая реализация должна определять, что понимается под строкой и позицией литер. Согласно этим определениям каждый Кобол-компилятор воспринимает исходную программу, записанную в этом формате представления, и создает выходную распечатку исходной программы в формате представления.

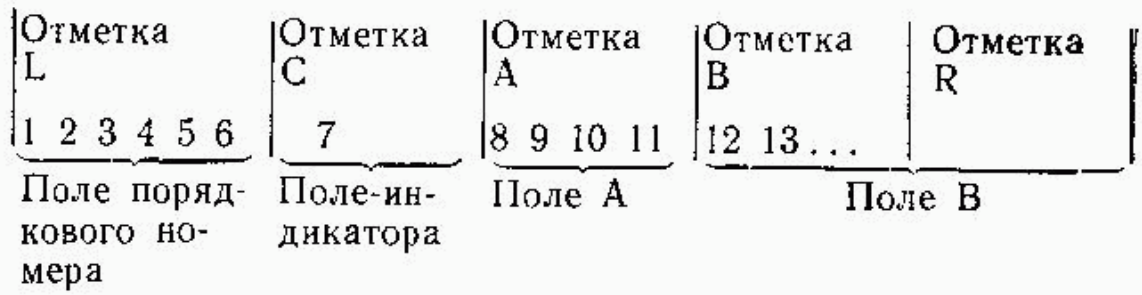
Правила простановки пробелов, приведенные при рассмотрении формата представления, имеют преимущество по отношению ко всем другим правилам для простановки пробелов.

Разделы исходной программы должны быть упорядочены следующим образом: раздел идентификации, раздел оборудования, раздел данных, раздел процедур. Каждый раздел должен быть записан в соответствии с правилами для формата представления.

7.2. Описание формата представления

Формат представления для строки изображен ниже.

GOST EXPERT ГОСТ 22558-89,



Отметка L находится левее самой левой позиции литеры в строке.

Отметка С находится между шестой и седьмой позициями литер в строке.

Отметка А находится между седьмой и восьмой позициями литер в строке.

Отметка В находится между одиннадцатой и двенадцатой позициями литер в строке.

Отметка R находится справа от самой правой позиции литеры в строке.

Поле порядкового номера занимает шесть позиций литер (1—6) и расположено между отметками L и С.

Поле индикатора занимает одну позицию литеры, начиная от отметки С.

Поле А занимает четыре позиции литеры, начиная от отметки А.

Поле В занимает конечное число позиций литер, оговоренное реализацией, начиная от отметки В.

7.2.1. Порядковые номера

Поле порядкового номера может быть использовано для обозначения строк исходной программы. Порядковый номер определяется пользователем и может состоять из любых литер из набора литер машины. Допускается дублирование порядковых номеров и появление их не в последовательном порядке.

7.2.2. Продолжение строк

Каждое предложение, статья или фраза, требующие более одной строки, могут быть продолжены посредством помещения в поле В следующей строки (строк). Эти последующие строки называются строками продолжения. Строка, подлежащая продолжению, называется продолжаемой строкой. Каждое слово, литерал или строка литер шаблона может быть расчленена таким образом, что ее часть попадает в строку продолжения.

Дефис в поле индикатора строки указывает, что первая литера, отличная от пробела в поле В данной строки, является преемником последней литеры (отличной от пробела) предыдущей строки без какого бы то ни было пробела между ними. Однако, если продолжаемая строка содержит нечисловой литерал без заключающей кавычки, то первой отличной от пробела литерой в поле В строки

GOST EXPERT ГОСТ 22558-89,

продолжения должен быть знак кавычки и продолжение начинается литерой, непосредственно следующей за знаком кавычки. Все пробелы в конце продолжаемой строки считаются частью литерала. Поле А строки продолжения должно быть заполнено пробелами.

Если в поле индикатора строки дефис отсутствует, то считается, что за последней литерой предыдущей строки следует пробел.

Обе литеры, составляющие разделитель «==», должны располагаться в одной строке.

7.2.3. Строка пробелов

Строка пробелов — строка, в которой от отметки С до отметки R включительно находятся пробелы. Строка пробелов допустима в любом месте исходной программы (см. п. 7.2.2 настоящей части).

7.2.4. Строки комментария

Строка комментария — строка, содержащая литеру * или / в поле индикатора строк. Строка комментария допустима в любом месте исходной программы после заголовка раздела идентификации и в любом месте библиотечного текста Кобола. В поле А и в поле В этой строки может быть включена любая комбинация литер из набора литер машины. Звездочка или дробная черта и литеры в поле А и В воспроизводятся при распечатке, но служат только для документации и не проверяются на синтаксическую правильность. Литера / в поле индикатора строки вызывает печать комментария на следующей странице распечатки исходной программы; звездочка в поле индикатора строки вызывает печать комментария на следующей доступной строке распечатки.

7.2.5. Псевдотекст

Строка литер и ограничители, образующие псевдотекст, могут начинаться либо от отметки А, либо от отметки В. Однако, если имеется дефис в поле индикатора строки, которая следует за открывающим ограничителем псевдотекста, поле А этой строки должно быть заполнено пробелами и к формированию слов текста применяются обычные правила продолжения строк (см. п. 7.2.2 настоящей части).

7.3. **Форматы раздела, секции, параграфа**

7.3.1. Заголовок раздела

Заголовок раздела начинается в поле А.

7.3.2. Заголовок секции

Заголовок секции начинается в поле А.

Секция состоит из нуля, одного или нескольких параграфов в разделах оборудования и процедур или из нуля, одной или нескольких статей в разделе данных.

GOST EXPERT ГОСТ 22558-89,

7.3.3. Заголовок параграфа, имя параграфа и параграф

Параграф состоит из имени параграфа, за которым следует разделитель точка, и нуля, одного или нескольких предложений или из заголовка параграфа, за которым следует одна или несколько статей.

Заголовок или имя параграфа начинается от отметки А и может быть в любой строке, следующей за первой строкой раздела или секции.

Первое предложение или статья параграфа начинается в той же строке, что и заголовок параграфа или имя параграфа, или в поле В следующей непустой строки, не являющейся строкой комментария. Следующие предложения или статьи либо начинаются в поле В той же строки, что и предыдущее предложение или статья, или в поле В следующей непустой строки, не являющейся строкой комментария.

Предложения или статьи параграфа могут быть продолжены в последующих строках (см. п. 7.2.2 настоящей части).

7.4. **Статьи раздела данных**

Каждая статья раздела данных начинается с индикатора уровня или номера уровня, затем следуют пробел, имя субъекта статьи, если оно указано, и последовательность независимых фраз, описывающих данное. Последняя фраза всегда заканчивается разделителем точка (точка с последующим пробелом).

Имеется два типа статей раздела данных: начинающиеся с индикатора уровня и начинающиеся с номера уровня.

Индикатором уровня может быть любое из зарезервированных слов FD (ОФ), SD (ОС), CD (ОК), RD (ОО).

В статьях раздела данных, которые начинаются с индикатора уровня, индикатор уровня начинается в поле А, за ним следует хотя бы один пробел и затем имя субъекта статьи и соответствующая описывающая информация.

Статьи раздела данных, которые начинаются номерами уровней, называются статьями описания данных.

Номером уровня может быть любое целое из следующего множества: 01, 02, . . . , 49, 66, 77, 88. Номера уровня в диапазоне 01, 02, . . . , 09 записываются либо в виде одной цифры, либо в виде нуля, за которым следует значащая цифра. По крайней мере один пробел должен отделять номер уровня от следующих за ним слов.

В статьях описания данных, имеющих номера уровней 01 или 77, номер уровня начинается в поле А, за ним следует хотя бы один пробел, затем соответствующее имя записи или данного, если они определены, и соответствующая описывающая информация.

GOST EXPERT ГОСТ 22558-89,

Статьи описаний данных могут быть записаны с отступами по отношению к отметке А. Каждая новая статья описания данного может начинаться через любое число пробелов вправо от отметки А, за исключением статей описания данного с номером уровня 01 или 77, которые должны начинаться в поле А. Размер допустимого отступа вправо зависит от величины поля В, которая определяется реализацией. Статьи при выводе записываются с отступами только тогда, когда они записаны с отступами на вводе. Запись с отступами не влияет на значимость номеров уровней.

7.5. **Декларативы**

Никакой другой текст не должен появляться в той же строке, что и ключевое слово DECLARATIVES (ДЕКЛАРАТИВЫ) или ключевые слова END DECLARATIVES (КОНЕЦ ДЕКЛАРАТИВ), которые обрамляют декларативную часть раздела процедур. Слова DECLARATIVES (ДЕКЛАРАТИВЫ) и END DECLARATIVES (КОНЕЦ ДЕКЛАРАТИВ) должны начинаться в поле А, за ними должен следовать разделитель точка.

7.6. **Заголовок конца программы**

Заголовок конца программы должен начинаться в поле А.

8. ЗАРЕЗЕРВИРОВАННЫЕ СЛОВА КОБОЛА

ACCEPT
ACCESS
ADD
ADVANCING
AFTER
ALL
ALPHABET
ALPHABETIC
ALPHABETIC-LOWER
ALPHABETIC-UPPER
ALPHANUMERIC
ALPHANUMERIC-EDITED
ALSO
ALTER
ALTERNATE
AND
ANY
ARE
AREA
AREAS
ASCENDING
ASSIGN
AT
AUTHOR
BEFORE
BINARY
BLANK
BLOCK
BOTTOM
BY
CALL
CANCEL
CD
CF
CH
CHARACTER
CHARACTERS
CLASS
CLOCK-UNITS
CLOSE
CODE
CODE-SET
COLLATING
COLUMN
COMMA
COMMON
COMMUNICATION

GOST EXPERT ГОСТ 22558-89,

COMP
COMPUTATIONAL
COMPUTE
CONFIGURATION
CONTAINS
CONTENT
CONTINUE
CONTROL
CONTROLS
CONVERTING
COPY
CORR
CORRESPONDING
COUNT
CURRENCY
DATA
DATE
DATE-COMPILED
DATE-WRITTEN
DAY
DAY-OF-WEEK
DE
DEBUG-CONTENTS
DEBUG-ITEM
DEBUG-LINE
DEBUG-NAME
DEBUG-SUB-1
DEBUG-SUB-2
DEBUG-SUB-3
DEBUGGING
DECIMAL-POINT
DECLARATIVES
DELETE
DELIMITED
DELIMITER
DEPENDING
DESCENDING
DESTINATION
DETAIL
DISABLE
DISPLAY
DIVIDE
DIVISION
DOWN
DUPLICATES
DYNAMIC

EGI
ELSE
EMI
ENABLE
END
END-ADD
END-CALL
END-COMPUTE
END-DELETE
END-DIVIDE
END-EVALUATE
END-IF
END-MULTIPLY
END-OF-PAGE
END-PERFORM
END-READ
END-RECEIVE
END-RETURN
END-REWRITE
END-SEARCH
END-START
END-STRING
END-SUBTRACT
END-UNSTRING
END-WRITE
ENTER
ENVIRONMENT
EOP
EQUAL
ERROR
ESI
EVALUATE
EVERY
EXCEPTION
EXIT
EXTEND
EXTERNAL

FALSE
FD
FILE
FILE-CONTROL
FILLER
FINAL
FIRST
FOOTING

GOST EXPERT ГОСТ 22558-89,

FOR
FROM

GENERATE
GIVING
GLOBAL
GO
GREATER
GROUP

HEADING
HIGH-VALUE
HIGH-VALUES

I-O
I-O-CONTROL
IDENTIFICATION
IF
IN
INDEX
INDEXED
INDICATE
INITIAL
INITIALIZE
INITIATE
INPUT
INPUT-OUTPUT
INSPECT
INSTALLATION
INTO
INVALID
IS

JUST
JUSTIFIED

KEY

LABEL
LAST
LEADING
LEFT
LENGTH
LESS
LIMIT
LIMITS
LINAGE
LINAGE-COUNTER
LINE
LINE-COUNTER
LINES
LINKAGE
LOCK
LOW-VALUE
LOW-VALUES

MEMORY
MERGE
MESSAGE
MODE
MODULES
MOVE
MULTIPLE
MULTIPLY

NATIVE
NEGATIVE
NEXT
NO
NOT
NUMBER
NUMERIC
NUMERIC-EDITED

OBJECT-COMPUTER
OCCURS
OF
OFF
OMITTED
ON
OPEN
OPTIONAL
OR
ORDER
ORGANIZATION
OTHER
OUTPUT
OVERFLOW

PACKED-DECIMAL
PADDING
PAGE

GOST EXPERT ГОСТ 22558-89,

PAGE-COUNTER
PERFORM
PF
RH
PIC
PICTURE
PLUS
POINTER
POSITION
POSITIVE
PRINTING
PROCEDURE
PROCEDURES
PROCEED
PROGRAM
PROGRAM-ID
PURGE

QUEUE
QUOTE
QUOTES

RANDOM
RD
READ
RECEIVE
RECORD
RECORDS
REDEFINES
REEL
REFERENCE
REFERENCES
RELATIVE
RELEASE
REMAINDER
REMOVAL
RENAMES
REPLACE
REPLACING
REPORT
REPORTING
REPORTS
RERUN
RESERVE
RESET
RETURN
REVERSED
REWIND
REWRITE
RF
RH
RIGHT
ROUNDED
RUN

SAME
SD
SEARCH
SECTION
SECURITY
SEGMENT
SEGMENT-LIMIT
SELECT
SEND
SENTENCE
SEPARATE
SEQUENCE
SEQUENTIAL
SET
SIGN
SIZE
SORT
SORT-MERGE
SOURCE
SOURCE-COMPUTER
SPACE
SPACES
SPECIAL-NAMES
STANDARD
STANDARD-1
STANDARD-2
STANDARD-R
START
STATUS
STOP
STRING
SUB-QUEUE-1
SUB-QUEUE-2
SUB-QUEUE-3
SUBTRACT
SUM
SUPPRESS

GOST EXPERT ГОСТ 22558-89,

SYMBOLIC
SYNC
SYNCHRONIZED

TABLE
TALLYING
TAPE
TERMINAL
TERMINATE
TEST
TEXT
THAN
THEN
THROUGH
THRU
TIME
TIMES
TO
TOP
TRAILING
TRUE
TYPE

UNIT
UNSTRING
UNTIL
UP
UPON
USAGE
USE
USING

VALUE
VALUES
VARYING

WHEN
WITH
WORDS
WORKING-STORAGE
WRITE

ZERO
ZEROES
ZEROS

\+
\-
*
/
**
>
<
=
>=
<=

GOST EXPERT ГОСТ 22558-89,

АВТОР
АДРЕСАТ
АДРЕСАТОВ
АЛФАВИТ

БЕЗ
БЛОКЕ
БОЛЬШЕ
БУКВЕННОЕ
БЦ
БЦР

В
ВАЛЮТНЫЙ
ВВОД
ВВОД-ВЫВОД
ВВОДА
ВВОДА-ВЫВОДА
ВЕДУЩИЕ
ВЕРНУТЬ
ВЕРСТКА
ВЕРХНЕЕ
ВКЛ
ВКЛЮЧЕНО
ВЛЕВО
ВНЕШНЕЕ
ВНУТРЕННИЙ
ВОЗРАСТАНИЮ
ВОЙТИ
ВПРАВО
ВРЕМЯ
ВСЕ
ВСЕМИ
ВСЕХ
ВХОДНОЙ
ВХОДНОЙ-ВЫХОДНОЙ
ВХОДНЫХ
ВХОДНЫХ-ВЫХОДНЫХ
ВЫВОД
ВЫВОДА
ВЫДАТЬ
ВЫДАЧИ
ВЫДЕЛЕНО
ВЫЗВАТЬ
ВЫЙТИ
ВЫКЛ
ВЫКЛЮЧЕНО
ВЫПОЛНИТЬ
ВЫХОДНОЙ
ВЫХОДНЫХ
ВЫЧ
ВЫЧИСЛЕНИЙ
ВЫЧИСЛИТЬ
ВЫЧИТАЯ

ГЕНЕРИРОВАТЬ
ГЛОБАЛЬНО
ГЛОБАЛЬНОЕ
ГРАНИЦА
ГРУППА
ГРУППУ

ДАННОЕ
ДАННЫЕ
ДАННЫЕ-ОТЛАДКИ
ДАННЫХ
ДАТА
ДАТА-НАПИСАНИЯ
ДАТА-ТРАНСЛЯЦИИ
ДАТУ
ДВОИЧНОЕ
ДЕКЛАРАТИВ
ДЕКЛАРАТИВЫ
ДЕНЬ
ДЕНЬ-НЕДЕЛИ
ДЕСЯТИЧНАЯ
ДЕСЯТИЧНОЕ
ДИНАМИЧЕСКИЙ
ДЛИНА
ДЛЯ
ДО
ДОПОЛНИТЕЛЬНЫЙ
ДОПОЛНЯЕМЫЙ
ДОПОЛНЯЕМЫХ
ДОСТУП
ДУБЛИРОВАНИЕМ

ЕДИНИЦ-ВРЕМЕНИ
ЕСЛИ
ЕСТЬ

GOST EXPERT ГОСТ 22558-89,

ЗАВИСИМОСТИ
ЗАГОЛОВОК
ЗАКОНЧИТЬ
ЗАКРЫТЬ
ЗАМЕНИТЬ
ЗАМЕНЯЯ
ЗАМКОМ
ЗАП
ЗАПИСЕЙ
ЗАПИСИ
ЗАПИСЬ
ЗАПОЛНИТЕЛЬ
ЗАПРЕТИТЬ
ЗАПЯТАЯ
ЗАТЕМ
ЗНАК
ЗНАЧ
ЗНАЧЕНИЕ
ЗНАЧЕНИЕ-ОТЛАДКИ
ЗО
ЗС

И
ИДЕНТИФИКАЦИИ
ИЗ
ИЗМЕНИТЬ
ИКГ
ИКС
ИКЩ
ИЛИ
ИМЯ-ОТЛАДКИ
ИНАЧЕ
ИНДЕКСА
ИНДЕКСИРУЕТСЯ
ИНДЕКСНАЯ
ИНДЕКС-ОТЛАДКИ-1
ИНДЕКС-ОТЛАДКИ-2
ИНДЕКС-ОТЛАДКИ-3
ИНИЦИИРОВАТЬ
ИСКАТЬ
ИСПОЛЬЗОВАТЬ
ИСПОЛЬЗУЯ
ИСТИНА
ИСТОЧНИК
ИСХОДНАЯ-МАШИНА

К
КАВЫЧКА
КАВЫЧКИ
КАЖДОЕ
КАЖДЫЕ
КАЖДЫЙ
КАТУШКЕ
КАТУШКИ
КАТУШКУ
КЛАСС
КЛЮЧ
КЛЮЧА
КО
КОГДА
КОДОМ
КОММУНИКАЦИЙ
КОНЕЦ
КОНЕЦ-ВЕРНУТЬ
КОНЕЦ-ВЫЗВАТЬ
КОНЕЦ-ВЫПОЛНИТЬ
КОНЕЦ-ВЫЧИСЛИТЬ
КОНЕЦ-ЕСЛИ
КОНЕЦ-ИСКАТЬ
КОНЕЦ-ОБНОВИТЬ
КОНЕЦ-ОТНЯТЬ
КОНЕЦ-ОЦЕНИТЬ
КОНЕЦ-ПИСАТЬ
КОНЕЦ-ПОДВЕСТИ
КОНЕЦ-ПОЛУЧИТЬ
КОНЕЦ-РАЗДЕЛИТЬ
КОНЕЦ-РАЗОБРАТЬ
КОНЕЦ-СЛОЖИТЬ
КОНЕЦ-СОБРАТЬ
КОНЕЦ-УДАЛИТЬ
КОНЕЦ-УМНОЖИТЬ
КОНЕЦ-ЧИТАТЬ
КОНФИГУРАЦИИ
КОНЦА
КОНЦЕ
КОНЦОВКА
КОНЦУ
КОПИРОВАТЬ
КС

ЛИТЕР
ЛИТЕРА

GOST EXPERT ГОСТ 22558-89,

ЛИТЕРЫ
ЛОЖЬ
ЛЮБОЕ

МЕНЬШЕ
МЕНЯЯ
МЕТКИ
МОДУЛЕЙ

НА
НАЗНАЧИТЬ
НАИБОЛЬШЕЕ-ЗНАЧЕНИЕ
НАИБОЛЬШИЕ-ЗНАЧЕНИЯ
НАИМЕНЬШЕЕ-ЗНАЧЕНИЕ
НАИМЕНЬШИЕ-ЗНАЧЕНИЯ
НАЧАЛЕ
НАЧАЛЬНАЯ
НАЧАЛЬНОГО
НАЧАТЬ
НЕ
НЕОБЯЗАТЕЛЬНОГО
НЕТ
НИЖНЕЕ
НОМЕР
НУЛИ
НУЛЬ

ОБЛАСТЕЙ
ОБЛАСТЬ
ОБНОВИТЬ
ОБОРУДОВАНИЯ
ОБЩАЯ
ОГРАНИЧИВАЯСЬ
ОГРАНИЧИТЕЛЬ
ОДНОЙ
ОК
ОКРУГЛЯЯ
ОО
ОПРЕДЕЛЯЕТ
ОПУЩЕНЫ
ОРГАНИЗАЦИЯ
ОС
ОСВОБОДИТЬ
ОСОБО
ОСТАНОВИТЬ
ОСТАТОК
ОТ
ОТДЕЛЬНО
ОТКЛЮЧИТЬ
ОТКРЫТЬ
ОТЛАДКИ
ОТНОСИТЕЛЬНАЯ
ОТНОСИТЕЛЬНЫЙ
ОТНЯТЬ
ОТРИЦАТЕЛЬНО
ОТЧЕТ
ОТЧЕТА
ОТЧЕТОВ
ОТЧЕТЫ
ОФ
ОЦЕНИТЬ
ОЧЕРЕДЬ
ОЧИСТИТЬ
ОШИБКЕ
ОШИБКИ

ПАМЯТИ
ПЕРВЫЙ
ПЕРЕДАТЬ
ПЕРЕИМЕНОВЫВАЕТ
ПЕРЕЙТИ
ПЕРЕМЕННОЕ
ПЕРЕМОТКИ
ПЕРЕОПРЕДЕЛЯЕТ
ПЕРЕПОЛНЕНИИ
ПЕРЕПОЛНЕНИЯ
ПЕРЕПРОГОН
ПЕРЕХОДА
ПЕЧАТЬ
ПИСАТЬ
ПЛЮС
ПО
ПОВТОРЯЕТСЯ
ПОДАВИТЬ
ПОДВЕСТИ
ПОДОЧЕРЕДЬ-1
ПОДОЧЕРЕДЬ-2
ПОДОЧЕРЕДЬ-3
ПОЗИЦИЯ
ПОЛЕ
ПОЛНОМОЧИЯ
ПОЛОЖИТЕЛЬНО

GOST EXPERT ГОСТ 22558-89,

ПОЛУЧАЯ
ПОЛУЧИТЬ
ПОЛЯ
ПОМЕСТИТЬ
ПОСЛАТЬ
ПОСЛЕ
ПОСЛЕДНИЙ
ПОСЛЕДОВАТЕЛЬНАЯ
ПОСЛЕДОВАТЕЛЬНЫЙ
ПРЕВРАЩАЯ
ПРЕДЛОЖЕНИЕ
ПРЕДПРИЯТИЕ
ПРИ
ПРИБАВЛЯЯ
ПРИНЯТЬ
ПРОБЕЛ
ПРОБЕЛЫ
ПРОВЕРКОЙ
ПРОГРАММА
ПРОГРАММНЫЙ
ПРОГРАММЫ
ПРОДВИЖЕНИЯ
ПРОДОЛЖИТЬ
ПРОИЗВОЛЬНЫЙ
ПРОПИСНЫЕ
ПРОСМОТРЕТЬ
ПРОЦЕДУР
ПРОЦЕДУРА
ПРОЦЕДУРАХ
ПРОЦЕДУРЫ

РАБОТУ
РАБОЧАЯ-МАШИНА
РАБОЧЕЙ-ПАМЯТИ
РАВЕН
РАВНО
РАЗ
РАЗА
РАЗДЕЛ
РАЗДЕЛИТЬ
РАЗМЕР
РАЗМЕРОМ
РАЗОБРАТЬ
РАЗРЕШИТЬ
РЕВЕРСНО
РЕЖИМЕ
РЕЗЕРВИРОВАТЬ

С
СБРОСИТЬ
СВЯЗИ
СДВИНУТО
СЕГМЕНТ
СЕГМЕНТОВ
СЕКЦИЯ
СИМВОЛИЧЕСКАЯ
СИМВОЛИЧЕСКИЙ
СЛЕДУЮЩАЯ
СЛЕДУЮЩЕЕ
СЛЕДУЮЩЕЙ
СЛЕДУЮЩУЮ
СЛИТЬ
СЛОВ
СЛОЖИТЬ
СОБРАТЬ
СООБЩЕНИЕ
СООБЩЕНИЙ
СООБЩЕНИЯ
СООТВ
СООТВЕТСТВЕННО
СОРТ
СОРТИРОВАТЬ
СОРТИРОВКИ
СОРТИРОВКИ-СЛИЯНИЯ
СОСТОЯНИЕ
СОСТОЯНИЯ
СПЕЦИАЛЬНЫЕ-ИМЕНА
ССЫЛКАХ
ССЫЛКУ
СТАНДАРТ-А
СТАНДАРТ-Р
СТАНДАРТ-М
СТАНДАРТНОЙ
СТАНДАРТНЫ
СТОЛБЦА
СТРАНИЦЕ
СТРАНИЦЫ
СТРОК
СТРОКА-ОТЛАДКИ
СТРОКИ
СТРОКУ
СТРОЧНЫЕ

GOST EXPERT ГОСТ 22558-89,

СУММА
СЧЕТ
СЧЕТЧИК-ВЕРСТКИ
СЧЕТЧИК-СТРАНИЦ
СЧЕТЧИК-СТРОК
СЧИТАЯ

ТАБЛИЦА
ТАКЖЕ
ТЕКСТА
ТЕРМИНАЛ
ТЕРМИНАЛА
ТИП
ТОМ
ТОМА
ТОЧКА

УБЫВАНИЮ
УДАЛЕНИЕМ
УДАЛИТЬ
УЗ
УК
УКАЗАТЕЛЬ
УМНОЖИТЬ
УПРАВЛЕНИЕ
УПРАВЛЕНИЕ-ВВОДОМ-ВЫВОДОМ
УПРАВЛЕНИЕ-ФАЙЛАМИ
УПРАВЛЯЕМАЯ
УПРАВЛЯЕМЫЙ
УСТАНОВИТЬ

ФАЙЛА
ФАЙЛОВ
ФР
ФРАГМЕНТ

ЧИСЛО
ЧИСЛОВОЕ
ЧИТАТЬ
ЧР

Ш
ШАБЛОН

> < + − * / ** = <= >=

Часть 5. ФОРМАТЫ ЯЗЫКА

1. СВОДКА ФОРМАТОВ АНГЛИЙСКОЙ НОТАЦИИ

1.1. **Общий формат раздела идентификации**

IDENTIFICATION DIVISION.

PROGRAM-ID. имя-программы [IS { COMMON / INITIAL } PROGRAM].

[AUTHOR. [статья-комментарий] . . .]
[INSTALLATION. [статья-комментарий] . . .]
[DATE-WRITTEN. [статья-комментарий] . . .]
[DATE-COMPILED. [статья-комментарий] . . .]
[SECURITY. [статья-комментарий] . . .]

GOST EXPERT ГОСТ 22558-89,

1.2. Общий формат раздела оборудования

```
[ENVIRONMENT DIVISION.
[CONFIGURATION SECTION.
[SOURCE-COMPUTER.  [имя-машины [WITH DEBUGGING
        MODE]. ]]
[OBJECT-COMPUTER. [имя-машины
   [MEMORY SIZE целое-1 { WORDS
                        { CHARACTERS
                        { MODULES
   [PROGRAM COLLATING SEQUENCE IS имя-алфавита-1]
   [SEGMENT-LIMIT IS номер-сегмента]. ]]
[SPECIAL-NAMES. [[имя-реализации-1
{ IS мнемоническое-имя-1 [ON STATUS IS имя-условия-1
                         [OFF STATUS IS имя-условия-2]]
  IS мнемоническое-имя-2 [OFF STATUS IS имя-условия-2
                         [ON STATUS IS имя-условия-1]]
  ON STATUS IS имя-условия-1 [OFF STATUS IS
                         имя-условия-2]
  OFF STATUS IS имя-условия-2
  [ON STATUS IS имя-условия-1]                          } ]...
  ALPHABET имя-алфавита-1 IS
{ STANDARD-1
  STANDARD-2
  STANDARD-R
  NATIVE
  имя-реализации-2
  { литерал-1 [ { THROUGH } литерал-2 ]                 } ...
              [ { THRU    }           ]
              [ {ALSO литерал-3} ... ]                  } ]...
  [SYMBOLIC CHARACTERS {{(символическая-литера-1} ...
    { IS  } {целое-1} ... ) ... [IN имя-алфавита-2]}] ...
    { ARE }
  [CLASS имя-класса-1 IS
    {литерал-4 [ { THROUGH } литерал-5 ]} ... ] ...
```


GOST EXPERT ГОСТ 22558-89,

```
[CURRENCY SIGN IS литерал-6]
[DECIMAL-POINT IS COMMA]. ]]]
[INPUT-OUTPUT SECTION.
FILE-CONTROL.
  {статья-управления-файлом} . . .
[I-O-CONTROL.
  [[RERUN[ON {имя-файла-1 }]]
             {имя-реализации-1}
       EVERY{ {[END OF]{REEL }}                }OF имя-файла-2 }] . . .
             {целое-1  {UNIT }}                {               }
             {целое-2 CLOCK-UNITS              {               }
             {имя-условия-1                    {               }
  [ SAME[ RECORD ]AREA FOR имя-файла-3
        [ SORT   ]
        [ SORT-MERGE ]
      {имя-файла-4} . . . ] . . .
[MULTIPLE FILE TAPE CONTAINS {имя-файла-5
  [POSITION целое-3]} . . . ] . . . ]]]
```

1.3. **Общий формат статьи управления файлом**

1.3.1. Последовательный файл

```
SELECT [OPTIONAL] имя-файла-1
     ASSIGN TO {имя-реализации-1 } . . .
               {литерал-1        }
[RESERVE целое-1 [AREA ]]
                 [AREAS]
[[ORGANIZATION IS] SEQUENTIAL]
[PADDING CHARACTER IS {имя-данного-1}]
                      {литерал-2    }
[ RECORD DELIMITER IS {STANDARD-1   }]
                      {имя-реализации-2}
[ACCESS MODE IS SEQUENTIAL]
[FILE STATUS IS имя-данного-2].
```


GOST EXPERT ГОСТ 22558-89,

1.3.2. Относительный файл

SELECT [OPTIONAL] имя-файла-1

ASSIGN TO {имя-реализации-1 / литерал-1} ...

[RESERVE целое-1 [AREA / AREAS]]

[ORGANIZATION IS] RELATIVE

[ACCESS MODE IS {SEQUENTIAL [RELATIVE KEY IS имя-данного-1] / {RANDOM / DYNAMIC} RELATIVE KEY IS имя-данного-1}]

[FILE STATUS IS имя-данного-2].

1.3.3. Индексный файл

SELECT [OPTIONAL] имя-файла-1

ASSIGN TO {имя-реализации-1 / литерал-1} ...

[RESERVE целое-1 [AREA / AREAS]]

[ORGANIZATION IS] INDEXED

[ACCESS MODE IS [{SEQUENTIAL / RANDOM / DYNAMIC}]]

RECORD KEY IS имя-данного-1

[ALTERNATE RECORD KEY IS имя-данного-2 [WITH DUPLICATES]]...

[FILE STATUS IS имя-данного-3].

1.3.4. Сортируемый-сливаемый файл

SELECT имя-файла-1 ASSIGN TO {имя-реализации-1 / литерал-1} ... .

1.3.5. Файл отчетов

SELECT [OPTIONAL] имя-файла-1

ASSIGN TO {имя-реализации-1 / литерал-1} ...

[RESERVE целое-1 [AREA / AREAS]]

[[ORGANIZATION IS] SEQUENTIAL]

[PADDING CHARACTER {имя-данного-1 / литерал-2}]

GOST EXPERT ГОСТ 22558-89,

[RECORD DELIMITER IS {STANDARD-1 / имя-реализации-2}]
[ACCESS MODE IS SEQUENTIAL]
[FILE STATUS IS имя-данного-2].

1.4. Общий формат раздела данных

[DATA DIVISION.
[FILE SECTION.

[статья-описания-файла {статья-описания-записи}...
статья-описания-сортируемого-сливаемого-файла {статья-описания-записи}...
статья-описания-файла-отчетов] ...]

[WORKING-STORAGE SECTION.
[статья-описания-уровня-77 / статья-описания-записи] ...]
[LINKAGE SECTION.
[статья-описания-уровня-77 / статья-описания-записи] ...]
[COMMUNICATION SECTION.
[статья-описания-коммуникации [статья-описания-записи] ...] ...]
[REPORT SECTION.
[статья-описания-отчета {статья-описания-группы-отчета}...]...]]

1.5. Общий формат статьи описания файла

1.5.1. Последовательный файл

FD имя-файла-1
[IS EXTERNAL]
[IS GLOBAL]
[BLOCK CONTAINS [целое-1 TO] целое-2 {RECORDS / CHARACTERS}]
[RECORD {CONTAINS целое-3 CHARACTERS
IS VARYING IN SIZE [[FROM целое-4] [TO целое-5] CHARACTERS] [DEPENDING ON имя-данного-1]
CONTAINS целое-6 TO целое-7 CHARACTERS}]

GOST EXPERT ГОСТ 22558-89,

```
[ LABEL { RECORD IS  } { STANDARD } ]
        { RECORDS ARE} { OMITTED  }
[VALUE OF {имя-реализации-1 IS {имя-данного-2}} ...]
          {              {литерал-1  }}
[ DATA { RECORD IS  } {имя-данного-3} ... ]
       { RECORDS ARE}
[ LINAGE IS {имя-данного-4} LINES
            {целое-8      }
    [WITH FOOTING AT {имя-данного-5}]
                     {целое-9      }
    [LINES AT TOP    {имя-данного-6}]
                     {целое-10     }
    [LINES AT BOTTOM {имя-данного-7}]]
             {целое-11     }
[CODE-SET IS имя-алфавита-1].
```

1.5.2. Относительный файл

```
FD имя-файла-1
   [IS EXTERNAL]
   [IS GLOBAL]
   [ BLOCK CONTAINS [целое-1 TO] целое-2 { RECORDS    } ]
                                         { CHARACTERS }
   [ RECORD { CONTAINS целое-3 CHARACTERS                                  } ]
            { IS VARYING IN SIZE [[FROM целое-4]                           }
            {  [TO целое-5]              CHARACTERS]                       }
            {       [DEPENDING ON имя-данного-1]                           }
            { CONTAINS целое-6 TO целое-7                                  }
            {                     CHARACTERS                               }
   [ LABEL { RECORD IS  } { STANDARD } ]
           { RECORDS ARE} { OMITTED  }
   [VALUE OF {имя-реализации-1 IS {имя-данного-2}} ...]
             {                    {литерал-1  }}
   [ DATA { RECORD IS  } {имя-данного-3} ... ] .
          { RECORDS ARE}
```

1.5.3. Индексный файл

```
FD имя-файла-1
   [IS EXTERNAL]
   [IS GLOBAL]
```


GOST EXPERT ГОСТ 22558-89,

```
[BLOCK CONTAINS [целое-1 TO] целое-2 {RECORDS / CHARACTERS}]
[RECORD {CONTAINS целое-3 CHARACTERS
         IS VARYING IN SIZE [[FROM целое-4]
                             [TO целое-5] CHARACTERS]
                            [DEPENDING ON имя-данного-1]
         CONTAINS целое-6 TO целое-7 CHARACTERS}]
[LABEL {RECORD IS / RECORDS ARE} {STANDARD / OMITTED}]
[VALUE OF {имя-реализации-1 IS {имя-данного-2 / литерал-1}} ...]
[DATA {RECORD IS / RECORDS ARE} {имя-данного-3} ...].
```

1.5.4. Сортируемый-сливаемый файл

```
SD имя-файла-1
[RECORD {CONTAINS целое-1 CHARACTERS
         IS VARYING IN SIZE [[FROM целое-2]
                            [TO целое-3] CHARACTERS]
                           [DEPENDING ON имя-данного-1]
         CONTAINS целое-4 TO целое-5
                           CHARACTERS}]
[DATA {RECORD IS / RECORDS ARE} {имя-данного-2} ...].
```

1.5.5. Файл отчетов

```
FD имя-файла-1
  [IS EXTERNAL]
  [IS GLOBAL]
  [BLOCK CONTAINS [целое-1 TO] целое-2 {RECORDS / CHARACTERS}]
  [RECORD {CONTAINS целое-3 CHARACTERS
           CONTAINS целое-4 TO целое-5 CHARACTERS}]
  [LABEL {RECORD IS / RECORDS ARE} {STANDARD / OMITTED}]
  [VALUE OF {имя-реализации-1 IS {имя-данного-2 / литерал-1}} ...]
  [CODE-SET IS имя-алфавита-1]
```


GOST EXPERT ГОСТ 22558-89,

```
{REPORT IS  }
{REPORTS ARE} {имя-отчета-1} . . . .
```

1.6. Общий формат статьи описания данного

Формат 1

```
номер-уровня [имя-данного-1]
             [FILLER      ]
  [REDEFINES имя-данного-2]
  [IS EXTERNAL]
  [IS GLOBAL]
  [{PICTURE} IS строка-литер ]
  [{PIC    }                 ]
  [                                                    ]
  [  [USAGE IS] {BINARY                }               ]
  [             {COMPUTATIONAL         }               ]
  [             {COMP                  }               ]
  [             {DISPLAY               }               ]
  [             {INDEX                 }               ]
  [             {PACKED-DECIMAL        }               ]
  [                                                    ]
  [ [SIGN IS] {LEADING } [SEPARATE CHARACTER] ]
  [           {TRAILING}                      ]
  [ OCCURS целое-2 TIMES                                                  ]
  [    [{ASCENDING } KEY IS {имя-данного-3} . . . ] . . .                 ]
  [    [{DESCENDING}                              ]                       ]
  [  [INDEXED BY {имя-индекса-1} . . . ]                                  ]
  [ OCCURS целое-1 TO целое-2 TIMES DEPENDING ON                          ]
  [          имя-данного-4                                                ]
  [    [{ASCENDING } KEY IS {имя-данного-3} . . . ] . . .                 ]
  [    [{DESCENDING}                              ]                       ]
  [  [INDEXED BY {имя-индекса-1} . . . ]                                  ]
  [{SYNCHRONIZED} [LEFT ]]
  [{SYNC        } [RIGHT]]
  [{JUSTIFIED} RIGHT]
  [{JUST     }       ]
  [BLANK WHEN ZERO]
  [VALUE IS литерал-1].
```


GOST EXPERT ГОСТ 22558-89,

Формат 2

66 имя-данного-1 RENAMES имя-данного-2

[{THROUGH / THRU} имя-данного-3].

Формат 3

88 имя-условия-1

{VALUE IS / VALUES ARE} {литерал-1 [{THROUGH / THRU} литерал-2]} ...

1.7. Общий формат статьи описания коммуникации

Формат 1

CD имя-коммуникации-1

FOR [INITIAL] INPUT

[[SYMBOLIC QUEUE IS имя-данного-1]
[SYMBOLIC SUB-QUEUE-1 IS имя-данного-2]
[SYMBOLIC SUB-QUEUE-2 IS имя-данного-3]
[SYMBOLIC SUB-QUEUE-3 IS имя-данного-4]
[MESSAGE DATE IS имя-данного-5]
[MESSAGE TIME IS имя-данного-6]
[SYMBOLIC SOURCE IS имя-данного-7]
[TEXT LENGTH IS имя-данного-8]
[END KEY IS имя-данного-9]
[STATUS KEY IS имя-данного-10]
[MESSAGE COUNT IS имя-данного-11]]
[имя-данного-1, имя-данного-2, имя-данного-3,
имя-данного-4, имя-данного-5, имя-данного-6,
имя-данного-7, имя-данного-8, имя-данного-9,
имя-данного-10, имя-данного-11]

Формат 2

CD имя-коммуникации-1 FOR OUTPUT

[DESTINATION COUNT IS имя-данного-1]
[TEXT LENGTH IS имя-данного-2]
[STATUS KEY IS имя-данного-3]
[DESTINATION TABLE OCCURS целое-1 TIMES [INDEXED
BY {имя-индекса-1} ...]]
[ERROR KEY IS имя-данного-4]
[SYMBOLIC DESTINATION IS имя-данного-5].

GOST EXPERT ГОСТ 22558-89,

Формат 3

CD имя-коммуникации-1 FOR [INITIAL] I-O

[[MESSAGE DATE IS имя-данного-1]
[MESSAGE TIME IS имя-данного-2]
[SYMBOLIC TERMINAL IS имя-данного-3]
[TEXT LENGTH IS имя-данного-4]
[END KEY IS имя-данного-5]
[STATUS KEY IS имя-данного-6]]
[имя-данного-1, имя-данного-2, имя-данного-3,
имя-данного-4, имя-данного-5, имя-данного-6]].

1.8. **Общий формат статьи описания отчета**

RD имя-отчета-1

[IS GLOBAL]

[CODE литерал-1]

[{ CONTROL IS / CONTROLS ARE } { {имя-данного-1} ... / FINAL [имя-данного-1] ... }]

[PAGE [LIMIT IS / LIMITS ARE] целое-1 [LINE / LINES] [HEADING целое-2]

[FIRST DETAIL целое-3] [LAST DETAIL целое-4]

[FOOTING целое-5]].

1.9. **Общий формат статьи описания группы отчета**

Формат 1

01 [имя-данного-1]

[LINE NUMBER IS { целое-1 [ON NEXT PAGE] / PLUS целое-2 }]

[NEXT GROUP IS { целое-3 / PLUS целое-4 / NEXT PAGE }]

GOST EXPERT ГОСТ 22558-89,

```
                  { { REPORT HEADING } }
                  { { RH             } }
                  { { PAGE HEADING   } }
                  { { PH             } }
                  { { CONTROL HEADING } { имя-данного-2 } }
                  { { CH              } { FINAL         } }
TYPE IS           { { DETAIL         } }
                  { { DE             } }
                  { { CONTROL FOOTING } { имя-данного-3 } }
                  { { CF              } { FINAL         } }
                  { { PAGE FOOTING   } }
                  { { PF             } }
                  { { REPORT FOOTING } }
                  { { RF             } }
  [[USAGE IS] DISPLAY].
```

Формат 2

```
номер-уровня [имя-данного-1]
  [ LINE NUMBER IS { целое-1 [ON NEXT PAGE] } ]
                   { PLUS целое-2           }
   [[USAGE IS] DISPLAY].
```

Формат 3

```
номер-уровня [имя-данного-1]
  { PICTURE } IS строка-литер
  { PIC     }
  [[USAGE IS] DISPLAY]
  [ [SIGN IS] { LEADING  } [SEPARATE CHARACTER] ]
              { TRAILING }
  [ { JUSTIFIED } RIGHT ]
  { JUST      }
  [BLANK WHEN ZERO]
  [ LINE NUMBER IS { целое-1 [ON NEXT PAGE] } ]
                   { PLUS целое-2           }
  [COLUMN NUMBER IS целое-3]
```


GOST EXPERT ГОСТ 22558-89,

```
{ SOURCE IS идентификатор-1
  VALUE IS литерал-1
  [SUM {идентификатор-2} ... [UPON
       {имя-данного-2}...]]...
       [RESET ON { имя-данного-3 }]
                 { FINAL         }
[GROUP INDICATE].
```

1.10. **Общий формат раздела процедур**

Формат 1

```
[PROCEDURE DIVISIONS [USING {имя-данного-1} ...].
[DECLARATIVES.
{имя-секции SECTION [номер-сегмента].
    оператор USE
[имя-параграфа.
  [предложение] ... ] ... } ...
END DECLARATIVES.]
{имя-секции SECTION [номер-сегмента].
[имя-параграфа.
  [предложение]...]...}...]
```

Формат 2

```
[PROCEDURE DIVISION [USING {имя-данного-1}...].
{имя-параграфа.
  [предложение] ... } ... ]
```

1.11. **Общий формат глаголов Кобола**

1.11.1

```
ACCEPT идентификатор-1 [FROM мнемоническое-имя-1]

                       { DATE        }
ACCEPT идентификатор-2 FROM { DAY         }
                       { DAY-OF-WEEK }
                       { TIME        }

ACCEPT имя-коммуникации-1 MESSAGE COUNT
```

1.11.2

```
ADD {идентификатор-1} ... TO {идентификатор-2
    {литерал-1      }          [ROUNDED]} ...
  [ON SIZE ERROR повелительный-оператор-1]
```


GOST EXPERT ГОСТ 22558-89,

[NOT ON SIZE ERROR повелительный-оператор-2]
[END-ADD]

ADD {идентификатор-1 / литерал-1} ... TO {идентификатор-2 / литерал-2}
GIVING {идентификатор-3 [ROUNDED]} ...
[ON SIZE ERROR повелительный-оператор-1]
[NOT ON SIZE ERROR повелительный-оператор-2]
[END-ADD]

ADD {CORRESPONDING / CORR} идентификатор-1 TO идентификатор-2
[ROUNDED]
ON SIZE ERROR повелительный-оператор-1]
[NOT ON SIZE ERROR повелительный-оператор-2]
[END-ADD]

1.11.3

ALTER {имя-процедуры-1 TO [PROCEED TO] имя-процедуры-2} ...

1.11.4

CALL {идентификатор-1 / литерал-1}
[USING {[BY REFERENCE] {идентификатор-2}... / BY CONTENT {идентификатор-2}...} ...]
[ON OVERFLOW повелительный-оператор-1] [END-CALL]

CALL {идентификатор-1 / литерал-1}
[USING {[BY REFERENCE] {идентификатор-2}... / BY CONTENT {идентификатор-2}...} ...]
[ON EXCEPTION повелительный-оператор-1]
[NOT ON EXCEPTION повелительный-оператор-2]
[END-CALL]

1.11.5

CANCEL {идентификатор-1 / литерал-1} ...

GOST EXPERT ГОСТ 22558-89,

1.11.6

CLOSE { имя-файла-1 [{ REEL / UNIT } [FOR REMOVAL] | WITH { NO REWIND / LOCK }] } ...

CLOSE {имя-файла-1 [WITH LOCK]} ...

1.11.7

COMPUTE {идентификатор-1 [ROUNDED]}... =
арифметическое-выражение-1
[ON SIZE ERROR повелительный-оператор-1]
[NOT ON SIZE ERROR повелительный-оператор-2]
[END-COMPUTE]

1.11.8

CONTINUE

1.11.9

DELETE имя-файла-1 RECORD
[INVALID KEY повелительный-оператор-1]
[NOT INVALID KEY повелительный-оператор-2]
[END-DELETE]

1.11.10

DISABLE { INPUT [TERMINAL] / I-O TERMINAL / OUTPUT } имя-коммуникации-1
[WITH KEY { идентификатор-1 / литерал-1 }]

1.11.11

DISPLAY { идентификатор-1 / литерал-1 } ... [UPON мнемоническое-имя-1]
[WITH NO ADVANCING]

1.11.12

DIVIDE { идентификатор-1 / литерал-1 } INTO {идентификатор-2
[ROUNDED]} ...
[ON SIZE ERROR повелительный-оператор-1]
[NOT ON SIZE ERROR повелительный-оператор-2]
[END-DIVIDE]

GOST EXPERT ГОСТ 22558-89,

DIVIDE {идентификатор-1 / литерал-1} INTO {идентификатор-2 / литерал-2}
GIVING {идентификатор-3 [ROUNDED]} ...
[ON SIZE ERROR повелительный-оператор-1]
[NOT ON SIZE ERROR повелительный-оператор-2]
[END-DIVIDE]

DIVIDE {идентификатор-1 / литерал-1} BY {идентификатор-2 / литерал-2}
GIVING {идентификатор-3 [ROUNDED]} ...
[ON SIZE ERROR повелительный-оператор-1]
[NOT ON SIZE ERROR повелительный-оператор-2]
[END-DIVIDE]

DIVIDE {идентификатор-1 / литерал-1} INTO {идентификатор-2 / литерал-2}
GIVING идентификатор-3 [ROUNDED]
REMAINDER идентификатор-4
[ON SIZE ERROR повелительный-оператор-1]
[NOT ON SIZE ERROR повелительный-оператор-2]
[END-DIVIDE]

DIVIDE {идентификатор-1 / литерал-1} BY {идентификатор-2 / литерал-2}
GIVING идентификатор-3 [ROUNDED]
REMAINDER идентификатор-4
[ON SIZE ERROR повелительный-оператор-1]
[NOT ON SIZE ERROR повелительный-оператор-2]
[END-DIVIDE]

1.11.13

ENABLE {INPUT [TERMINAL] / I-O TERMINAL / OUTPUT} имя-коммуникации-1
[WITH KEY {идентификатор-1 / литерал-1}]

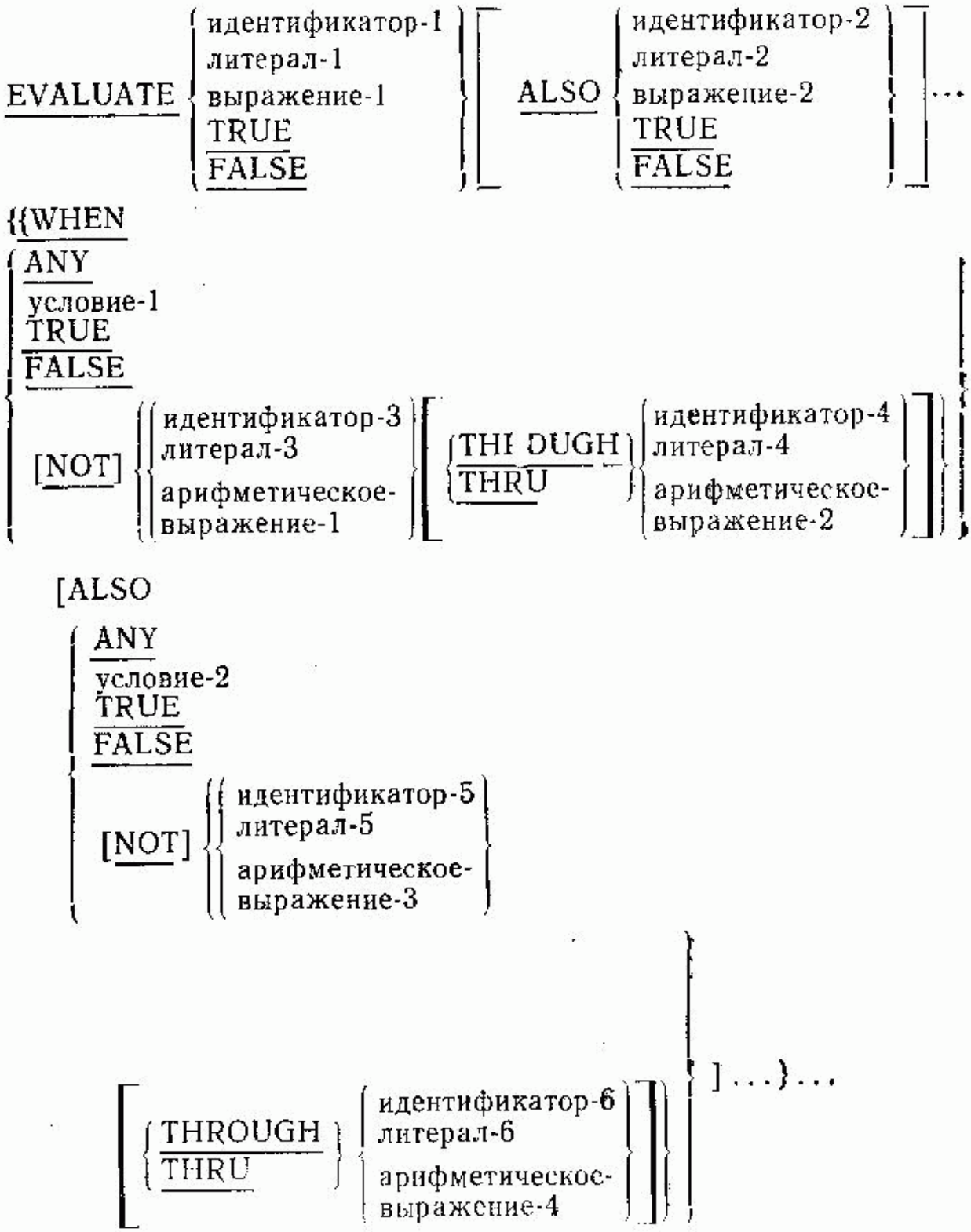
7 Зак. 251

GOST EXPERT ГОСТ 22558-89,

1.11.14

ENTER имя-языка-1 [имя-программного-модуля-1]

1.11.15



GOST EXPERT ГОСТ 22558-89,

повелительный-оператор-1} ...
[WHEN OTHER повелительный-оператор-2]
[END-EVALUATE]

1.11.16
EXIT
EXIT PROGRAM

1.11.17

GENERATE {имя-данного-1 / имя-отчета-1}

1.11.18
GO TO [имя-процедуры-1]
GO TO {имя-процедуры-1} ... DEPENDING ON
идентификатор-1

1.11.19

IF условие-1 THEN {{оператор-1} ... / NEXT SENTENCE}

{ELSE {оператор-2} ... [END-IF] / ELSE NEXT SENTENCE / END-IF}

1.11.20
INITIALIZE {идентификатор-1} ...

[REPLACING {{ALPHABETIC / ALPHANUMERIC / NUMERIC / ALPHANUMERIC-EDITED / NUMERIC-EDITED}

DATA BY {идентификатор-2 / литерал-1} ...]

1.11.21
INITIATE {имя-отчета-1} ...

GOST EXPERT ГОСТ 22558-89,

ГОСТ 22558-89,

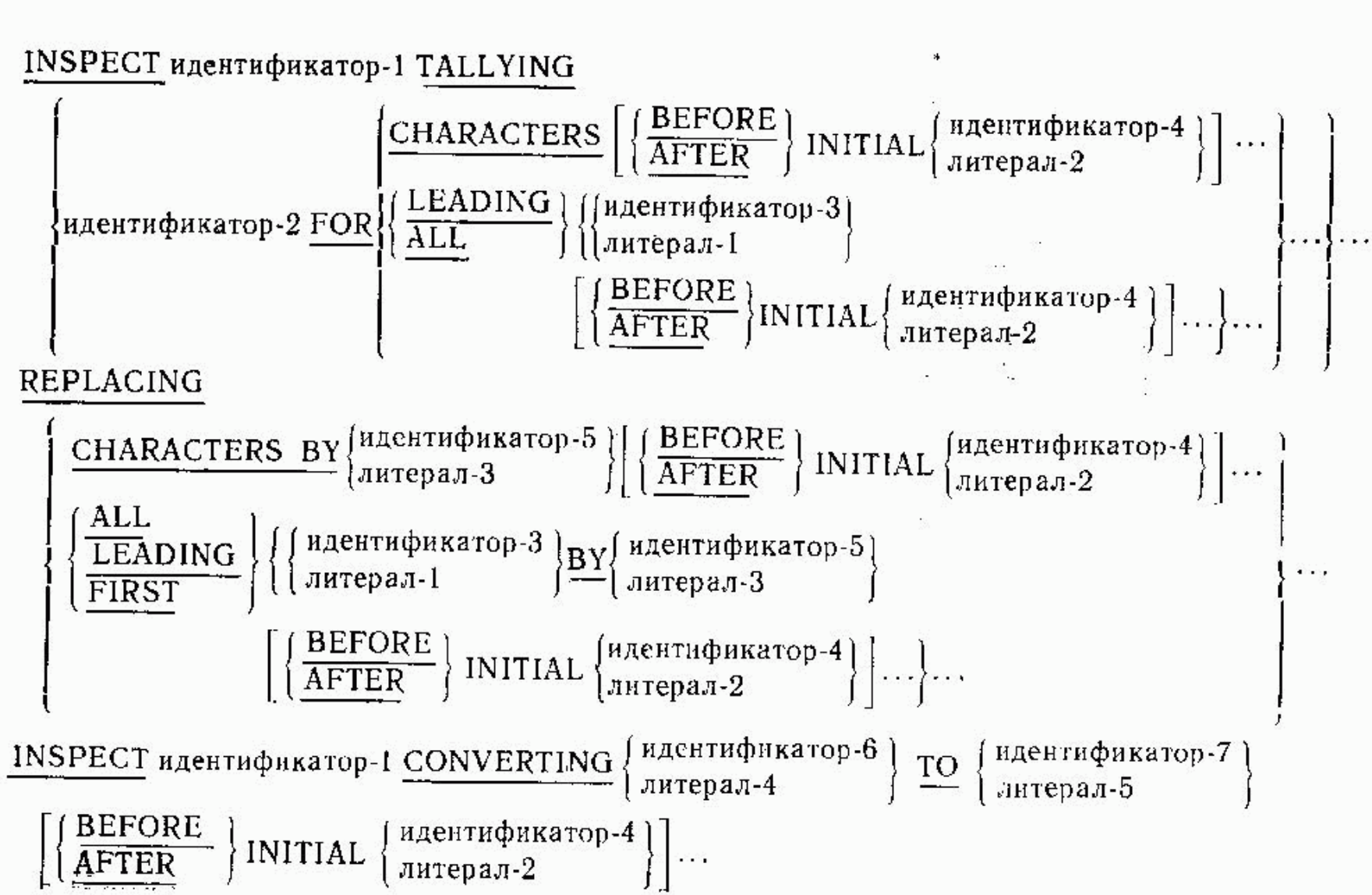
1.11.22

INSPECT идентификатор-1 TALLYING

{ идентификатор-2 FOR { { CHARACTERS [{ BEFORE / AFTER } INITIAL { идентификатор-4 / литерал-2 }] ... } ... / { { ALL / LEADING } { { идентификатор-3 / литерал-1 } [{ BEFORE / AFTER } INITIAL { идентификатор-4 / литерал-2 }] ... } ... } ... }

INSPECT идентификатор-1 REPLACING

{ CHARACTERS BY { идентификатор-5 / литерал-3 } [{ BEFORE / AFTER } INITIAL { идентификатор-4 / литерал-2 }] ... / { { ALL / LEADING / FIRST } { { идентификатор-3 / литерал-1 } BY { идентификатор-5 / литерал-3 } [{ BEFORE / AFTER } INITIAL { идентификатор-4 / литерал-2 }] ... } ... } ... }


INSPECT идентификатор-1 TALLYING
идентификатор-2 FOR
CHARACTERS
BEFORE
AFTER
INITIAL
идентификатор-4
литерал-2
ALL
LEADING
идентификатор-3
литерал-1
REPLACING
CHARACTERS BY
идентификатор-5
литерал-3
FIRST
BY
INSPECT идентификатор-1 CONVERTING
идентификатор-6
литерал-4
TO
идентификатор-7
литерал-5

1.11.23

MERGE имя-файла-1 [ON {ASCENDING / DESCENDING} KEY {имя-данного-1} ...]...
[COLLATING SEQUENCE IS имя-алфавита-1]
USING имя-файла-2 {имя-файла-3} ...
{OUTPUT PROCEDURE IS имя-процедуры-1 [{THROUGH / THRU} имя-процедуры-2] / GIVING {имя-файла-4} ...}

1.11.24

MOVE {идентификатор-1 / литерал-1} TO {идентификатор-2} ...

MOVE {CORRESPONDING / CORR} идентификатор-1 TO идентификатор-2

1.11.25

MULTIPLY {идентификатор-1 / литерал-1} BY {идентификатор-2 [ROUNDED]} ...
[ON SIZE ERROR повелительный-оператор-1]
[NOT ON SIZE ERROR повелительный-оператор-2]
[END-MULTIPLY]

MULTIPLY {идентификатор-1 / литерал-1} BY {идентификатор-2 / литерал-2}
GIVING {идентификатор-3 [ROUNDED]} ...
[ON SIZE ERROR повелительный-оператор-1]
[NOT ON SIZE ERROR повелительный-оператор-2]
[END-MULTIPLY]

GOST EXPERT ГОСТ 22558-89,

1.11.26

OPEN { INPUT {имя-файла-1 [REVERSED / WITH NO REWIND]}... / OUTPUT {имя-файла-2 [WITH NO REWIND]}... / I-O {имя-файла-3}... / EXTEND {имя-файла-4}... }...

OPEN { OUTPUT {имя-файла-1 [WITH NO REWIND]}... / EXTEND {имя-файла-2}... }...

OPEN { INPUT {имя-файла-1}... / OUTPUT {имя-файла-2}... / I-O {имя-файла-3}... / EXTEND {имя-файла-4}... }...

1.11.27

PERFORM [имя-процедуры-1 [{THROUGH / THRU} имя-процедуры-2]]
[повелительный-оператор-1 END-PERFORM]

PERFORM [имя-процедуры-1 [{THROUGH / THRU} имя-процедуры-2]]
{идентификатор-1 / целое-1} TIMES [повелительный-оператор-1
END-PERFORM]

PERFORM [имя-процедуры-1 [{THROUGH / THRU} имя-процедуры-2]]
[WITH TEST {BEFORE / AFTER}] UNTIL условие-1
[повелительный-оператор-1 END-PERFORM]

PERFORM [[имя-процедуры-1 [{THROUGH / THRU} имя-процедуры-2]]
[WITH TEST {BEFORE / AFTER}]
VARYING {идентификатор-2 / имя-индекса-1} FROM {идентификатор-3 / имя-индекса-2 / литерал-1}

GOST EXPERT ГОСТ 22558-89,

BY { идентификатор-4 / литерал-2 } UNTIL условие-1

[AFTER { идентификатор-5 / имя-индекса-3 } FROM { идентификатор-6 / имя-индекса-4 / литерал-3 }

BY { идентификатор-7 / литерал-4 } UNTIL условие-2]...

[повелительный-оператор-1 END-PERFORM]

1.11.28

PURGE имя-коммуникации-1

1.11.29

READ имя-файла-1 [NEXT] RECORD [INTO идентификатор-1]
[AT END повелительный-оператор-1]
[NOT AT END повелительный-оператор-2]
[END-READ]

READ имя-файла-1 RECORD [INTO идентификатор-1]
[INVALID KEY повелительный-оператор-3]
[NOT INVALID KEY повелительный-оператор-4]
[END-READ]

READ имя-файла-1 RECORD [INTO идентификатор-1]
[KEY IS имя-данного-1]
[INVALID KEY повелительный-оператор-3]
[NOT INVALID KEY повелительный-оператор-4]
[END-READ]

1.11.30

RECEIVE имя-коммуникации-1 { MESSAGE / SEGMENT }
INTO идентификатор-1
[NO DATA повелительный-оператор-1]
[WITH DATA повелительный-оператор-2]
[END-RECEIVE]

1.11.31

RELEASE имя-записи-1 [FROM идентификатор-1]

GOST EXPERT ГОСТ 22558-89,

1.11.32

RETURN имя-файла-1 RECORD [INTO идентификатор-1]
[AT END повелительный-оператор-1]
[NOT AT END повелительный-оператор-2]
[END-RETURN]

1.11.33

REWRITE имя-записи-1
[FROM идентификатор-1]

REWRITE имя-записи-1 [FROM идентификатор-1]
[INVALID KEY повелительный-оператор-1]
[NOT INVALID KEY повелительный-оператор-2]
[END-REWRITE]

1.11.34

SEARCH идентификатор-1 [VARYING {идентификатор-2 / имя-индекса-1}]
[AT END повелительный-оператор-1]
{WHEN условие-1 {повелительный-оператор-2 / NEXT SENTENCE}}...
[END-SEARCH]

SEARCH ALL идентификатор-1
[AT END повелительный-оператор-1]
WHEN {имя-данного-1 {IS EQUAL TO / IS =} {идентификатор-3 / литерал-1 / арифметическое-выражение-1} / имя-условия-1}

GOST EXPERT ГОСТ 22558-89,

```
[ AND { имя-данного-2 { IS EQUAL TO } { идентификатор-4 }
      {               { IS =        } { литерал-2      } } ] ...
      {               {               { арифметическое-  } }
      { имя-условия-2                 { выражение-2      } }
```

{повелительный-оператор-2 | NEXT SENTENCE}

[END-SEARCH]

1.11.35

SEND имя-коммуникации-1 FROM идентификатор-1

SEND имя-коммуникации-1

[FROM идентификатор-1] {WITH идентификатор-2 | WITH ESI | WITH EMI | WITH EGI}

```
[ {BEFORE} {идентификатор-3} [LINE ]  ]
[ {AFTER } {целое-1        } [LINES]  ]
[ {BEFORE} {мнемоническое-имя-1}      ]
[ {AFTER } {PAGE               }      ]
```

ADVANCING

[REPLACING LINE]

1.11.36

SET {имя-индекса-1 | идентификатор-1} ... TO {имя-индекса-2 | идентификатор-2 | целое-1}

SET {имя-индекса-3} ... {UP BY | DOWN BY} {идентификатор-3 | целое-2}

SET {{мнемоническое-имя-1} ... TO {ON | OFF}} ...

SET {имя-условия-1} ... TO TRUE

GOST EXPERT ГОСТ 22558-89,

1.11.37

SORT имя-файла-1 { ON { ASCENDING / DESCENDING } KEY {имя-данного-1} ... }...

[WITH DUPLICATES IN ORDER]

[COLLATING SEQUENCE IS имя-алфавита-1]

{ INPUT PROCEDURE IS имя-процедуры-1 [{ THROUGH / THRU } имя-процедуры-2] / USING {имя-файла-2} ... }

{ OUTPUT PROCEDURE IS имя-процедуры-3 [{ THROUGH / THRU } имя процедуры-4] / GIVING {имя-файла-3} ... }

1.11.38

START имя-файла-1

[[KEY { IS EQUAL TO / IS = / IS GREATER THAN / IS > / IS NOT LESS THAN / IS NOT < / IS GREATER THAN OR EQUAL TO / IS >= } имя-данного-1]

[INVALID KEY повелительный-оператор-1]

[NOT INVALID KEY повелительный-оператор-2]

[END-START]

GOST EXPERT ГОСТ 22558-89,

1.11.39

STOP { RUN / литерал-1 }

1.11.40

STRING {{ идентификатор-1 / литерал-1 }...

DELIMITED BY { идентификатор-2 / литерал-2 / SIZE }}...

INTO идентификатор-3

[WITH POINTER идентификатор-4]

[ON OVERFLOW повелительный-оператор-1]

[NOT ON OVERFLOW повелительный-оператор-2]

[END-STRING]

1.11.41

SUBTRACT { идентификатор-1 / литерал-1 }... FROM {идентификатор-2

[ROUNDED]}...

[ON SIZE ERROR повелительный-оператор-1]

[NOT ON SIZE ERROR повелительный-оператор-2]

[END-SUBTRACT]

SUBTRACT { идентификатор-1 / литерал-1 }... FROM { идентификатор-2 / литерал-2 }

GIVING {идентификатор-3 [ROUNDED]}...

[ON SIZE ERROR повелительный-оператор-1]

[NOT ON SIZE ERROR повелительный-оператор-2]

[END-SUBTRACT]

SUBTRACT { CORRESPONDING / CORR } идентификатор-1

GOST EXPERT ГОСТ 22558-89,

FROM идентификатор-2 [ROUNDED]
[ON SIZE ERROR повелительный-оператор-1]
[NOT ON SIZE ERROR повелительный-оператор-2]
[END-SUBTRACT]

1.11.42
SUPPRESS PRINTING

1.11.43
TERMINATE {имя-отчета-1} . . .

1.11.44
UNSTRING идентификатор-1
[DELIMITED BY [ALL] { идентификатор-2 / литерал-1 }
[OR [ALL] { идентификатор-3 / литерал-2 }] . . .]
INTO {идентификатор-4 [DELIMITER IN идентификатор-5]
[COUNT IN идентификатор-6]} . . .
[WITH POINTER идентификатор-7]
[TALLYING IN идентификатор-8]
[ON OVERFLOW повелительный-оператор-1]
[NOT ON OVERFLOW повелительный-оператор-2]
[END-UNSTRING]

1.11.45
USE [CLOBAL] AFTER STANDARD { EXCEPTION / ERROR }
PROCEDURE
ON { {имя-файла-1} . . . / INPUT / OUTPUT / I-O / EXTEND }

GOST EXPERT ГОСТ 22558-89,

USE AFTER STANDARD { EXCEPTION / ERROR }

PROCEDURE ON { {имя-файла-1} ... / OUTPUT / EXTEND }

USE [GLOBAL] BEFORE REPORTING идентификатор-1

USE FOR DEBUGGING

ON { имя-коммуникации-1 / [ALL REFERENCES OF] идентификатор-1 / имя-файла-1 / имя-процедуры-1 / ALL PROCEDURES } ...

1.11.46

WRITE имя-записи-1 [FROM идентификатор-1]

[{ BEFORE / AFTER } ADVANCING { { идентификатор-2 / целое-1 } [LINE / LINES] / { мнемоническое-имя-1 / PAGE } }]

[AT { END-OF-PAGE / EOP } повелительный-оператор-1]

[NOT AT { END-OF-PAGE / EOP } повелительный-оператор-2]

[END-WRITE]

WRITE имя-записи-1 [FROM идентификатор-1]

[INVALID KEY повелительный-оператор-1]

[NOT INVALID KEY повелительный-оператор-2]

[END-WRITE]

GOST EXPERT ГОСТ 22558-89,

1.12. Общий формат операторов COPY (КОПИРОВАТЬ) и REPLACE (ЗАМЕНИТЬ)

1.12.1

COPY имя-текста-1 [{OF / IN} имя-библиотеки-1]

[REPLACING { ==псевдотекст-1== / идентификатор-1 / литерал-1 / слово-1 }

BY { ==псевдотекст-2== / идентификатор-2 / литерал-2 / слово-2 } ...]

REPLACE {==псевдотекст-1== BY ==псевдотекст-2==} ...

REPLACE OFF

1.13. Общий формат условий

1.13.1. Условие отношения

{ идентификатор-1 / литерал-1 / арифметическое-выражение-1 / имя-индекса-1 }

{ IS [NOT] GREATER THAN
IS [NOT] >
IS [NOT] LESS THAN
IS [NOT] <
IS [NOT EQUAL TO
IS [NOT] =
IS GREATER THAN OR EQUAL TO
IS >=
IS LESS THAN OR EQUAL TO
IS <= }

{ идентификатор-2 / литерал-2 / арифметическое-выражение-2 / имя-индекса-2 }

GOST EXPERT ГОСТ 22558-89,

1.13.2. Условие класса

идентификатор-1 IS [NOT] { NUMERIC / ALPHABETIC / ALPHABETIC-LOWER / ALPHABETIC-UPPER / имя-класса-1 }

1.13.3. Условие имени-условия

имя-условия-1

1.13.4. Условие состояния переключателя

имя-условия-1

1.13.5. Условие знака

арифметическое-выражение-1 IS [NOT] { POSITIVE / NEGATIVE / ZERO }

1.13.6. Отрицание условия

NOT условие-1

1.13.7. Комбинированное условие

условие-1 { { AND / OR } условие-2 } ...

1.13.8. Сокращенное комбинированное условие отношения

условие-отношения { { AND / OR } [NOT] [знак-операции-отношения] объект } ...

GOST EXPERT ГОСТ 22558-89,

1.14. Общий формат уточнения

Формат 1

{имя-данного-1 / имя-условия-1} { {{OF / IN} имя-данного-2} ... [{OF / IN} {имя-файла-1 / имя-коммуникации-1}] / {OF / IN} {имя-файла-1 / имя-коммуникации-1} }

Формат 2

имя-параграфа-1 {IN / OF} имя-секции-1

Формат 3

имя-текста-1 {IN / OF} имя-библиотеки-1

Формат 4

LINAGE-COUNTER {IN / OF} имя-файла-2

Формат 5

{PAGE-COUNTER / LINE-COUNTER} {IN / OF} имя-отчета-1

Формат 6

имя-данного-3 { {IN / OF} имя-данного-4 [{IN / OF} имя-отчета-2] / {IN / OF} имя-отсчета-2 }

1.15. Прочие форматы

1.15.1. Индексирование

{имя-данного-1 / имя-условия-1} ({ целое-1 / имя-данного-2 [{±} целое-2] / имя-индекса-1 [{±} целое-3] } ...)

GOST EXPERT ГОСТ 22558-89,

1.15.2. Модификация ссылки

имя-данного-1 (позиция-самой-левой-литеры: [длина])

1.15.3. Идентификатор

имя-данного-1 [{OF / IN} имя-данного-2] ... [{OF / IN} {имя-коммуникации-1 / имя-файла-1 / имя-отчета-1}]

[({индекс} ...)] [(позиция-самой-левой-литеры: [длина])]

1.16. Общий формат вложенных исходных программ

IDENTIFICATION DIVISION.

PROGRAM-ID. имя-программы-1 [IS INITIAL PROGRAM].

[ENVIRONMENT DIVISION. содержимое-раздела-оборудования]

[DATA DIVISION. содержимое-раздела-данных]

[PROCEDURE DIVISION. содержимое-раздела-процедур]

[[вложенная-исходная-программа] ...

END PROGRAM имя-программы-1.]

1.17. Общий формат вложенной исходной программы

IDENTIFICATION DIVISION.

PROGRAM-ID. имя-программы-2 [IS {COMMON / INITIAL}] PROGRAM].

[ENVIRONMENT DIVISION. содержимое-раздела-оборудования]

[DATA DIVISION. содержимое-раздела-данных]

[PROCEDURE DIVISION. содержимое-раздела-процедур]

[вложенная-исходная-программа] ...

END PROGRAM имя-программы-2.

1.18. Общий формат последовательности исходных программ

IDENTIFICATION DIVISION.

PROGRAM-ID. имя-программы-3 [IS INITIAL PROGRAM].

[ENVIRONMENT DIVISION. содержимое-раздела-оборудования]

GOST EXPERT ГОСТ 22558-89,

[DATA DIVISION. содержимое-раздела-данных]
[PROCEDURE DIVISION. содержимое-раздела-процедур]
{вложенная-исходная-программа]...
END PROGRAM имя-программы-3.}...
IDENTIFICATION DIVISION.
PROGRAM-ID. имя-программы-4 [IS INITIAL PROGRAM].
[ENVIRONMENT DIVISION. содержимое-раздела-
оборудования]
[DATA DIVISION. содержимое-раздела-данных]
[PROCEDURE DIVISION. содержимое-раздела-процедур]
[[вложенная-исходная-программа]...
END PROGRAM имя-программы-4.]

2. СВОДКА ФОРМАТОВ РУССКОЙ НОТАЦИИ

2.1. Общий формат раздела идентификации

РАЗДЕЛ ИДЕНТИФИКАЦИИ.
[ПРОГРАММА. имя-программы [{ОБЩАЯ | НАЧАЛЬНАЯ}].]
[АВТОР. [статья-комментарий]...]
[ПРЕДПРИЯТИЕ. [статья-комментарий]...]
[ДАТА-НАПИСАНИЯ. [статья-комментарий]...]
[ДАТА-ТРАНСЛЯЦИИ. [статья-комментарий]...]
[ПОЛНОМОЧИЯ. [статья-комментарий]...]

2.2. Общий формат раздела оборудования

[РАЗДЕЛ ОБОРУДОВАНИЯ.
[СЕКЦИЯ КОНФИГУРАЦИИ.
[ИСХОДНАЯ-МАШИНА. [имя-машины
[В РЕЖИМЕ ОТЛАДКИ].]]
[РАБОЧАЯ-МАШИНА. [имя-машины
[РАЗМЕР ПАМЯТИ целое-1 {СЛОВ | ЛИТЕР | МОДУЛЕЙ}]]
[ПРОГРАММНЫЙ АЛФАВИТ имя-алфавита]
[ГРАНИЦА СЕГМЕНТОВ номер-сегмента].]]

GOST EXPERT ГОСТ 22558-89,

[СПЕЦИАЛЬНЫЕ-ИМЕНА. [[имя-реализации-1

{ ЕСТЬ мнемоническое-имя-1 [{ ВКЛЮЧЕНО / ВКЛ } имя-условия-1 [{ ВЫКЛЮЧЕНО / ВЫКЛ } имя-условия-2]]

ЕСТЬ мнемоническое-имя-2 [{ ВЫКЛЮЧЕНО / ВЫКЛ } имя-условия-2 [{ ВКЛЮЧЕНО / ВКЛ } имя-условия-1]] }] ...

{ ВКЛЮЧЕНО / ВКЛ } имя-условия-1 [{ ВЫКЛЮЧЕНО / ВЫКЛ } имя-условия-2]

{ ВЫКЛЮЧЕНО / ВЫКЛ } имя-условия-2 [{ ВКЛЮЧЕНО / ВКЛ } имя-условия-1] }

[АЛФАВИТ имя-алфавита-1

{ СТАНДАРТ-А / СТАНДАРТ-М / СТАНДАРТ-Р / ВНУТРЕННИЙ / имя-реализации-2 }] ...

{ литерал-1 [ПО литерал-2 / {ТАКЖЕ литерал-3} ...] } ...

[СИМВОЛИЧЕСКАЯ ЛИТЕРА {{{символическая-литера-1}...
ЕСТЬ {целое-1} ... } ... [ИЗ имя-алфавита-2]}] ...

[КЛАСС имя-класса-1 ЕСТЬ {литерал-4 [ПО литерал-5]}...] ...

[ВАЛЮТНЫЙ ЗНАК литерал-6]

[ДЕСЯТИЧНАЯ ТОЧКА ЗАПЯТАЯ].]]]

GOST EXPERT ГОСТ 22558-89,

[СЕКЦИЯ ВВОДА-ВЫВОДА.
УПРАВЛЕНИЕ-ФАЙЛАМИ.
{статья-управления-файлом} ...
[УПРАВЛЕНИЕ-ВВОДОМ-ВЫВОДОМ.

[[ПЕРЕПРОГОН [НА { имя-файла-1 / имя-реализации-1 }]]

{ { КАЖДЫЙ КОНЕЦ { КАТУШКИ / ТОМА } } имя-файла-2
  { КАЖДЫЕ целое-1 ЗАПИСЕЙ }
  КАЖДЫЕ целое-2 ЕДИНИЦ-ВРЕМЕНИ
  КАЖДОЕ имя-условия-1 }] ...

[ОБЩАЯ ОБЛАСТЬ [ЗАПИСИ / СОРТИРОВКИ / СОРТИРОВКИ-СЛИЯНИЯ]

ДЛЯ имя-файла-3 {имя-файла-4}...] ...

[НА ОДНОЙ КАТУШКЕ {имя-файла-5 [ПОЗИЦИЯ целое-3]} ...] ... .]]]

2.3. **Общий формат статьи управления файлом**

2.3.1. Последовательный файл
ДЛЯ [НЕОБЯЗАТЕЛЬНОГО] имя-файла-1

НАЗНАЧИТЬ { имя-реализации-1 / литерал-1 } ...

[РЕЗЕРВИРОВАТЬ целое-1 ОБЛАСТЕЙ]

[[ОРГАНИЗАЦИЯ] ПОСЛЕДОВАТЕЛЬНАЯ]

[ЛИТЕРА ЗАПОЛНИТЕЛЬ { имя-данного-1 / литерал-2 }]

[ОГРАНИЧИТЕЛЬ ЗАПИСИ { СТАНДАРТ-А / имя-реализации-2 }]

[ДОСТУП ПОСЛЕДОВАТЕЛЬНЫЙ]

[СОСТОЯНИЕ ФАЙЛА имя-данного-2].

GOST EXPERT ГОСТ 22558-89,

2.3.2. Относительный файл

ДЛЯ [НЕОБЯЗАТЕЛЬНОГО] имя-файла-1

НАЗНАЧИТЬ { имя-реализации-1 / литерал-1 } ...

[РЕЗЕРВИРОВАТЬ целое-1 ОБЛАСТЕЙ]

[ОРГАНИЗАЦИЯ] ОТНОСИТЕЛЬНАЯ

[ДОСТУП { ПОСЛЕДОВАТЕЛЬНЫЙ [ОТНОСИТЕЛЬНЫЙ КЛЮЧ имя-данного-1] / { ПРОИЗВОЛЬНЫЙ / ДИНАМИЧЕСКИЙ } ОТНОСИТЕЛЬНЫЙ КЛЮЧ имя данного-1 }]

[СОСТОЯНИЕ ФАЙЛА имя-данного-2].

2.3.3. Индексный файл

ДЛЯ [НЕОБЯЗАТЕЛЬНОГО] имя-файла-1

НАЗНАЧИТЬ { имя-реализации-1 / литерал-1 } ...

[РЕЗЕРВИРОВАТЬ целое-1 ОБЛАСТЕЙ]

[ОРГАНИЗАЦИЯ] ИНДЕКСНАЯ

[ДОСТУП { ПОСЛЕДОВАТЕЛЬНЫЙ / ПРОИЗВОЛЬНЫЙ / ДИНАМИЧЕСКИЙ }]

КЛЮЧ ЗАПИСИ имя-данного-1

[ДОПОЛНИТЕЛЬНЫЙ КЛЮЧ ЗАПИСИ имя-данного-2 [С ДУБЛИРОВАНИЕМ]] ...

[СОСТОЯНИЕ ФАЙЛА имя-данного-3].

2.3.4. Сортируемый-сливаемый файл

ДЛЯ имя-файла-1 НАЗНАЧИТЬ { имя-реализации-1 / литерал-1 } ...

GOST EXPERT ГОСТ 22558-89,

2.3.5. Файл отчетов

ДЛЯ [НЕОБЯЗАТЕЛЬНОГО] имя-файла-1

НАЗНАЧИТЬ { имя-реализации-1 / литерал-1 } ...

[РЕЗЕРВИРОВАТЬ целое-1 ОБЛАСТЕЙ]

{[ОРГАНИЗАЦИЯ] ПОСЛЕДОВАТЕЛЬНАЯ]

[ЛИТЕРА ЗАПОЛНИТЕЛЬ { имя-данного-1 / литерал-2 }]

[ОГРАНИЧИТЕЛЬ ЗАПИСИ { СТАНДАРТ-А / имя-реализации-2 }]

[ДОСТУП ПОСЛЕДОВАТЕЛЬНЫЙ]

[СОСТОЯНИЕ ФАЙЛА имя-данного-2].

2.4. **Общий формат раздела данных**

[РАЗДЕЛ ДАННЫХ.

[СЕКЦИЯ ФАЙЛОВ.

[статья-описания-файла {статья-описания-записи} ...
  статья-описания-сортируемого-сливаемого-файла
                {статья-описания-записи} ...
  статья-описания-файла отчетов] ...]

[СЕКЦИЯ РАБОЧЕЙ-ПАМЯТИ.

[статья-описания-уровня-77 / статья-описания-записи] ...]

[СЕКЦИЯ СВЯЗИ.

[статья-описания-уровня-77 / статья-описания-записи] ...]

[СЕКЦИЯ КОММУНИКАЦИЙ.

[статья-описания-коммуникации [статья-описания-записи]...] ...]

[СЕКЦИЯ ОТЧЕТОВ.

[статья-описания-отчета {статья-описания-группы-отчета}...] ...]]

GOST EXPERT ГОСТ 22558-89,

2.5. Общий формат статьи описания файла

2.5.1. Последовательный файл

ОФ имя-файла-1

[ВНЕШНЕЕ]

[ГЛОБАЛЬНОЕ]

[В БЛОКЕ [ОТ целое-1 ДО] целое-2 {ЗАПИСЕЙ / ЛИТЕР}]

[В ЗАПИСИ {целое-3 ЛИТЕР / ПЕРЕМЕННОЕ ЧИСЛО / [[ОТ целое-4] [ДО целое-5] ЛИТЕР] [В ЗАВИСИМОСТИ ОТ имя-данного-1] / ОТ целое-6 ДО целое-7 ЛИТЕР}]

[МЕТКИ {СТАНДАРТНЫ / ОПУЩЕНЫ}]

[{ЗНАЧЕНИЕ / ЗНАЧ} {имя-реализации-1 {имя-данного-2 / литерал-1}} ...]

[ЗАПИСИ ДАННЫХ {имя-данного-3} ...]

[ВЕРСТКА {имя-данного-4 / целое-8} СТРОК

[КОНЦОВКА ОТ {имя-данного-5 / целое-9} [ ]

[ВЕРХНЕЕ ПОЛЕ {имя-данного-6 / целое-10}]

[НИЖНЕЕ ПОЛЕ {имя-данного-7 / целое-11}]]]

[АЛФАВИТ имя-алфавита-1].

2.5.2. Относительный файл

ОФ имя-файла-1

[ВНЕШНЕЕ]

[ГЛОБАЛЬНОЕ]

[В БЛОКЕ [ОТ целое-1 ДО] целое-2 {ЗАПИСЕЙ / ЛИТЕР}]

GOST EXPERT ГОСТ 22558-89,

[В ЗАПИСИ { целое-3 ЛИТЕР
ПЕРЕМЕННОЕ ЧИСЛО
[[ОТ целое-4] [ДО целое-5] ЛИТЕР]
[В ЗАВИСИМОСТИ ОТ имя-данного-1]
ОТ целое-6 ДО целое-7 ЛИТЕР }]

[МЕТКИ { СТАНДАРТНЫ / ОПУЩЕНЫ }]

[{ ЗНАЧЕНИЕ / ЗНАЧ } { имя-реализации-1 { имя-данного-2 / литерал-1 } } ...]

[ЗАПИСИ ДАННЫХ {имя-данного-3} ...].

2.5.3. Индексный файл

ОФ имя-файла-1

[ВНЕШНЕЕ]

[ГЛОБАЛЬНОЕ]

[В БЛОКЕ [ОТ целое-1 ДО] целое-2 { ЗАПИСЕЙ / ЛИТЕР }]

[В ЗАПИСИ { целое-3 ЛИТЕР
ПЕРЕМЕННОЕ ЧИСЛО [[ОТ целое-4]
[ДО целое-5] ЛИТЕР]
[В ЗАВИСИМОСТИ ОТ имя-данного-1]
ОТ целое-6 ДО целое-7 ЛИТЕР }]

[МЕТКИ { СТАНДАРТНЫ / ОПУЩЕНЫ }]

[{ ЗНАЧЕНИЕ / ЗНАЧ } { имя-реализации-1 { имя-данного-2 / литерал-1 } } ...]

[ЗАПИСИ ДАННЫХ {имя-данного-3} ...].

2.5.4. Сортируемый-сливаемый файл

ОС имя-файла-1

[В ЗАПИСИ { целое-1 ЛИТЕР
ПЕРЕМЕННОЕ ЧИСЛО
[[ОТ целое-2] [ДО целое-3] ЛИТЕР]
[В ЗАВИСИМОСТИ ОТ имя-данного-1]
ОТ целое-4 ДО целое-5 ЛИТЕР }]

GOST EXPERT ГОСТ 22558-89,

[{ ЗАПИСЬ / ЗАПИСИ } ДАННЫХ {имя-данного-3} ...] .

2.5.5. Файл отчетов

ОФ имя-файла-1

[ВНЕШНЕЕ]

[ГЛОБАЛЬНОЕ]

[В БЛОКЕ [ОТ целое-1 ДО] целое-2 { ЗАПИСЕЙ / ЛИТЕР }]

[В ЗАПИСИ { целое-3 ЛИТЕР / ОТ целое-4 ДО целое-5 ЛИТЕР }]

[МЕТКИ { СТАНДАРТНЫ / ОПУЩЕНЫ }]

[{ ЗНАЧЕНИЕ / ЗНАЧ } { имя-реализации-1 { имя-данного-2 / литерал-2 } } ...]

[АЛФАВИТ имя-алфавита-1]

{ ОТЧЕТ / ОТЧЕТЫ } {имя-отчета-1} ... .

2.6. Общий формат статьи описания данного

Формат 1

номер-уровня { имя-данного-1 / ЗАПОЛНИТЕЛЬ / ЗАП }

[ПЕРЕОПРЕДЕЛЯЕТ имя-данного-2]

[ГЛОБАЛЬНОЕ]

[ВНЕШНЕЕ]

[{ ШАБЛОН / Ш } строка-литер]

GOST EXPERT ГОСТ 22558-89,

[ДВОИЧНОЕ / ДЕСЯТИЧНОЕ ДЛЯ {ВЫЧИСЛЕНИЙ / ВЫЧ / ВЫДАЧИ / ИНДЕКСА}]

[[ЗНАК] {ПЕРВЫЙ / ПОСЛЕДНИЙ} [ОТДЕЛЬНО]]

[ПОВТОРЯЕТСЯ целое-2 РАЗ
[ПО {ВОЗРАСТАНИЮ / УБЫВАНИЮ} КЛЮЧА {имя-данного-3} ...] ...
[ИНДЕКСИРУЕТСЯ {имя-индекса-1} ...]
ПОВТОРЯЕТСЯ ОТ целое-1 ДО целое-2 РАЗ
В ЗАВИСИМОСТИ ОТ имя-данного-4
[ПО {ВОЗРАСТАНИЮ / УБЫВАНИЮ} КЛЮЧА {имя-данного-3} ...] ...
[ИНДЕКСИРУЕТСЯ {имя-индекса-1} ...]]

[ВЫДЕЛЕНО [ВЛЕВО / ВПРАВО]]

[СДВИНУТО ВПРАВО]

[ПРОБЕЛ КОГДА НУЛЬ]

[{ЗНАЧЕНИЕ / ЗНАЧ} литерал-1].

Формат 2

66 имя-данного-1 ПЕРЕИМЕНОВЫВАЕТ имя-данного-2
[ПО имя-данного-3].

Формат 3

88 имя-условия-1 {ЗНАЧЕНИЕ / ЗНАЧ} {литерал-1 [ПО литерал-2]} ...

GOST EXPERT ГОСТ 22558-89,

2.7. Общий формат статьи описания коммуникации

Формат 1

ОК имя-коммуникации-1 ДЛЯ [НАЧАЛЬНОГО] ВВОДА

[[СИМВОЛИЧЕСКАЯ ОЧЕРЕДЬ имя-данного-1]
[СИМВОЛИЧЕСКАЯ ПОДОЧЕРЕДЬ-1 имя-данного-2]
[СИМВОЛИЧЕСКАЯ ПОДОЧЕРЕДЬ-2 имя-данного-3]
[СИМВОЛИЧЕСКАЯ ПОДОЧЕРЕДЬ-3 имя-данного-4]
[ДАТА СООБЩЕНИЯ имя-данного-5]
[ВРЕМЯ СООБЩЕНИЯ имя-данного-6]
[СИМВОЛИЧЕСКИЙ ИСТОЧНИК имя-данного-7]
[ДЛИНА ТЕКСТА имя-данного-8]
[КЛЮЧ КОНЦА имя-данного-9]
[КЛЮЧ СОСТОЯНИЯ имя-данного-10]
[ЧИСЛО СООБЩЕНИЙ имя-данного-11]]
[имя-данного-1, имя-данного-2, имя-данного-3, имя-данного-4, имя-данного-5, имя-данного-6, имя-данного-7, имя-данного-8,, имя-данного-9, имя-данного-10, имя данного-11]

Формат 2

ОК имя-коммуникации-1 ДЛЯ ВЫВОДА
[ЧИСЛО АДРЕСАТОВ имя-данного-1]
[ДЛИНА ТЕКСТА имя-данного-2]
[КЛЮЧ СОСТОЯНИЯ имя-данного-3]
[ТАБЛИЦА АДРЕСАТОВ ПОВТОРЯЕТСЯ целое-1 РАЗ
[ИНДЕКСИРУЕТСЯ {имя-индекса-1}...]]
[КЛЮЧ ОШИБКИ имя-данного-4]
[СИМВОЛИЧЕСКИЙ АДРЕСАТ имя-данного-5].

GOST EXPERT ГОСТ 22558-89,

Формат 3

ОК имя-коммуникации-1
ДЛЯ [НАЧАЛЬНОГО ВВОДА-ВЫВОДА
[[ДАТА СООБЩЕНИЯ имя-данного-1]
[ВРЕМЯ СООБЩЕНИЯ имя-данного-2]
[СИМВОЛИЧЕСКИЙ ТЕРМИНАЛ имя-данного-3]
[ДЛИНА ТЕКСТА имя-данного-4]
[КЛЮЧ КОНЦА имя-данного-5]
[КЛЮЧ СОСТОЯНИЯ имя-данного-6]]
[имя-данного-1, имя-данного-2, имя-данного-3,
имя-данного-4, имя-данного-5, имя-данного-6]

2.8. Общий формат статьи описания отчета

ОО имя-отчета-1
[ГЛОБАЛЬНОЕ]
[С КОДОМ литерал-1]
[УПРАВЛЕНИЕ ПО {имя-данного-1} ...
КОНЦУ [имя-данного-1] ...]
[РАЗМЕР СТРАНИЦЫ целое-1 СТРОК
[ЗАГОЛОВОК целое-2] [ПЕРВЫЙ ФРАГМЕНТ целое-3]
[ПОСЛЕДНИЙ ФРАГМЕНТ целое-4] [КОНЦОВКА
целое-5]].

2.9. Общий формат статьи описания группы отчета

Формат 1

01 [имя-данного-1]
[НОМЕР СТРОКИ {целое-1
ПЛЮС целое-2
[НА СЛЕДУЮЩЕЙ СТРАНИЦЕ]}]
[СЛЕДУЮЩАЯ ГРУППА {целое-3
ПЛЮС целое-4
НА СЛЕДУЮЩЕЙ СТРАНИЦЕ}]

GOST EXPERT ГОСТ 22558-89,

{ { ЗАГОЛОВОК ОТЧЕТА / ЗО }
{ ЗАГОЛОВОК СТРАНИЦЫ / ЗС }
{ УПРАВЛЯЕМЫЙ ЗАГОЛОВОК / УЗ } ПО { имя-данного-2 / КОНЦУ }
ТИП { { ФРАГМЕНТ / ФР }
{ УПРАВЛЯЕМАЯ КОНЦОВКА / УК } ПО { имя-данного-3 / КОНЦУ }
{ КОНЦОВКА СТРАНИЦЫ / КС }
{ КОНЦОВКА ОТЧЕТА / КО } }

[ДЛЯ ВЫДАЧИ].

Формат 2

номер-уровня [имя-данного-1]

[НОМЕР СТРОКИ { целое-1 [НА СЛЕДУЮЩЕЙ СТРАНИЦЕ] / ПЛЮС целое 2 }]

[ДЛЯ ВЫДАЧИ].

Формат 3

номер-уровня [имя-данного-1]

{ ШАБЛОН / Ш } строка-литер

[ДЛЯ ВЫДАЧИ]

[[ЗНАК] { ПЕРВЫЙ / ПОСЛЕДНИЙ } [ОТДЕЛЬНО]]

GOST EXPERT ГОСТ 22558-89,

[СДВИНУТО ВПРАВО]

[ПРОБЕЛ КОГДА НУЛЬ]

[НОМЕР СТРОКИ { целое-1 [НА СЛЕДУЮЩЕЙ СТРАНИЦЕ] / ПЛЮС целое-2 }]

[НОМЕР СТОЛБЦА целое-3]

{ ИСТОЧНИК идентификатор-1 /
ЗНАЧЕНИЕ / ЗНАЧ } литерал-1 /
{СУММА {идентификатор-2} ...
[ДЛЯ {имя-данного-2} ...]} ...
[СБРОСИТЬ ПО { имя-данного-3 / КОНЦУ }] }

[ОПРЕДЕЛЯЕТ ГРУППУ].

2.10. **Общий формат раздела процедур**

Формат 1

[РАЗДЕЛ ПРОЦЕДУР [ИСПОЛЬЗУЯ {имя-данного} ...].

[ДЕКЛАРАТИВЫ.

{СЕКЦИЯ имя-секции [номер-сегмента].

Оператор ИСПОЛЬЗОВАТЬ.

[имя-параграфа.
[предложение] ...] ... } ...

КОНЕЦ ДЕКЛАРАТИВ.

{СЕКЦИЯ имя-секции [номер-сегмента].

[имя-параграфа.
[предложение] ...] ... } ...

Формат 2

[РАЗДЕЛ ПРОЦЕДУР [ИСПОЛЬЗУЯ {имя-данного-1}...].

{имя-параграфа.
[предложение] ... } ...]

GOST EXPERT ГОСТ 22558-89,

2.11. Общий формат глаголов Кобола

2.11.1

ПРИНЯТЬ идентификатор-1 [С мнемоническое-имя-1]

ПРИНЯТЬ В идентификатор-2 { ДАТУ / ДЕНЬ / ДЕНЬ-НЕДЕЛИ / ВРЕМЯ }

ПРИНЯТЬ ЧИСЛО СООБЩЕНИЙ имя-коммуникации-1

2.11.2

СЛОЖИТЬ { идентификатор-1 / литерал-1 } ... С {идентификатор-2 [ОКРУГЛЯЯ]} ...
[ПРИ ПЕРЕПОЛНЕНИИ повелительный-оператор-1]
[БЕЗ ПЕРЕПОЛНЕНИЯ повелительный-оператор-2]
[КОНЕЦ-СЛОЖИТЬ]

СЛОЖИТЬ { идентификатор-1 / литерал-1 } ... С { идентификатор-2 / литерал-2 }
ПОЛУЧАЯ {идентификатор-3 [ОКРУГЛЯЯ]} ...
[ПРИ ПЕРЕПОЛНЕНИИ повелительный-оператор-1]
[БЕЗ ПЕРЕПОЛНЕНИЯ повелительный-оператор-2]
[КОНЕЦ-СЛОЖИТЬ]

СЛОЖИТЬ { СООТВЕТСТВЕННО / СООТВ } идентификатор-1
С идентификатор-2 [ОКРУГЛЯЯ]
[ПРИ ПЕРЕПОЛНЕНИИ повелительный-оператор-1]
[БЕЗ ПЕРЕПОЛНЕНИЯ повелительный-оператор-2]
[КОНЕЦ-СЛОЖИТЬ]

2.11.3

ИЗМЕНИТЬ {имя-процедуры-1 [ДЛЯ ПЕРЕХОДА] К имя-процедуры-2} ...

GOST EXPERT ГОСТ 22558-89,

2.11.4

ВЫЗВАТЬ { идентификатор-1 / литерал-1 }

[ИСПОЛЬЗУЯ

{ [ССЫЛКУ НА] {идентификатор-2} ... / ЗНАЧЕНИЕ {идентификатор-2} ... } ...]

[ПРИ ПЕРЕПОЛНЕНИИ повелительный-оператор-1]

[КОНЕЦ-ВЫЗВАТЬ]

ВЫЗВАТЬ { идентификатор-1 / литерал-1 }

[ИСПОЛЬЗУЯ

{ [ССЫЛКУ НА] {идентификатор-2} ... / ЗНАЧЕНИЕ {идентификатор-2} ... } ...]

[ПРИ ОШИБКЕ повелительный-оператор-1]

[БЕЗ ОШИБКИ повелительный-оператор-2]

[КОНЕЦ-ВЫЗВАТЬ]

2.11.5

ОСВОБОДИТЬ { идентификатор-1 / литерал-1 } ...

2.11.6

ЗАКРЫТЬ { { КАТУШКУ / ТОМ } имя-файла-1 [С УДАЛЕНИЕМ] / имя-файла-1 [БЕЗ ПЕРЕМОТКИ / С ЗАМКОМ] } ...

ЗАКРЫТЬ {имя-файла-1 [С ЗАМКОМ]} ...

2.11.7

ВЫЧИСЛИТЬ {идентификатор-1 [ОКРУГЛЯЯ]} ... =

арифметическое-выражение-1

[ПРИ ПЕРЕПОЛНЕНИИ повелительный-оператор-1]

GOST EXPERT ГОСТ 22558-89,

[БЕЗ ПЕРЕПОЛНЕНИЯ повелительный-оператор-2]
[КОНЕЦ-ВЫЧИСЛИТЬ]

2.11.8
ПРОДОЛЖИТЬ

2.11.9
УДАЛИТЬ ЗАПИСЬ имя-файла-1
[ПРИ ОШИБКЕ КЛЮЧА повелительный-оператор-1]
[БЕЗ ОШИБКИ КЛЮЧА повелительный-оператор-2]
[КОНЕЦ-УДАЛИТЬ]

2.11.10
ЗАПРЕТИТЬ { ВВОД [С ТЕРМИНАЛА] / ВВОД-ВЫВОД С ТЕРМИНАЛА / ВЫВОД }
имя-коммуникации-1
[КЛЮЧ { идентификатор-1 / литерал-1 }]

2.11.11
ВЫДАТЬ { идентификатор-1 / литерал-1 } ... [НА мнемоническое-имя-1]
[БЕЗ ПРОДВИЖЕНИЯ]

2.11.12
РАЗДЕЛИТЬ НА { идентификатор-1 / литерал-1 }
{идентификатор-2 [ОКРУГЛЯЯ]} ...
[ПРИ ПЕРЕПОЛНЕНИИ повелительный-оператор-1]
[БЕЗ ПЕРЕПОЛНЕНИЯ повелительный-оператор-2]
[КОНЕЦ-РАЗДЕЛИТЬ]

РАЗДЕЛИТЬ НА { идентификатор-1 / литерал-1 } { идентификатор-2 / литерал-2 }
[ПОЛУЧАЯ {идентификатор-3 [ОКРУГЛЯЯ]} ...
[ПРИ ПЕРЕПОЛНЕНИИ повелительный-оператор-1]

GOST EXPERT ГОСТ 22558-89,

[БЕЗ ПЕРЕПОЛНЕНИЯ повелительный-оператор-2]
[КОНЕЦ-РАЗДЕЛИТЬ]

РАЗДЕЛИТЬ { идентификатор-1 / литерал-1 } НА { идентификатор-2 / литерал-2 }

ПОЛУЧАЯ {идентификатор-3 [ОКРУГЛЯЯ]} . . .
[ПРИ ПЕРЕПОЛНЕНИИ повелительный-оператор-1]
[БЕЗ ПЕРЕПОЛНЕНИЯ повелительный-оператор-2]
[КОНЕЦ-РАЗДЕЛИТЬ]

РАЗДЕЛИТЬ НА { идентификатор-1 / литерал-1 } { идентификатор-2 / литерал-2 }

ПОЛУЧАЯ идентификатор-3 [ОКРУГЛЯЯ]
ОСТАТОК идентификатор-4
[ПРИ ПЕРЕПОЛНЕНИИ повелительный-оператор-1]
[БЕЗ ПЕРЕПОЛНЕНИЯ повелительный-оператор-2]
[КОНЕЦ-РАЗДЕЛИТЬ]

РАЗДЕЛИТЬ { идентификатор-1 / литерал-1 } НА { идентификатор-2 / литерал-2 }

ПОЛУЧАЯ идентификатор-3 [ОКРУГЛЯЯ] ОСТАТОК
идентификатор-4
[ПРИ ПЕРЕПОЛНЕНИИ повелительный-оператор-1]
[БЕЗ ПЕРЕПОЛНЕНИЯ повелительный-оператор-2]
[КОНЕЦ-РАЗДЕЛИТЬ]

2.11.13

РАЗРЕШИТЬ { ВВОД [С ТЕРМИНАЛА] / ВВОД-ВЫВОД С ТЕРМИНАЛА / ВЫВОД }
имя-коммуникации-1

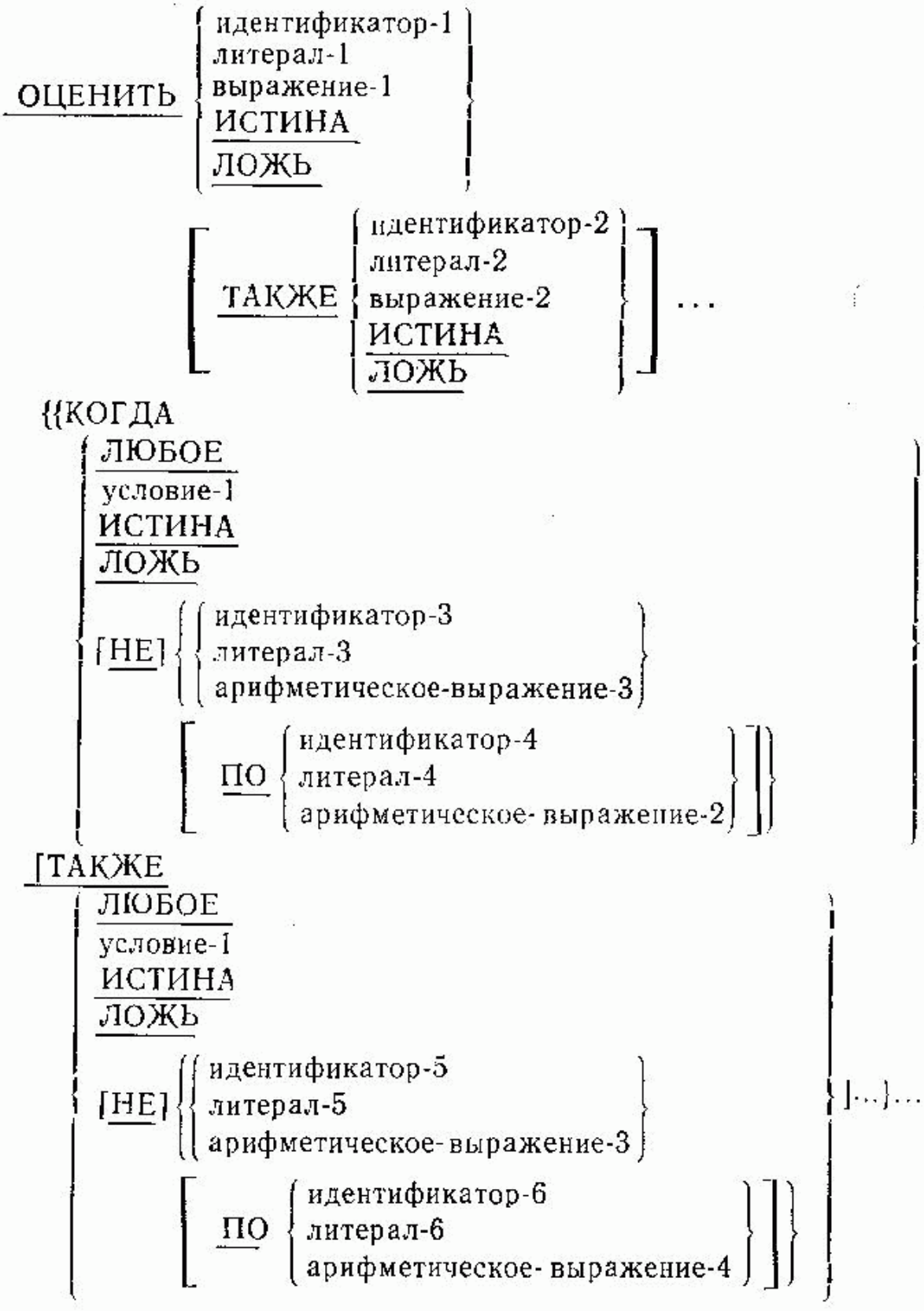
[КЛЮЧ { идентификатор-1 / литерал-1 }]

GOST EXPERT ГОСТ 22558-89,

2.11.14

ВОЙТИ В имя-языка-1 [имя-программного модуля-1].

2.11.15



GOST EXPERT ГОСТ 22558-89,

повелительный-оператор-1} . . .

[ИНАЧЕ повелительный-оператор-2]

[КОНЕЦ-ОЦЕНИТЬ]

2.11.16

ВЫЙТИ

ВЫЙТИ ИЗ ПРОГРАММЫ

2.11.17

ГЕНЕРИРОВАТЬ { имя-данного-1 / имя-отчета-1 }

2.11.18

ПЕРЕЙТИ К [имя-процедуры-1]

ПЕРЕЙТИ К {имя-процедуры-1} . . . В ЗАВИСИМОСТИ ОТ идентификатор-1

2.11.19

ЕСЛИ условие-1 ТО { {оператор-1} . . . / СЛЕДУЮЩЕЕ ПРЕДЛОЖЕНИЕ }

{ ИНАЧЕ {оператор-2} . . . [КОНЕЦ-ЕСЛИ] / ИНАЧЕ СЛЕДУЮЩЕЕ ПРЕДЛОЖЕНИЕ / КОНЕЦ-ЕСЛИ }

2.11.20

ИНИЦИИРОВАТЬ {идентификатор-1} . . .

[ЗАМЕНЯЯ { { БУКВЕННОЕ / БЦ / ЧИСЛОВОЕ / БЦР / ЧР } ДАННОЕ НА { идентификатор-2 / литерал-1 } } . . .]

2.11.21

НАЧАТЬ {имя-отчета-1} . . .

GOST EXPERT ГОСТ 22558-89,

230 ГОСТ 22558-89

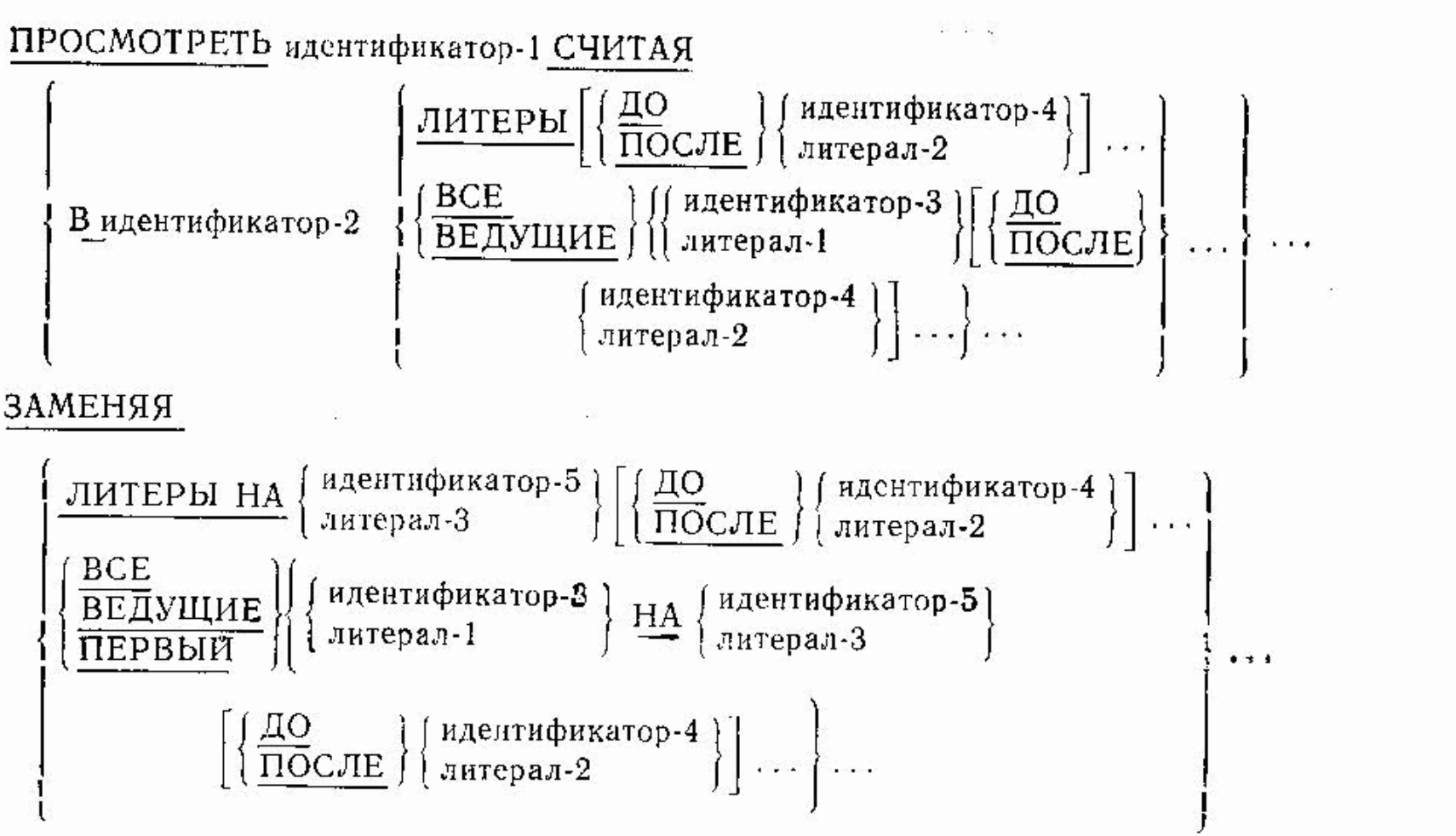
2.11.22

ПРОСМОТРЕТЬ идентификатор-1 СЧИТАЯ

{ В идентификатор-2 { { ЛИТЕРЫ [{ ДО / ПОСЛЕ } { идентификатор-4 / литерал-2 }] ... } / { { ВСЕ / ВЕДУЩИЕ } { { идентификатор-3 / литерал-1 } [{ ДО / ПОСЛЕ } { идентификатор-4 / литерал-2 }] ... } ... } }

ПРОСМОТРЕТЬ идентификатор-1 ЗАМЕНЯЯ

{ ЛИТЕРЫ НА { идентификатор-5 / литерал-3 } [{ ДО / ПОСЛЕ } { идентификатор-4 / литерал-2 }] ... / { { ВСЕ / ВЕДУЩИЕ / ПЕРВЫЙ } { { идентификатор-3 / литерал-1 } НА { идентификатор-5 / литерал-3 } [{ ДО / ПОСЛЕ } { идентификатор-4 / литерал-2 }] ... } ... } ... }


ПРОСМОТРЕТЬ идентификатор-1 СЧИТАЯ
В идентификатор-2
ЛИТЕРЫ
ДО
ПОСЛЕ
идентификатор-4
литерал-2
ВСЕ
ВЕДУЩИЕ
идентификатор-3
литерал-1
ЗАМЕНЯЯ
ЛИТЕРЫ НА
идентификатор-5
литерал-3
ПЕРВЫЙ
НА
ПРОСМОТРЕТЬ идентификатор-1 ПРЕВРАЩАЯ
идентификатор-6
литерал-4
В
идентификатор-7
литерал-5

2.11.23

СЛИТЬ имя-файла-1

{ ПО { ВОЗРАСТАНИЮ / УБЫВАНИЮ } КЛЮЧА {имя-данного-1} ... } ...

[АЛФАВИТ имя-алфавита-1]

ИСПОЛЬЗУЯ имя-файла-2 {имя-файла-3} ...

{ ПРОЦЕДУРА ВЫВОДА имя-процедуры-1 [ПО имя-процедуры-2] / ПОЛУЧАЯ {имя-файла-4} ... }

2.11.24

ПОМЕСТИТЬ { идентификатор-1 / литерал-1 } В {идентификатор-2} ...

ПОМЕСТИТЬ { СООТВЕТСТВЕННО / СООТВ } идентификатор-1

В идентификатор-2

2.11.25

УМНОЖИТЬ { идентификатор-1 / литерал-1 } НА {идентификатор-2

[ОКРУГЛЯЯ]} ...

[ПРИ ПЕРЕПОЛНЕНИИ повелительный-оператор-1]

[БЕЗ ПЕРЕПОЛНЕНИЯ повелительный-оператор-2]

[КОНЕЦ-УМНОЖИТЬ]

УМНОЖИТЬ { идентификатор-1 / литерал-1 } НА { идентификатор-2 / литерал-2 }

ПОЛУЧАЯ {идентификатор-3 [ОКРУГЛЯЯ]} ...

[ПРИ ПЕРЕПОЛНЕНИИ повелительный-оператор-1]

[БЕЗ ПЕРЕПОЛНЕНИЯ повелительный-оператор-2]

[КОНЕЦ-УМНОЖИТЬ]

GOST EXPERT ГОСТ 22558-89,

2.11.26

ОТКРЫТЬ { ВХОДНОЙ { имя-файла-1 [РЕВЕРСНО | БЕЗ ПЕРЕМОТКИ] } ...
ВЫХОДНОЙ {имя-файла-2 [БЕЗ ПЕРЕМОТКИ]} ...
ВХОДНОЙ-ВЫХОДНОЙ {имя-файла-3} ...
ДОПОЛНЯЕМЫЙ {имя-файла-4} ... } ...

ОТКРЫТЬ { ВХОДНОЙ {имя-файла-1} ...
ВЫХОДНОЙ {имя-файла-2} ...
ВХОДНОЙ-ВЫХОДНОЙ {имя-файла-3} ...
ДОПОЛНЯЕМЫЙ {имя-файла-4} ... } ...

ОТКРЫТЬ { ВЫХОДНОЙ {имя-файла-1 [БЕЗ ПЕРЕМОТКИ]} ...
ДОПОЛНЯЕМЫЙ {имя-файла-2} ... } ...

2.11.27

ВЫПОЛНИТЬ {имя-процедуры-1 [ПО имя-процедуры-2]}
[повелительный-оператор-1 КОНЕЦ-ВЫПОЛНИТЬ]

ВЫПОЛНИТЬ [имя-процедуры-1 [ПО имя-процедуры-2]]
{ идентификатор-1 | целое-1 } { РАЗ | РАЗА }
[повелительный-оператор-1 КОНЕЦ-ВЫПОЛНИТЬ]

ВЫПОЛНИТЬ [имя-процедуры-1 [ПО имя-процедуры-2]]
[С ПРОВЕРКОЙ В { НАЧАЛЕ | КОНЦЕ }] ДО условие-1
[повелительный-оператор-1 КОНЕЦ-ВЫПОЛНИТЬ]

ВЫПОЛНИТЬ [имя-процедуры-1 [ПО имя-процедуры-2]]
[С ПРОВЕРКОЙ В { НАЧАЛЕ | КОНЦЕ }]

GOST EXPERT ГОСТ 22558-89,

МЕНЯЯ { идентификатор-2 / имя-индекса-1 } ОТ { идентификатор-3 / имя-индекса-2 / литерал-1 }

НА { идентификатор-4 / литерал-2 } ДО условие-1

[ЗАТЕМ { идентификатор-5 / имя-индекса-3 } ОТ { идентификатор-6 / имя-индекса-4 / литерал-3 }

НА { идентификатор-7 / литерал-4 } ДО условие-2] ...

[повелительный-оператор-1 КОНЕЦ-ВЫПОЛНИТЬ]

2.11.28

ОЧИСТИТЬ имя-коммуникации-1

2.11.29

ЧИТАТЬ [СЛЕДУЮЩУЮ] ЗАПИСЬ имя-файла-1
[В идентификатор-1]
[В КОНЦЕ повелительный-оператор-1]
[НЕ В КОНЦЕ повелительный-оператор-2]
[КОНЕЦ-ЧИТАТЬ]

ЧИТАТЬ ЗАПИСЬ имя-файла-1 [В идентификатор-1]
[ПРИ ОШИБКЕ КЛЮЧА повелительный-оператор-3]
[БЕЗ ОШИБКИ КЛЮЧА повелительный-оператор-4]
[КОНЕЦ-ЧИТАТЬ].

ЧИТАТЬ ЗАПИСЬ имя-файла-1 [В идентификатор-1]
[КЛЮЧ имя-данного-1]
[ПРИ ОШИБКЕ КЛЮЧА повелительный-оператор-3]
[БЕЗ ОШИБКИ КЛЮЧА повелительный-оператор-4]
[КОНЕЦ-ЧИТАТЬ]

GOST EXPERT ГОСТ 22558-89,

2.11.30

ПОЛУЧИТЬ { СЕГМЕНТ / СООБЩЕНИЕ } имя-коммуникации В
идентификатор-1
[НЕТ ДАННЫХ повелительный-оператор-1]
[ЕСТЬ ДАННЫЕ повелительный-оператор-2]
[КОНЕЦ-ПОЛУЧИТЬ]

2.11.31

ПЕРЕДАТЬ имя-записи-1 [ИЗ ПОЛЯ идентификатор-1]

2.11.32

ВЕРНУТЬ ЗАПИСЬ имя-файла-1 [В идентификатор-1]
[В КОНЦЕ повелительный-оператор-1]
[НЕ В КОНЦЕ повелительный-оператор-2]
[КОНЕЦ-ВЕРНУТЬ]

2.11.33

ОБНОВИТЬ имя-записи-1 [ИЗ ПОЛЯ идентификатор-1]
ОБНОВИТЬ имя-записи-1 [ИЗ ПОЛЯ идентификатор-1]
[ПРИ ОШИБКЕ КЛЮЧА повелительный-оператор-1]
[БЕЗ ОШИБКИ КЛЮЧА повелительный-оператор-2]
[КОНЕЦ-ОБНОВИТЬ]

2.11.34

ИСКАТЬ В идентификатор-1 [МЕНЯЯ { идентификатор-2 / имя-индекса-1 }]
[В КОНЦЕ повелительный-оператор-1]
{ КОГДА условие-1 { повелительный-оператор-2 / СЛЕДУЮЩЕЕ-ПРЕДЛОЖЕНИЕ } } ...
[КОНЕЦ-ИСКАТЬ]
ИСКАТЬ ОСОБО В идентификатор-1 [В КОНЦЕ
повелительный-оператор-1]

GOST EXPERT ГОСТ 22558-89,

КОГДА { имя-данного-1 { РАВНО / = } { идентификатор-3 / литерал-1 / арифметическое-выражение-1 } / имя-условия-1 }

[И { имя-данного-2 { РАВНО / = } { идентификатор-4 / литерал-2 / арифметическое-выражение-2 } / имя-условия-2 }] ...

{ повелительный-оператор-2 / СЛЕДУЮЩЕЕ ПРЕДЛОЖЕНИЕ }

[КОНЕЦ-ИСКАТЬ]

2.11.35

ПОСЛАТЬ имя-коммуникации-1 ИЗ ПОЛЯ идентификатор-1

ПОСЛАТЬ имя-коммуникации-1 [ИЗ ПОЛЯ идентификатор-1]

{ С идентификатор-2 / С ИКС / С ИКЩ / С ИКГ }

[{ ДО / ПОСЛЕ } ПРОДВИЖЕНИЯ { { идентификатор-3 / целое-1 } СТРОК / мнемоническое-имя-1 СТРАНИЦЫ }]

[ЗАМЕНЯЯ СТРОКУ]

2.11.36

УСТАНОВИТЬ { идентификатор-1 / имя-индекса-1 } ... НА { идентификатор-2 / имя-индекса-2 / целое-1 }

GOST EXPERT ГОСТ 22558-89,

УСТАНОВИТЬ {имя-индекса-3} ... { ПРИБАВЛЯЯ / ВЫЧИТАЯ }

{ идентификатор-3 / целое-2 }

УСТАНОВИТЬ {{мнемоническое-имя-1} ... НА

{ ВКЛЮЧЕНО / ВЫКЛЮЧЕНО } } ...

УСТАНОВИТЬ {имя-условия-1} ... НА ИСТИНА

2.11.37

СОРТИРОВАТЬ имя-файла-1

{ ПО { ВОЗРАСТАНИЮ / УБЫВАНИЮ } КЛЮЧА {имя-данного-1} ... } ...

[С ДУБЛИРОВАНИЕМ]

[АЛФАВИТ имя-алфавита-1]

{ ПРОЦЕДУРА ВВОДА имя-процедуры-1 [ПО имя-процедуры-2] / ИСПОЛЬЗУЯ {имя-файла-2} ... }

{ ПРОЦЕДУРА ВЫВОДА имя-процедуры-3 [ПО имя-процедуры-4] / ПОЛУЧАЯ {имя-файла-3} ... }

2.11.38

ПОДВЕСТИ ЗАПИСЬ имя-файла-1

[КЛЮЧ { РАВЕН / = / БОЛЬШЕ / > / НЕ МЕНЬШЕ / НЕ < / БОЛЬШЕ ИЛИ РАВЕН / >= } имя-данного-1]

[ПРИ ОШИБКЕ КЛЮЧА повелительный-оператор-1]

[БЕЗ ОШИБКИ КЛЮЧА повелительный-оператор-2]

[КОНЕЦ-ПОДВЕСТИ]

GOST EXPERT ГОСТ 22558-89,

2.11.39

ОСТАНОВИТЬ { РАБОТУ / литерал-1 }

2.11.40

СОБРАТЬ { { идентификатор-1 / литерал-1 } ... ОГРАНИЧИВАЯСЬ

{ идентификатор-2 / литерал-2 / РАЗМЕРОМ } ...

В идентификатор-3 [УКАЗАТЕЛЬ идентификатор-4]
[ПРИ ПЕРЕПОЛНЕНИИ повелительный-оператор-1]
[БЕЗ ПЕРЕПОЛНЕНИЯ повелительный-оператор-2]
[КОНЕЦ-СОБРАТЬ]

2.11.41

ОТНЯТЬ { литерал-1 / идентификатор-1 } ... ОТ {идентификатор-2
[ОКРУГЛЯЯ]} ...
[ПРИ ПЕРЕПОЛНЕНИИ повелительный-оператор-1]
[БЕЗ ПЕРЕПОЛНЕНИЯ повелительный-оператор-2]
[КОНЕЦ-ОТНЯТЬ]

ОТНЯТЬ { литерал-1 / идентификатор-1 } ... ОТ { литерал-2 / идентификатор-2 }
ПОЛУЧАЯ {идентификатор-3 [ОКРУГЛЯЯ]} ...
[ПРИ ПЕРЕПОЛНЕНИИ повелительный-оператор-1]
[БЕЗ ПЕРЕПОЛНЕНИЯ повелительный-оператор-2]
[КОНЕЦ-ОТНЯТЬ]

ОТНЯТЬ { СООТВЕТСТВЕННО / СООТВ } идентификатор-1
ОТ идентификатор-2 [ОКРУГЛЯЯ]
[ПРИ ПЕРЕПОЛНЕНИИ повелительный-оператор-1]

GOST EXPERT ГОСТ 22558-89,

[БЕЗ ПЕРЕПОЛНЕНИЯ повелительный-оператор-2]

[КОНЕЦ-ОТНЯТЬ]

2.11.42

ПОДАВИТЬ ПЕЧАТЬ

2.11.43

ЗАКОНЧИТЬ {имя-отчета-1}...

2.11.44

РАЗОБРАТЬ идентификатор-1

[ОГРАНИЧИВАЯСЬ [ВСЕМИ] { идентификатор-2 / литерал-1 }

[ИЛИ [ВСЕМИ] { идентификатор-3 / литерал-2 }]...]

В {идентификатор-4 [ОГРАНИЧИТЕЛЬ В идентификатор-5]

[СЧЕТ В идентификатор-6]}...

[УКАЗАТЕЛЬ идентификатор-7]

[СЧИТАЯ В идентификатор-8]

[ПРИ ПЕРЕПОЛНЕНИИ повелительный-оператор-1]

[БЕЗ ПЕРЕПОЛНЕНИЯ повелительный-оператор-2]

[КОНЕЦ-РАЗОБРАТЬ]

2.11.45

ИСПОЛЬЗОВАТЬ [ГЛОБАЛЬНО] ПОСЛЕ СТАНДАРТНОЙ ПРОЦЕДУРЫ ОШИБКИ

ДЛЯ { {имя-файла-1}... / ВХОДНЫХ / ВЫХОДНЫХ / ВХОДНЫХ-ВЫХОДНЫХ / ДОПОЛНЯЕМЫХ }

ИСПОЛЬЗОВАТЬ ПОСЛЕ СТАНДАРТНОЙ ПРОЦЕДУРЫ ОШИБКИ

ДЛЯ { {имя-файла-1}... / ВЫХОДНЫХ / ДОПОЛНЯЕМЫХ }...

GOST EXPERT ГОСТ 22558-89,

ИСПОЛЬЗОВАТЬ [ГЛОБАЛЬНО] ДО ВЫДАЧИ
идентификатор-1

ИСПОЛЬЗОВАТЬ ДЛЯ ОТЛАДКИ ПРИ

{ имя-коммуникации-1
| [ВСЕХ ССЫЛКАХ НА] идентификатор-1
| имя-файла-1
| имя-процедуры-1
| ВСЕХ ПРОЦЕДУРАХ } ...

2.11.46

ПИСАТЬ имя-записи-1 [ИЗ ПОЛЯ идентификатор-1]

[{ ДО
| ПОСЛЕ } ПРОДВИЖЕНИЯ

[{ { идентификатор-2
| { целое-1 } СТРОК
| { мнемоническое-имя-1
| { СТРАНИЦЫ }]]

[В КОНЦЕ СТРАНИЦЫ повелительный-оператор-1]

[НЕ В КОНЦЕ СТРАНИЦЫ повелительный-оператор-2]

[КОНЕЦ-ПИСАТЬ]

ПИСАТЬ имя-записи-1 [ИЗ ПОЛЯ идентификатор-1]

[ПРИ ОШИБКЕ КЛЮЧА повелительный-оператор-1]

[БЕЗ ОШИБКИ КЛЮЧА повелительный-оператор-2]

[КОНЕЦ-ПИСАТЬ]

2.12. Общий формат операторов КОПИРОВАТЬ и ЗАМЕНИТЬ

2.12.1

КОПИРОВАТЬ имя-текста-1 [ИЗ имя-библиотеки-1]

[ЗАМЕНЯЯ { { == псевдотекст-1 == } | идентификатор-1 | литерал-1 | слово-1 }]

GOST EXPERT ГОСТ 22558-89,

НА { == псевдотекст-2 == | идентификатор-2 | литерал-2 | слово-2 } ...

2.12.2

ЗАМЕНИТЬ {==псевдотекст-1==НА
==псевдотекст-2==} ...

ОТКЛЮЧИТЬ ЗАМЕНИТЬ

2.13. Общий формат условий

2.13.1. Условие отношения

{ идентификатор-1 | литерал-1 | арифметическое-выражение-1 | имя-индекса-1 }
{ [НЕ] БОЛЬШЕ | [НЕ] МЕНЬШЕ | [НЕ] РАВНО | [НЕ] > | [НЕ] < | [НЕ] = | БОЛЬШЕ ИЛИ РАВНО | >= | МЕНЬШЕ ИЛИ РАВНО | <= }
{ идентификатор-2 | литерал-2 | арифметическое-выражение-2 | имя-индекса-2 }

2.13.2. Условие класса

идентификатор-1 [НЕ] { ЧИСЛОВОЕ | БУКВЕННОЕ | СТРОЧНЫЕ | ПРОПИСНЫЕ | имя-класса-1 }

2.13.3. Условие имени-условия

имя-условия-1

2.13.4. Условие состояния переключателя

имя-условия-1

GOST EXPERT ГОСТ 22558-89,

2.13.5. Условие знака

арифметическое-выражение-1 [НЕ] { ПОЛОЖИТЕЛЬНО / ОТРИЦАТЕЛЬНО / НУЛЬ }

2.13.6. Отрицание условия

НЕ условие-1

2.13.7. Комбинированное условие

условие-1 { { И / ИЛИ } условие-2 } ...

2.13.8. Сокращенное комбинированное условие отношения

условие-отношения { { И / ИЛИ } [НЕ] [знак-операции-отношения] объект } ...

2.14. ***Общий формат уточнения***

Формат 1

{ имя-данного-1 / имя-условия-1 } { {ИЗ имя-данного-2} ... [ИЗ { имя-файла-1 / имя-коммуникации-1 }] / ИЗ { имя-файла-1 / имя-коммуникации-1 } }

Формат 2

имя-параграфа-1 ИЗ имя-секции-1

Формат 3

имя-текста-1 ИЗ имя-библиотеки-1

Формат 4

СЧЕТЧИК-ВЕРСТКИ ИЗ имя-файла-2

Формат 5

{ СЧЕТЧИК-СТРАНИЦ / СЧЕТЧИК-СТРОК } ИЗ имя-отчета-1

GOST EXPERT ГОСТ 22558-89,

Формат 6

имя-данного-3 { ИЗ имя-данного-4 [ИЗ имя-отсчета-2] / ИЗ имя-отчета-2 }

2.15. Прочие форматы

2.15.1. Индексирование

{ имя-данного-1 / имя-условия-1 } ({ целое-1 / имя-данного-2 [{±} целое-2] / имя-индекса-1]{±} целое-3] } ...)

2.15.2. Модификация ссылки

имя-данного-1 (позиция-самой-левой-литеры: [длина])

2.15.3. Идентификатор

имя-данного-1 [ИЗ имя-данного-2] ...

[ИЗ { имя-коммуникации-1 / имя-файла-1 / имя-отчета-1 }]

[({индекс} ...)] [(позиция-самой-левой-литеры: [длина])

2.16. Общий формат вложенных исходных программ

РАЗДЕЛ ИДЕНТИФИКАЦИИ.

ПРОГРАММА. имя-программы-1 [НАЧАЛЬНАЯ].

[РАЗДЕЛ ОБОРУДОВАНИЯ.
содержимое-раздела-оборудования]

[РАЗДЕЛ ДАННЫХ. содержимое-раздела-данных]

[РАЗДЕЛ ПРОЦЕДУР. содержимое-раздела-процедур]

[[вложенная-исходная-программа] ...

КОНЕЦ ПРОГРАММЫ имя-программы-1.]

2.17. Общий формат вложенной-исходной-программы

РАЗДЕЛ ИДЕНТИФИКАЦИИ.

ПРОГРАММА. имя-программы-2 [{ ОБЩАЯ / НАЧАЛЬНАЯ }].

[РАЗДЕЛ ОБОРУДОВАНИЯ.
содержимое-раздела-оборудования]

GOST EXPERT ГОСТ 22558-89,

[РАЗДЕЛ ДАННЫХ. содержимое-раздела-данных]

[РАЗДЕЛ ПРОЦЕДУР. содержимое-раздела-процедур]

[вложенная-исходная-программа] . . .

КОНЕЦ ПРОГРАММЫ имя-программы-2.

2.18. **Общий формат последовательности исходных программ**

{РАЗДЕЛ ИДЕНТИФИКАЦИИ.

ПРОГРАММА. имя-программы-3 [НАЧАЛЬНАЯ].

[РАЗДЕЛ ОБОРУДОВАНИЯ.
содержимое-раздела-оборудования]

[РАЗДЕЛ ДАННЫХ. содержимое-раздела-данных]

[РАЗДЕЛ ПРОЦЕДУР. содержимое-раздела-процедур]

[вложенная-исходная-программа] . . .

КОНЕЦ ПРОГРАММЫ имя-программы-3.} . . .

РАЗДЕЛ ИДЕНТИФИКАЦИИ.

ПРОГРАММА. имя-программы-4 [НАЧАЛЬНАЯ].

[РАЗДЕЛ ОБОРУДОВАНИЯ.
содержимое-раздела-оборудования]

[РАЗДЕЛ ДАННЫХ. содержимое-раздела-данных]

[РАЗДЕЛ ПРОЦЕДУР. содержимое-раздела-процедур]

[[вложенная-исходная-программа] . . .

КОНЕЦ ПРОГРАММЫ имя-программы-4.]

Часть 6. ЯДРО

1. ВВЕДЕНИЕ В МОДУЛЬ ЯДРА

1.1. **Назначение**

Ядро включает средства для внутренней обработки данных в рамках структуры четырех разделов программы на Коболе. Ядро также включает средства определения таблиц последовательных элементов данных и доступа к элементу по его относительной позиции в таблице. Ядро включает возможности отладки, состоящие из переключателя, действующего во время компиляции, и отладочных строк.

GOST EXPERT ГОСТ 22558-89,

1.2. **Характеристика уровней**

Уровень 1 ядра предоставляет ограниченные возможности параграфа SPECIAL-NAMES (СПЕЦИАЛЬНЫЕ-ИМЕНА) и статьи описания данных. В разделе процедур уровень 1 ядра предоставляет ограниченные возможности операторов ACCEPT (ПРИНЯТЬ), ADD (СЛОЖИТЬ), ALTER (ИЗМЕНИТЬ), DISPLAY (ВЫДАТЬ), DIVIDE (РАЗДЕЛИТЬ), IF (ЕСЛИ), MOVE (ПОМЕСТИТЬ), MULTIPLY (УМНОЖИТЬ), PERFORM (ВЫПОЛНИТЬ) и SUBTRACT (ОТНЯТЬ) и полные возможности операторов CONTINUE (ПРОДОЛЖИТЬ), ENTER (ВОЙТИ), EXIT (ВЫЙТИ), GO TO (ПЕРЕЙТИ) и STOP (ОСТАНОВИТЬ). Уровень 1 ядра не обеспечивает всех возможностей Кобола для уточнений, правил образования имен данных и стандартных констант. Уровень 1 ядра предоставляет возможности доступа к элементам трехмерных таблиц фиксированной длины. Уровень 1 ядра предоставляет возможности отладки, состоящие из переключателя, действующего во время компиляции, и отладочных строк.

Уровень 2 ядра предоставляет полные возможности параграфа SPECIAL-NAMES (СПЕЦИАЛЬНЫЕ-ИМЕНА) и статьи описания данных. В разделе процедур уровень 2 предоставляет полные возможности операторов ACCEPT (ПРИНЯТЬ), ADD (СЛОЖИТЬ), ALTER (ИЗМЕНИТЬ), COMPUTE (ВЫЧИСЛИТЬ), DISPLAY (ВЫДАТЬ), DIVIDE (РАЗДЕЛИТЬ), EVALUATE (ОЦЕНИТЬ), IF (ЕСЛИ), INITIALIZE (ИНИЦИИРОВАТЬ), INSPECT (ПРОСМОТРЕТЬ), MOVE (ПОМЕСТИТЬ), MULTIPLY (УМНОЖИТЬ), PERFORM (ВЫПОЛНИТЬ), SEARCH (ИСКАТЬ), SET (УСТАНОВИТЬ), STRING (СОБРАТЬ), SUBTRACT (ОТНЯТЬ), UNSTRING (РАЗОБРАТЬ). Уровень 2 ядра предоставляет полные возможности уточнений, правил образования имен данных и стандартных констант. Уровень 2 ядра предоставляет возможности доступа к элементам таблиц до семи измерений.

1.3. **Ограничения по уровням, касающиеся общих характеристик языка**

1.3.1. Набор литер

Литера Кобола двоеточие (:) не включена в уровень 1.

Литера Кобола двоеточие (:) допустима на уровне 2.

1.3.2. Представление имен

Уточнение не допускается на уровне 1.

На уровне 1 все определяемые пользователем слова, к которым имеется обращение, за исключением номеров уровней и номеров сегментов, должны быть уникальны. На уровне 2 допускается 50 уточнителей. На этом уровне определяемые пользователем слова могут быть не уникальны.

GOST EXPERT ГОСТ 22558-89,

1.3.3. Стандартные константы

На уровне 1 могут быть использованы следующие стандартные константы: ZERO (НУЛЬ), ZEROS (НУЛИ), ZEROES (НУЛИ), SPACE (ПРОБЕЛ), SPACES (ПРОБЕЛЫ), HIGH-VALUE (НАИБОЛЬШЕЕ-ЗНАЧЕНИЕ), HIGH-VALUES (НАИБОЛЬШИЕ-ЗНАЧЕНИЯ), LOW-VALUE (НАИМЕНЬШЕЕ-ЗНАЧЕНИЕ), LOW-VALUES (НАИМЕНЬШИЕ-ЗНАЧЕНИЯ), QUOTE (КАВЫЧКА) и QUOTES (КАВЫЧКИ).

На уровне 2 могут быть использованы следующие стандартные константы: ZERO (НУЛЬ), ZEROS (НУЛИ), ZEROES (НУЛИ), SPACE (ПРОБЕЛ), SPACES (ПРОБЕЛЫ), HIGH-VALUE (НАИБОЛЬШЕЕ-ЗНАЧЕНИЕ), HIGH-VALUES (НАИБОЛЬШИЕ-ЗНАЧЕНИЯ), LOW-VALUE (НАИМЕНЬШЕЕ-ЗНАЧЕНИЕ), LOW-VALUES (НАИМЕНЬШИЕ-ЗНАЧЕНИЯ), символическая литера, ALL (ВСЕ) литерал, ALL (ВСЕ) стандартная константа и ALL (ВСЕ) символическая литера, QUOTE (КАВЫЧКА), QUOTES (КАВЫЧКИ).

1.3.4. Индексы

На уровне 1 допускаются один, два или три индекса.

На уровне 2 допускаются от одного до семи индексов.

1.3.5. Модификация ссылки

Модификация ссылки допускается только на уровне 2.

1.3.6. Формат представления

На уровне 1 слово, числовой литерал или строка-литер шаблона не могут разбиваться для переноса их части на следующую строку.

На уровне 2 слово, числовой литерал или строка-литер шаблона могут быть разбиты для переноса на следующую строку.

2. ИСХОДНАЯ ПРОГРАММА НА КОБОЛЕ

2.1. Общее описание

Исходная программа на Коболе представляет собой набор синтаксически правильных операторов Кобола.

2.2. Организация

За исключением операторов COPY (КОПИРОВАТЬ), REPLACE (ЗАМЕНИТЬ) и заголовка конца программы, операторы, статьи, параграфы и секции исходной Кобол-программы делятся на 4 раздела, расположенных в следующем порядке:

(1) раздел идентификации;
(2) раздел оборудования;
(3) раздел данных;
(4) раздел процедур.

GOST EXPERT ГОСТ 22558-89,

Конец исходной Кобол-программы обозначается либо заголовком конца программы, если он специфицирован, либо отсутствием строк исходной программы.

2.3. **Структура**

Далее дается общий формат и порядок следования статей и операторов, составляющих исходную Кобол-программу.

2.3.1. Общий формат

раздел-идентификации
[раздел-оборудования]
[раздел-данных]
[раздел-процедур]
[заголовок-конца-программы]

2.3.2. Синтаксические правила

(1) Общие термины раздел-идентификации, раздел-оборудования, раздел-данных, раздел-процедур и заголовок-конца-программы означают соответственно раздел идентификации Кобола, раздел оборудования Кобола, раздел данных Кобола, раздел процедур Кобола и заголовок конца программы Кобола.

2.3.3. Общие правила

(1) Начало раздела в программе идентифицируется соответственно заголовком раздела. Конец раздела идентифицируется:

а) заголовком следующего раздела программы;

б) заголовком конца программы;

в) такой физической позицией, после которой нет больше строк исходной программы.

(2) В последовательности программ все отдельно компилируемые программы должны быть ограничены заголовком конца программы за исключением последней программы в последовательности.

2.4. **Заголовок конца программы**

2.4.1. Назначение

Заголовок конца программы идентифицирует конец именованной исходной Кобол-программы.

2.4.2. Общий формат

END PROGRAM имя-программы.
КОНЕЦ ПРОГРАММЫ имя-программы.

2.4.3. Синтаксические правила

(1) Имя-программы должно формироваться по правилам образования слов, определенных пользователем.

(2) Имя-программы должно быть идентично имени программы, названному в предшествующем параграфе PROGRAM-ID (ПРОГРАММА) (п. 3.3 настоящей части).

GOST EXPERT ГОСТ 22558-89,

2.4.4. Общие правила

(1) Заголовок конца программы идентифицирует конец указанной исходной Кобол-программы.

(2) Следующим оператором после заголовка конца программы может быть только заголовок раздела идентификации программы, компилируемой отдельно от программы, ограниченной заголовком конца программы.

3. РАЗДЕЛ ИДЕНТИФИКАЦИИ В ЯДРЕ

3.1. Общее описание

Раздел идентификации идентифицирует программу. Раздел идентификации обязателен в каждой исходной Кобол-программе. К тому же пользователь может включать дату написания программы и другую требуемую информацию в параграфы, общий формат которых показан ниже.

3.2. Организация

Заголовки параграфов идентифицируют тип информации, содержащейся в параграфе. Имя программы должно быть дано в первом параграфе, которым является параграф PROGRAM-ID (ПРОГРАММА). Другие параграфы необязательны и могут быть включены в раздел по желанию пользователя в порядке, указанном в общем формате.

Параграф AUTHOR (АВТОР), параграф INSTALLATION (ПРЕДПРИЯТИЕ), параграф DATE-WRITTEN (ДАТА-НАПИСАНИЯ), параграф DATE-COMPILED (ДАТА-ТРАНСЛЯЦИИ) и параграф SECURITY (ПОЛНОМОЧИЯ) рассматриваются в настоящем стандарте как устаревшие и будут удалены в следующей редакции стандарта.

3.2.1. Структура

Далее следует общий формат параграфов раздела идентификации, определяющий порядок их следования в исходной программе.

В пп. 3.3 и 3.4 настоящей части определяются параграф PROGRAM-ID (ПРОГРАММА) и параграф DATE-COMPILED (ДАТА-ТРАНСЛЯЦИИ). Определения остальных параграфов не приводятся, так как они аналогичны.

3.2.1.1. *Общий формат*

IDENTIFICATION DIVISION.
PROGRAM-ID. имя-программы.
[AUTHOR. [статья-комментарий] . . .]
[INSTALLATION. [статья-комментарий] . . .]
[DATE-WRITTEN. [статья-комментарий] . . .]
[DATE-COMPILED. [статья-комментарий] . . .]

GOST ГОСТ 22558-89,

[SECURITY. [статья-комментарий] . . .]

РАЗДЕЛ ИДЕНТИФИКАЦИИ.

ПРОГРАММА. имя-программы.

[АВТОР. [статья-комментарий] . . .]

[ПРЕДПРИЯТИЕ. [статья-комментарий] . . .]

[DATA-НАПИСАНИЯ. [статья-комментарий] . . .]

[ДАТА-ТРАНСЛЯЦИИ. [статья-комментарий] . . .]

[ПОЛНОМОЧИЯ. [статья-комментарий] . . .]

3.2.1.2. *Синтаксическое правило*

(1) Статья-комментарий может быть любой комбинацией литер из набора литер машины. Перенос слов статьи-комментария посредством использования дефиса в поле индикатора не допускается, хотя статья-комментарий может располагаться на одной или нескольких строках.

3.3. **Параграф PROGRAM-ID (ПРОГРАММА)**

3.3.1. Назначение

Параграф PROGRAM-ID (ПРОГРАММА) указывает имя, с помощью которого идентифицируется программа.

3.3.2. Общий формат

PROGRAM-ID. имя-программы.

ПРОГРАММА. имя-программы.

3.3.3. Синтаксическое правило

(1) Имя-программы должно формироваться по правилам образования слов, определенных пользователем.

3.3.4. Общее правило

(1) Имя-программы идентифицирует исходную программу, объектную программу и все относящиеся к ней выдачи.

3.4. **Параграф DATE-COMPILED (ДАТА-ТРАНСЛЯЦИИ)**

3.4.1. Назначение

Параграф DATE-COMPILED (ДАТА-ТРАНСЛЯЦИИ) обеспечивает включение даты компиляции в выдачи раздела идентификации исходной программы. Параграф DATE-COMPILED (ДАТА-ТРАНСЛЯЦИИ) рассматривается в настоящем стандарте как устаревший элемент и будет удален в следующей редакции стандарта.

GOST EXPERT ГОСТ 22558-89,

3.4.2. Общий формат

DATE-COMPILED. [статья-комментарий] ...

ДАТА-ТРАНСЛЯЦИИ. [статья-комментарий] ...

3.4.3. Синтаксическое правило

(1) Статья-комментарий может представлять любую комбинацию литер из допустимого набора литер. Перенос слов статьи-комментария с помощью дефиса в поле индикатора не разрешается, однако статья-комментарий может занимать более одной строки.

3.4.4. Общее правило

(1) Имя параграфа DATE-COMPILED (ДАТА-ТРАНСЛЯЦИИ) вызывает во время компиляции программы включение текущей даты в выдаваемый текст. Если параграф DATE-COMPILED (ДАТА-ТРАНСЛЯЦИИ) указан, он заменяется во время компиляции параграфом следующей формы:

DATE-COMPILED. текущая-дата.

ДАТА-ТРАНСЛЯЦИИ. текущая-дата.

4. РАЗДЕЛ ОБОРУДОВАНИЯ В ЯДРЕ

4.1. Общее описание

Раздел оборудования определяет стандартный метод описания таких аспектов обработки данных, которые зависят от физических характеристик конкретной машины. Раздел оборудования не обязателен в исходной Кобол-программе.

4.2. Секция конфигурации

Секция конфигурации располагается в разделе оборудования исходной программы. Секция конфигурации связана с характеристиками исходной машины и рабочей машины. В этой секции также обеспечиваются средства спецификации валютного знака; выбора символа для десятичной точки; спецификации специальных символических литер; установления соответствия между мнемоническими именами, задаваемыми пользователем, и именами реализации; установления соответствия имени алфавита набору литер и (или) основной последовательности; установление соответствия имени класса набору литер. Секция конфигурации не обязательна в разделе оборудования исходной Кобол-программы.

Общий формат секции конфигурации приводится ниже.

CONFIGURATION SECTION.

[SOURCE-COMPUTER. [статья-исходной-машины]]

[OBJECT-COMPUTER. [статья-объектной-машины]]

GOST EXPERT ГОСТ 22558-89,

[SPECIAL-NAMES. [статья-специальных-имен]]

СЕКЦИЯ КОНФИГУРАЦИИ.

[ИСХОДНАЯ-МАШИНА. [статья-исходной-машины]]

[РАБОЧАЯ-МАШИНА. [статья-объектной-машины]]

[СПЕЦИАЛЬНЫЕ-ИМЕНА. [статья-специальных-имен]]

Секция конфигурации не должна содержаться в программе, которая косвенно или непосредственно содержится в другой программе.

Статьи, явно или неявно присутствующие в секции конфигурации программы, включающей другие программы, относятся к каждой содержащейся программе.

4.3. Параграф SOURCE-COMPUTER (ИСХОДНАЯ-МАШИНА)

4.3.1. Назначение

Параграф SOURCE-COMPUTER (ИСХОДНАЯ-МАШИНА) описывает вычислительную машину, на которой должна компилироваться программа.

4.3.2. Общий формат

SOURCE-COMPUTER. [имя-машины [WITH DEBUGGING MODE].]

ИСХОДНАЯ-МАШИНА. [имя-машины [В РЕЖИМЕ ОТЛАДКИ].]

4.3.3. Синтаксическое правило

Имя-машины является системным именем.

4.3.4. Общие правила

(1) Все фразы, явно или неявно указанные в параграфе SOURCE-COMPUTER (ИСХОДНАЯ-МАШИНА), применимы как в программе, в которой они указаны, так и в любой содержащейся в ней программе.

(2) Если параграф SOURCE-COMPUTER (ИСХОДНАЯ-МАШИНА) не определен и программа не содержится в другой программе, содержащей такой параграф, то машина, на которой будет компилироваться исходная программа, является исходной машиной.

(3) Если параграф SOURCE-COMPUTER (ИСХОДНАЯ-МАШИНА) определен, но фраза, указывающая машину, не определена, то машина, на которой будет компилироваться исходная программа, будет исходной машиной.

(4) Если в программе указана фраза WITH DEBUGGING MODE (В РЕЖИМЕ ОТЛАДКИ), то все отладочные строки компилируются так, как это определено в настоящем документе для ядра (п. 7.1 настоящей части).

GOST EXPERT ГОСТ 22558-89,

(5) Если в программе не указана фраза WITH DEBUGGING MODE (В РЕЖИМЕ ОТЛАДКИ) и программа не включается в другую программу, содержащую такую фразу, то отладочные строки компилируются как комментарии.

4.4. Параграф OBJECT-COMPUTER (РАБОЧАЯ-МАШИНА)

4.4.1. Назначение

Параграф OBJECT-COMPUTER (РАБОЧАЯ-МАШИНА) обеспечивает средства описания машины, на которой программа будет выполняться. Фраза MEMORY SIZE (РАЗМЕР ПАМЯТИ) рассматривается в настоящем стандарте как устаревший элемент стандарта и будет удалена в последующей редакции стандарта.

4.4.2. Общий формат

OBJECT-COMPUTER. [имя-машины

[MEMORY SIZE целое-1 { WORDS / CHARACTERS / MODULES }]

[PROGRAM COLLATING SEQUENCE IS имя-алфавита-1].]

РАБОЧАЯ-МАШИНА. [имя-машины

[РАЗМЕР ПАМЯТИ целое-1 { СЛОВ / ЛИТЕР / МОДУЛЕЙ }]

[ПРОГРАММНЫЙ АЛФАВИТ имя-алфавита-1].]

4.4.3. Синтаксические правила

(1) Имя-машины является системным именем.

4.4.4. Общие правила

(1) Имя-машины может служить для определения конфигурации оборудования; в этом случае имя-машины и ее подразумеваемая конфигурация задаются каждой реализацией. Определение конфигурации содержит специальную информацию, касающуюся размера памяти.

Если подмножество, задаваемое пользователем, меньше минимальной конфигурации, требуемой для выполнения объектной программы, необходимые меры определяются реализацией.

(2) Все фразы параграфа OBJECT-COMPUTER (РАБОЧАЯ-МАШИНА) применяются к программе, в которой они явно или неявно определены, и к любой содержащейся в ней программе.

(3) При отсутствии параграфа OBJECT-COMPUTER (РАБОЧАЯ-МАШИНА) в программе и во всех программах, включающих данную программу, рабочая машина определяется реализацией.

(4) Если параграф OBJECT-COMPUTER (РАБОЧАЯ-МАШИНА) указан, но фраза, определяющая рабочую машину, не указана, рабочая машина определяется реализацией.

GOST EXPERT ГОСТ 22558-89,

(5) При наличии фразы PROGRAM COLLATING SEQUENCE (ПРОГРАММНЫЙ АЛФАВИТ) в программе используется основная последовательность, заданная именем-алфавита, указанным в этой фразе.

(6) При отсутствии фразы PROGRAM COLLATING SEQUENCE (ПРОГРАММНЫЙ АЛФАВИТ) используется внутренняя основная последовательность.

(7) Если задана фраза PROGRAM COLLATING SEQUENCE (ПРОГРАММНЫЙ АЛФАВИТ), то для определения значения истинности нечисловых сравнений, явно указанных в условиях отношения (п. 6.3.1.1 настоящей части), явно указанных в условиях имени-условия (п. 6.3.1.3 настоящей части) или неявно указанных наличием фразы CONTROL (УПРАВЛЕНИЕ) в статье описания отчета (ч. 13, п. 3.7), используется основная последовательность, определяемая именем алфавита.

(8) Программная основная последовательность, определенная в параграфе OBJECT-COMPUTER (РАБОЧАЯ-МАШИНА), применяется к любым нечисловым ключам сортировки или слияния, если в соответствующих операторах SORT (СОРТИРОВАТЬ) или MERGE (СЛИТЬ) не задана фраза COLLATING SEQUENCE (АЛФАВИТ) (ч. 11, пп. 4.1, 4.4).

4.5. Параграф SPECIAL-NAMES (СПЕЦИАЛЬНЫЕ-ИМЕНА)

4.5.1. Назначение

Параграф SPECIAL-NAMES (СПЕЦИАЛЬНЫЕ-ИМЕНА) обеспечивает средства спецификации валютного знака; выбора символа для десятичной точки; спецификации специальных символических литер; установления соответствия между мнемоническими именами, задаваемыми пользователем, и именами реализации; установления соответствия имени алфавита набору литер или основной последовательности; установления соответствия имени класса набору литер.

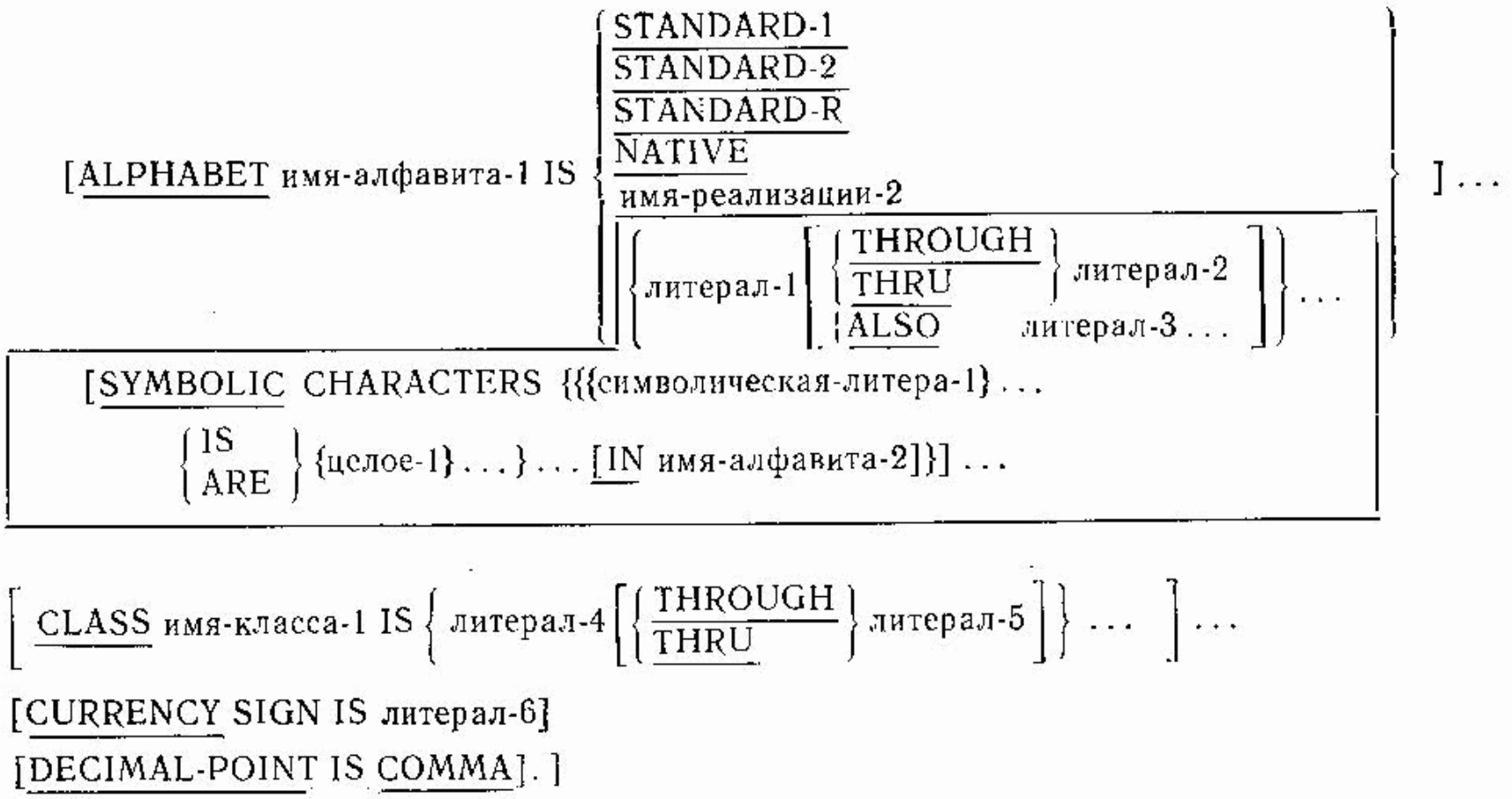
4.5.2. Общий формат

SPECIAL-NAMES. [[имя-реализации-1

{ IS мнемоническое-имя-1 [ON STATUS IS имя-условия-1 [OFF STATUS IS имя-условия-2]]
IS мнемоническое-имя-2 [OFF STATUS IS имя-условия-2 [ON STATUS IS имя-условия-1]]
ON STATUS IS имя-условия-1 [OFF STATUS IS имя-условия-2]
OFF STATUS IS имя-условия-2 [ON STATUS IS имя-условия-1] } ...

GOST EXPERT ГОСТ 22558-89,

ГОСТ 22558-89,



СПЕЦИАЛЬНЫЕ-ИМЕНА. [[имя-реализации-1

{ ЕСТЬ мнемоническое-имя-1 [{ ВКЛЮЧЕНО / ВКЛ } имя-условия-1
[{ ВЫКЛЮЧЕНО / ВЫКЛ } имя-условия-2]]
ЕСТЬ мнемоническое-имя-2 [{ ВЫКЛЮЧЕНО / ВЫКЛ } имя-условия-2
[{ ВКЛЮЧЕНО / ВКЛ } имя-условия-1]]
{ ВКЛЮЧЕНО / ВКЛ } имя-условия-1 [{ ВЫКЛЮЧЕНО / ВЫКЛ } имя-условия-2]
{ БЫКЛЮЧЕНО / ВЫКЛ } имя-условия-2 [{ ВКЛЮЧЕНО / ВКЛ } имя-условия-1] }]...

GOST ГОСТ 22558-89,

4.5.3. Синтаксические правила

(1) Если имя-реализации-1 относится к внешнему переключателю, то соответствующее мнемоническое имя может быть указано только в операторе SET (УСТАНОВИТЬ).

(2) Если имя-реализации-1 не относится к внешнему переключателю, то соответствующее мнемоническое имя может быть указано только в операторах ACCEPT (ПРИНЯТЬ), DISPLAY (ВЫДАТЬ), SEND (ПОСЛАТЬ), WRITE (ПИСАТЬ).

(3) Во фразе имя-алфавита одно и то же значение литерала может указываться только один раз.

(4) Если во фразе имя-алфавита указаны литералы, то:

а) числовые литералы должны быть целыми без знака и иметь значения от 1 до целого, равного количеству литер во внутреннем наборе литер;

б) нечисловые литералы, указанные после слов THROUGH (ПО) и ALSO (ТАКЖЕ), должны состоять из одной литеры.

(5) Литерал-1, литерал-2, литерал-3, литерал-4 и литерал-5 не должны совпадать по написанию со стандартными константами, являющимися символическими литерами.

(6) Слова THROUGH и THRU эквивалентны.

(7) Во фразе SYMBOLIC CHARACTERS (СИМВОЛИЧЕСКАЯ ЛИТЕРА) одна и та же литера может встречаться только один раз.

(8) Соответствие между символической-литерой-1 и целым-1 определяется позицией во фразе SYMBOLIC CHARACTERS (СИМВОЛИЧЕСКАЯ ЛИТЕРА). Первая символическая литера-1 соответствует первому целому-1, вторая — второму и так далее.

(9) Между экземплярами символического-имени-1 и экземплярами целого-1 должно быть взаимно однозначное соответствие.

(10) Порядковая позиция, определяемая целым-1, должна существовать во внутреннем наборе кодов литер. Если указана фраза IN (ИЗ), то эта порядковая позиция должна существовать в наборе литер, определяемом именем-алфавита-2.

(11) Правила для литералов фразы литерал-4 следующие:

а) числовые литералы должны быть целыми без знака и иметь значения от 1 до целого, равного количеству литер во внутреннем наборе литер;

б) нечисловые литералы, указанные после слова THROUGH (ПО), должны состоять из одной литеры.

(12) Литерал не должен быть стандартной константой.

4.5.4. Общие правила

(1) Все фразы, указанные в параграфе SPECIAL-NAMES

GOST EXPERT ГОСТ 22558-89,

(СПЕЦИАЛЬНЫЕ-ИМЕНА) некоторой программы, относятся ко всем содержащимся в ней программам. На определенные в данном параграфе имена условий можно ссылаться из любой включенной в эту программу программы.

(2) Если имя-реализации-1 относится к внешнему переключателю, то состояния переключателя должны быть связаны с именами условий и опрашиваются посредством проверки имен условий (п. 6.3.1.4 настоящей части).

(3) Если имя-реализации-1 относится к внешнему переключателю, то состояние этого переключателя может быть изменено оператором SET (УСТАНОВИТЬ) формата 3, в качестве операнда которого указывается мнемоническое имя, связанное с этим переключателем (п. 6.2.3 настоящей части). Разработчик определяет, какие внешние переключатели можно использовать в операторе SET (УСТАНОВИТЬ).

(4) Указание имени-алфавита-1 обеспечивает средства присвоения некоторого имени заданному набору кодов литер и (или) основной последовательности. Если указанное имя-алфавита-1 используется во фразе PROGRAM COLLATING SEQUENCE (ПРОГРАММНЫЙ АЛФАВИТ) (см. п. 4.4 настоящей части) или во фразе COLLATING SEQUENCE (АЛФАВИТ) оператора SORT (СОРТИРОВАТЬ) или MERGE (СЛИТЬ) (ч. 11, пп. 4.1, 4.4), фраза имя-алфавита задает основную последовательность. Если имя-алфавита-1 используется во фразе SYMBOLIC CHARACTERS (СИМВОЛИЧЕСКАЯ ЛИТЕРА) или во фразе CODE-SET (АЛФАВИТ) в описании файла (ч. 7, п. 3.4) то фраза ALPHABET (имя-алфавита) определяет набор кодов литер:

а) если задана фраза STANDARD-R (СТАНДАРТ-Р), набор кодов литер или соответствующая основная последовательность определяется в соответствии с упорядоченностью русского алфавита.

Если задана фраза STANDARD-1 (СТАНДАРТ-А), набор кодов литер или соответствующая основная последовательность определяется Американским Национальным Стандартным Кодом для Обмена Информацией, X3.4—1968.

Если задана фраза STANDARD-2 (СТАНДАРТ-М), то набор кодов определяется Международной версией 7-битового кода, определенной в Международном стандарте 646, 7-битовый Код литер для обмена информацией.

Если между отдельными литерами стандартного и внутреннего наборов отсутствует соответствие, это соответствие устанавливает реализация;

б) если задана фраза NATIVE (ВНУТРЕННИЙ), используется внутренний набор кодов литер или внутренняя основная последовательность;

GOST EXPERT ГОСТ 22558-89,

в) если задана фраза имя-реализации-2, набор кодов литер или соответствующая основная последовательность, а также их соответствие внутренней основной последовательности определяются реализацией. Реализация также определяет соответствие между литерами или литерой набора кодов литер, специфицированного фразой имя-реализации-2, и литерами внутреннего набора кодов литер;

г) если фраза имя-алфавита содержит литералы, соответствующее имя-алфавита не может упоминаться во фразе CODE-SET (АЛФАВИТ) (ч. 7, п. 3.4), при этом основная последовательность определяется следующим образом:

значение каждого числового литерала определяет порядковый номер литеры во внутреннем наборе литер; это значение не должно превосходить количество литер во внутреннем наборе литер;

значение каждого нечислового литерала определяет явную литеру во внутреннем наборе литер. Если нечисловой литерал содержит несколько литер, каждая его литера, начиная с самой левой, ставится в соответствие последовательным возрастающим позициям в определяемой основной последовательности;

порядок, в котором литералы появляются во фразе имя-алфавита, точно определяет используемую основную последовательность (в возрастающем порядке);

не указанные явно литералами литеры внутренней основной последовательности занимают позиции в основной последовательности, следующие за явно заданными во фразе имя-алфавита литерами. Относительный порядок литер, не указанных в определяемой основной последовательности, соответствует внутренней основной последовательности;

если задана фраза THROUGH (ПО), последовательность литер внутреннего набора, начиная от литеры, заданной значением литерала-1, и кончая литерой, заданной значением литерала-2, занимает последовательные позиции определяемой основной последовательности. Кроме того, последовательность литер, заданных фразой THRU (ПО), может задавать литеры внутреннего набора либо в возрастающем, либо в убывающем порядке;

если указана фраза ALSO (ТАКЖЕ), литеры внутреннего набора, заданные значением литерала-1, литерала-3, литерала-4 . . . , ставятся в соответствие одной и той же позиции определяемой основной последовательности или набора кодов литер, используемых для представления данных, и если имя-алфавита-1 используется во фразе SYMBOLIC CHARACTERS (СИМВОЛИЧЕСКАЯ ЛИТЕРА), то только литерал-1 используется для представления литеры во внутреннем наборе литер.

GOST EXPERT ГОСТ 22558-89,

(5) Литера, которая занимает самую старшую позицию в задаваемой программной основной последовательности, ставится в соответствие стандартной константе HIGH-VALUE (НАИБОЛЬШЕЕ-ЗНАЧЕНИЕ), кроме случая, когда эта последняя стандартная константа используется в качестве литерала в параграфе SPECIAL-NAMES (СПЕЦИАЛЬНЫЕ-ИМЕНА).

Если таких литер несколько, то стандартной константе HIGH-VALUE (НАИБОЛЬШЕЕ-ЗНАЧЕНИЕ) ставится в соответствие литера, указанная последней из них.

(6) Литера, которая занимает самую младшую позицию в задаваемой программной основной последовательности, ставится в соответствие стандартной константе LOW-VALUE (НАИМЕНЬШЕЕ-ЗНАЧЕНИЕ), кроме случая, когда эта стандартная константа используется в качестве литерала в параграфе SPECIAL-NAMES (СПЕЦИАЛЬНЫЕ-ИМЕНА). Если таких литер несколько, стандартной константе LOW-VALUE (НАИМЕНЬШЕЕ-ЗНАЧЕНИЕ) ставится в соответствие литера, указанная первой из них.

(7) Стандартные константы HIGH-VALUE (НАИБОЛЬШЕЕ-ЗНАЧЕНИЕ) и LOW-VALUE (НАИМЕНЬШЕЕ-ЗНАЧЕНИЕ), определенные как литералы в параграфе SPECIAL-NAMES (СПЕЦИАЛЬНЫЕ-ИМЕНА), соответствуют литерам, имеющим наибольшую и наименьшую позиции во внутренней основной последовательности.

(8) Если не указана фраза IN (ИЗ), то символическая-литера-1 представляет литеру, чья порядковая позиция во внутреннем наборе определяется целым-1. Если фраза IN (ИЗ) указана, то целое-1 определяет порядковую позицию литеры, находящейся в наборе литер, названном именем-алфавита-2.

(9) Внутреннее представление символической-литеры-1 является внутренним представлением литеры внутреннего набора.

(10) Фраза CLASS (КЛАСС) обеспечивает соответствие имени, указанному во фразе набору литер. На имя-класса-1 можно ссылаться только в условии класса. Литеры, указанные значениями литералов в этой фразе, определяют набор литер, из которых состоит значение данного, принадлежащего к этому классу.

Значение каждого литерала специфицирует

порядковый номер литеры во внутреннем наборе, если литерал числовой. Его значение не должно превышать числа литер во внутреннем наборе;

действительное значение литеры во внутреннем наборе, если литерал нечисловой. Если значение нечислового литерала состоит из нескольких литер, то каждая литера литерала включается в набор литер, идентифицируемый именем-класса-1.

GOST EXPERT ГОСТ 22558-89,

Если фраза THROUGH (ПО) указана, то все литеры внутреннего набора, начиная с литеры, указанной значением литерала-4, и кончая литерой, указанной значением литерала-5, включаются в набор литер, идентифицируемый именем-класса-1. Кроме того, литеры, указанные во фразе THROUGH (ПО), могут определять литеры внутреннего набора в возрастающей или убывающей последовательности.

(11) Литерал-6, указанный во фразе CURRENCY SIGN (ВАЛЮТНЫЙ ЗНАК), используется во фразе PICTURE (ШАБЛОН) для представления валютного символа. Литерал должен представлять одну литеру и не должен совпадать ни с одной из следующих литер:

а) цифры: от 0 до 9;

б) буквы прописные A, B, C, D, P, R, S, V, X, Y, Z (А, Б, В, Д, З, Р, К, М, П, Т, Х) и строчные или пробел;

в) специальные • литеры, +, —, ,, ., ;, (,), », /, =. Если эта фраза отсутствует, во фразе PICTURE (ШАБЛОН) используется только валютный символ из набора литер Кобола.

(12) Фраза DECIMAL POINT IS COMMA (ДЕСЯТИЧНАЯ ТОЧКА ЗАПЯТАЯ) означает, что в строке-литер фразы PICTURE (ШАБЛОН) и в числовых литералах функции запятой и точки меняются местами.

5. РАЗДЕЛ ДАННЫХ В ЯДРЕ

5.1. Общее описание

Раздел данных описывает данные, которые обрабатываются объектной программой. Раздел данных не обязателен в исходной Кобол-программе.

5.2. Секция рабочей памяти

Секция рабочей памяти входит в раздел данных исходной программы. Секция рабочей памяти описывает записи и их элементы, которые не являются частью файлов данных.

Секция рабочей памяти составляется из заголовка секции, за которым следуют статьи описания записей и (или) статьи описания несвязанных данных. Общий формат секции рабочей памяти следующий:

WORKING-STORAGE SECTION.

[статья-описания-уровня-77 | статья-описания-записи] ...

СЕКЦИЯ РАБОЧЕЙ-ПАМЯТИ.

[статья-описания-уровня-77 | статья-описания-записи] ...

GOST EXPERT ГОСТ 22558-89,

5.2.1. Несвязанные данные рабочей памяти

Данные и константы в рабочей памяти, которые не имеют никакой иерархической связи друг с другом и которые в дальнейшем не потребуется подразделять, нет необходимости группировать в записи. Они классифицируются и определяются как несвязанные элементарные данные. Каждое из этих данных определяется отдельной статьей описания данного, которая начинается со специального номера уровня 77. В каждой статье описания таких данных обязательны следующие фразы: номер уровня 77, имя-данного, PICTURE (ШАБЛОН) или USAGE IS INDEX (ДЛЯ ИНДЕКСА).

Другие фразы описания данного не обязательны, но могут, при необходимости, дополнять описание данного.

5.2.2. Записи рабочей памяти

Элементы данных и константы в рабочей памяти, которые находятся в определенной иерархической связи друг с другом, должны быть сгруппированы в записи согласно правилам образования описания записи. Элементы данных, не находящиеся в иерархической связи друг с другом, могут быть описаны как записи из одного элементарного данного. Фразы, допустимые в описании записей в секции файлов, могут быть использованы для описания записей рабочей памяти.

5.2.3. Структура описания записи

Описание записи состоит из набора статей описания данных, которые описывают характеристики отдельной записи. Каждая статья описания данного состоит из номера уровня, за которым может следовать имя данного или фраза FILLER (ЗАПОЛНИТЕЛЬ), после чего следует, если это требуется, ряд независимых фраз. Описание записи может иметь иерархическую структуру, поэтому фразы, используемые в отдельных статьях, могут сильно отличаться, в зависимости от наличия подчиненных статей. Структура описания записи и разрешенные элементы в статье описания записи объясняются в ч. 4, п. 4.3.2 и в п. 5.3 настоящей части.

5.2.4. Начальные значения

Начальное значение любого данного в секции рабочей памяти, за исключением индексного данного, задается в описании данного фразой VALUE (ЗНАЧЕНИЕ). Начальное значение индексного данного и данного, не связанного с фразой VALUE (ЗНАЧЕНИЕ), не определено.

5.3. **Статья описания данного**

5.3.1. Назначение

Статья описания данного задает характеристики отдельного данного.

GOST EXPERT ГОСТ 22558-89,

5.3.2. Общий формат

Формат 1

номер-уровня [имя-данного-1 | FILLER]

[REDEFINES имя-данного-2]

[{ PICTURE | PIC } IS строка-литер]

[[USAGE IS] { BINARY | COMPUTATIONAL | COMP | DISPLAY | INDEX | PACKED-DECIMAL }]

[[SIGN IS] { LEADING | TRAILING } [SEPARATE CHARACTER]]

[OCCURS целое-2 TIMES

[{ ASCENDING | DESCENDING } KEY IS {имя-данного-3} ...] ...

[INDEXED BY {имя-индекса-1} ...]

OCCURS целое-1 TO целое-2 TIMES DEPENDING ON имя-данного-4

[{ ASCENDING | DESCENDING } KEY IS {имя-данного-3} ...] ...

[INDEXED BY {имя-индекса-1} ...]]

[{ SYNCHRONIZED | SYNC } [LEFT | RIGHT]]

[{ JUSTIFIED | JUST } RIGHT]

[BLANK WHEN ZERO]

[VALUE IS литерал-1].

GOST EXPERT ГОСТ 22558-89,

номер-уровня [имя-данного-1 | ЗАПОЛНИТЕЛЬ | ЗАП]

[ПЕРЕОПРЕДЕЛЯЕТ имя-данного-2]

[{ ШАБЛОН | Ш } строка-литер]

[ДВОИЧНОЕ | ДЕСЯТИЧНОЕ ДЛЯ { ВЫЧИСЛЕНИЙ | ВЫЧ | ВЫДАЧИ | ИНДЕКСА }]

[[ЗНАК] { ПЕРВЫЙ | ПОСЛЕДНИЙ } [ОТДЕЛЬНО]]

[ПОВТОРЯЕТСЯ целое-2 РАЗ

[[ПО { ВОЗРАСТАНИЮ | УБЫВАНИЮ } КЛЮЧА {имя-данного-3} . . .] . . .
[ИНДЕКСИРУЕТСЯ {имя-индекса-1} . . .]
ПОВТОРЯЕТСЯ ОТ целое-1 ДО целое-2 РАЗ
В ЗАВИСИМОСТИ ОТ имя-данного-4
[ПО { ВОЗРАСТАНИЮ | УБЫВАНИЮ } КЛЮЧА {имя-данного-3} . . .] . . .
[ИНДЕКСИРУЕТСЯ {имя-индекса-1} . . .]]

[ВЫДЕЛЕНО [ВЛЕВО | ВПРАВО]]

[СДВИНУТО ВПРАВО]

[ПРОБЕЛ КОГДА НУЛЬ]

GOST EXPERT ГОСТ 22558-89,

[{ЗНАЧЕНИЕ / ЗНАЧ} литерал-1].

Формат 2

66 имя-данного-1 RENAMES имя-данного-2

[{THROUGH / THRU} имя-данного-3] .

66 имя-данного-1 ПЕРЕИМЕНОВЫВАЕТ имя-данного-2

[ПО имя-данного-3].

Формат 3

88 имя-условия-1 {VALUE IS / VALUES ARE}

{ литерал-1 [{THROUGH / THRU} литерал-2]} . . . .

88 имя-условия-1 {ЗНАЧЕНИЕ / ЗНАЧ} {литерал-1 [ПО литерал-2]} . . . .

5.3.3. Синтаксические правила

(1) Номер уровня в формате 1 может быть любым числом от 01 до 49, или 77.

(2) В формате 1 имя-данного-1 или фраза FILLER (ЗАПОЛНИТЕЛЬ), если она указана, должны непосредственно следовать за номером уровня. Фраза REDEFINES (ПЕРЕОПРЕДЕЛЯЕТ), если она указана, должна непосредственно следовать за именем-данного-1 или фразой FILLER (ЗАПОЛНИТЕЛЬ), если они указаны, иначе она должна непосредственно следовать за номером уровня. Остальные фразы могут следовать в любом порядке.

(3) Фраза PICTURE (ШАБЛОН) должна быть задана для каждого элементарного данного, за исключением индексного данного, для которого употребление этой фразы запрещено.

(4) Слова THRU и THROUGH эквивалентны.

5.3.4. Общие правила

(1) Фразы SYNCHRONIZED (ВЫДЕЛЕНО), PICTURE (ШАБЛОН), JUSTIFIED (СДВИНУТО) и BLANK WHEN ZERO (ПРОБЕЛ КОГДА НУЛЬ) могут задаваться только для элементарного данного.

GOST EXPERT ГОСТ 22558-89,

(2) Формат 3 используется для имени условия. Каждое имя условия определяется отдельной статьей с номером уровня 88. В формате 3 указывается имя-условия, а также одно значение, несколько значений или диапазон значений, связанных с именем-условия. Статьи имени условия для конкретной условной переменной должны следовать за статьей, описывающей данное, к которому относится имя-условия. Имя-условия может относиться к любому элементарному или групповому данному, за исключением другого имени условия, индексного данного, данного с номером уровня 66, или группы, содержащей данные с описанием, включающим фразы JUSTIFIED (СДВИНУТО), SYNCHRONIZED (ВЫДЕЛЕНО) или фразу об использовании (кроме варианта USAGE IS DISPLAY (ДЛЯ ВЫДАЧИ)).

(3) Несколько статей с номером уровня 01, подчиненных одному и тому же заданному индикатору уровня, отличному от индикатора уровня RD (ОО) для статьи описания отчета, переопределяют одно и то же поле памяти.

5.4. Фраза BLANK WHEN ZERO (ПРОБЕЛ КОГДА НУЛЬ)

5.4.1. Назначение

Фраза BLANK WHEN ZERO (ПРОБЕЛ КОГДА НУЛЬ) позволяет заменить данное пробелом, когда его значение является нулевым.

5.4.2. Общий формат

BLANK WHEN ZERO

ПРОБЕЛ КОГДА НУЛЬ

5.4.3. Синтаксические правила

(1) Фраза BLANK WHEN ZERO (ПРОБЕЛ КОГДА НУЛЬ) может быть использована только для элементарного данного, шаблон которого определен как числовой или числовой редактируемый (п. 5.9 настоящей части).

(2) Числовые и числовые редактируемые данные, описанные с фразой BLANK WHEN ZERO (ПРОБЕЛ КОГДА НУЛЬ), должны быть явно или неявно описаны как USAGE IS DISPLAY (ДЛЯ ВЫДАЧИ).

5.4.4. Общие правила

(1) Если в описании данного указана фраза BLANK WHEN ZERO (ПРОБЕЛ КОГДА НУЛЬ), и значение этого данного равно нулю, это данное заменяется пробелами.

(2) Если в данном, имеющем числовой шаблон, использована фраза BLANK WHEN ZERO (ПРОБЕЛ КОГДА НУЛЬ), категория данного считается числовой редактируемой.

GOST EXPERT ГОСТ 22558-89,

5.5. Фраза имя-данного или FILLER (ЗАПОЛНИТЕЛЬ)

5.5.1. Назначение

Имя данного задает имя описываемого данного. Слово FILLER (ЗАПОЛНИТЕЛЬ) задает элементарное данное в логической записи, к которому явно нельзя обращаться.

5.5.2. Общий формат

```
[ имя-данного-1 ]
[ FILLER        ]

[ имя-данного-1 ]
[ ЗАПОЛНИТЕЛЬ   ]
[ ЗАП           ]
```

5.5.3. Синтаксические правила

(1) В секциях рабочей памяти, коммуникации, связи или в секции файлов имя-данного или слово FILLER (ЗАПОЛНИТЕЛЬ), если одно из них указано, должны непосредственно следовать в статье описания данного за номером уровня.

(2) ЗАП является сокращением слова ЗАПОЛНИТЕЛЬ.

5.5.4. Общие правила

(1) Если эта фраз опущена, то считается, что в описании данного указана фраза FILLER (ЗАПОЛНИТЕЛЬ).

(2) Слово FILLER (ЗАПОЛНИТЕЛЬ) может быть использовано в качестве имени элементарного данного в записи. К данному FILLER (ЗАПОЛНИТЕЛЬ) запрещено явное обращение.

Тем не менее, слово FILLER (ЗАПОЛНИТЕЛЬ) может использоваться в качестве условной переменной, что не требует явного обращения к данному, названному FILLER (ЗАПОЛНИТЕЛЬ) (а только к его значению).

5.6. Фраза JUSTIFIED (СДВИНУТО)

5.6.1. Назначение

Фраза JUSTIFIED (СДВИНУТО) задает нестандартное расположение данных в поле, отведенном для данного.

5.6.2. Общий формат

```
{ JUSTIFIED } RIGHT
{ JUST      }

СДВИНУТО ВПРАВО
```

5.6.3. Синтаксические правила

(1) Фраза JUSTIFIED (СДВИНУТО) может быть задана только на уровне элементарного данного.

(2) Фраза JUST является сокращением JUSTIFIED.

GOST EXPERT ГОСТ 22558-89,

(3) Фраза JUSTIFIED (СДВИНУТО) не может быть задана для данного, описанного как числовое, числовое редактируемое или буквенно-цифровое редактируемое.

(4) Фраза JUSTIFIED (СДВИНУТО) не может быть задана для индексного данного.

5.6.4. Общие правила

(1) Если принимающее данное описывается с фразой JUSTIFIED (СДВИНУТО) и пересылаемое данное по размеру больше принимающего, то отбрасываются самые левые литеры. Если описание принимающего данного имеет JUSTIFIED (СДВИНУТО) и его размер превышает размер пересылаемого данного, то пересылаемое значение располагается с выравниванием к самой правой позиции литеры и дополняется слева пробелами.

(2) Если фраза JUSTIFIED (СДВИНУТО) опущена, применяются стандартные правила выравнивания элементарных данных (см. ч. 4, п. 4.3.6).

5.7. **Номер уровня**

5.7.1. Назначение

Номер уровня указывает на иерархию данных внутри логической записи. Кроме того, он используется, чтобы определить статьи для имен-условий, данных рабочей памяти, данных связи, а также для указания переименований.

5.7.2. Общий формат

номер-уровня

5.7.3. Синтаксические правила

(1) Номер-уровня обязательно должен быть первым элементом в каждой статье описания данного.

(2) Статьи описания данных, подчиненных статьям FD (ОФ), CD (ОК) или SD (ОС), должны иметь номера уровней от 01 до 49, 66 или 88.

(3) Статьи описания данных в секции рабочей памяти и секции связи должны иметь номера уровней от 1 до 49, 66, 77 или 88.

5.7.4. Общие правила

(1) Номер уровня 01 определяет первую статью в каждом описании записи.

(2) Отдельным статьям, для которых понятие уровня не имеет смысла, присваиваются специальные номера-уровня:

а) номер-уровня 77 присваивается для определения несвязанных данных рабочей памяти, несвязанных данных, используемых для связи, и может быть использован только в формате 1 статьи описания данных (см. п. 5.3 настоящей части);

GOST EXPERT ГОСТ 22558-89,

б) номер-уровня 66 определяет статью переименования и может использоваться только в формате 2 статьи описания данных (см. п. 5.3 настоящей части);

в) номер-уровня 88 присваивается статьям, которые определяют имена условий, связанных с условными переменными, и может быть использован только в формате 3 статьи описания данных (см. п. 5.3 настоящей части).

(3) Несколько статей уровня 01, подчиненных одному и тому же индикатору уровня, отличному от RD (OO), неявно переопределяют одно и то же поле памяти.

5.8. **Фраза OCCURS (ПОВТОРЯЕТСЯ)**

5.8.1. Назначение

Фраза OCCURS (ПОВТОРЯЕТСЯ) исключает необходимость отдельных статей описания для повторяющихся данных и дает необходимую информацию для индексирования.

5.8.2. Общий формат

Формат 1

OCCURS целое-2 TIMES

[{ASCENDING / DESCENDING} KEY IS {имя-данного-2} ...] ...

[INDEXED BY {имя-индекса-1} ...]

ПОВТОРЯЕТСЯ целое-2 РАЗ

[ПО {ВОЗРАСТАНИЮ / УБЫВАНИЮ} КЛЮЧА {имя-данного-2} ...] ...

[ИНДЕКСИРУЕТСЯ {имя-индекса-1} ...]

Формат 2

OCCURS целое-1 TO целое-2 TIMES DEPENDING ON имя-данного-1

[{ASCENDING / DESCENDING} KEY IS {имя-данного-2} ...] ...

[INDEXED BY {имя-индекса-1} ...]

ПОВТОРЯЕТСЯ ОТ целое-1 ДО целое-2 РАЗ

В ЗАВИСИМОСТИ ОТ имя-данного-1

GOST EXPERT ГОСТ 22558-89,

[ПО {ВОЗРАСТАНИЮ / УБЫВАНИЮ} КЛЮЧА {имя-данного-2} ...] ...
[ИНДЕКСИРУЕТСЯ {имя-индекса-1} ...]

5.8.3. Синтаксические правила

(1) Фраза OCCURS (ПОВТОРЯЕТСЯ) не может быть задана в статьях описания данных с номерами уровней 01, 66, 77, 88 и в статьях описания данных, имеющих подчиненными переменно повторяющиеся данные.

(2) Имя-данного-1 имя-данного-2 могут быть уточнены.

(3) Имя-данного-2 должно быть либо именем статьи, содержащей фразу OCCURS (ПОВТОРЯЕТСЯ), либо именем статьи, подчиненной статье, содержащей фразу OCCURS (ПОВТОРЯЕТСЯ).

(4) Имя-данного-2 должно быть указано без требуемого для него индексирования.

(5) Если указаны целое-1 и целое-2, то целое-1 должно быть больше или равно 0, а целое-2 должно быть больше целого-1.

(6) Имя-данного-1 должно быть описано как целое.

(7) В формате 2 данное, определенное именем-данного-1, не должно располагаться в области памяти, занимаемой данным, статья описания которого содержит фразу OCCURS (ПОВТОРЯЕТСЯ).

(8) Если фраза OCCURS (ПОВТОРЯЕТСЯ) указана в статье описания данного, включенного в статью описания записи, содержащую фразу EXTERNAL (ВНЕШНЕЕ), то имя-данного-1, если оно указано, должно ссылаться на данное, описанное в том же разделе данных, со свойством EXTERNAL (ВНЕШНЕЕ).

(9) Если фраза OCCURS (ПОВТОРЯЕТСЯ) указана в статье описания данного, подчиненного статье, содержащей фразу GLOBAL (ГЛОБАЛЬНОЕ), то имя-данного-1, если оно указано, должно быть глобальным именем и должно ссылаться на данное, описанное в том же разделе данных.

(10) За статьей описания данного, содержащей фразу OCCURS (ПОВТОРЯЕТСЯ) формата 2, могут следовать только подчиненные ей статьи описания данных.

(11) Данное, определяемое именем-данного-2, не должно содержать фразу OCCURS (ПОВТОРЯЕТСЯ) за исключением случая, когда имя-данного-2 является субъектом статьи.

(12) Между статьями описания данных, указанными именами данных в варианте KEY IS (ПО ВОЗРАСТАНИЮ/УБЫВА-

GOST EXPERT ГОСТ 22558-89,

[ПО {ВОЗРАСТАНИЮ / УБЫВАНИЮ} КЛЮЧА {имя-данного-2} ...] ...
[ИНДЕКСИРУЕТСЯ {имя-индекса-1} ...]

5.8.3. Синтаксические правила

(1) Фраза OCCURS (ПОВТОРЯЕТСЯ) не может быть задана в статьях описания данных с номерами уровней 01, 66, 77, 88 и в статьях описания данных, имеющих подчиненными переменно повторяющиеся данные.

(2) Имя-данного-1 имя-данного-2 могут быть уточнены.

(3) Имя-данного-2 должно быть либо именем статьи, содержащей фразу OCCURS (ПОВТОРЯЕТСЯ), либо именем статьи, подчиненной статье, содержащей фразу OCCURS (ПОВТОРЯЕТСЯ).

(4) Имя-данного-2 должно быть указано без требуемого для него индексирования.

(5) Если указаны целое-1 и целое-2, то целое-1 должно быть больше или равно 0, а целое-2 должно быть больше целого-1.

(6) Имя-данного-1 должно быть описано как целое.

(7) В формате 2 данное, определенное именем-данного-1, не должно располагаться в области памяти, занимаемой данным, статья описания которого содержит фразу OCCURS (ПОВТОРЯЕТСЯ).

(8) Если фраза OCCURS (ПОВТОРЯЕТСЯ) указана в статье описания данного, включенного в статью описания записи, содержащую фразу EXTERNAL (ВНЕШНЕЕ), то имя-данного-1, если оно указано, должно ссылаться на данное, описанное в том же разделе данных, со свойством EXTERNAL (ВНЕШНЕЕ).

(9) Если фраза OCCURS (ПОВТОРЯЕТСЯ) указана в статье описания данного, подчиненного статье, содержащей фразу GLOBAL (ГЛОБАЛЬНОЕ), то имя-данного-1, если оно указано, должно быть глобальным именем и должно ссылаться на данное, описанное в том же разделе данных.

(10) За статьей описания данного, содержащей фразу OCCURS (ПОВТОРЯЕТСЯ) формата 2, могут следовать только подчиненные ей статьи описания данных.

(11) Данное, определяемое именем-данного-2, не должно содержать фразу OCCURS (ПОВТОРЯЕТСЯ) за исключением случая, когда имя-данного-2 является субъектом статьи.

(12) Между статьями описания данных, указанными именами данных в варианте KEY IS (ПО ВОЗРАСТАНИЮ/УБЫВА-

GOST EXPERT ГОСТ 22558-89,

НИЮ КЛЮЧА), и предметом этой статьи не должно быть статьи, содержащей фразу OCCURS (ПОВТОРЯЕТСЯ).

(13) Фраза INDEXED BY (ИНДЕКСИРУЕТСЯ) обязательна, если обращение к предмету этой статьи или статьи, ей подчиненной, осуществляется с помощью имени-индекса. Имя-индекса, идентифицируемое этой фразой, не определяется в исходной программе, так как его размещение и формат зависят от машины, и, не будучи данными, имена индексов не могут быть связаны с какой-либо структурой данных.

(14) Имя-индекса-1 должно быть однозначным.

5.8.4. Общие правила

(1) Исключая фразу OCCURS (ПОВТОРЯЕТСЯ), все фразы описания данных, относящиеся к данному, описание которого включает фразу OCCURS (ПОВТОРЯЕТСЯ), применяются к каждому вхождению описанного данного.

(2) Число вхождений предмета статьи определяется следующим образом:

а) в формате 1 значение целого-2 представляет точное число повторений;

б) в формате 2 текущее значение данного, именуемого именем-данного-1, представляет число повторений.

Формат 2 указывает, что предмет статьи имеет переменное число повторений. Значение целого-2 представляет максимальное число повторений, а значение целого-1 – минимальное. Это не означает, что размер предмета статьи переменный, а говорит лишь только о том, что переменно число его повторений.

Всякий раз, когда ссылаются на предмет статьи или на подчиненное ему данное или на данное, которому подчиняется предмет статьи, значение данного, определенное именем-данного-1, должно принадлежать диапазону от целого-1 до целого-2. Значения данных, номера вхождений которых превышают значение имени-данного-1, не определены.

(3) При обращении к групповому данному, содержащему данное, в статье описания которого содержится фраза OCCURS (ПОВТОРЯЕТСЯ) формата 2, используется только часть области таблицы, определяемая следующим образом:

а) если данное, определяемое именем-данного-1, не входит в группу, то используется только часть области таблицы, определяемая значением имени-данного-1 в момент начала обработки;

б) если данное, определенное именем-данного-1, входит в группу и групповое данное пересылается, то в пересылке будет использоваться только часть области таблицы, определенная значением имени-данного-1 в момент начала работы. Если груп-

GOST EXPERT ГОСТ 22558-89,

па есть принимающее данное, то будет использоваться максимальная длина группы.

(4) Вариант DESCENDING (ASCENDING) KEY (ПО УБЫВАНИЮ/ВОЗРАСТАНИЮ КЛЮЧА) указывает, что повторяющиеся данные упорядочены в возрастающем или убывающем порядке в соответствии со значениями имени-данного-2, имени-данного-3 и т. д. Возрастающий или убывающий порядок определяется в соответствии с правилами сравнения операндов (пп. 6.3.1.1.1 и 6.3.1.1.2 настоящей части). Имена-данных перечисляются в порядке уменьшения значимости.

(5) Если формат 2 используется в статье описания записи и соответствующее описание файла или описание сортируемого-сливаемого файла содержит вариант VARYING (ПЕРЕМЕННОЕ ЧИСЛО) во фразе RECORD (В ЗАПИСИ), то запись имеет переменную длину. Если вариант DEPENDING ON (В ЗАВИСИМОСТИ ОТ) во фразе RECORD (В ЗАПИСИ) не указан, то значение данного, определяемого именем-данного-1 во фразе OCCURS (ПОВТОРЯЕТСЯ), должно быть установлено на число записываемых повторений до выполнения любого из операторов RELEASE (ПЕРЕДАТЬ), REWRITE (ОБНОВИТЬ) или WRITE (ПИСАТЬ).

5.9. **Фраза PICTURE (ШАБЛОН)**

5.9.1. Назначение

Фраза PICTURE (ШАБЛОН) описывает общие характеристики и требования редактирования элементарного данного.

5.9.2. Общий формат

$$\left\{\begin{matrix}\underline{\text{PICTURE}}\\ \underline{\text{PIC}}\end{matrix}\right\} \text{ IS строка-литер}$$

$$\left\{\begin{matrix}\underline{\text{ШАБЛОН}}\\ \underline{\text{Ш}}\end{matrix}\right\} \text{ строка-литер}$$

5.9.3. Синтаксические правила

(1) Фраза PICTURE (ШАБЛОН) может быть задана только на уровне элементарного данного.

(2) Строка-литер содержит отдельные допустимые комбинации литер из набора литер, используемых в качестве символов шаблона. Для каждой категории элементарного данного определены допустимые комбинации символов шаблона.

(3) Для символов шаблона A, B, P(М), S(З), V(Т), X, Z(П), CR(КР), DB (ДБ) строчные буквы эквивалентны в описании строки литер шаблона прописным буквам. Остальные строчные буквы не эквивалентны соответствующим прописным буквам.

(4) Строка-литер может содержать не более 30 литер.

GOST EXPERT ГОСТ 22558-89,

(5) Фраза PICTURE (ШАБЛОН) должна быть определена для каждого элементарного данного, за исключением индексного данного или предмета фразы RENAMES (ПЕРЕИМЕНОВЫВАЕТ). В этих случаях использование фразы PICTURE (ШАБЛОН) запрещено.

(6) PIC (Ш) есть сокращение слова PICTURE (ШАБЛОН).

(7) Звездочка в строке-литер, используемая как символ подавления нуля, и фраза BLANK WHEN ZERO (ПРОБЕЛ КОГДА НУЛЬ) не могут появляться в одной и той же статье.

5.9.4. Общие правила

(1) Имеется пять категорий данных, которые могут быть описаны фразой PICTURE (ШАБЛОН): буквенная, числовая, буквенно-цифровая, буквенно-цифровая редактируемая и числовая редактируемая.

(2) Данное определяется как буквенное, если:

строка-литер его шаблона содержит только символ А;

его значение, будучи представлено в стандартном формате данных, является любой комбинацией одной или более буквенных литер алфавита.

(3) Данное определяется как числовое, если:

строка литер шаблона данного состоит только из символов 9, P(M), S(З) и V(Т), при этом количество цифровых позиций, задаваемых строкой-литер шаблона, должно быть от 1 до 18 включительно;

значение данного, будучи представлено в стандартном формате данных, может быть комбинацией цифр; оно может также включать знак числа.

(4) Данное определяется как буквенно-цифровое, если:

строка-литер шаблона данного ограничивается некоторыми комбинациями символов А, Х, 9, при этом данное обрабатывается так, будто строка-литер шаблона содержит только литеры Х. Строка-литер шаблона, которая состоит только из литер А или только из литер 9, не определяет буквенно-цифровое данное;

значение данного, будучи представлено в стандартном формате данных, состоит из любых литер набора литер машины.

(5) Данное определяется как буквенно-цифровое редактируемое если:

строка-литер шаблона данного представляется некоторыми комбинациями символов А, Х, 9, В, 0 (нуль) или /, при этом должна содержать по крайней мере один из символов А или Х и по крайней мере один из символов В, 0 (нуль) или /;

значение данного, будучи представлено в стандартном формате данных, состоит из любых литер набора литер машины.

(6) Данное определяется как числовое редактируемое, если:

GOST EXPERT ГОСТ 22558-89,

строка-литер его шаблона представляется некоторыми комбинациями символов B, /, P(M), V(T), Z(П), 0, 9, ., „, *, +, —, CR(КР), DB (ДБ) и символа валютного знака; допустимые комбинации этих символов определяются правилами предшествования символов и редактирования. Максимальное число цифровых позиций данного, определяемых строкой-литер, равно 18. Строка-литер должна содержать по крайней мере один из символов 0, B, /, Z(П), *, +, „ ., —, CR(КР), DB(ДБ) или символ валютного знака;

содержимое каждой из позиций литер должно соответствовать указанному для нее символу шаблона.

(7) Размер элементарного данного (число позиций литер в терминах стандартного формата данных, занимаемых элементарным данным) определяется числом допустимых символов, определяющих позиции литер.

Целое, заключенное в круглые скобки и следующее за символами A, „, X, 9, P(M), Z(П), *, +, B, 0, /, — или валютным знаком, указывает число повторений символа. Символы S(З), V(T), ., CR(КР), DB(ДБ) могут появляться в одном шаблоне только один раз.

(8) Функции символов, используемых для описания элементарного данного, поясняются ниже:

A — каждый символ A в строке-литер представляет позицию литеры, которая может содержать только букву или пробел.

B — каждый символ B в строке-литер представляет позицию, в которую будет помещена литера пробела.

P(M) — каждый символ P(M) указывает на подразумеваемую десятичную позицию; кроме того, если точка не находится между цифрами, представляющими число, литеры P(M) используются для указания места подразумеваемой десятичной точки. Литеры P(M) не учитываются при определении размера данного. Литеры P(M) учитываются при определении максимального числа цифровых позиций (не превышающего 18) в числовых редактируемых данных или в данных, которые появляются в качестве операндов в арифметических операторах. Литеры P(M) могут появляться в строке-литер шаблона только справа или только слева в виде непрерывной последовательности. Так как литера P(M) предполагает наличие подразумеваемой десятичной точки (слева от всех P(M), если все P(M) являются самыми левыми литерами шаблона, и справа от всех P(M), если все P(M) являются самыми правыми литерами шаблона), символ подразумеваемой точки V(T) является излишним. Литера P(M) и литера вставки «.» не могут использоваться одновременно в одной и той же строке-литер шаблона.

Если в некоторой операции ссылаются на данное, шаблон которого содержит символ P(M), то в ней используется алгебраическое

GOST EXPERT ГОСТ 22558-89,

значение данного, а не его действительное литерное представление. Это алгебраическое значение предполагает наличие десятичной точки на указанной позиции и нуля на десятичной позиции, указанной символом P(M). Размер значения определяется общим числом десятичных позиций, представленных в шаблоне. Вышесказанное относится к следующим операциям:

любым операциям, требующим пересылки числового операнда;

оператору MOVE (ПОМЕСТИТЬ), в котором пересылаемый операнд является числовым и его шаблон содержит символ P(M);

оператору MOVE (ПОМЕСТИТЬ), в котором пересылаемый операнд является числовым редактируемым и его шаблон содержит символ P(M), а принимающий операнд является числовым или числовым редактируемым;

операции сравнения, в которой оба операнда числовые.

Во всех других операциях десятичные позиции, определенные при помощи символа P(M), игнорируются и не учитываются при подсчете размера операнда.

S(З) — символ S(З) используется в строке-литер для указания наличия знака числа, не определяя при этом позицию или представление знака. Символ S(З) должен быть записан как самая левая литера в шаблоне. Символ S(З) не учитывается при определении размера элементарного данного, если статья описания этого данного не содержит фразу SEPARATE CHARACTER (ЗНАК ОТДЕЛЬНО) (п. 5.12 настоящей части).

V(Т) — символ V(Т) используется в строке-литер для указания расположения подразумеваемой десятичной точки и может появляться в строке-литер только один раз. Символ V(Т) не представляет позицию литеры и поэтому не учитывается при определении размера элементарного данного. Если подразумеваемая десятичная точка находится справа от самого правого символа в строке, символ V(Т) является излишним и может быть опущен.

X — каждый символ X в строке-литер представляет позицию литеры, содержащую любую допустимую литеру из набора литер машины и подсчитывается при определении размера данного.

Z(П) — каждый символ Z(П) в строке-литер представляет левые ведущие позиции цифр числа, которые должны быть заменены пробелами, если содержимое этих позиций окажется нулем. Каждый символ Z(П) подсчитывается при определении размера данного.

9 — каждый символ 9 в строке-литер представляет позицию литеры, содержащую цифру и учитывается при определении размера данного.

0 — каждый символ 0 (нуль) в строке-литер представляет позицию литеры, в которую будет помещен числовой нуль. 0 подсчитывается при определении размера данного.

GOST EXPERT ГОСТ 22558-89,

/ — каждый символ / (дробная черта) в строке-литер представляет позицию литеры, в которую будет помещена дробная черта. Символ / подсчитывается при определении размера данного.

, — каждый символ , (запятая) в строке-литер представляет позицию литеры, в которую будет помещена литера , . Эта позиция литеры подсчитывается при определении размера данного.

. — символ . (точка) появляется в строке-литер в качестве редактирующего символа, представляющего десятичную точку, по которой происходит выравнивание и, кроме того, представляет позицию, в которую будет помещена литера «.». Литера «.» подсчитывается при определении размера данного. Для данной программы функции точки и запятой меняются, если в параграфе SPECIAL-NAMES (СПЕЦИАЛЬНЫЕ-ИМЕНА) приведена фраза DECIMAL POINT IS COMMA (ДЕСЯТИЧНАЯ ТОЧКА ЗАПЯТАЯ); в этом случае правила для точки применяются к запятой и правила для запятой применяются к точке, где бы они ни встречались во фразе PICTURE (ШАБЛОН).

+, —, CR(КР), DB(ДБ) — эти символы при использовании представляют позицию литеры, в которую будет помещен редактирующий символ, управляемый знаком. Эти символы являются взаимно исключающими в любой строке-литер, и каждая литера, используемая в качестве символа редактирования, подсчитывается при определении размера данного.

*— каждый символ * (звездочка) в строке-литер представляет позицию ведущей числовой литеры, в которую, если содержимое этой позиции окажется нулем, будет помещена звездочка. Каждый символ * подсчитывается при определении размера данного.

Валютный символ — валютный символ в строке-литер представляет позицию литеры в данном, в которую должен быть помещен валютный знак. Валютный символ в строке-литер представляется либо валютным знаком, либо литерой, определенной во фразе CURRENCY SIGN (ВАЛЮТНЫЙ ЗНАК) в параграфе SPECIAL-NAMES (СПЕЦИАЛЬНЫЕ-ИМЕНА).

Валютный символ учитывается при определении размера данного.

5.9.5. Правила редактирования

(1) Имеются два общих метода редактирования во фразе PICTURE (ШАБЛОН): либо путем вставки, либо путем подавления и замещения. Допустимы четыре типа редактирования вставкой:

простая вставка;
специальная вставка;
фиксированная вставка;
плавающая вставка.

GOST EXPERT ГОСТ 22558-89,

Имеются два типа редактирования посредством подавления и замещения нулей;

подавление нуля и замещение пробелами;

подавление нуля и замещение звездочками.

(2) Тип редактирования, который может выполняться над данным, зависит от категории, к которой данное принадлежит. Ниже определяется тип редактирования, разрешенный для данной категории;

Категория	Тип редактирования
Буквенная	Никакой
Числовая	Никакой
Буквенно-цифровая	Никакой
Буквенно-цифровая редактируемая	Простая вставка 0, B и /
Числовая редактируемая	Все типы с учетом правила 3

(3) Во фразе PICTURE (ШАБЛОН) редактирование плавающей вставкой и редактирование подавлением и замещением нуля взаимно исключаются. Во фразе PICTURE (ШАБЛОН) может быть использован только один тип замещения при подавлении нуля.

(4) Редактирование простой вставкой указывается символами шаблона: , (запятая), B (пробел), / (дробная черта) и 0 (нуль). Литеры вставки учитываются при определении размера данного и представляют позицию в данном, в которую будет вставлена литера, соответствующая символу шаблона. Если литера вставки , (запятая) является последним символом в шаблоне, фраза PICTURE (ШАБЛОН) должна быть последней фразой статьи описания данного и за ней непосредственно должен следовать разделитель точка. В результате в статье описания данного появляется комбинация «,.» или две последовательные точки, если используется фраза DECIMAL-POINT IS COMMA (ДЕСЯТИЧНАЯ ТОЧКА ЗАПЯТАЯ).

(5) При редактировании специальной вставкой в качестве литеры вставки используется символ . (точка), которая кроме того служит для выравнивания. Литера вставки для явной десятичной точки учитывается при определении размера данного. В одной и той же строке-литер шаблона не разрешается использование подразумеваемой десятичной точки, представленной символом V (Т), и явной десятичной точки, представленной символом вставки. Если литера вставки является последней литерой шаблона, то фраза PICTURE (ШАБЛОН) должна быть последней фразой в статье описания данного и за ней непосредственно должен следовать разделитель точка. В результате в статье описания данного появляются две последовательные точки или комбинация,. , если использу-

GOST EXPERT ГОСТ 22558-89,

ется фраза DECIMAL-POINT IS COMMA (ДЕСЯТИЧНАЯ ТОЧКА ЗАПЯТАЯ).

Результатом редактирования специальной вставкой является появление литеры специальной вставки в значении данного на позиции, указанной в строке-литер шаблона.

(6) Редактирование фиксированной вставкой указывается валютным символом и символами редактирования, управляемыми знаком: +, —, CR(KP), DB(ДБ).

В данной строке-литер шаблона может быть использован только один валютный символ и только один из символов редактирования, управляемых знаком. Если используются символы CR (KP) и DB (ДБ), они представляют две самые правые позиции литер и учитываются при определении размера данного. Если используется символ + или —, он должен быть самой левой или самой правой позицией литеры, которая учитывается при определении размера данного. Валютный знак должен быть самой левой позицией литеры, кроме случаев, когда ему может предшествовать символ + или символ —; валютный знак учитывается при определении размера данного. При редактировании фиксированной вставкой в редактируемое данное помещается литера вставки, управляемая знаком, на позицию литеры, которую эта литера занимает в строке-литер шаблона.

Зависимость результатов редактирования фиксированной вставкой от значения данного приведена ниже.

Редактирующий символ в строке-литер шаблона	Литеры вставки, появляющиеся в редактируемом данном	
	Данное положительное или нуль	Данное отрицательное
+	+	—
—	Пробел	—
CR (KP)	2 пробела	CR (KP)
DB (ДБ)	То же	DB (ДБ)

(7) Редактирование плавающей вставкой указывается валютным символом и редактирующими символами, управляемыми знаком + или —, которые являются литерами плавающей вставки, взаимно исключающими друг друга в данной строке-литер шаблона.

Редактирование плавающей вставкой указывается в строке-литер шаблона цепочкой из двух или более литер плавающей вставки. Цепочка литер плавающей вставки может содержать любые символы фиксированной вставки или предшествовать литерам

GOST EXPERT ГОСТ 22558-89,

фиксированной вставки. Литеры простой вставки рассматриваются как часть цепочки литер плавающей вставки. Если литерой плавающей вставки является валютный символ, то цепочка литер плавающей вставки может содержать литеры фиксированной вставки CR (КР) или DB (ДБ) непосредственно справа от этой цепочки.

Самая левая литера цепочки литер плавающей вставки представляет левую границу для размещения литеры плавающей вставки. Самая правая литера цепочки литер плавающей вставки представляет правую границу для размещения литеры плавающей вставки.

Вторая слева литера плавающей вставки представляет левую границу числового данного, которое может быть помещено в поле данного. Ненулевые числовые данные могут располагаться, начиная от этой позиции, на любой позиции справа от этой границы.

Строка-литер шаблона определяет два способа редактирования плавающей вставкой. Первый из них заключается в том, что некоторые или все ведущие позиции цифр слева от десятичной точки представляются литерой вставки; при втором способе позиции всех цифр в строке-литер шаблона представляются литерой вставки.

Если литеры вставки находятся только слева от десятичной точки, то в результате редактирования единственная литера вставки размещается на позиции литеры, непосредственно предшествующей десятичной точке или первой ненулевой цифре в позициях данного, представленных цепочкой символов вставки (в зависимости от того, что встречается ранее в данном). Позиции литер, предшествующие литере вставки, замещаются пробелами.

Если все позиции цифровых литер в строке-литер шаблона представляются литерой вставки, результат редактирования зависит от значения данных. Если значение равно нулю, все данное будет содержать пробелы. Если значение не равно нулю, результат будет таким, как если бы литеры вставки находились только слева от десятичной точки.

Если все позиции цифровых литер в строке-литер шаблона представляются литерами вставки, то по крайней мере одна из литер вставки должна находиться слева от десятичной точки.

Если цепочкой литер плавающей вставки являются управляющие символы редактирования + или —, то вставляемые литеры зависят от значения данного:

Редактирующий символ в строке-литер шаблона	Литеры вставки, появляющиеся в редактируемом данном	
	Данное положительно или нуль	Данное отрицательно
+	+	—
—	Пробел	—

GOST EXPERT ГОСТ 22558-89,

Если все позиции цифровых литер в строке-литер шаблона представлены литерами вставки, результат зависит от значения данного. Если значение нуль, то данное будет содержать пробелы. Если значение не нуль, то результат будет таким же, как и в случае, когда вставляемыми литерами являются литеры только слева от десятичной точки.

Во избежание отбрасывания литер при перемещении данных, минимальный размер строки-литер шаблона для принимающего поля должен равняться числу литер пересылаемого данного плюс число литер фиксированной вставки в принимающем поле, плюс 1 для литеры плавающей вставки. Если же отбрасывание литер произошло, то значением данных для редактирования будет значение, полученное после отбрасывания литер.

(8) Подавление ведущих нулей на позициях цифр указывается использованием литер Z(П) или* в качестве символов подавления в строке-литер шаблона. Эти символы являются взаимно исключающими в одной и той же строке-литер шаблона. Каждый символ подавления учитывается при определении размера данного. Если используется Z(П), литерой замещения будет пробел; если используется звездочка, литерой замещения будет *.

Подавление и замещение нулей указывается в строке-литер шаблона цепочкой из одного или более символов подавления нуля в ведущих позициях цифр, которые должны быть заменены, если соответствующая позиция литеры в данных содержит нуль. Литеры простой вставки, находящиеся в этой цепочке символов или непосредственно справа от нее, рассматриваются как часть этой цепочки.

В строке-литер шаблона имеется два способа представления подавления нуля: при первом способе некоторые или все ведущие позиции цифр слева от десятичной точки представляются символами подавления; другой способ заключается в представлении всех позиций цифр в строке-литер шаблона символами подавления.

Если символы подавления оказались только слева от десятичной точки, то любой ведущий нуль в данном на позиции символа подавления в строке-литер шаблона заменяется на литеру замещения. Подавление заканчивается на первой ненулевой цифре из представленных строкой символов подавления или на десятичной точке, в зависимости от того, что встретилось первым.

Если все позиции цифр в строке-литер шаблона представляются символами подавления и значение данных отлично от нуля, то результат является таким же, как в случае, если бы литеры подавления находились только слева от десятичной точки. Если значение является нулевым, то различаются следующие два случая: если символом подавления является Z(П), все данное, включая литеры редактирования, заменяется пробелами; если символом по-

GOST EXPERT ГОСТ 22558-89,

давления является *, все данное, кроме явной десятичной точки, заменяется звездочками;

(9) символы +, —, *, Z(П) и валютный знак при использовании в качестве плавающих литер замещения являются взаимно исключающими внутри одной строки-литер.

5.9.6. Правила предшествования

В таблице показан порядок предшествования при использовании литер в качестве символов строки-литер шаблона.

Символ × в клетке таблицы указывает, что символы, используемые как заголовки соответствующих столбцов, могут предшествовать (но необязательно непосредственно) в строке шаблона символам, указанным в начале соответствующих строк таблицы. Символ «—» указывает на недопустимость такого предшествования. Заключенные в фигурные скобки символы являются взаимно исключающими. Валютный знак указывается с помощью символов «В. зн.».

В строку литер шаблона должны входить по крайней мере один из символов A, X, Z (П), 9 или *, или по крайней мере два из символов +, —, или В.зн.

Символы неплавающей вставки + и —, символы плавающей вставки Z (П), *, +, — и В. зн. и символ Р (М) встречаются дважды в таблице. Для каждого символа левый столбец и верхний ряд представляют использование этого символа слева от позиции десятичной точки. Правый столбец и нижний ряд представляют использование символа справа от позиции десятичной точки.

5.10. **Фраза REDEFINES (ПЕРЕОПРЕДЕЛЯЕТ)**

5.10.1. Назначение

Фраза REDEFINES (ПЕРЕОПРЕДЕЛЯЕТ) позволяет одну и ту же область памяти машины описывать различными статьями описания данных.

5.10.2. Общий формат

номер-уровня [имя-данного-1 / FILLER] REDEFINES имя-данного-2

номер-уровня [имя-данного-1 / ЗАП / ЗАПОЛНИТЕЛЬ] ПЕРЕОПРЕДЕЛЯЕТ имя-данного-2

В вышеприведенном формате номер-уровня, имя-данного-1 и FILLER (ЗАПОЛНИТЕЛЬ) не являются частью фразы REDEFINES (ПЕРЕОПРЕДЕЛЯЕТ) и приведены для наглядности.

5.10.3. Синтаксические правила

(1) Фраза REDEFINES (ПЕРЕОПРЕДЕЛЯЕТ), если она указана, должна следовать непосредственно за субъектом статьи.

(2) Номера уровней имени-данного-1 и имени-данного-2 должны быть одинаковыми | и не могут быть равными 66 или 88 | .

GOST EXPERT ГОСТ 22558-89,

Допустимость предшествования первого символа

Второй символ	Символы фиксированной вставки: В	0	/	,	.	{+ −}	{+ −}	{CR (КР) DB (ДВ)}	В. зн.	Символы плавающей вставки: {Z (П) *}	{Z (П) *}	{+ −}	{+ −}	В. зн.	В. зн.	Другие символы: У	А Х	S (З)	V (Т)	P (М)	P (М)
Символы фиксированной вставки	В	×	×	×	×	×	×	—	×	×	×	×	×	—	×	×	×		×		×
	0	×	×	×	×	×	—	—	×	×	×	×	×	—	×	×	×		×		×
	/	×	×	×	×	×	—	—	×	×	×	×	×	—	×	×	×		×		×
	,	×	×	×	×	—	—	—	×	×	×	×	×	—	×	×	×		×		×
	.	×	×	×	—	—	—	—	×	×	×	×	×	—	×	×	×		×		×
	{+ −}	×	×	×	×	—	—	—	×	×	×	×	×	—	×	×	×		×		×
	{+ −}	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—		—		—
	{CR (КР) DB (ДВ)}	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—		—		—
	В. зн.	×	×	×	×	×	×	—	×	×	×	×	×	—	×	×	×		×		×
Символы плавающей вставки	{Z (П) *}	×	×	×	×	×	×	—	×	×	×	×	×	—	×	×	×		×		×
	{Z (П) *}	×	×	×	×	×	×	—	×	×	×	×	×	—	×	×	×		×		×
	{+ −}	×	×	×	×	×	×	—	×	×	×	×	×	—	×	×	×		×		×
	{+ −}	×	×	×	×	×	×	—	×	×	×	×	×	—	×	×	×		×		×
	В. зн.	×	×	×	×	×	×	—	×	×	×	×	×	—	×	×	×		×		×
	В. зп.	×	×	×	×	×	×	—	×	×	×	×	×	—	×	×	×		×		×

GOST EXPERT ГОСТ 22558-89,

Продолжение

Второй символ	Допустимость предшествования первого символа																							
	Символы фиксированной вставки									Символы плавающей вставки					Другие символы									
	B	0	/	,	-	{+ −}	{+ −}	{CR (КР) DB (ДВ)}	В. зн	{Z * (П)}	{+ −}	{+ −}	В. зн	В. зн	9	A X	S (З)	V (Т)	P (М)	P (М)
Другие символы																								
9	X	X	X	X	X	X	—	—	X	—	X	—	—	—	X	X	X	X	—	X
A X	X	X	X	—	—	—	—	—	—	—	—	—	—	—	X	X	—	—	—	—
S(3)	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
V(T)	X	X	X	—	—	—	—	—	X	—	—	—	X	—	X	—	X	—	—	—
P(M)	X	X	X	—	—	—	—	—	X	—	—	—	X	—	X	—	X	—	X	—
P(M)	—	—	—	—	X	—	—	—	X	—	—	—	—	—	—	—	—	X	—	X

GOST EXPERT ГОСТ 22558-89,

(3) В секции файлов эта фраза не должна использоваться в статьях уровня 01.

(4) В секции коммуникаций эта фраза не должна использоваться в статьях уровня 01.

(5) Статья описания данного для имени-данного-2 не может содержать фразу OCCURS (ПОВТОРЯЕТСЯ).

Однако имя-данного-2 может подчиняться данному, статья описания которого содержит фразу OCCURS (ПОВТОРЯЕТСЯ). При этом, однако, имя-данного-2 во фразе REDEFINES (ПЕРЕОПРЕДЕЛЯЕТ) не должно индексироваться.

Как исходное определение, так и переопределение не должны включать данные переменного размера.

(6) Если данное имя-данного-2 описано как внешнее либо имеет уровень, отличный от 01, то число содержащихся в нем позиций литер должно быть больше или равно числу позиций литер данного, определенного именем-данного-1. Если данное имя-данного-2 имеет уровень 01 и не является внешним, то такого ограничения нет.

(7) Имя-данного-2 не должно уточняться, даже если оно неоднозначно, так как в этом случае благодаря расположению фразы REDEFINES (ПЕРЕОПРЕДЕЛЯЕТ) в исходной программе двусмысленности не будет.

(8) Допустимы множественные переопределения одних и тех же позиций литер. Множественные переопределения одних и тех же позиций литер должны использовать имя-данного статьи, в которой впервые была определена область памяти.

(9) Статьи, дающие новые описания позиций литер, не должны содержать фразу VALUE (ЗНАЧЕНИЕ), за исключением статей описаний имен-условий.

(10) Между статьями описания имени-данного-2 и субъекта статьи не может появляться статья, имеющая номер-уровня, численно меньший номера уровня имени-данного-2 и субъекта статьи.

(11) Статьи, дающие новые описания позиций литер, должны следовать за статьей, определяющей область имени-данного-2, без промежуточных статей, описывающих новые позиции литер.

(12) Для уровня 1 ядра имя-данного-2 не может быть подчинено статье, содержащей фразу REDEFINES (ПЕРЕОПРЕДЕЛЯЕТ).

Для уровня 2 ядра имя-данного-2 может быть подчинено статье, содержащей фразу REDEFINES (ПЕРЕОПРЕДЕЛЯЕТ).

5.10.4. Общие правила

(1) Распределение памяти начинается с имени-данного-2 и продолжается на область памяти, достаточную для размещения данного, определяемого именем-данного-1 или фразой FILLER (ЗАПОЛНИТЕЛЬ).

GOST EXPERT ГОСТ 22558-89,

то же время имя-данного-3 не может входить в группу, определяемую именем-данного-2.

5.11.4. Общие правила

(1) Если имя-данного-3 задано, то имя-данного-1 является именем группы, которая содержит все элементарные данные, начиная от имени-данного-2 (если имя-данного-2 является элементарным данным) или первого элементарного данного в имени-данного-2 (если имя-данного-2 является именем группы) и заканчивая именем-данного-3 (если имя-данного-3 является элементарным данным) или последним элементарным данным в имени-данного-3 (если имя-данного-3 относится к групповому данному).

(2) Если имя-данного-3 не указано, то все свойства данного, определенного именем-данного-2, становятся свойствами данного, определенного именем-данного-1.

5.12. **Фраза SIGN (ЗНАК)**

5.12.1. Назначение

Фраза SIGN (ЗНАК) определяет позицию и способ представления знака числа, если эти сведения необходимо явно задать.

5.12.2. Общий формат

[SIGN IS] $\left\{\begin{matrix}\text{LEADING}\\\text{TRAILING}\end{matrix}\right\}$ [SEPARATE CHARACTER]

[ЗНАК] $\left\{\begin{matrix}\text{ПЕРВЫЙ}\\\text{ПОСЛЕДНИЙ}\end{matrix}\right\}$ [ОТДЕЛЬНО]

5.12.3. Синтаксические правила

(1) Фраза может быть задана только в статьях описания числовых данных, строка литер шаблона которых содержит литеру S(Э), или в статьях описания групповых данных, содержащих хоть одну статью описания числовых данных вышеуказанного типа.

(2) Статьи описания числовых данных, содержащие фразу SIGN (ЗНАК), должны быть описаны явно или неявно с использованием DISPLAY (ДЛЯ ВЫДАЧИ).

(3) Если в описании файла указана фраза CODE-SET (АЛФАВИТ), то статьи описания входящих в его записи числовых данных со знаком должны содержать фразу SEPARATE (ОТДЕЛЬНО).

5.12.4. Общие правила

(1) Фраза SIGN (ЗНАК) необязательна; она определяет позицию и способ представления знака числового данного, в статье описания которого она встречается, или числовых данных, подчиненных групповому данному, статья описания которого содержит эту фразу. Фраза SIGN (ЗНАК) применяется только к статьям описания числовых данных, строка литер шаблона которых содер-

GOST EXPERT ГОСТ 22558-89,

жит символ S (З), указывающий на наличие знака числа, но не на его способ представления или позицию.

(2) Если фраза SIGN (ЗНАК) указана для группового данного, подчиненного другому групповому данному, для которого фраза SIGN (ЗНАК) тоже указана, то фраза, определенная на младшем уровне иерархии, подавляет фразу SIGN (ЗНАК), указанную на старшем уровне иерархии.

(3) Если фраза SIGN (ЗНАК) указана в статье описания элементарного числового данного, подчиненного групповому данному, для которого фраза SIGN (ЗНАК) тоже указана, то фраза SIGN (ЗНАК), определенная в статье описания элементарного числового данного подавляет фразу SIGN (ЗНАК) группового данного.

(4) Если статья описания числового данного не содержит фразу SIGN (ЗНАК), а строка литер его шаблона содержит символ S(З), то позиция и представление знака (по умолчанию) определяются реализацией. Общие правила п. 5.12.4 не применимы к таким данным.

(5) Если вариант SEPARATE (ОТДЕЛЬНО) не указан, то предполагается, что знак числа связывается с первой (или, соответственно, последней) позицией цифры элементарного числового данного; символ S (З) в строке-литер шаблона не учитывается при определении размера данного (в терминах числа литер стандартного формата данного); представление знака данного определяет реализация.

(6) Если вариант SEPARATE (ОТДЕЛЬНО) указан, то знак числа будет занимать первую (или, соответственно, последнюю) позицию литеры элементарного числового данного; эта позиция литеры не является позицией цифры; символ S (З) в строке-литер шаблона учитывается при определении размера данного (в терминах числа литер стандартного формата данного); положительный или отрицательный знак числа представляют литеры стандартного формата данных + или —.

(7) Все необходимые преобразования при выполнении вычислений или сравнений для данных, статьи описания которых содержат фразу SIGN (ЗНАК) и символ S (З) в строке литер-шаблона, выполняются автоматически.

5.13. Фраза SYNCHRONIZED (ВЫДЕЛЕНО)

5.13.1. Назначение

Фраза SYNCHRONIZED (ВЫДЕЛЕНО) определяет выравнивание элементарного данного по естественным границам машинной памяти (см. ч. 4, п. 4.3.7)

5.13.2. Общий формат

$$\left\{\begin{matrix}\underline{\text{SYNCHRONIZED}}\\ \underline{\text{SYNC}}\end{matrix}\right\}\left[\begin{matrix}\underline{\text{LEFT}}\\ \underline{\text{RIGHT}}\end{matrix}\right]$$

GOST EXPERT ГОСТ 22558-89,

ВЫДЕЛЕНО [ВЛЕВО / ВПРАВО]

5.13.3. Синтаксические правила

(1) Эта фраза может появляться только в статье описания элементарного данного.

(2) SYNC является сокращением слова SYNCHRONIZED.

5.13.4. Общие правила

(1) Фраза определяет, что при создании внутреннего формата данного процессор Кобола должен разместить это данное в последовательных единицах памяти таким образом, что никакое другое данное не может появиться ни в одной единице памяти между левой и правой естественными границами, заключающими это данное. Если данное по размеру занимает не всю память между заключающими его естественными границами, то неиспользованные единицы памяти (или части их) не могут быть использованы для другого данного.

Такая неиспользованная память включается:

а) в размер любой группы данных, к которой принадлежит элементарное данное;

б) в распределяемые позиции литер, когда групповое данное является объектом фразы REDEFINES (ПЕРЕОПРЕДЕЛЯЕТ). Неиспользуемые позиции литер не включаются в переопределяемые позиции, если объектом фразы REDEFINES (ПЕРЕОПРЕДЕЛЯЕТ) является элементарное данное.

(2) Фраза SYNCHRONIZED (ВЫДЕЛЕНО), в которой не указано ни RIGHT (ВПРАВО), ни LEFT (ВЛЕВО), означает, что элементарное данное нужно расположить между естественными границами таким образом, чтобы достичь наиболее эффективного его использования. Особенности размещения определяются реализацией.

(3) SYNCHRONIZED LEFT (ВЫДЕЛЕНО ВЛЕВО) определяет расположение элементарного данного, начиная с левой крайней позиции литеры, в естественных границах памяти, в которую оно помещается.

(4) SYNCHRONIZED RIGHT (ВЫДЕЛЕНО ВПРАВО) указывает, что элементарное данное размещается вплотную к правой позиции литеры в естественных границах памяти, в которую оно помещается.

(5) При обращении в исходной программе к данному, имеющему в описании фразу SYNCHRONIZED (ВЫДЕЛЕНО), при определении любого действия, зависящего от размера (например выравнивания, усечения, переполнения), используется размер данного, задаваемый фразой PICTURE (ШАБЛОН).

GOST EXPERT ГОСТ 22558-89,

(6) Если описание данного содержит фразу SYNCHRONIZED (ВЫДЕЛЕНО) и знак числа, то последний появляется в позиции, знака, явно или неявно указанной фразой SIGN (ЗНАК).

(7) Если фраза SYNCHRONIZED (ВЫДЕЛЕНО) указана для данного внутри области действия фразы OCCURS (ПОВТОРЯЕТСЯ), то выделяется каждое вхождение повторяющегося данного, для которого при этом порождается необходимый неявный заполнитель.

(8) Фраза SYNCHRONIZED (ВЫДЕЛЕНО) зависит от оборудования и в дополнение к правилам (1)—(7) при реализации должно быть точно определено, как обрабатываются связанные с этой фразой элементарные данные в зависимости от формата внешнего представления содержащих их записей или групп и порождения необходимых неявных заполнителей (если элементарное данное, непосредственно предшествующее данному, содержащему фразу SYNCHRONIZED (ВЫДЕЛЕНО), не оканчивается на естественной границе). Автоматически порождаемые позиции заполнителя включаются в размер каждой группы, в которой содержится заполнитель, и в переопределяемые поля машинной памяти, если группа данных, частью которой является заполнитель, появляется как объект фразы REDEFINES (ПЕРЕОПРЕДЕЛЯЕТ).

(9) Реализация может задавать автоматическое выравнивание внутри записи для любого внутреннего формата данных, за исключением данных, использование которых указано DISPLAY (ДЛЯ ВЫДАЧИ). Однако запись в целом может быть выделена.

(10) Правила выделения записей в файлах данных, вызывающие выделение элементарных данных, должны быть оговорены реализацией.

5.14. Фраза об использовании

5.14.1. Назначение

Фраза об использовании определяет формат данного в памяти машины.

5.14.2. Общий формат

USAGE IS { BINARY / COMPUTATIONAL / COMP / DISPLAY / INDEX / PACKED-DECIMAL }

GOST EXPERT ГОСТ 22558-89,

{ ДВОИЧНОЕ
ДЕСЯТИЧНОЕ
ДЛЯ { ВЫЧИСЛЕНИЙ
ВЫЧ
ВЫДАЧИ
ИНДЕКСА } }

5.14.3. Синтаксические правила

(1) Фраза об использовании может быть указана в любой статье описания данного [с номером уровня, отличным от 66 или 88].

(2) Если фраза об использовании указана в статье описания группового данного, она также может быть указана в статье описания любого подчиненного элементарного или группового данного, но в обоих случаях должно быть указано одинаковое использование.

(3) Элементарное данное, описание которого содержит фразы BINARY (ДВОИЧНОЕ), COMPUTATIONAL (ДЛЯ ВЫЧИСЛЕНИЙ) или PACKED-DECIMAL (ДЕСЯТИЧНОЕ), или элементарное данное, подчиненное групповому данному, описание которого содержит указанные фразы, должно быть описано строкой-литер шаблона, определяющей числовые данные, т. е. строкой-литер шаблона, содержащей только символы P (М), S (З), V (Т), 9.

(4) COMP (ВЫЧ) является сокращением слова COMPUTATIONAL (ВЫЧИСЛЕНИЙ).

(5) На индексное данное явно можно ссылаться только в операторах SEARCH (ИСКАТЬ) или SET (УСТАНОВИТЬ), условии отношения, во фразе USING (ИСПОЛЬЗУЯ) заголовка раздела процедур или во фразе USING (ИСПОЛЬЗУЯ) оператора CALL (ВЫЗВАТЬ).

(6) Фразы BLANK WHEN ZERO (ПРОБЕЛ КОГДА НУЛЬ), JUSTIFIED (СДВИНУТО), PICTURE (ШАБЛОН), SYNCHRONIZED (ВЫДЕЛЕНО) и VALUE (ЗНАЧЕНИЕ) не должны указываться для данного, которое используется как индексное.

(7) Элементарное данное, описанное фразой INDEX (ДЛЯ ИНДЕКСА), не должно быть условной переменной.

5.14.4. Общие правила

(1) Если фраза об использовании написана на уровне группового данного, она относится к каждому элементарному данному группы.

(2) Фраза указывает способ представления данного в памяти машины. Она не влияет на использование данного, хотя спецификации для некоторых операторов в разделе процедур могут ограничить фразу об использовании для операндов, к которым производится обращение. Фраза об использовании может влиять на вы-

GOST EXPERT ГОСТ 22558-89,

бор основания системы счисления или на тип представления данного.

(3) Фраза BINARY (ДВОИЧНОЕ) означает, что двоичное основание используется для представления числового данного в памяти машины. Каждая реализация определяет точное воздействие фразы BINARY (ДВОИЧНОЕ) на выравнивание и представление данного в памяти машины, включая представление алгебраического знака. Реализацией должно быть распределено достаточно памяти для размещения десятичных значений максимального диапазона, определяемого строкой литер шаблона.

(4) Фраза COMPUTATIONAL (ДЛЯ ВЫЧИСЛЕНИЙ) определяет используемые реализацией основание и формат представления числового данного в памяти машины. Конкретная реализация определяет точное воздействие фразы COMPUTATIONAL (ДЛЯ ВЫЧИСЛЕНИЙ) на выравнивание и представление данного в памяти машины, включая представление алгебраического знака и допустимый диапазон значений данных.

(5) Фраза DISPLAY (ДЛЯ ВЫДАЧИ) (указанная явно или неявно) определяет, что для представления данных в памяти машины используется стандартный формат данных и что данные выравниваются на границе символа.

(6) Если для элементарного данного или любой группы, к которой принадлежит данное, не задана фраза об использовании, то подразумевается использование DISPLAY (ДЛЯ ВЫДАЧИ).

(7) Фраза INDEX (ДЛЯ ИНДЕКСА) определяет данное как индексное данное, содержащее значение, соответствующее номеру вхождения элемента таблицы. Точное воздействие фразы INDEX (ДЛЯ ИНДЕКСА) на выравнивание и представление данного в памяти машины, включая действительные значения, присвоенные любому номеру вхождения элемента таблицы, определяет конкретная реализация.

(8) При выполнении оператора MOVE (ПОМЕСТИТЬ) или оператора ввода-вывода, использующих групповое данное, содержащее индексное данное, никакого преобразования индексного данного не происходит.

(9) Фраза PACKED-DECIMAL (ДЕСЯТИЧНОЕ) указывает, что для представления числового данного в памяти машины используется десятичное основание. Более того, она указывает, что каждая десятичная позиция должна занимать наименьшую возможную область памяти машины. Каждая реализация определяет точное воздействие фразы PACKED-DECIMAL (ДЕСЯТИЧНОЕ) на выравнивание и представление данного в памяти машины, включая представление любого алгебраического знака. Для размещения максимального значения, соответствующего десятичным позициям числа, определяемого строкой литер шаблона, реализацией должен быть выделен достаточный объем памяти.

GOST EXPERT ГОСТ 22558-89,

5.15. Фраза VALUE (ЗНАЧЕНИЕ)

5.15.1. Назначение

Фраза VALUE (ЗНАЧЕНИЕ) задает начальное значение данных рабочей памяти, начальное значение данных в секции коммуникаций | и значения, связанные с именем-условия. |

5.12.2. Общий формат

Формат 1

VALUE IS литерал-1

{ ЗНАЧЕНИЕ / ЗНАЧ } литерал-1

Формат 2

{ VALUE IS / VALUES ARE } { литерал-2 [{ THROUGH / THRU } литерал-3] } ...

{ ЗНАЧЕНИЕ / ЗНАЧ } { литерал-2 [ПО литерал-3] } ...

5.12.3. Синтаксические правила

(1) Для данного, шаблон которого определяет числовое данное со знаком, литерал, представляющий значение, должен иметь знак.

(2) Все числовые литералы, приведенные во фразе VALUE (ЗНАЧЕНИЕ), должны принадлежать диапазону значений, определяемых фразой PICTURE (ШАБЛОН), и не должны приводить к усечению значащих цифр. Размер нечисловых литералов не должен превышать размер данного, определяемый фразой PICTURE (ШАБЛОН).

(3) Слова THROUGH и THRU эквивалентны. ЗНАЧ является сокращением слова ЗНАЧЕНИЕ.

(4) Фраза VALUE (ЗНАЧЕНИЕ) не может быть употреблена ни в одной статье, являющейся частью описания или переопределения внешней записи данных, | за исключением статьи имени-условия |.

5.15.4. Общие правила

(1) Фраза VALUE (ЗНАЧЕНИЕ) не должна противоречить другим фразам в описании данного или фразам в описании данных внутри иерархии, которой принадлежит данное, описание которого содержит эту фразу. Применяются следующие правила:

а) если категория данного числовая, все литералы во фразе VALUE (ЗНАЧЕНИЕ) должны быть числовыми литералами. Если литерал определяет значение данного из рабочей памяти, этот

GOST EXPERT ГОСТ 22558-89,

литерал выравнивается согласно стандартным правилам выравнивания (см. ч. 4, п. 4.3.6);

б) если категория данного буквенная, буквенно-цифровая, буквенно-цифровая редактируемая или числовая редактируемая, то все литералы во фразе VALUE (ЗНАЧЕНИЕ) должны быть нечисловыми литералами. Литералы выравниваются как буквенно-цифровые данные (см. ч. 4, п. 4.3.6).

Литеры редактирования в строке литер шаблона учитываются при определении размера данного, но не влияют на установку начального значения данного (см. п. 5.9 настоящей части); поэтому значение редактируемого данного представляется в отредактированной форме;

в) присвоение начального значения выполняется независимо от наличия фразы BLANK WHEN ZERO (ПРОБЕЛ КОГДА НУЛЬ) или JUSTIFIED (СДВИНУТО).

5.15.5. Правила для имен-условий

(1) В статье имени условия фраза VALUE (ЗНАЧЕНИЕ) обязательна. Фраза VALUE (ЗНАЧЕНИЕ) и само имя-условия являются единственными допустимыми фразами в такой статье.

(2) Формат 2 может использоваться только в связи с именами условий (см. ч. 4, п. 4.2.2.1.1.1). При использовании фразы THRU (ПО) литерал-2 должен быть меньше, чем литерал-3.

5.15.6. Статьи описания данных, отличные от имен-условий

Должны выполняться следующие правила.

(1) Правила употребления фразы VALUE (ЗНАЧЕНИЕ) различны для соответствующих секций раздела данных:

а) в секции файлов на уровне 1 фраза VALUE (ЗНАЧЕНИЕ) не может использоваться. На уровне 2 она допустима только в статьях имен-условий, поэтому начальные значения данных секции файлов не определены;

б) фраза VALUE (ЗНАЧЕНИЕ) не может использоваться в секции связи на уровне 1.

В секции связи она допустима только в статьях имен-условий;

в) в секции рабочей памяти и секции коммуникаций фраза VALUE (ЗНАЧЕНИЕ) должна использоваться в статьях имен-условий. Фраза VALUE (ЗНАЧЕНИЕ) в секции рабочей памяти и секции коммуникаций оказывает свое действие только тогда, когда программа устанавливается в свое начальное состояние. Если фраза VALUE (ЗНАЧЕНИЕ) указана в описании данного, то ему присваивается указанное значение. Если фраза VALUE

GOST EXPERT ГОСТ 22558-89,

(ЗНАЧЕНИЕ) не указана, начальное значение данного не определено.

(2) Фраза VALUE (ЗНАЧЕНИЕ) не должна входить в статью описания данного, которая содержит фразу REDEFINES (ПЕРЕОПРЕДЕЛЯЕТ), или в статью, которая подчиняется статье, содержащей такую фразу. Это правило не применяется к статьям имен-условий.

(3) Если фраза VALUE (ЗНАЧЕНИЕ) используется в статье на уровне группы, то литерал должен быть стандартной константой или нечисловым литералом. В этом случае области памяти для группы присваивается начальное значение без учета особенностей элементарных или групповых данных, содержащихся внутри этой группы; при этом фраза VALUE (ЗНАЧЕНИЕ) не может задаваться на подчиненных уровнях внутри этой же группы.

(4) Фраза VALUE (ЗНАЧЕНИЕ) не должна задаваться для группы, содержащей данные, описания которых включают фразы JUSTIFIED (СДВИНУТО), SYNCHRONIZED (ВЫДЕЛЕНО) или фразы об использовании (кроме варианта USAGE IS DISPLAY (ДЛЯ ВЫДАЧИ)).

(5) Если фраза VALUE (ЗНАЧЕНИЕ) указана в статье описания данного, связанного с переменно повторяющимся данным, то инициация начального значения происходит так, как если бы значение данного, указанного в варианте DEPENDING ON (В ЗАВИСИМОСТИ ОТ) фразы OCCURS (ПОВТОРЯЕТСЯ) было равным максимальному числу вхождений. Переменно повторяющимися данными являются следующие:

а) групповое данное, содержащее переменно повторяющееся данное;

б) переменно повторяющееся данное;

в) данное, подчиненное переменно повторяющемуся данному.

Если фраза VALUE (ЗНАЧЕНИЕ) указана для данного, указанного в варианте DEPENDING ON (В ЗАВИСИМОСТИ ОТ), то это значение инициируется после инициации значений данного с переменным числом вхождений.

(6) Если фраза VALUE (ЗНАЧЕНИЕ) формата 1 указана в статье описания данного, содержащей фразу OCCURS (ПОВТОРЯЕТСЯ), или в статье, подчиненной статье с фразой OCCURS (ПОВТОРЯЕТСЯ), то каждому вхождению элемента данного будет присвоено указанное значение.

6. РАЗДЕЛ ПРОЦЕДУР В ЯДРЕ

6.1. *Общее описание*

Раздел процедур содержит процедуры, которые должна выполнять объектная программа (см. ч. 4, п. 6.4). Раздел процедур не обязателен в исходной Кобол-программе.

GOST EXPERT ГОСТ 22558-89,

Общие форматы раздела процедур ядра показаны ниже.

Формат 1

PROCEDURE DIVISION.

{имя-секции SECTION.

[имя-параграфа.

[предложение] ...] ... }...

РАЗДЕЛ ПРОЦЕДУР.

{СЕКЦИЯ имя-секции.

[имя-параграфа.

[предложение] ...] ... }...

Формат 2

PROCEDURE DIVISION.

{имя-параграфа.

[предложение] ... }...

РАЗДЕЛ ПРОЦЕДУР

{имя-параграфа.

[предложение] ... }...

6.2. **Арифметические выражения**

6.2.1. Определение арифметического выражения

Арифметическим выражением может быть идентификатор числового элементарного данного; числовой литерал; стандартная константа ZERO (ZEROS, ZEROES) (НУЛЬ (НУЛИ)), а также идентификаторы, стандартные константы, литералы и выражения, разделенные знаками арифметических операций, либо арифметическое выражение, заключенное в скобки. Любому арифметическому выражению может предшествовать унарная операция. Допустимые комбинации переменных, числовых литералов, скобок и знаков арифметических операций приведены в п. 6.2.3 настоящей части. Все идентификаторы и литералы, встречающиеся в арифметическом выражении, должны представлять либо числовые элементарные данные, либо числовые литералы.

6.2.2. Знаки арифметических операций

Имеется пять знаков бинарных арифметических операций и два знака унарных арифметических операций, которые могут быть использованы в арифметических выражениях. Они изображаются специальными литерами, с обеих сторон которых должен стоять пробел. Ниже приведены знаки операций и их смысл.

GOST EXPERT ГОСТ 22558-89,

Знак бинарной операции	Смысл операции
+	Сложение
—	Вычитание
*	Умножение
/	Деление
**	Возведение в степень

Знак унарной операции	Смысл операции
+	Соответствует умножению на числовой литерал +1
—	Соответствует умножению на числовой литерал —1

6.2.3. Правила вычисления арифметических выражений

Арифметические выражения вычисляются по таким правилам:

(1) для определения порядка выполнения указанных в арифметических выражениях действий могут применяться скобки. Выражения внутри скобок вычисляются первыми; внутри вложенных скобок вычисление происходит, начиная от самых внутренних скобок к внешним. Если скобки не используются или выражения, заключенные в скобки, находятся на одном и том же уровне вложенности, применяется следующий порядок старшинства выполнения операций: первая по старшинству операция — унарный плюс или минус; вторая — возведение в степень, третья — умножение или деление, четвертая — сложение или вычитание;

(2) скобки используются в следующих случаях: для исключения логической неоднозначности, при появлении последовательных операций одного и того же иерархического уровня или для изменения последовательности выполнения операций, установленной порядком старшинства. Если последовательность выполнения не определена скобками, порядок выполнения последовательных операций одного и того же иерархического уровня определен слева направо;

(3) ниже указаны способы, которыми идентификаторы, литералы, операции и скобки могут образовывать арифметическое выражение.

GOST EXPERT ГОСТ 22558-89,

Первый элемент арифметического выражения	Второй элемент арифметического выражения				
	Идентификатор или литерал	+ — * / **	Унарный плюс или минус	(	)
Идентификатор или литерал	—	Р	—	—	Р
*/ ** + —	Р	—	Р	Р	—
Унарный плюс или минус	Р	—	—	Р	—
(	Р	—	Р	Р	—
)	—	Р	—	—	Р

Примечание. Буква Р представляет допустимую пару элементов, знак «—» — недопустимую;

(4) арифметическое выражение может начинаться только с идентификатора, литерала или символов (, +, — и может заканчиваться только), идентификатором или литералом. Между открывающими и закрывающими скобками в арифметическом выражении должно быть взаимно однозначное соответствие, так что каждая открывающая скобка должна быть слева от соответствующей ей закрывающей скобки. Если первая операция в арифметическом выражении унарная и это арифметическое выражение непосредственно следует за идентификатором или другим арифметическим выражением, ей должна непосредственно предшествовать левая скобка;

(5) вычисление экспоненты в арифметическом выражении производится по следующим правилам:

а) если значение выражения, которое должно быть возведено в степень, равно нулю, степень должна иметь значение большее нуля. В противном случае возникает условие переполнения (п. 6.4.2 настоящей части);

б) если в результате вычисления может получиться и положительное, и отрицательное число, в качестве результата возвращается положительное;

в) если не существует действительного числа, которое может быть результатом вычисления, возникает условие переполнения;

(6) арифметические выражения позволяют пользователю комбинировать арифметические операции без ограничений на совокупность операндов и (или) получаемых данных. В каждой реализации указываются способы, которые используются в обработке арифметических выражений.

GOST EXPERT ГОСТ 22558-89,

6.3. Условные выражения

Условные выражения задают условия, которые проверяются в объектной программе для выбора альтернативных путей управления в зависимости от значения истинности условия. Условное выражение имеет значение истинности «истина» или «ложь». Условные выражения задаются в операторах EVALUATE (ОЦЕНИТЬ), IF (ЕСЛИ), PERFORM (ВЫПОЛНИТЬ) и SEARCH (ИСКАТЬ). Есть две категории условий, связанных с условными выражениями: простые условия и сложные условия. Каждое из них может заключаться в любое число парных скобок, в результате чего категория условия не меняется.

6.3.1. Простые условия

Простыми условиями являются условия отношения, класса, имени-условия, состояния-переключателя и знака. Простое условие может иметь значение истинности «истина» или «ложь». Заключение в скобки простых условий не меняет их значения истинности.

6.3.1.1. *Условия отношения*

Условие отношения вызывает сравнение двух операндов, каждый из которых может быть либо данным, представленным идентификатором, либо литералом, либо значением арифметического выражения, либо именем индекса. Условие отношения имеет значение истинности «истина», если между операндами имеет место указанное отношение. Допускается сравнение двух числовых операндов независимо от формата, определяемого индивидуальными фразами об использовании. Однако для всех других случаев сравнения описания операндов должны задавать одинаковое использование. Если хоть один из операндов является групповым данным, сравнение выполняется по правилам сравнения нечисловых операндов.

Общий формат для условия отношения следующий:

GOST EXPERT ГОСТ 22558-89,

GOST EXPERT ГОСТ 22558-89,

Первый операнд (идентификатор-1, литерал-1 [, арифметическое-выражение-1] или имя-индекса-1) называется субъектом условия; второй операнд (идентификатор-2, литерал-2 [, арифметическое-выражение-2] или имя-индекса-2) называется объектом условия. Условие отношения должно содержать по крайней мере один идентификатор.

Знак операции отношения задает тип сравнения, которое должно быть произведено в условии отношения. Знаки операций отношения и зарезервированные слова должны обрамляться пробелами. NOT (НЕ) (если оно используется) и последующее ключевое слово или литера отношения представляют один знак операции отношения, определяющий тип сравнения для определения значения истинности. Следующие знаки операций отношения эквивалентны: IS NOT GREATER THAN (НЕ БОЛЬШЕ) эквивалентно IS LESS THAN OR EQUAL TO (МЕНЬШЕ ИЛИ РАВНО); IS NOT LESS THAN (НЕ МЕНЬШЕ) эквивалентно IS GREATER THAN OR EQUAL TO (БОЛЬШЕ ИЛИ РАВНО). Смысл знаков операций отношения показан ниже.

Значение	Знак операции отношения	
Больше или не больше	IS [NOT] GREATER THAN IS [NOT] >	[НЕ] БОЛЬШЕ [НЕ] >
Меньше или не меньше	IS [NOT] LESS THAN IS [NOT] <	[НЕ] МЕНЬШЕ [НЕ] <
Равно или не равно	IS [NOT] EQUAL TO IS [NOT] =	[НЕ] РАВНО [НЕ] =
Больше или равно	IS GREATER THAN OR EQUAL TO IS > =	БОЛЬШЕ ИЛИ РАВНО > =
Меньше или равно	IS LESS THAN OR EQUAL TO IS < =	МЕНЬШЕ ИЛИ РАВНО < =

6.3.1.1.1. *Сравнение числовых операндов*

Для операндов числовой категории производится сравнение алгебраических значений операндов. Длина литерала или операндов

GOST EXPERT . ГОСТ 22558-89,

арифметического выражения в терминах числа цифр не существенна. Нуль рассматривается как единственное значение, не зависящее от знака.

Сравнение числовых операндов допускается независимо от способа описания их использования. Числовые операнды, не имеющие знака, рассматриваются при сравнении как положительные.

6.3.1.1.2. *Сравнение нечисловых операндов*

Для нечисловых операндов или для случая, когда один операнд числовой, а второй — нечисловой, производится сравнение в соответствии с указанной основной последовательностью литер. В последнем случае числовой операнд должен быть целым литералом или данным, описанным как целое и:

(1) если нечисловой операнд является элементарным данным или нечисловым литералом, числовой операнд рассматривается так, как будто выполнено его перемещение в элементарное буквенно-цифровое данное того же размера, что и числовое данное (в терминах литер стандартного формата данных); значение этого буквенно-цифрового данного затем сравнивается с нечисловым операндом (см. п. 5.9.4, правило (8), п. 6.19 настоящей части);

(2) если нечисловой операнд является групповым данным, числовой операнд рассматривается так, как будто выполнено его перемещение в групповое данное того же размера, что и числовое данное (в терминах литер стандартного формата данных), и значение этого группового данного затем сравнивается с нечисловым операндом (см. п. 5.9.4, правило (8), п. 6.19 настоящей части);

(3) числовой операнд, не представляющий целое число, не может сравниваться с нечисловым операндом.

Размер операнда равен числу литер стандартного формата данных в операнде. Числовые и нечисловые операнды могут сравниваться только тогда, когда их использование явно или неявно является одинаковым.

Правила сравнения для операндов равного размера и операндов неравного размера различны.

(1) Если операнды имеют равный размер, то литеры операндов в соответствующих позициях сравниваются попарно, начиная от самой левой позиции, пока не встретятся неравные литеры или не будет достигнут правый конец.

Если все пары литер совпадают вплоть до последней пары, операнды считаются равными.

Для первой появившейся пары несовпадающих литер определяется их относительная позиция в основной последовательности. Считается, что операнд, содержащий литеру, занимающую более высокую позицию в основной последовательности, является большим.

GOST EXPERT ГОСТ 22558-89,

(2) Если операнды имеют разный размер, сравнение литер производится так, как будто операнд, содержащий меньшее число литер, дополнен справа пробелами до получения операндов одинаковой длины.

6.3.1.1.3. *Сравнения, содержащие имена индексов и (или) индексные данные*

Проверка отношения может быть выполнена только между:

(1) двумя именами индексов. Результат такой же, как если бы сравнивались номера вхождений;

(2) именем индекса и данным (отличным от индексного данного) или литералом. Номер вхождения, который соответствует имени индекса, сравнивается с данным или литералом;

(3) индексным данным и именем индекса или другим индексным данным. Значения сравниваются без преобразования.

6.3.1.2. *Условие класса*

Условие класса определяет, является ли операнд числовым, буквенным, буквенным прописным, буквенным строчным или содержит лишь литеры из множества литер, заданного фразой CLASS (КЛАСС) параграфа SPECIAL-NAMES (СПЕЦИАЛЬНЫЕ-ИМЕНА) раздела оборудования. Класс операнда определяется следующим образом:

(1) операнд числовой, если он состоит из литер 1, 2, 3 . . . , 9,0 со знаком или без;

(2) операнд буквенный, если он состоит полностью из прописных букв, пробелов, из строчных букв и пробела, или любой комбинации прописных и строчных букв или любой комбинации прописных, строчных букв и пробела;

(3) операнд буквенный строчный, если он полностью состоит из строчных букв и пробела;

(4) операнд буквенный прописной, если он полностью состоит из прописных букв и пробелов;

(5) операнд соответствует имени-класса, если он состоит лишь из литер, перечисленных в определении имени-класса в параграфе SPECIAL-NAMES (СПЕЦИАЛЬНЫЕ-ИМЕНА).

Общий формат для условия класса следующий:

идентификатор-1 IS [NOT] { NUMERIC / ALPHABETIC / ALPHABETIC-LOWER / ALPHABETIC-UPPER / имя-класса-1 }

GOST EXPERT ГОСТ 22558-89,

идентификатор-1 [НЕ] { ЧИСЛОВОЕ / БУКВЕННОЕ / СТРОЧНЫЕ / ПРОПИСНЫЕ / имя-класса-1 }

Проверяемый операнд должен быть описан с использованием DISPLAY (ДЛЯ ВЫДАЧИ).

NOT (НЕ) (если оно используется) и следующее за ним ключевое слово задает одно условие класса, которое определяет выполняемую для установления истинности проверку. NOT NUMERIC (НЕ ЧИСЛОВОЕ) указывает проверку истинности для установления того, является ли операнд нечисловым.

Проверка на NUMERIC (ЧИСЛОВОЕ) не может применяться к данному, описанному как буквенное или как групповое данное, составленное из элементарных данных, описания которых указывают на наличие знака числа. Если описание проверяемого данного не содержит указания на знак числа, то проверяемое данное будет определено как числовое только в том случае, когда оно является числовым без знака. Если описание данного содержит указание на знак числа, проверяемое данное будет определено как числовое только в том случае, если его значение является числовой величиной и имеет правильный знак. Правильным знаком числового данного, описанного с фразой SIGN IS SEPARATE (ЗНАК ОТДЕЛЬНО), является одна из литер + или — в стандартном формате данных; представление правильного знака данных, описание которых не содержит фразу SIGN IS SEPARATE (ЗНАК ОТДЕЛЬНО), определяется реализацией.

Проверка на ALPHABETIC (БУКВЕННОЕ) не может применяться для данных, описанных как числовые. Результат проверки «истина», если содержимое данного, представленного идентификатором-1, состоит только из буквенных литер.

Проверка на ALPHABETIC-LOWER (СТРОЧНЫЕ) не может применяться для данных, описанных как числовые. Результат проверки «истина», если содержимое данного, представленного идентификатором-1, состоит только из строчных букв и пробелов.

Проверка на ALPHABETIC-UPPER (ПРОПИСНЫЕ) не может применяться для данных, описанных как числовые. Результат проверки «истина», если содержимое данного, представленного идентификатором-1, состоит только из прописных букв и пробелов.

Проверка с использованием имени-класса-1 не может применяться для данных, описанных как числовые.

GOST EXPERT ГОСТ 22558-89,

6.3.1.3. *Условие имени-условия (условная переменная)*

Для условной переменной в условии имени-условия проверяется равенство ее значения одному из значений, связанных с именем-условия.

Общий формат для условия имени-условия:

имя-условия-1

Если имя-условия связано с диапазоном или с несколькими диапазонами значений, то для условной переменной проверяется, попадает ли ее значение в диапазоны, включая концы диапазонов.

Правила для сравнения условной переменной со значением имени-условия те же, что и для условий отношения.

Результатом проверки является значение «истина», если одно из значений, соответствующих имени-условия-1, равно значению соответствующей условной переменной.

6.3.1.4. *Условие состояния переключателя*

Условие состояния-переключателя определяет, является ли состояние определяемого реализацией переключателя состоянием «включено» или «выключено». Имя-реализации и связанное с ним состояние «включено» или «выключено» должны быть названы в параграфе SPECIAL-NAMES (СПЕЦИАЛЬНЫЕ-ИМЕНА) раздела оборудования. Общий формат для условия состояния-переключателя:

имя-условия-1

Результат проверки есть «истина», если переключатель установлен в позицию, соответствующую имени-условия-1.

6.3.1.5. *Условие знака*

Условие знака определяет, является ли алгебраическое значение арифметического выражения меньшим нуля, большим нуля или равным нулю. Общий формат для условия знака:

арифметическое-выражение-1 IS [NOT] { POSITIVE / NEGATIVE / ZERO }

арифметическое-выражение-1 [НЕ] { ПОЛОЖИТЕЛЬНО / ОТРИЦАТЕЛЬНО / НУЛЬ }

NOT (НЕ) (если оно используется) и следующее за ним ключевое слово определяют одно условие знака, которое задает алгебраическую проверку для определения значения истинности;

GOST EXPERT ГОСТ 22558-89,

например, NOT ZERO (НЕ НУЛЬ) является проверкой истинности на ненулевое (положительное или отрицательное) значение.

Операнд положителен, если его значение больше нуля, отрицателен, если его значение меньше нуля, и равен нулю, если его значение равно нулю. Арифметическое выражение должно содержать по крайней мере один идентификатор.

6.3.2. Сложные условия

Сложные условия образуются комбинацией простых условий и (или) сложных условий с помощью логических связок (логических операций AND (И) и OR (ИЛИ)) или отрицания этих условий с помощью логического отрицания (логическая операция NOT (НЕ)). Значением истинности сложного условия, заключенного или незаключенного в скобки, является то значение, которое следует в результате выполнения установленных логических операций над отдельными значениями истинности простых условий.

Знаки логических операций и их значения следующие:

AND (И) — логическая конъюнкция; значение истинности будет «истина», если оба соединенных ею условия истинны; «ложь», если одно или оба соединенных условия ложны;

OR (ИЛИ) — логическая дизъюнкция; значение истинности есть «истина», если одно или оба соединенных этой связкой условий истинны; «ложь», если оба эти условия ложны;

NOT (НЕ) — логическое отрицание или инверсия значения истинности; значение истинности есть «истина», если соответствующее условие ложно; «ложь», если это условие истинно. До и после знаков логических операций должен стоять пробел.

6.3.2.1. *Отрицание условий*

Отрицание условия производится с помощью логической операции NOT (НЕ), которая отрицает значение истинности условия, к которому применяется. Таким образом, значением истинности отрицания условия будет «истина» тогда и только тогда, когда значение истинности исходного условия «ложь»; значением истинности отрицания условия будет «ложь» тогда и только тогда, когда значение истинности исходного условия «истина»; заключение отрицания условия в скобки не меняет его значение истинности.

Общий формат отрицания условия:

NOT условие-1

НЕ условие-1

6.3.2.2. *Комбинированные условия*

Комбинированные условия получаются посредством связывания условий одним из знаков логических операций AND (И) или OR (ИЛИ). Общий формат комбинированных условий:

GOST EXPERT ГОСТ 22558-89,

условие-1 { { AND / OR } условие-2 } ...

условие-1 { { И / ИЛИ } условие-2 } ...

6.3.2.3. *Старшинство логических операций и использование скобок*

При отсутствии парных скобок в сложном условии старшинство (т. е. степень связывания) логических операций задает условия, к которым применяются заданные логические операции, и предполагает эквивалентное связывание скобками. Порядок старшинства следующий: NOT (НЕ), AND (И), OR (ИЛИ). Таким образом условие вида «условие-1 OR NOT условие-2 AND условие-3» («условие-1 ИЛИ НЕ условие-2 И условие-3») эквивалентно условию «условие-1 OR ((NOT условие-2) AND условие-3)» («условие-1 ИЛИ ((НЕ условие-2) И условие-3)»).

Когда в сложном условии используются скобки, они определяют связывание условий со знаками логических операций. Таким образом, скобки могут использоваться для изменения обычного старшинства операций, как указано выше. Например, приведенное выше сложное условие будет иметь другой смысл, если его задать в виде (условие-1 OR (NOT условие-2)) AND условие-3 ((условие-1 ИЛИ (НЕ условие-2)) И условие-3) (п. 6.3.4 настоящей части).

В табл. 1 приведены способы комбинирования условий и логических операций и правила употребления скобок. Между открывающими и закрывающими скобками должно быть соответствие.

Как видно из табл. 1, пара элементов OR NOT (ИЛИ НЕ) является допустимой, в то время как пара NOT OR (НЕ ИЛИ) недопустима; пара NOT ((НЕ ( ) допустима, а NOT NOT (НЕ НЕ) недопустима.

6.3.3. Сокращенные комбинированные условия отношений

Если простые условия отношения или их отрицания комбинируются с помощью логических связок в последовательности, при которой условия отношения содержат совпадающие с предыдущим субъект или субъект и знак операции отношения и при этом не используются скобки, то любое условие отношения, за исключением первого, может быть сокращено следующим образом:

(1) может быть опущен субъект условия отношения;

(2) могут быть опущены субъект и знак операции отношения в условии отношения.

GOST EXPERT ГОСТ 22558-89,

Таблица 1

Элемент	Может располагаться в условном выражении		Последовательность элементов слева направо	
	первым	последним	Элементу, если он не первый, может непосредственно предшествовать только	За элементом, если он не последний, может непосредственно следовать только
Простое-условие	Да	Да	OR (ИЛИ), NOT (НЕ) AND (И), (	OR (ИЛИ), AND (И),)
OR (ИЛИ) AND (И)	Нет	Нет	Простое-условие,)	Простое-условие, NOT (НЕ), (
NOT (НЕ)	Да	Нет	OR (ИЛИ), AND (И), (	Простое-условие, (
(	Да	Нет	OR (ИЛИ), NOT (НЕ) AND (И), (	Простое-условие, NOT (НЕ), (
)	Нет	Да	Простое-условие,)	OR (ИЛИ), AND (И),)

Формат для сокращенного условия

условие-отношения $\left\{\left\{\begin{matrix}\underline{\text{AND}}\\ \underline{\text{OR}}\end{matrix}\right\}\right.$ [NOT] [знак-операции-отношения] объект$\}$ ...

условие-отношения $\left\{\left\{\begin{matrix}\underline{\text{И}}\\ \underline{\text{ИЛИ}}\end{matrix}\right\}\right.$ [НЕ] [знак-операции-отношения] объект$\}$ ...

Внутри последовательности условий отношения могут использоваться оба вида приведенных выше сокращений. Действие такого сокращенного условия такое же, как если бы опущенные части были взяты из ближайшего предшествующего полного условия отношения внутри того же предложения. Соответствия сокращенных и полных условий приведены ниже.

Вставка опущенного субъекта и (или) операции отношения завершается, как только в сложном условии встречается полное простое условие.

Интерпретация, применяемая к слову NOT (НЕ) в сокращенном комбинированном условии отношения, следующая:

(1) если за словом NOT (НЕ) непосредственно следует знак операции GREATER (БОЛЬШЕ) (>), LESS (МЕНЬШЕ) (<), EQUAL (РАВНО) (=), тогда NOT (НЕ) является частью знака операции отношения;

(2) в остальных случаях NOT (НЕ) интерпретируется как знак логической операции и, следовательно, подразумеваемая

GOST EXPERT ГОСТ 22558-89,

вставка субъекта или операции отношения приводит к отрицанию условия отношения.

Ниже приводятся примеры сокращений для комбинированных условий отношения и их отрицаний с соответствующими эквивалентами.

Сокращенное комбинированное условие_отношения	Эквивалент
a>b AND NOT <c OR d А>Б И НЕ<Г ИЛИ Д	((a>b) AND (a NOT<c)) OR (a NOT<d) ((А>Б) И (А НЕ<Г)) ИЛИ (А НЕ<Д)
a NOT EQUAL b OR c А НЕ РАВНО Б ИЛИ Г	(a NOT EQUAL b) OR (a NOT EQUAL c) (А НЕ РАВНО Б) ИЛИ (А НЕ РАВНО Г)
NOT a=b OR c НЕ А=Б ИЛИ Г	(NOT (a=b)) OR (a=c) (НЕ (А=Б)) ИЛИ (А=Г)
NOT (a GREATER b OR <c) НЕ (А БОЛЬШЕ Б ИЛИ <Г)	NOT ((a GREATER b) OR (a<c)) НЕ ((А БОЛЬШЕ Б) ИЛИ (А<Г))
NOT (a NOT>b AND c AND NOT d) НЕ (А НЕ>Б И Г И НЕ Д)	NOT ((((a NOT>b) AND (a NOT>c)) AND (NOT (a NOT>d)))) НЕ ((((А НЕ>Б) И (А НЕ>Г)) И (НЕ (А НЕ>Д))))

6.3.4. Порядок вычисления условий

Скобки явные и неявные задают уровень вложенности в сложном условии. Два или более условий, связанных только знаком логической операции AND (И) или только знаком логической операции OR (ИЛИ) на одном и том же уровне вложенности, устанавливают один уровень иерархии в сложном условии. Таким образом, полное сложное условие может рассматриваться как вложенная структура уровней иерархии. При этом само сложное условие соответствует самому внешнему уровню иерархии. В этом контексте вычисление условий в полном сложном условии начинается слева и продолжается по следующим правилам, рекурсивно применимым, когда необходимо.

(1) На одном уровне иерархии связанные составляющие условия вычисляются в порядке слева направо. Вычисление на данном иерархическом уровне завершается сразу, как только значение истинности для него становится определенным независимо от того, все ли связанные в рамках этого уровня составляющие условия вычислены.

Значения арифметических выражений вычисляются тогда и только тогда, когда вычисляются содержащие их условия. Ана-

GOST EXPERT ГОСТ 22558-89,

логично, отрицания условий вычисляются тогда и только тогда, когда необходимо вычислить сложное условие, которое они задают (см. п. 6.2.3 настоящей части).

Применение вышеперечисленных правил показано на рис. 1—4. Эти схемы не определяют требования к реализации.

6.4. Общие фразы и правила для форматов операторов

Ниже приводится описание общих фраз и условий, которые относятся к нескольким различным операторам.

6.4.1. Фраза ROUNDED (ОКРУГЛЯЯ)

Если после выравнивания по десятичной точке число позиций в дробной части результата арифметической операции оказывается большим, чем число позиций, заданных для дробной части идентификатора результата, производится усечение цифр в соответствии с размером, заданным для идентификатора результата. Если при этом старшая усекаемая цифра больше или равна 5, то абсолютное значение идентификатора результата при наличии фразы ROUNDED (ОКРУГЛЯЯ) увеличивается на единицу в самом младшем разряде.

Если младшие позиции целого в значении идентификатора результата представляются в его шаблоне литерой P(M), то производится округление или отбрасывание цифр относительно самой правой из позиций цифр, для которых отведена память.

6.4.2. Фраза ON SIZE ERROR (ПРИ ПЕРЕПОЛНЕНИИ)

Условие переполнения возникает в следующих случаях:

(1) нарушение правил вычисления степени всегда приводит к завершению выполнения арифметической операции и всегда вызывает условие переполнения (см. п. 6.2.3 настоящей части);

(2) деление на нуль всегда вызывает условие переполнения;

(3) если после выравнивания позиции десятичной точки абсолютное значение результата превышает наибольшее значение, которое может содержаться в соответствующем идентификаторе результата, возникает ошибка переполнения.

В случае, если для результирующего идентификатора задана фраза USAGE IS BINARY (ДВОИЧНОЕ), наибольшее значение, которое может в нем содержаться, определяется на основании строки-литер фразы PICTURE (ШАБЛОН) для соответствующего десятичного числа. Если задана фраза ROUNDED (ОКРУГЛЯЯ), округление выполняется перед проверкой условия переполнения.

Если задана фраза ON SIZE ERROR (ПРИ ПЕРЕПОЛНЕНИИ) и после выполнения арифметических операций, заданных в арифметическом операторе, возникает условие переполнения, значения соответствующих результирующих идентификаторов оста-

GOST EXPERT ГОСТ 22558-89,

ются такими же, какими они были до начала вычисления арифметического выражения. Значения результирующих идентификаторов, для которых не возникало условие переполнения, будут такими же, как если бы условие переполнения не влияло бы ни на какой результирующий идентификатор. После завершения арифметической операции управление передается повелительному оператору, заданному во фразе ON SIZE ERROR (ПРИ ПЕРЕПОЛНЕНИИ), и выполнение продолжается в соответствии с правилами для каждого оператора, указанного в этом повелительном операторе. Если им является оператор ветвления процедур или условный оператор, который сам вызывает явную передачу управления, управление передается в соответствии с правилами для данного оператора, иначе после завершения выполнения повелительного оператора, заданного во фразе ON SIZE ERROR (ПРИ ПЕРЕПОЛНЕНИИ) управление передается на конец арифметического оператора, а фраза NOT ON SIZE ERROR (БЕЗ ПЕРЕПОЛНЕНИЯ), если она и задана, игнорируется.

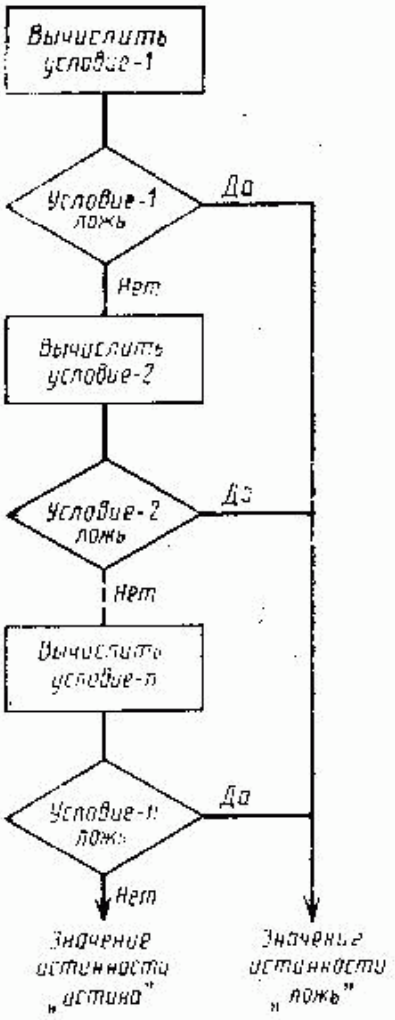


Рис. 1. Вычисление уровня иерархии

условие-1 AND условие-2 AND ... условие-n
(условие-1 И условие-2 И ... условие-n)

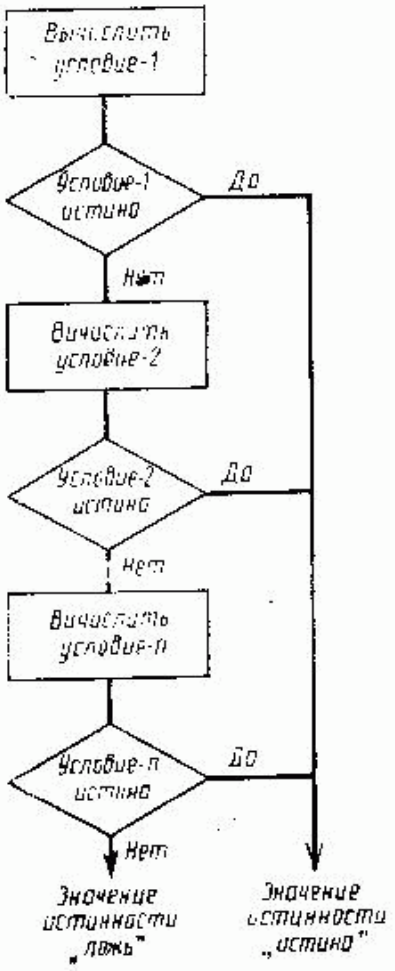


Рис. 2. Вычисление уровня иерархии

условие-1 OR условие-2 OR ... условие-n
(условие-1 ИЛИ условие-2 ИЛИ ... условие-n)

GOST EXPERT ГОСТ 22558-89,

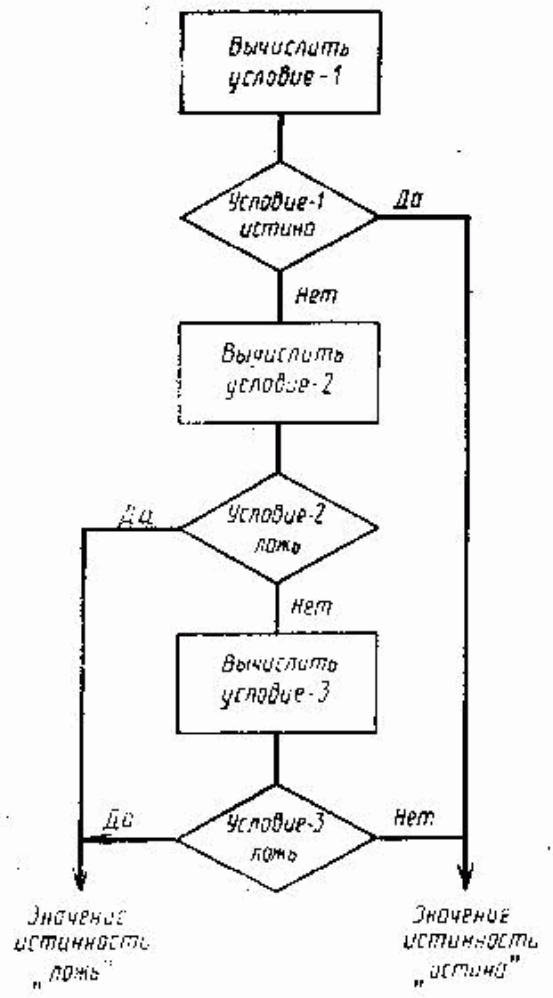


Рис. 3. Вычисление выражения
условие-1 OR условие-2 AND условие-3
(условие-1 ИЛИ условие-2 И условие-3)

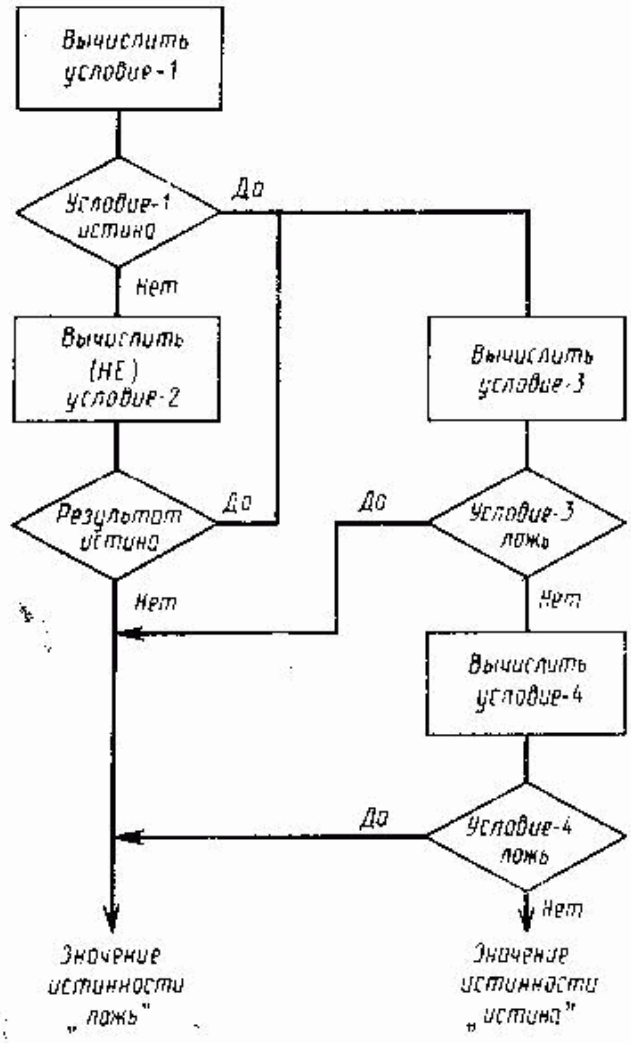


Рис. 4. Вычисление выражения
(условие-1 OR NOT условие-2) AND условие-3 AND условие-4 ((условие-1 ИЛИ НЕ условие-2) И условие-3 И условие-4)

Если фраза ON SIZE ERROR (ПРИ ПЕРЕПОЛНЕНИИ) не задана и во время выполнения арифметической операции в арифметическом операторе возникает условие переполнения, значения результирующих идентификаторов не определены. Значения результирующих идентификаторов, для которых ситуация переполнения не возникала, будут такими же, как если бы ситуация переполнения не возникала бы ни для какого результирующего идентификатора. После завершения арифметических операций управление передается на конец арифметического оператора, а фраза NOT ON SIZE ERROR (БЕЗ ПЕРЕПОЛНЕНИЯ), если она и задана, игнорируется.

Если при выполнении арифметических операций в арифметическом операторе переполнения не возникает, фраза ON SIZE ERROR (ПРИ ПЕРЕПОЛНЕНИИ), если она и задана, игнорируется, а управление передается на конец арифметического операто-

GOST EXPERT ГОСТ 22558-89,

ра или, если задана фраза NOT ON SIZE ERROR (БЕЗ ПЕРЕПОЛНЕНИЯ), повелительному оператору, определенному в этой фразе. Во втором случае выполнение продолжается в соответствии с правилами выполнения каждого оператора, указанного в этом повелительном операторе. Если выполняется оператор ветвления процедуры или условный оператор, управление передается в соответствии с правилами этого оператора, в противном случае после завершения выполнения повелительного оператора, заданного во фразе NOT ON SIZE ERROR (БЕЗ ПЕРЕПОЛНЕНИЯ), управление передается на конец арифметического оператора.

Если при выполнении операторов ADD (СЛОЖИТЬ) и SUBTRACT (ОТНЯТЬ) с фразой CORRESPONDING (СООТВЕТСТВЕННО) хоть одна отдельная операция вырабатывает условие переполнения, повелительный-оператор фразы ON SIZE ERROR (ПРИ ПЕРЕПОЛНЕНИИ) не выполняется до тех пор, пока не завершатся все сложения или вычитания.

6.4.3. Фраза CORRESPONDING (СООТВЕТСТВЕННО)

Идентификаторы групповых данных для удобства изложения обозначим d_1 и d_2. Два данных, одно из d_1 и одно из d_2, определяются как соответствующие, если выполняются следующие условия:

(1) данное в d_1 и данное в d_2 не определяются ключевым словом FILLER (ЗАПОЛНИТЕЛЬ) и имеют одно и то же имя и одни и те же уточнители вплоть до d_1 и d_2, но исключая последние;

(2) в случае использования оператора MOVE CORRESPONDING (ПОМЕСТИТЬ СООТВЕТСТВЕННО) по крайней мере одно из данных является элементарным и соответствующая пересылка допустима по правилам пересылки; оба данных являются элементарными числовыми данными в случае оператора ADD CORRESPONDING (СЛОЖИТЬ СООТВЕТСТВЕННО) или SUBTRACT CORRESPONDING (ОТНЯТЬ СООТВЕТСТВЕННО);

(3) ни d_1, ни d_2 не могут быть описаны с номером уровня 66, 77 или 88 или с фразой USAGE IS INDEX (ДЛЯ ИНДЕКСА);

(4) данные, подчиненные d_1 или d_2 и содержащие в описании одну из фраз REDEFINES (ПЕРЕОПРЕДЕЛЯЕТ), RENAMES (ПЕРЕИМЕНОВЫВАЕТ), OCCURS (ПОВТОРЯЕТСЯ), USAGE IS INDEX (ДЛЯ ИНДЕКСА), или подчиненные данным, описанным с такими фразами, не рассматриваются как соответствующие. Ссылки на d_1 и d_2 не могут модифицироваться;

(5) имя каждого данного, удовлетворяющего вышеперечисленным условиям, должно быть однозначным, с учетом его подразумеваемого уточнения.

GOST EXPERT ГОСТ 22558-89,

6.4.4. Арифметические операторы

Арифметическими операторами являются операторы ADD (СЛОЖИТЬ), COMPUTE (ВЫЧИСЛИТЬ), DIVIDE (РАЗДЕЛИТЬ), MULTIPLY (УМНОЖИТЬ), SUBTRACT (ОТНЯТЬ). Они имеют несколько общих особенностей.

(1) Описания операндов могут быть различными. При вычислении обеспечивается необходимое преобразование и выравнивание по десятичной точке.

(2) Максимальный размер каждого операнда — 18 десятичных цифр. Композиция операндов, представляющая собой гипотетическое данное, порождаемое суперпозицией указанных операндов оператора, выровненных по их десятичным точкам, не должна содержать более 18 цифр (пп. 6.6, 6.11, 6.20, 6.26 настоящей части).

6.4.5. Перекрывающиеся операнды

Если посылаемое и принимающее данное в любом операторе имеют общую часть или всю область памяти, то, даже если они не определены в одной статье описания данного, результат выполнения такого оператора не определен. Кроме того, будут не определены результаты выполнения некоторых операторов, в которых посылаемые и принимающие данные определены одной и той же статьей описания данного. Эти случаи будут рассматриваться в общих правилах соответствующих операторов.

6.4.6. Несколько результатов арифметических операторов

Операторы ADD (СЛОЖИТЬ), COMPUTE (ВЫЧИСЛИТЬ), DIVIDE (РАЗДЕЛИТЬ), MULTIPLY (УМНОЖИТЬ) и SUBTRACT (ОТНЯТЬ) могут иметь несколько результатов.

Эти операторы выполняются так, как если бы они были заданы следующим образом:

(1) оператор, при выполнении которого осуществляется доступ к данным, которые являются частью начальных вычислений этого оператора, выполняет необходимые арифметические операции или комбинирование этих данных и запоминает результат этой операции в промежуточном поле памяти. Элементы, которые принимают участие в начальном вычислении, определяются правилами конкретных операторов;

(2) выполняется последовательность операторов, выполнение которых передает или комбинирует значение в этом промежуточном поле с каждым отдельным результирующим данным. Эти операторы рассматриваются, как если бы они были записаны слева направо в той же последовательности, в которой заданы множественные результаты.

Результат оператора

ADD a, b, c, TO c, d(c), e

(СЛОЖИТЬ А, Б, М, С М, Д(М), Е) эквивалентен результатам

GOST EXPERT ГОСТ 22558-89,

следующих операторов:
ADD a, b, c GIVING temp
ADD temp TO c
ADD temp TO d(c)
ADD temp TO e
(СЛОЖИТЬ А, Б, М ПОЛУЧАЯ пром
СЛОЖИТЬ пром С М
СЛОЖИТЬ пром С Д (М)
СЛОЖИТЬ пром С Е)
и результат оператора
MULTIPLY a(i) BY i, a(i)
УМНОЖИТЬ А(М) НА М, А(М)
эквивалентен
MOVE a(i) TO temp
MULTIPLY temp BY i
MULTIPLY temp BY a(i)
(ПОМЕСТИТЬ А(М) в пром
УМНОЖИТЬ пром НА М
УМНОЖИТЬ пром НА А(М)
где в обоих случаях «temp» (пром) есть обозначение промежуточного результата, обеспечиваемого реализацией.

6.4.7. Несовместимые данные

Если значения данных, к которым происходит обращение в разделе процедур, не соответствуют классу, определенному при описании этих данных с фразой PICTURE (ШАБЛОН), результат обращения не определен. Исключение составляет условие класса (см. п. 6.3.1.2 настоящей части).

6.5. **Оператор ACCEPT (ПРИНЯТЬ)**

6.5.1. Назначение

Оператор ACCEPT (ПРИНЯТЬ) позволяет передать небольшой объем информации от некоторого устройства в указанное данное.

6.5.2. Общий формат

Формат 1

```
ACCEPT идентификатор-1      [FROM мнемоническое-имя-1]
ПРИНЯТЬ идентификатор-1     [С мнемоническое-имя-1]
```

Формат 2

```
ACCEPT идентификатор-2 FROM { DATE
                            { DAY
                            { DAY-OF-WEEK
                            { TIME
```


GOST EXPERT ГОСТ 22558-89,

ПРИНЯТЬ В идентификатор-2 { ДАТУ | ДЕНЬ | ДЕНЬ-НЕДЕЛИ | ВРЕМЯ }

6.5.3. Синтаксическое правило

Мнемоническое-имя-1 должно указываться в параграфе SPECIAL-NAMES (СПЕЦИАЛЬНЫЕ-ИМЕНА) раздела оборудования и должно связываться с устройством оборудования.

6.5.4. Общие правила

Формат 1

(1) Оператор ACCEPT (ПРИНЯТЬ) вызывает передачу данного с указанного устройства; это данное замещает значение данного, представленного идентификатором-1.

Любое преобразование данных, требуемое при их перемещении с внешнего устройства в данное, представленное идентификатором-1, определяется реализацией.

(2) Размер одной передачи для каждого устройства оборудования определяет реализация.

(3) Если устройство может передавать данные того же размера, что и размер принимающих данных, передаваемое данное запоминается в поле принимающего данного.

(4) Если устройство не может передавать данные такого же размера, как и принимающие данные, то:

а) если размер принимающего данного (или части принимающего данного, еще не занятой передаваемым данным) превышает размер передаваемого данного, передаваемое данное запоминается в принимающем поле выровненным влево (или в части принимающего поля, которая еще не занята) и запрашивается дополнительное данное. На уровне 1 осуществляется только одна передача данных;

б) если размер передаваемого данного превышает размер принимающего поля (или части принимающего поля, еще не занятой передаваемым данным), в поле принимающего данного или в оставшейся его части запоминаются только самые левые позиции передаваемого данного; остальные его литеры, не помещающиеся в принимающем данном, игнорируются.

(5) если фраза FROM (С) не задана, то используется устройство, которое определяется реализацией как стандартное.

GOST EXPERT ГОСТ 22558-89,

(6) при использовании формата 2 оператор ACCEPT (ПРИНЯТЬ) вызывает передачу запрашиваемой информации в данное, заданное идентификатором-2, согласно правилам оператора MOVE (ПОМЕСТИТЬ); дата, день, день недели и время автоматически обеспечиваются реализацией и не описываются в Кобол-программе.

(7) дата содержит год столетия, месяц года и день месяца.

Последовательность кодов, представляющая DATE (ДАТУ), такова, как если бы это данное было описано как элементарное целое числовое данное размером в шесть цифр, первые две из которых представляют год столетия, следующие две — месяц года, и последние две — число месяца. Например, дата 1 июля 1987 года изображается последовательностью цифр 870701.

(8) день содержит год столетия и порядковый номер дня в году.

Последовательность кодов, представляющая DAY (ДЕНЬ), такова, как если бы это данное было описано как элементарное целое числовое данное размером в пять цифр, первые две из которых (слева направо) представляют год столетия и последние три — порядковый номер дня в году. Например, 1 июля 1987 года представляется последовательностью цифр 87182.

(9) Время содержит часы, минуты, секунды и сотые доли секунды. Время отсчитывается на 24-часовой основе, начиная от полуночи. Последовательность кодов, представляющих TIME (ВРЕМЯ), такова, как если бы это данное было описано как элементарное целое число без знака размером в 8 цифр, из которых первые две (слева направо) представляют часы, следующие две — минуты, следующие две — секунды и последние две — сотые доли секунды. Минимальное значение данного (ВРЕМЯ) равно 00000000, максимальное — 23595999. Если система не поддерживает возможности работы с долями секунды, те разряды, которые не могут быть определены, полагаются равными нулю.

(10) DAY-OF-WEEK (ДЕНЬ-НЕДЕЛИ) состоит из одного данного, содержимое которого представляет день недели. DAY-OF-WEEK (ДЕНЬ-НЕДЕЛИ) доступен Кобол-программе, как если бы он был описан в ней как элементарное целое числовое данное без знака из одной цифры.

В DAY-OF-WEEK (ДЕНЬ-НЕДЕЛИ) значением 1 задается понедельник, значением 2 вторник, . . . , 7 — воскресенье.

6.6. **Оператор ADD (СЛОЖИТЬ)**

6.6.1. Назначение

Оператор ADD (СЛОЖИТЬ) позволяет просуммировать два или более числовых операнда и запомнить результат.

GOST EXPERT ГОСТ 22558-89,

6.6.2. Общий формат

Формат 1

ADD { идентификатор-1 / литерал-1 } ...
  TO {идентификатор-2 [ROUNDED]} ...
  [ON SIZE ERROR повелительный-оператор-1]
  [NOT ON SIZE ERROR повелительный-оператор-2]
  [END-ADD]

СЛОЖИТЬ { идентификатор-1 / литерал-1 } ...
  С {идентификатор-2 [ОКРУГЛЯЯ]} ...
  [ПРИ ПЕРЕПОЛНЕНИИ повелительный-оператор-1]
  [БЕЗ ПЕРЕПОЛНЕНИЯ повелительный-оператор-2]
  [КОНЕЦ-СЛОЖИТЬ]

Формат 2

ADD { идентификатор-1 / литерал-1 } ... TO { идентификатор-2 / литерал-2 }
  GIVING {идентификатор-3 [ROUNDED]} ...
  [ON SIZE ERROR повелительный-оператор-1]
  [NOT ON SIZE ERROR повелительный-оператор-2]
[END-ADD]

СЛОЖИТЬ { идентификатор-1 / литерал-1 } ... С { идентификатор-2 / литерал-2 }
  ПОЛУЧАЯ {идентификатор-3 [ОКРУГЛЯЯ]} ...
  [ПРИ ПЕРЕПОЛНЕНИИ повелительный-оператор-1]
  [БЕЗ ПЕРЕПОЛНЕНИЯ повелительный-оператор-2]
  [КОНЕЦ-СЛОЖИТЬ]

Формат 3

ADD { CORRESPONDING / CORR } идентификатор-1
  TO идентификатор-2 [ROUNDED]
  [ON SIZE ERROR повелительный-оператор-1]

GOST EXPERT ГОСТ 22558-89,

[NOT ON SIZE ERROR повелительный-оператор-2]
[END-ADD]

СЛОЖИТЬ { СООТВЕТСТВЕННО / СООТВ } идентификатор-1 С
идентификатор-2
[ОКРУГЛЯЯ] [ПРИ ПЕРЕПОЛНЕНИИ
повелительный-оператор-1]
[БЕЗ ПЕРЕПОЛНЕНИЯ повелительный-оператор-2]
[КОНЕЦ-СЛОЖИТЬ]

6.6.3. Синтаксические правила

(1) В форматах 1 и 2 каждый идентификатор должен представлять элементарное числовое данное, за исключением идентификаторов, которые следуют за словом GIVING (ПОЛУЧАЯ) и должны представлять элементарное числовое или элементарное числовое редактируемое данное. В формате 3 каждый идентификатор должен представлять групповое данное.

(2) Каждый литерал должен быть числовым литералом.

(3) Максимальный размер каждого операнда — восемнадцать десятичных цифр.

а) В формате 1 композиция операндов определяется в результате суммирования всех операндов данного оператора.

б) В формате 2 композиция операндов определяется путем использования всех операндов оператора, кроме следующих за словом GIVING (ПОЛУЧАЯ).

в) В формате 3 композиция операндов определяется отдельно для каждой пары соответствующих данных.

(4) CORR (СООТВ) есть сокращение слова CORRESPONDING (СООТВЕТСТВЕННО).

6.6.4. Общие правила

(1) Если используется формат 1, то значения операндов, предшествующих слову TO (С), складываются вместе, затем сумма запоминается в промежуточном поле памяти. Значение этого промежуточного поля складывается со значением данного, определенного идентификатором-2, и результат запоминается в поле, заданном идентификатором-2. Этот процесс повторяется для каждого последующего вхождения идентификатора-2 в той же последовательности слева направо, в которой заданы вхождения идентификатора-2.

GOST EXPERT ГОСТ 22558-89,

(2) Если используется формат 2, то значения операндов, предшествующих слову GIVING (ПОЛУЧАЯ), складываются вместе, затем сумма запоминается как новое значение каждого из результирующих идентификаторов, заданных посредством идентификатора-3.

(3) Если используется формат 3, данные из группы, определенной идентификатором-1, складываются с соответствующими данными из группы, определенной идентификатором-2 и запоминаются в них.

(4) Выделение достаточного поля для выполнения вычислений без потери значащих цифр обеспечивается реализацией.

(5) Дополнительные правила и объяснения, относящиеся к этому оператору, приводятся в соответствующих параграфах (см. ч. 4, п. 6.4.3; пп. 6.4.1—6.4.6 настоящей части).

6.7. Оператор ALTER (ИЗМЕНИТЬ)

6.7.1. Назначение

Оператор ALTER (ИЗМЕНИТЬ) модифицирует определенную ранее последовательность операторов.

Оператор ALTER (ИЗМЕНИТЬ) в настоящем стандарте является устаревшим элементом, и будет удален в следующей редакции стандарта.

6.7.2. Общий формат

ALTER {имя-процедуры-1 TO [PROCEED TO]
имя-процедуры-2} [...]
ИЗМЕНИТЬ {имя-процедуры-1 [ДЛЯ ПЕРЕХОДА] К
имя-процедуры-2} [...]

6.7.3. Синтаксические правила

(1) Имя-процедуры-1 является именем параграфа, который содержит только одно предложение, состоящее из оператора GO TO (ПЕРЕЙТИ) без фразы DEPENDING ON (В ЗАВИСИМОСТИ ОТ).

(2) Имя-процедуры-2 является именем параграфа или секции в разделе процедур.

6.7.4. Общие правила

(1) Во время выполнения объектной программы оператор ALTER (ИЗМЕНИТЬ) модифицирует оператор GO TO (ПЕРЕЙТИ) в параграфе, названном имя-процедуры-1, заменяя указанное в нем имя-процедуры именем-процедуры-2. Модифицируемые операторы GO TO (ПЕРЕЙТИ) в независимых сегментах могут в некоторых случаях восстанавливаться в начальное состояние (ч. 16, п. 1.4.3).

(2) К оператору GO TO (ПЕРЕЙТИ), указанному в секции с номером сегмента, большим или равным 50, нельзя обращаться в

GOST EXPERT ГОСТ 22558-89,

операторе ALTER (ИЗМЕНИТЬ), указанном в секции с другим номером сегмента.

Все другие применения оператора ALTER (ИЗМЕНИТЬ) являются допустимыми и выполняются даже тогда, когда оператор GO TO (ПЕРЕЙТИ), на который ссылается оператор ALTER (ИЗМЕНИТЬ), находится в еще не вызванном для выполнения сегменте программы (ч. 16).

6.8. Оператор COMPUTE (ВЫЧИСЛИТЬ)

6.8.1. Назначение

Оператор COMPUTE (ВЫЧИСЛИТЬ) присваивает одному или нескольким данным значение арифметического выражения.

6.8.2. Общий формат

COMPUTE {идентификатор-1 [ROUNDED]}... =
арифметическое-выражение-1
[ON SIZE ERROR повелительный-оператор-1]
[NOT ON SIZE ERROR повелительный-оператор-2]
[END-COMPUTE]

ВЫЧИСЛИТЬ {идентификатор-1 [ОКРУГЛЯЯ]}... =
арифметическое-выражение-1
[ПРИ ПЕРЕПОЛНЕНИИ повелительный-оператор-1]
[БЕЗ ПЕРЕПОЛНЕНИЯ повелительный-оператор-2]
[КОНЕЦ-ВЫЧИСЛИТЬ]

6.8.3. Синтаксические правила

(1) Идентификатор-1 должен ссылаться либо на элементарное числовое данное, либо на элементарное числовое редактируемое данное.

6.8.4. Общие правила

(1) Арифметическое выражение, состоящее из единственного идентификатора или литерала, позволяет установить значение идентификатора-1 равным значению идентификатора или литерала.

(2) Если для результата оператора указано несколько идентификаторов, предшествующих знаку равенства, то после вычисления значения арифметического выражения оно запоминается как новое значение каждого из данных, заданных посредством идентификатора-1.

(3) Оператор COMPUTE (ВЫЧИСЛИТЬ) позволяет сочетать без ограничений на композиции операндов и результирующих данных арифметические операции, производимые арифметическими операторами ADD (СЛОЖИТЬ), SUBTRACT (ОТНЯТЬ), MULTIPLY (УМНОЖИТЬ) и DIVIDE (РАЗДЕЛИТЬ).

GOST EXPERT ГОСТ 22558-89,

Каждая реализация определяет приемы обработки арифметических выражений.

(4) Дополнительные правила и объяснения, относящиеся к этому оператору, даются в соответствующих параграфах (см. ч. 4, п. 6.4.3; пп. 6.4.1, 6.4.2 6,4.4—6.4.6 настоящей части).

6.9. **Оператор CONTINUE (ПРОДОЛЖИТЬ)**

6.9.1. Назначение

Оператор CONTINUE (ПРОДОЛЖИТЬ) задает отсутствие операции. Он указывает, что никакой выполнимый оператор не присутствует.

6.9.2. Общий формат

CONTINUE

ПРОДОЛЖИТЬ

6.9.3. Синтаксическое правило

(1) Оператор CONTINUE (ПРОДОЛЖИТЬ) может использоваться в любом месте условного оператора, в котором может использоваться повелительный оператор.

6.9.4. Общее правило

(1) Оператор CONTINUE (ПРОДОЛЖИТЬ) не влияет на выполнение программы.

6.10. **Оператор DISPLAY (ВЫДАТЬ)**

6.10.1. Назначение

Оператор DISPLAY (ВЫДАТЬ) обеспечивает передачу небольшого объема данных на некоторое подходящее устройство оборудования.

6.10.2. Общий формат

DISPLAY { идентификатор-1 / литерал-1 } ... [UPON мнемоническое-имя-1]
[WITH NO ADVANCING]

ВЫДАТЬ { идентификатор-1 / литерал-1 } ... [НА мнемоническое-имя-1]
[БЕЗ ПРОДВИЖЕНИЯ]

6.10.3. Синтаксические правила

(1) Мнемоническое-имя-1 связывается с устройством оборудования в параграфе SPECIAL-NAMES (СПЕЦИАЛЬНЫЕ-ИМЕНА) раздела оборудования.

(2) Если литерал-1 числовой, то он должен быть целым без знака.

GOST EXPERT ГОСТ 22558-89,

6.10.4. Общие правила

(1) В результате выполнения оператора DISPLAY (ВЫДАТЬ) значение каждого операнда передается на устройство оборудования в порядке, указанном в списке. Любое преобразование данных, которое может потребоваться при выводе идентификатора-1 или литерала-1 на внешнее устройство, определяется реализацией.

(2) Размер одной передачи для каждого устройства определяется реализацией.

(3) Если в качестве одного из операндов указана стандартная константа, то выдается только единственное ее вхождение.

(4) Если устройство может принимать данные такого же размера, как размер передаваемого данного, то данное передается на устройство.

(5) Если устройство оборудования не может принимать данных такого размера, как размер передаваемого данного, выполняется одно из следующих действий:

а) если размер передаваемого данного превышает размер данного, которое может принять устройство за одну передачу, то данное выдается на устройство, начиная с самых левых литер, выровненным влево, а оставшиеся данные затем передаются согласно общим правилам (4) и (5) до тех пор, пока не будут переданы все данные. На уровне 1 разрешается только одна передача данных;

б) если размер данного, которое может принять устройство, превышает размер передаваемого данного, передаваемое данное на принимающем устройстве запоминается выровненным влево.

(6) Если оператор DISPLAY (ВЫДАТЬ) содержит более одного операнда, размер пересылаемого данного определяется как сумма размеров соответствующих операндов, и значения операндов выдаются в последовательности, соответствующей перечислению операндов без изменения положения текущей позиции внешнего устройства при переходе на последующие операнды.

(7) Если фраза UPON (НА) не используется, то используется стандартное выводное устройство реализации.

(8) Если задана фраза WITH NO ADVANCING (БЕЗ ПРОДВИЖЕНИЯ), положение текущей позиции устройства оборудования после вывода последнего операнда не будет автоматически переноситься на следующую строку либо изменяться любым другим способом. Если устройство способно устанавливать текущее положение на конкретные позиции литер, оно остается позиционированным на позицию литеры, непосредственно следующую за последней литерой последнего выданного операнда. Если устройство не обеспечивает установку положения текущей позиции на заданные позиции литер, можно управлять (если возможно) лишь вертикальным позиционированием. Если уст-

GOST EXPERT ГОСТ 22558-89,

ройство оборудования допускает наложение печати, это может вызвать такое наложение.

(9) Если фраза WITH NO ADVANCING (БЕЗ ПРОДВИЖЕНИЯ) не задана, то после того как последний операнд будет передан на устройство оборудования, положением текущей позиции устройства станет самая левая позиция следующей выводимой устройством строки.

(10) Если устройство оборудования не поддерживает вертикальное позиционирование, операционная система будет игнорировать явно или неявно заданное вертикальное позиционирование.

6.11. Оператор DIVIDE (РАЗДЕЛИТЬ)

6.11.1. Назначение

Оператор DIVIDE (РАЗДЕЛИТЬ) позволяет разделить числовое данное и присвоить некоторым данным значения частного и остатка.

6.11.2. Общий формат

Формат 1

DIVIDE { идентификатор-1 / литерал-1 } INTO {идентификатор-2
[ROUNDED]}...
[ON SIZE ERROR повелительный-оператор-1]
[NOT ON SIZE ERROR повелительный-оператор-2]
[END-DIVIDE]

РАЗДЕЛИТЬ НА { идентификатор-1 / литерал-1 } {идентификатор-2
{ОКРУГЛЯЯ]}...
[ПРИ ПЕРЕПОЛНЕНИИ повелительный-оператор-1]
{БЕЗ ПЕРЕПОЛНЕНИЯ повелительный-оператор-2]
[КОНЕЦ-РАЗДЕЛИТЬ]

Формат 2

DIVIDE { идентификатор-1 / литерал-1 } INTO { идентификатор-2 / литерал-2 }
GIVING {идентификатор-3 [ROUNDED]}...
[ON SIZE ERROR повелительный-оператор-1]
[NOT ON SIZE ERROR повелительный-оператор-2]
[END-DIVIDE]

GOST EXPERT ГОСТ 22558-89,

РАЗДЕЛИТЬ НА { идентификатор-1 / литерал-1 } { идентификатор-2 / литерал-2 }
ПОЛУЧАЯ {идентификатор-3 [ОКРУГЛЯЯ]}...
[ПРИ ПЕРЕПОЛНЕНИИ повелительный-оператор-1]
[БЕЗ ПЕРЕПОЛНЕНИЯ повелительный-оператор-2]
[КОНЕЦ-РАЗДЕЛИТЬ]

Формат 3

DIVIDE { идентификатор-1 / литерал-1 } BY { идентификатор-2 / литерал-2 }
GIVING {идентификатор-3 [ROUNDED]}...
[ON SIZE ERROR повелительный-оператор-1]
[NOT ON SIZE ERROR повелительный-оператор-2]
[END-DIVIDE]

РАЗДЕЛИТЬ { идентификатор-1 / литерал-1 } НА { идентификатор-2 / литерал-2 }
ПОЛУЧАЯ {идентификатор-3 [ОКРУГЛЯЯ]}...
[ПРИ ПЕРЕПОЛНЕНИИ повелительный-оператор-1]
[БЕЗ ПЕРЕПОЛНЕНИЯ повелительный-оператор-2]
[КОНЕЦ-РАЗДЕЛИТЬ]

Формат 4

DIVIDE { идентификатор-1 / литерал-1 } INTO { идентификатор-2 / литерал-2 }
GIVING идентификатор-3 [ROUNDED]
REMAINDER идентификатор-4
[ON SIZE ERROR повелительный-оператор-1]
[NOT ON SIZE ERROR повелительный-оператор-2]
[END-DIVIDE

РАЗДЕЛИТЬ НА { идентификатор-1 / литерал-1 } { идентификатор-2 / литерал-2 }
ПОЛУЧАЯ идентификатор-3 [ОКРУГЛЯЯ]
ОСТАТОК идентификатор-4
[ПРИ ПЕРЕПОЛНЕНИИ повелительный-оператор-1]

GOST EXPERT ГОСТ 22558-89,

[БЕЗ ПЕРЕПОЛНЕНИЯ повелительный-оператор-2]

[КОНЕЦ-РАЗДЕЛИТЬ]

Формат 5

DIVIDE { идентификатор-1 / литерал-1 } BY { идентификатор-2 / литерал-2 }

GIVING идентификатор-3

REMAINDER идентификатор-4

[ON SIZE ERROR повелительный-оператор-1]

[NOT ON SIZE ERROR повелительный-оператор-2]

[END-DIVIDE]

РАЗДЕЛИТЬ { идентификатор-1 / литерал-1 } НА { идентификатор-2 / литерал-2 }

ПОЛУЧАЯ идентификатор-3 [ОКРУГЛЯЯ] ОСТАТОК идентификатор-4

[ПРИ ПЕРЕПОЛНЕНИИ повелительный-оператор-1]

[БЕЗ ПЕРЕПОЛНЕНИЯ повелительный-оператор-2]

[КОНЕЦ-РАЗДЕЛИТЬ]

6.11.3. Синтаксические правила

(1) Каждый идентификатор должен относиться к числовому элементарному данному, за исключением идентификаторов, указанных после слов GIVING (ПОЛУЧАЯ) [или REMAINDER (ОСТАТОК),] которые могут представлять элементарное числовое редактируемое данное или элементарное числовое данное.

(2) Каждый литерал должен быть числовым.

(3) Максимальный размер каждого операнда или их композиции не должен превышать восемнадцати десятичных цифр.

6.11.4. Общие правила

(1) Если используется формат 1, литерал-1 или значение идентификатора-1 запоминается во временном данном. Далее значение идентификатора-2 делится на значение этого данного. Значение делимого (значение идентификатора-2) заменяется этим частным; аналогично значение каждого последующего вхождения идентификатора-2 делится в порядке перечисления на вышеупомянутое временное данное, и полученное частное замещает значение делимого.

(2) Если используется формат 2, литерал-2 или значение идентификатора-2 делится на литерал-1 или значение идентификатора-1. Результат запоминается в каждом вхождении идентификатора-3.

GOST EXPERT ГОСТ 22558-89,

(3) Если используется формат 3, литерал-1 или значение идентификатора-1 делится на литерал-2 или значение идентификатора-2 и результат запоминается в каждом вхождении идентификатора-3.

(4) Если используется формат 4, литерал-2 или значение идентификатора-2 делится на литерал-1, или значение идентификатора-2 и результат запоминается как значение идентификатора-3. Далее вычисляется остаток, и результат запоминается как значение идентификатора-4. Если идентификатор-4 индексирован, индексы вычисляются непосредственно перед присваиванием ему значения.

(5) Если используется формат 5, литерал-1 или значение идентификатора-1 делится на литерал-2 или значение идентификатора-2. Далее выполнение деления продолжается как в вышеописанном формате 4.

(6) Форматы 4 и 5 используются, когда требуется запомнить остаток операции деления как значение идентификатора-4. Остаток в Коболе определен как результат вычитания произведения частного (идентификатор-3) на делитель из делимого. Если идентификатор-3 определен как числовое редактируемое данное, для вычисления остатка используется промежуточное поле, содержащее неотредактированное частное. Если используется фраза ROUNDED (ОКРУГЛЯЯ), для вычисления остатка используется промежуточное поле, которое содержит неокругленное частное оператора DIVIDE (РАЗДЕЛИТЬ), но с отброшенными избыточными позициями. Промежуточное поле определяется как числовое, которое содержит столько же разрядов, такое же положение десятичной точки и то же наличие или отсутствие знака, что и частное (идентификатор-3).

(7) Если используются форматы 4 и 5, точность остатка определяется вычислением, описанным выше. Если необходимо, при запоминании данного, представляемого идентификатором-4, производится выравнивание по десятичной точке и усечение (без округления).

(8) При использовании в форматах 4 и 5 фразы ON SIZE ERROR (ПРИ ПЕРЕПОЛНЕНИИ) имеют место следующие правила:

а) если имеет место переполнение частного и указана фраза ON SIZE ERROR (ПРИ ПЕРЕПОЛНЕНИИ), значение остатка не определено. При этом значения данных, определяемых идентификатором-3 и идентификатором-4, остаются неизменными;

б) если имеет место переполнение остатка и указана фраза ON SIZE ERROR (ПРИ ПЕРЕПОЛНЕНИИ), значение данного, представленного идентификатором-4, не меняется. При на-

GOST EXPERT ГОСТ 22558-89,

личии нескольких результатов пользователь должен сам определить, какая именно ситуация встретилась.

(9) Дополнительные правила и объяснения, относящиеся к этому оператору, приводятся в соответствующих параграфах (см. ч. 4, п. 6.4.3 , а также ч. 6, пп. 6.4.1, 6.4.2, 6.4.4—6.4.6).

6.12. **Оператор ENTER (ВОЙТИ)**

6.12.1. Назначение

Оператор ENTER (ВОЙТИ) обеспечивает возможность использования в одной и той же программе нескольких языков программирования.

Оператор ENTER (ВОЙТИ) является устаревшим элементом в этом стандарте и будет удален из следующей редакции стандарта.

6.12.2. Общий формат

ENTER имя-языка-1 [имя-программного-модуля-1].

ВОЙТИ В имя-языка-1 [имя-программного-модуля-1].

6.12.3. Синтаксические правила

(1) Имя-языка может относиться к любому из языков программирования, определенных реализацией как языки, в которые можно войти через Кобол. Имя-языка-1 определяется реализацией.

(2) Имя-программного-модуля-1 является словом Кобола, к которому можно обращаться только в операторе ENTER (ВОЙТИ).

(3) После последнего оператора другого языка должно следовать предложение ENTER COBOL (ВОЙТИ В КОБОЛ), указывающее компилятору точку возврата в исходную Кобол-программу.

6.12.4. Общие правила

(1) Операторы другого языка выполняются в объектной программе так, как если бы они компилировались в объектную программу вслед за оператором ENTER (ВОЙТИ).

(2) Все детали записи в других языках определяются реализацией.

(3) Если операторы языка, указанного именем-языка, не могут быть записаны непосредственно вслед за оператором ENTER (ВОЙТИ), то имя-программного-модуля указывает порцию программы в этом языке, выполняемую в данном месте программы. Если указанные операторы могут быть приведены непосредственно, имя-программного-модуля-1 не указывается.

6.13. **Оператор EVALUATE (ОЦЕНИТЬ)**

6.13.1. Назначение

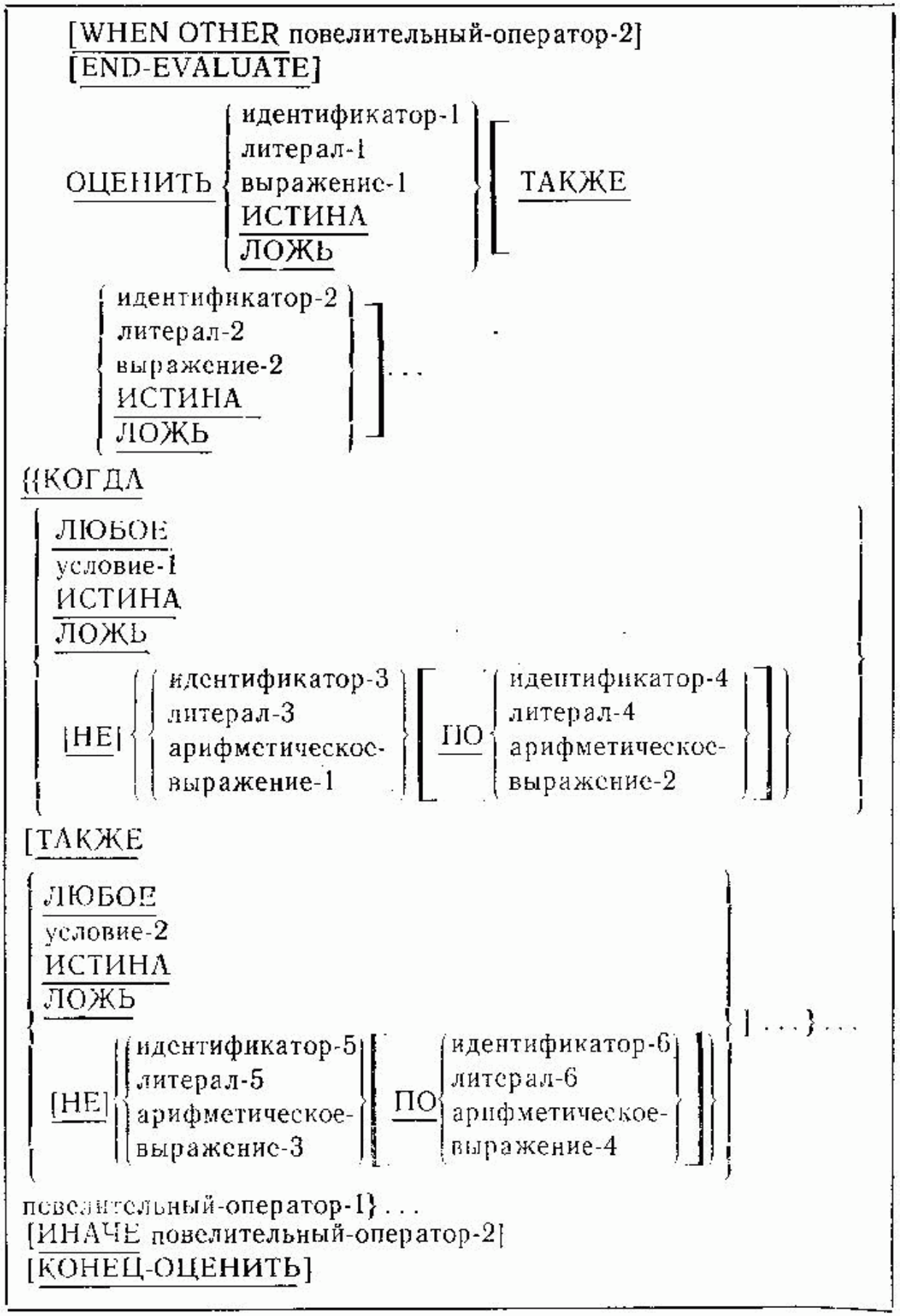
Оператор EVALUATE (ОЦЕНИТЬ) описывает структуры со многими ветвлениями и соединениями. Он может вызывать оценку нескольких условий. Соответствующие действия объектной программы зависят от результатов этих оценок.

GOST EXPERT ГОСТ 22558-89,

6.13.2. Общий формат

```
EVALUATE { идентификатор-1 | литерал-1 | выражение-1 | TRUE | FALSE }
    [ ALSO { идентификатор-2 | литерал-2 | выражение-2 | TRUE | FALSE } ] ...
{{WHEN
  { ANY | условие-1 | TRUE | FALSE
  | [NOT] { { идентификатор-3 | литерал-3 | арифметическое-выражение-1 }
      [ { THROUGH | THRU } { идентификатор-4 | литерал-4 | арифметическое-выражение-2 } ] } ... }
[ALSO
  { ANY | условие-2 | TRUE | FALSE
  | [NOT] { { идентификатор-5 | литерал-5 | арифметическое-выражение-3 }
      [ { THROUGH | THRU } { идентификатор-6 | литерал-6 | арифметическое-выражение-4 } ] } ... } ...
  повелительный-оператор-1} ...
```


GOST EXPERT ГОСТ 22558-89,



GOST EXPERT ГОСТ 22558-89,

6.13.3. Синтаксические правила

(1) Каждый из операндов или слов TRUE (ИСТИНА) и FALSE (ЛОЖЬ), заданных перед первой фразой WHEN (КОГДА) оператора EVALUATE (ОЦЕНИТЬ), является субъектом выбора, а все вместе — множеством субъектов выбора.

(2) Каждый из операндов или слов TRUE (ИСТИНА), FALSE (ЛОЖЬ), и ANY (ЛЮБОЕ), заданные во фразе WHEN (КОГДА) оператора EVALUATE (ОЦЕНИТЬ), рассматриваются от отдельности как объект выбора, а указанные вместе в одной фразе WHEN (КОГДА)—как множество объектов выбора.

(3) Слова THROUGH и THRU эквивалентны.

(4) Оба операнда одной фразы THROUGH (ПО) должны быть одного и того же класса. Связанные таким образом два операнда составляют один объект выбора.

(5) Количество объектов выбора в каждом множестве объектов выбора должно быть равно количеству субъектов выбора.

(6) Каждый объект выбора в множестве объектов выбора должен соответствовать субъекту выбора, имеющему ту же порядковую позицию во множестве субъектов выбора, согласно следующим правилам:

а) идентификаторы, литералы или арифметические выражения, встречающиеся в объектах выбора, должны быть правильными операндами для сравнения с соответствующими операндами во множестве субъектов выбора (см. п. 6.3.11 настоящей части);

б) условие-1, условие-2 или слова TRUE (ИСТИНА) или FALSE (ЛОЖЬ), заданные как объекты выбора, должны соответствовать условным выражениям или словам TRUE (ИСТИНА) или FALSE (ЛОЖЬ) во множестве субъектов выбора;

в) слово ANY (ЛЮБОЕ) может соответствовать субъекту выбора любого типа.

6.13.4. Общие правила

(1) Оператор EVALUATE (ОЦЕНИТЬ) действует так, как если бы все объекты и субъекты выбора были вычислены и получили бы числовые или нечисловые значения, или значения диапазонов числовых или нечисловых значений, или значений истинности. Эти значения определяются следующим образом:

а) любому субъекту выбора, заданному идентификатором-1, идентификатором-2, и любому объекту выбора, заданному идентификатором-3, идентификатором-5, не содержащему фраз NOT (НЕ) или THRU (ПО), присваивается значение и класс данного, на который ссылается идентификатор;

б) любому субъекту выбора, заданному литералом-1, литералом-2, и любому объекту выбора, заданному литералом-3, ли-

GOST EXPERT ГОСТ 22558-89,

тералом-5 и не содержащему фраз NOT (НЕ) или THRU (ПО), присваивается значение и класс соответствующего литерала. Если литерал-2, литерал-5 является стандартной константой ZERO (НУЛЬ), ему присваивается класс соответствующего субъекта выбора;

в) любому субъекту выбора, заданному арифметическим выражением выражение-1, выражение-2, и любому объекту выбора без фраз NOT (НЕ) и THRU (ПО), представленному арифметическим-выражением-1, арифметическим-выражением-3, присваивается числовое значение в соответствии с правилами вычисления арифметических выражений (см. п. 6.2 настоящей части);

г) любому субъекту выбора, в котором выражение-1, выражение-2 являются условными выражениями, и любому объекту выбора, в котором заданы условие-1, условие-2, присваивается значение истинности в соответствии с правилами вычисления условных выражений (см. п. 6.3 настоящей части);

д) любому субъекту выбора и любому объекту выбора, заданному словами TRUE (ИСТИНА) или FALSE (ЛОЖЬ), присваивается значение истинности. Значение истинности «истина» присваивается элементам, для которых задано слово TRUE (ИСТИНА) и значение истинности «ложь» присваивается элементам, для которых задано слово FALSE (ЛОЖЬ);

е) любой объект выбора, заданный словом ANY (ЛЮБОЕ), далее не вычисляется;

ж) если фраза THRU (ПО) задана для объекта выбора без фразы NOT (НЕ), диапазон значений включает все допустимые значения субъекта выбора, которые больше или равны первому операнду и меньше или равны второму в соответствии с правилами сравнения (см. п. 6.3.1.1 настоящей части);

з) если для объекта выбора задана фраза NOT (НЕ), значениями, соответствующими этому элементу, являются все допустимые значения субъекта выбора, не равные значению или не содержащиеся в диапазоне значений, которые соответствовали бы элементу, если бы не была задана фраза NOT (НЕ).

(2) Далее выполнение оператора EVALUATE (ОЦЕНИТЬ) производится так, как если бы значения, присвоенные субъектам и объектам выбора, сравнивались, чтобы определить, удовлетворяют ли значения субъектов условиям, указанным соответствующими фразами WHEN (КОГДА). Это сравнение производится следующим образом:

а) каждый объект выбора из множества объектов выбора первой фразы WHEN (КОГДА) сравнивается с субъектом выбора, имеющим тот же порядковый номер во множестве субъек-

GOST EXPERT ГОСТ 22558-89,

тов выбора. Чтобы сравнение было удовлетворено, должно выполняться одно из следующих условий:

1) если сравниваемым элементам присваиваются числовые или нечисловые значения или диапазоны числовых или нечисловых значений, сравнение удовлетворяется, если в соответствии с правилами сравнения значение или одно из значений из диапазона, присвоенного объекту выбора, равно значению, присвоенному субъекту выбора (см. п. 6.3.1.1 настоящей части);

2) если сравниваемым элементам присваиваются значения истинности, сравнение удовлетворяется, если сравниваемым элементам соответствуют одинаковые значения истинности;

3) если сравниваемый объект выбора задан словом ANY (ЛЮБОЕ), сравнение удовлетворяется всегда, независимо от значения субъекта выбора;

б) если выше указанное сравнение удовлетворяется для каждого объекта выбора в сравниваемом множестве объектов выбора, фраза WHEN (КОГДА), содержащая это множество, считается удовлетворяющей множеству субъектов выбора;

в) если вышеуказанное сравнение не удовлетворяется для одного или более объектов выбора из сравниваемого множества объектов, это множество не удовлетворяет множеству субъектов выбора;

г) эти действия повторяются для множеств объектов выбора в том порядке, в котором эти множества задаются в исходной программе, до тех пор, пока либо встретится фраза WHEN (КОГДА), удовлетворяющая множеству субъектов выбора, либо будут исчерпаны все множества объектов выбора.

(3) После завершения операции сравнения, выполнение оператора EVALUATE (ОЦЕНИТЬ) продолжается следующим образом:

а) если выбрана некоторая фраза WHEN (КОГДА), начинается выполнение повелительного-оператора-1, следующего за этой фразой;

б) если ни одна фраза WHEN (КОГДА) не выбрана и задана фраза WHEN OTHER (ИНАЧЕ), выполнение продолжается с повелительного-оператора-2;

в) выполнение оператора EVALUATE (ОЦЕНИТЬ) завершается, когда выполнение достигает или конца повелительного-оператора-1 выбранной фразы WHEN (КОГДА), или конца повелительного-оператора-2, или если нет выбранной фразы WHEN (КОГДА), а фраза WHEN OTHER (ИНАЧЕ) не указана (см. ч. 4, п. 6.4.3).

GOST EXPERT ГОСТ 22558-89,

6.14. **Оператор EXIT (ВЫЙТИ)**

6.14.1. Назначение

Оператор EXIT (ВЫЙТИ) обеспечивает общую точку выхода для ряда процедур.

6.14.2. Общий формат

EXIT.

ВЫЙТИ.

6.14.3. Синтаксическое правило

(1) Оператор EXIT (ВЫЙТИ) должен составлять предложение, не содержащее других операторов, и быть единственным в параграфе.

6.14.4. Общее правило

(1) Оператор EXIT (ВЫЙТИ) служит для того, чтобы поставить в соответствие данной точке программы имя-процедуры. Оператор EXIT (ВЫЙТИ) не оказывает другого действия на компиляцию или выполнение программы.

6.15. **Оператор GO TO (ПЕРЕЙТИ)**

6.15.1. Назначение

Оператор GO TO (ПЕРЕЙТИ) приводит к передаче управления от одной точки раздела процедур к другой. Необязательность имени-процедуры-1 в формате 1 оператора GO TO (ПЕРЕЙТИ) в данном стандарте является устаревшей и поэтому будет устранена в следующей редакции стандарта.

6.15.2. Общий формат

Формат 1

GO TO [имя-процедуры-1]

ПЕРЕЙТИ К [имя-процедуры-1]

Формат 2

GO TO {имя-процедуры-1} . . . DEPENDING ON идентификатор-1

ПЕРЕЙТИ К {имя-процедуры-1} . . . В ЗАВИСИМОСТИ ОТ идентификатор-1

6.15.3. Синтаксические правила

(1) Идентификатор-1 представляет числовое элементарное данное, описанное как целое.

(2) Если к параграфу, содержащему оператор GO TO (ПЕРЕЙТИ), имеется обращение в операторе ALTER (ИЗМЕНИТЬ), то параграф может состоять только из заголовка параграфа, за которым следует единственный оператор GO TO (ПЕРЕЙТИ) в формате-1.

GOST ГОСТ 22558-89,

(3) Формат 1 оператора GO TO (ПЕРЕЙТИ) без имени-процедуры-1 разрешен только в параграфе, содержащем этот единственный оператор.

(4) Если оператор GO TO (ПЕРЕЙТИ), представленный форматом 1, появляется в последовательности повелительных операторов внутри предложения, он должен быть последним оператором в этой последовательности.

6.15.4. Общие правила

(1) При выполнении оператора GO TO (ПЕРЕЙТИ), представленного форматом 1, управление передается указанному имени-процедуры-1.

(2) Если имя-процедуры в формате 1 не определено, то оператор ALTER (ИЗМЕНИТЬ), ссылающийся на оператор GO TO (ПЕРЕЙТИ), должен быть выполнен до выполнения оператора GO TO (ПЕРЕЙТИ).

(3) При выполнении оператора GO TO (ПЕРЕЙТИ), представленного форматом 2, управление передается той из указанных в списке процедур, названных именем-процедуры-1, порядковый номер которой совпадает со значением идентификатора. Если значение идентификатора не равно одному из целых положительных (или без знака) 1, 2, ..., n, то управление передается следующему оператору.

6.16. **Оператор IF (ЕСЛИ)**

6.16.1. Назначение

Оператор IF (ЕСЛИ) приводит к вычислению условия (см. п. 6.3 настоящей части).

Последующие действия рабочей программы зависят от истинности условия.

6.16.2. Общий формат

IF условие-1 THEN { {оператор-1} ... / NEXT SENTENCE }

{ ELSE оператор-2 ... [END-IF] / ELSE NEXT SENTENCE / END-IF }

ЕСЛИ условие-1 { {оператор-1} ... / СЛЕДУЮЩЕЕ ПРЕДЛОЖЕНИЕ }

{ ИНАЧЕ {оператор-2} ... [КОНЕЦ-ЕСЛИ] / ИНАЧЕ СЛЕДУЮЩЕЕ ПРЕДЛОЖЕНИЕ / КОНЕЦ-ЕСЛИ }

6.16.3. Синтаксические правила

(1) Оператор-1 и оператор-2 представляют собой условные операторы или повелительные операторы, за которыми, воз-

GOST EXPERT ГОСТ 22558-89,

можно, следуют условные операторы. Дальнейшее описание правил управления выполнением оператора-1 и оператора-2 дается в другом месте (см. ч. 4, п. 6.4.3).

(2) Фраза ELSE NEXT SENTENCE (ИНАЧЕ СЛЕДУЮЩЕЕ ПРЕДЛОЖЕНИЕ) может быть опущена, если она непосредственно предшествует точке, ограничивающей предложение.

(3) Если задана фраза END-IF (КОНЕЦ-ЕСЛИ), фраза NEXT SENTENCE (СЛЕДУЮЩЕЕ ПРЕДЛОЖЕНИЕ) не должна задаваться.

6.16.4. Общие правила

(1) Область действия оператора IF (ЕСЛИ) может ограничиваться одним из следующих способов:

а) фразой END-IF (КОНЕЦ-ЕСЛИ) на том же уровне вложенности;

б) разделителем точкой;

в) если оператор вложенный, то фразой ELSE (ИНАЧЕ) оператора IF (ЕСЛИ) на более высоком уровне вложенности (см. ч. 4, п. 6.4.3).

(2) При выполнении оператора IF (ЕСЛИ) имеют место следующие передачи управления:

а) если условие истинно и задан оператор-1, управление передается первому оператору оператора-1 и выполнение продолжается в соответствии с правилами для каждого оператора из оператора-1. Если выполняется оператор ветвления процедур или условный оператор, который осуществляет явную передачу управления, управление передается явно в соответствии с правилами для этого оператора. После завершения выполнения оператора-1 фраза ELSE (ИНАЧЕ), даже если она указана, игнорируется и управление передается на конец оператора IF (ЕСЛИ);

б) если условие истинно и вместо оператора-1 задана фраза NEXT SENTENCE (СЛЕДУЮЩЕЕ ПРЕДЛОЖЕНИЕ), фраза ELSE (ИНАЧЕ), если она задана, игнорируется и управление передается на следующее выполнимое предложение;

в) если условие ложно и задан оператор-2, оператор-1 или заменяющая его фраза NEXT SENTENCE (СЛЕДУЮЩЕЕ ПРЕДЛОЖЕНИЕ) игнорируется, управление передается на первый оператор оператора-2 и выполнение продолжается в соответствии с правилами для каждого оператора из оператора-2. Если выполняется оператор ветвления процедур или условный оператор, который осуществляет явную передачу управления, управление передается явно в соответствии с правилами для этого оператора. После завершения выполнения оператора-2 управление передается на конец оператора IF (ЕСЛИ);

GOST EXPERT ГОСТ 22558-89,

г) если условие ложно, а фраза ELSE (ИНАЧЕ) не указана, оператор-1 игнорируется и управление передается на конец оператора IF (ЕСЛИ);

д) если условие ложно и задана фраза ELSE NEXT SENTENCE (ИНАЧЕ СЛЕДУЮЩЕЕ ПРЕДЛОЖЕНИЕ), оператор-1 игнорируется, а управление передается на следующее выполнимое предложение.

(3) Оператор-1 и (или) оператор-2 могут содержать оператор IF (ЕСЛИ). Оператор IF (ЕСЛИ) в этом случае называется вложенным. Более детальные правила вложенности даются в соответствующем параграфе (см. ч. 4, п. 6.4.3). Вложенные операторы могут рассматриваться как парные комбинации IF (ЕСЛИ), ELSE (ИНАЧЕ) и END-IF (КОНЕЦ-ЕСЛИ), задаваемые слева направо. Таким образом, любое встретившееся ELSE (ИНАЧЕ) или END-IF (КОНЕЦ-ЕСЛИ) рассматривается как связанное с непосредственно предшествующим IF (ЕСЛИ), которому еще нет парной фразы ELSE (ИНАЧЕ) или END-IF (КОНЕЦ-ЕСЛИ).

6.17. Оператор INITIALIZE (ИНИЦИИРОВАТЬ)

6.17.1. Назначение

Оператор INITIALIZE (ИНИЦИИРОВАТЬ) обеспечивает возможность присваивать выбранным типам полей данных заранее определенные значения, например, числовым данным нули, буквенно-цифровым данным — пробелы.

6.17.2. Общий формат

INITIALIZE {идентификатор-1} ...

```
[ REPLACING { ALPHABETIC          }
            { ALPHANUMERIC        }
            { NUMERIC             }
            { ALPHANUMERIC-EDITED }
            { NUMERIC-EDITED      }

  DATA BY { идентификатор-2 } } ... ]
          { литерал-1       } }
```

ИНИЦИИРОВАТЬ {идентификатор-1} ...

```
[ ЗАМЕНЯЯ { БУКВЕННОЕ  }
          { БЦ         }
          { ЧИСЛОВОЕ   } ДАННОЕ НА
          { БЦР        }
          { ЧР         }

  { идентификатор-2 } ] ... ]
  { литерал-1       }
```


GOST EXPERT ГОСТ 22558-89,

6.17.3. Синтаксические правила

(1) Литерал-1 и данное, на которое ссылается идентификатор-2, задают посылающую область; данное, на которое ссылается идентификатор-1, представляет принимающую область.

(2) Каждая категория, заданная во фразе REPLACING (ЗАМЕНЯЯ), должна быть допустимой для принимающего операнда в операторе MOVE (ПОМЕСТИТЬ), в котором литерал-1 или данное, на которое ссылается идентификатор-2, используется как посылаемый операнд (п. 6.19 настоящей части).

(3) Одна и та же категория не может повторяться во фразе REPLACING (ЗАМЕНЯЯ).

(4) Описание данного, заданного идентификатором-1 или подчиненного идентификатору-1, не должно содержать фразу OCCURS (ПОВТОРЯЕТСЯ) с вариантом DEPENDING (В ЗАВИСИМОСТИ ОТ).

(5) Индексное данное не может быть операндом оператора INITIALIZE (ИНИЦИИРОВАТЬ).

(6) Статья описания данного для данного, заданного идентификатором-1, не должна содержать фразу RENAMES (ПЕРЕИМЕНОВЫВАЕТ).

6.17.4. Общие правила

(1) Ключевое слово, следующее за словом REPLACING (ЗАМЕНЯЯ), соответствует категориям данных, как они определяются в этом документе (см. ч. 4, п. 4.3.3).

(2) В зависимости от того, ссылается ли идентификатор-1 на элементарное или групповое данное, все действия выполняются, как если бы была написана последовательность операторов MOVE (ПОМЕСТИТЬ), каждый из которых имеет в качестве принимающего поля элементарное данное, подчиняющееся следующим правилам:

если задана фраза REPLACING (ЗАМЕНЯЯ):

а) если идентификатор-1 ссылается на групповое данное, любое подчиненное ему элементарное данное инициируется только в том случае, если оно принадлежит к категории, заданной во фразе REPLACING (ЗАМЕНЯЯ);

б) если идентификатор-1 ссылается на элементарное данное, то это данное инициируется, только если оно принадлежит к категории, заданной во фразе REPLACING (ЗАМЕНЯЯ).

Эта инициация производится следующим образом.

Данное, на которое ссылается идентификатор-2 или литерал-1, действует как посылаемый операнд соответствующего неявного оператора MOVE (ПОМЕСТИТЬ).

GOST EXPERT ГОСТ 22558-89,

Это действие распространяется на все принимающие элементарные поля, включая все вхождения элементов таблицы в групповом данном за исключением полей, перечисленных в пунктах (3) и (4) общих правил.

(3) Действие оператора INITIALIZE (ИНИЦИИРОВАТЬ) не распространяется на индексные данные и элементарные данные FILLER (ЗАПОЛНИТЕЛЬ).

(4) Любое данное, которое подчиняется идентификатору принимающего данного и содержит фразу REDEFINES (ПЕРЕОПРЕДЕЛЯЕТ), или любое подчиненное ему данное исключается из операции инициации. Однако принимающее данное или содержащее его данное может иметь в своем описании фразу REDEFINES (ПЕРЕОПРЕДЕЛЯЕТ).

(5) Если оператор задан без фразы REPLACING (ЗАМЕНЯЯ), в данные буквенной, буквенно-цифровой и буквенно-цифровой редактируемой категорий посылаются пробелы, а в данные числовой и числовой редактируемой категории посылаются нули. В этом случае операция выполняется так, как если бы каждое принимающее данное было бы принимающей областью элементарного оператора MOVE (ПОМЕСТИТЬ) с пробелом или нулем в качестве посылаемого данного.

(6) Во всех случаях содержимое данных, на которые ссылается идентификатор-1, устанавливается на указанные значения в порядке (слева направо) появления идентификатора-1 в операторе INITIALIZE (ИНИЦИИРОВАТЬ). В пределах этой последовательности, если идентификатор-1 ссылается на групповое данное, подчиненные ему элементарные данные инициируются в порядке их определения внутри группы.

(7) Если идентификатор-1 занимает ту же область памяти, что идентификатор-2, результат выполнения оператора не определен, даже если эти данные описаны одной статьей описания данных (см. п. 6.4.5 настоящей части).

6.18. Оператор INSPECT (ПРОСМОТРЕТЬ)

6.18.1. Назначение

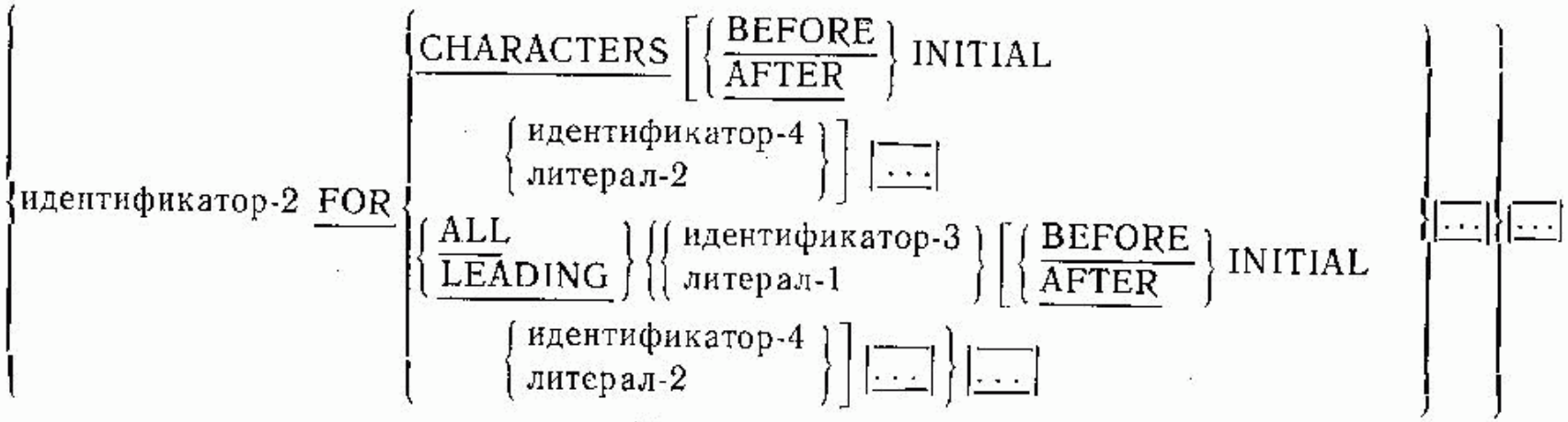
Оператор INSPECT (ПРОСМОТРЕТЬ) обеспечивает возможность подсчета или замены вхождений единичных литер или цепочек литер в данном.

6.18.2. Общий формат

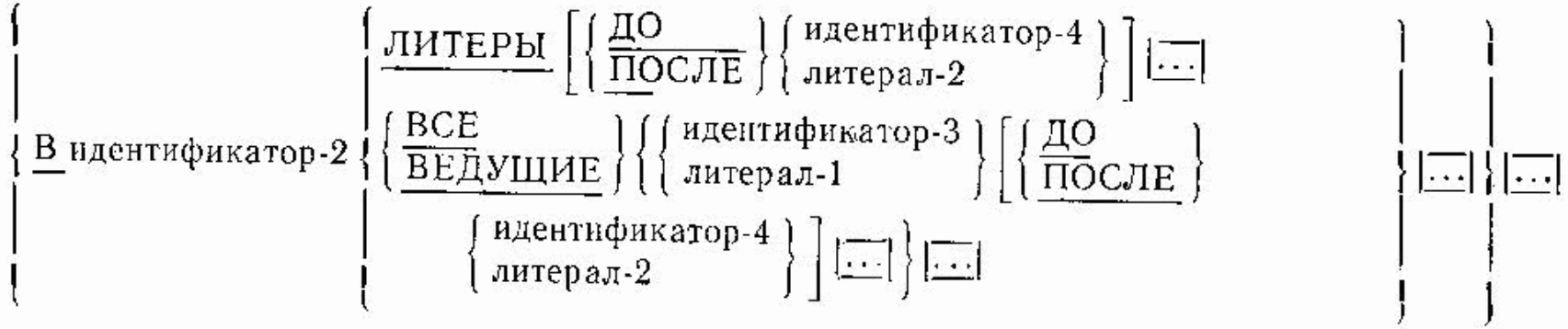
GOST EXPERT ГОСТ 22558-89,

Формат 1

INSPECT идентификатор-1 TALLYING



ПРОСМОТРЕТЬ идентификатор-1 СЧИТАЯ



ГОСТ 22558-89

340

Формат 2

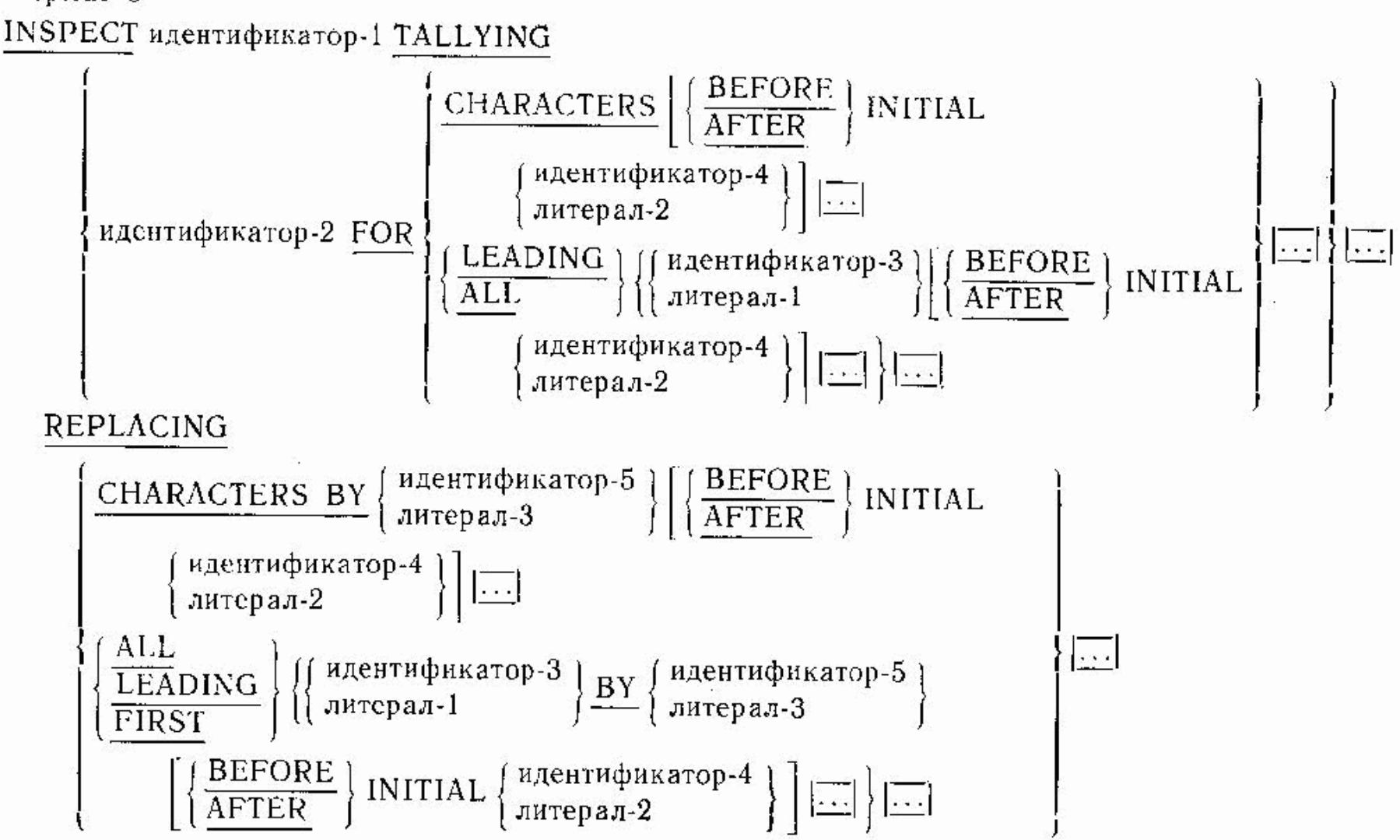
INSPECT идентификатор-1 REPLACING

{ CHARACTERS BY { идентификатор-5 / литерал-3 } [{ BEFORE / AFTER } INITIAL { идентификатор-4 / литерал-2 }] [...]
/ { ALL / LEADING / FIRST } { { идентификатор-3 / литерал-1 } BY { идентификатор-5 / литерал-3 } [{ BEFORE / AFTER } INITIAL { идентификатор-4 / литерал-2 }] [...] } [...] } [...]

ПРОСМОТРЕТЬ идентификатор-1 ЗАМЕНЯЯ

{ ЛИТЕРЫ НА { идентификатор-5 / литерал-3 } [{ ДО / ПОСЛЕ } { идентификатор-4 / литерал-2 }] [...]
/ { ВСЕ / ВЕДУЩИЕ / ПЕРВЫЙ } { { идентификатор-3 / литерал-1 } НА { идентификатор-5 / литерал-3 } [{ ДО / ПОСЛЕ } { идентификатор-4 / литерал-2 }] [...] } [...] } [...]

Формат 3


INSPECT идентификатор-1 TALLYING
идентификатор-2 FOR
CHARACTERS
BEFORE
AFTER
INITIAL
идентификатор-4
литерал-2
LEADING
ALL
идентификатор-3
литерал-1
REPLACING
CHARACTERS BY
идентификатор-5
литерал-3
FIRST
BY

GOST EXPERT ГОСТ 22558-89,

Формат 4

INSPECT идентификатор-1 CONVERTING { идентификатор-6 / литерал-4 } TO { идентификатор-7 / литерал-5 }

[{ BEFORE / AFTER } INITIAL { идентификатор-4 / литерал-2 }] ...

ПРОСМОТРЕТЬ идентификатор-1 ПРЕВРАЩАЯ { идентификатор-6 / литерал-4 } В { идентификатор-7 / литерал-5 }

[{ ДО / ПОСЛЕ } { идентификатор-4 / литерал-2 }] ...

ГОСТ 22558-89,

6.18.3. Синтаксические правила

Для всех форматов:

(1) Идентификатор-1 должен относиться к групповому данному или к элементарному данному любой категории с заданным (явно или неявно) использованием DISPLAY (ДЛЯ ВЫДАЧИ).

(2) Идентификатор-3, идентификатор-4 и так далее должны относиться к элементарному данному с заданным (явно или неявно) использованием DISPLAY (ДЛЯ ВЫДАЧИ).

(3) Каждый литерал должен быть нечисловым и может быть любой стандартной константой, кроме константы ALL (ВСЕ). Если литерал-1, литерал-2 или литерал-4 являются стандартными константами, они неявно относятся к однолитерным элементарным данным.

(4) Для каждой из фраз ALL (ВСЕ), LEADING (ВЕДУЩИЕ), CHARACTERS (ЛИТЕРЫ), FIRST (ПЕРВЫЙ), CONVERTING (ПРЕВРАЩАЯ) могут быть определены одна фраза BEFORE (ДО) и одна фраза AFTER (ПОСЛЕ).

(5) На уровне 1 ядра литерал-1, литерал-2, литерал-3 и данные, представляемые идентификатором-3, идентификатором-4 и идентификатором-5, должны состоять из одной литеры. Это ограничение не применяется на уровне 2, кроме случаев, специально оговоренных в синтаксических и общих правилах.

(6) При использовании форматов 1 и 3 идентификатор-2 должен представлять элементарное числовое данное.

(7) При использовании форматов 2 и 3 размер литерала-3 или данного, представляемого идентификатором-5, должны быть равны размеру литерала-1 или данного, представляемого идентификатором-3. Если вместо литерала-3 используется стандартная константа, размер данного, представленного стандартной константой, равен размеру литерала-1 или размеру данного, представляемого идентификатором-3.

(8) Если используется фраза CHARACTERS (ЛИТЕРЫ), то литерал-2, литерал-3 или значение данных, представляемых идентификатором-4, идентификатором-5, должны состоять из одной литеры.

(9) При использовании формата 4 размер литерала-5 или данного, представляемого идентификатором-7, должен быть равен размеру литерала-4 или данного, представляемого идентификатором-6. Если вместо литерала-5 используется стандартная константа, размер данного, представленного стандартной константой, равен размеру литерала-4 или размеру данного, представляемого идентификатором-6.

GOST EXPERT ГОСТ 22558-89,

(10) Одна и та же литера не должна содержаться дважды ни в литерале-4, ни в данном, представляемом идентификатором-6.

6.18.4. Общие правила

Для всех форматов:

(1) При просмотре (который включает цикл сравнения, установление границ для фраз BEFORE (ДО) и AFTER (ПОСЛЕ) и механизм подсчета или замещения) данное, представленное идентификатором-1, независимо от его класса, просматривается, начиная от своей самой левой позиции литеры, слева направо до самой правой позиции; выполняемые при этом действия и область их применения оговорены в общих правилах (5)—(7).

(2) При просмотре содержимое данных, указанных идентификатором-1, идентификатором-3, идентификатором-4, идентификатором-5, идентификатором-6, идентификатором-7, трактуется следующим образом:

а) если какое-либо из данных, представленных идентификатором-1, идентификатором-3, идентификатором-4, идентификатором-5, идентификатором-6 или идентификатором-7, описано как буквенное или буквенно-цифровое, то оператор INSPECT (ПРОСМОТРЕТЬ) обрабатывает значение каждого такого данного как строку литер;

б) если какое-либо из данных, представленных идентификатором-1, идентификатором-3, идентификатором-4, идентификатором-5, идентификатором-6 или идентификатором-7, описано как буквенно-цифровое редактируемое, числовое редактируемое или числовое без знака, то оператор INSPECT (ПРОСМОТРЕТЬ) обрабатывает значение каждого такого данного так, как переопределенное на буквенно-цифровое (см. общее правило 2а);

в) если какой-либо из перечисленных ранее идентификаторов представляет числовое данное со знаком, это данное рассматривается при выполнении оператора так, как если бы оно было помещено в числовое данное без знака того же размера, после чего к нему применяется правило 2б (п. 6.19 настоящей части). Если идентификатор-1 представляет числовое данное со знаком, первоначальное значение знака сохраняется после завершения оператора INSPECT (ПРОСМОТРЕТЬ).

(3) При изложении ниже правил (5)—(17) все, оговоренное для литерала-1, литерала-2, литерала-3, литерала-4 и литерала-5, применяется в равной мере к значениям данных, представленных идентификатором-3, идентификатором-4, идентификатором-5, идентификатором-6 и идентификатором-7 соответственно.

(4) Индексы, связанные с любым идентификатором, вычисляются один раз в начале выполнения оператора INSPECT (ПРОСМОТРЕТЬ).

GOST EXPERT ГОСТ 22558-89,

В форматах 1 и 2

(5) При просмотре значения данного, представленного идентификатором-1, каждое обнаруженное вхождение литерала-1 подсчитывается (при использовании формата 1) или заменяется вхождением литерала-3 (при использовании формата 2).

(6) Проверка вхождения литерала-1 в идентификатор-1 для подсчета или замены осуществляется по следующим правилам:

а) операнды фраз TALLYING (СЧИТАЯ) и REPLACING (ЗАМЕНЯЯ) рассматриваются в том порядке, в котором они представлены в операторе слева направо. Первый из литералов-1 сравнивается с соответствующим количеством первых слева позиций литер данного, представленного идентификатором-1. Литерал-1 совпадает с порцией данного, представляемого идентификатором-1, если их литеры попарно совпадают и:

1) если не определены ни LEADING (ВЕДУЩИЕ), ни FIRST (ПЕРВЫЙ) либо

2) если LEADING (ВЕДУЩИЕ) относится к литералу-1, а литерал-1 является ведущим вхождением, как определено в общих правилах (10) и (13), либо

3) если FIRST (ПЕРВЫЙ) относится к литералу-1, а литерал-1 является первым вхождением, как определено в общих правилах (10) и (13);

б) если для первого литерала-1 и первых позиций литер идентификатора-1 вхождение не найдено, проверяется вхождение следующего литерала-1 и так далее для тех же позиций литер идентификатора-1.

Если перебраны все указанные литералы-1 и вхождение не обнаружено, переходим к рассмотрению следующих позиций литер данного, представленного идентификатором-1, таким образом, что соседняя справа литера по отношению к самой левой из рассматриваемых в предыдущем цикле сравнений позиций литер данного, представленного идентификатором-1, становится самой левой рассматриваемой позицией этого данного, после чего снова выполняется цикл сравнений для всех указанных литералов-1 (начиная от первого);

в) если вхождение найдено, выполняется подсчет или замещение согласно правилам (10) и (13).

Соседняя справа позиция литеры по отношению к самой правой из рассматриваемых в предыдущем цикле сравнения позиций литер данного, представленного идентификатором-1, становится самой левой рассматриваемой позицией этого данного, после чего снова выполняется цикл сравнений, начиная с первого литерала-1;

г) процесс проверки вхождения продолжается до тех пор, пока самая правая позиция литеры данного, представленного идентификатором-1, не будет рассмотрена как удовлетворяющая провер-

GOST EXPERT ГОСТ 22558-89,

ке вхождения или как самая левая позиция этого данного, после чего оператор INSPECT (ПРОСМОТРЕТЬ) считается выполненным;

д) если указана фраза CHARACTERS (ЛИТЕРЫ), в описанном выше цикле сравнения вместо литерала-1 всегда участвует однолитерный подразумеваемый операнд, значение которого в каждый момент совпадает с самой левой рассматриваемой позицией данного, представленного идентификатором-1.

(7) Операции сравнения, определенные правилами (6), ограничиваются фразами BEFORE (ДО) и AFTER (ПОСЛЕ) следующим образом:

а) если не указаны фразы BEFORE (ДО) и AFTER (ПОСЛЕ), литерал-1 или подразумеваемый операнд фразы CHARACTERS (ЛИТЕРЫ) участвует в сравнении согласно предыдущему правилу с самой левой позиции идентификатора-1;

б) если указана фраза BEFORE (ДО), то самая левая позиция литеры этого данного становится самой левой рассматриваемой позицией; соседняя слева позиция литеры по отношению к первому вхождению литерала-2 в значение данного, представленного идентификатором-1, становится самой правой рассматриваемой позицией литеры этого данного;

в) если указана фраза AFTER (ПОСЛЕ), то соседняя справа позиция литеры по отношению к первому вхождению литерала-2 в данное, представленное идентификатором-1, становится самой левой рассматриваемой позицией этого данного, а его самая правая позиция становится самой правой рассматриваемой позицией.

Если вхождение литерала-2 в значение данного, представленного идентификатором-1, не установлено, цикл сравнения в случае фразы BEFORE (ДО) выполняется так, как если бы фраза BEFORE (ДО) не была задана; в случае фразы AFTER (ПОСЛЕ) считается, что вхождение литерала-1 в значение данного, представленного идентификатором-1, не обнаружено.

В формате 1

(8) Слова ALL (ВСЕ), LEADING (ВЕДУЩИЕ) рассматриваются как прилагательные, которые относятся ко всем последующим литералам-1 до появления другого прилагательного.

(9) При выполнении оператора INSPECT (ПРОСМОТРЕТЬ) значение данного, представленного идентификатором-2, не устанавливается в начальное состояние.

(10) Правила использования следующие:

а) Если задана фраза ALL (ВСЕ), значение данного, представленного идентификатором-2, увеличивается на единицу при каждом вхождении литерала-1 в значение данного, представленного идентификатором-1;

б) если задана фраза LEADING (ВЕДУЩИЕ), значение данного, представленного идентификатором-2, увеличивается на еди-

GOST EXPERT ГОСТ 22558-89,

ницу при каждом следующем вхождении литерала-1 в значение данного, на которое ссылается идентификатор-1, при условии, что слева не встретились позиции данного, представленного идентификатором-1, которые не совпали ни с одним из указанных литералов-1;

в) если задана фраза CHARACTERS (ЛИТЕРЫ), значение данного, на которое ссылается идентификатор-2, увеличивается на единицу при каждом просмотре литеры данного, представленного идентификатором-1.

(11) При использовании формата 1 в случае, когда идентификатор-1, идентификатор-3 или идентификатор-4 занимают ту же область памяти, что и идентификатор-2, результат выполнения оператора INSPECT (ПРОСМОТРЕТЬ) не определен, даже если эти идентификаторы описаны одной и той же статьей описания данных (см. п. 6.4.5 настоящей части).

В формате 2

(12) Фразы ALL (ВСЕ), LEADING (ВЕДУЩИЕ), FIRST (ПЕРВЫЙ) применяются к каждой следующей за ними фразе BY (НА) до появления очередного вхождения одной из перечисленных фраз.

(13) Правила замещения следующие:

а) если задана фраза CHARACTERS (ЛИТЕРЫ), каждая литера значения данного, представляемого идентификатором-1, заменяется на литерал-3;

б) если задана фраза ALL (ВСЕ), каждое вхождение литерала-1 в значение данного, представляемого идентификатором-1, заменяется на литерал-3;

в) если задана фраза LEADING (ВЕДУЩИЕ), каждое смежное вхождение литерала-1 в значение данного, представляемого идентификатором-1, заменяется на литерал-3 при условии, что слева не были обнаружены позиции этого данного, которые не совпали ни с одним из литералов-1;

г) когда задана фраза FIRST (ПЕРВЫЙ), самое левое вхождение литерала-1 в значение данного, представляемого идентификатором-1, заменяется литералом-3.

Это правило применяется ко всем последующим указаниям фразы FIRST (ПЕРВЫЙ) независимо от значения литерала-1.

(14) При использовании формата 2 в случае, когда идентификатор-3, идентификатор-4 или идентификатор-5 занимают одну и ту же область памяти, результат выполнения оператора INSPECT (ПРОСМОТРЕТЬ) не определен, даже если эти идентификаторы описаны одной и той же статьей описания данных (см. п. 6.4.5 настоящей части).

В формате 3

(15) Оператор INSPECT (ПРОСМОТРЕТЬ) интерпретируется и выполняется так, как будто заданы два следующих друг за дру-

GOST EXPERT ГОСТ 22558-89,

гом оператора INSPECT (ПРОСМОТРЕТЬ) для одного и того же идентификатора-1, один из которых имеет формат 1 с фразой TALLYING (СЧИТАЯ), тождественной соответствующей фразе оператора в формате 3, а другой — формат 2 с фразой REPLACING (ЗАМЕНЯЯ), тождественной соответствующей фразе оператора в формате 3. К оператору в формате 1 применяются общие правила сравнения и подсчета. К оператору в формате 2 применяются общие правила сравнения и замены.

Индексы, связанные с любым идентификатором в операторе формата 2, вычисляются только один раз до выполнения оператора в формате 1.

В формате 4

(16) Оператор INSPECT (ПРОСМОТРЕТЬ) интерпретируется и выполняется так, как будто задан оператор INSPECT (ПРОСМОТРЕТЬ) в формате 2 для одного и того же идентификатора-1 с последовательностью фраз ALL (ВСЕ) — по одной для каждой литеры литерала-4. Результат оператора такой же, как если бы в каждой фразе ALL (ВСЕ) в качестве литерала-1 указывалась единственная литера литерала-4, а в качестве литерала-3 — соответствующая единственная литера литерала-5. Соответствие между литерами литерала-4 и литерами литерала-5 осуществляется по порядковому номеру литеры в данном.

(17) Если идентификатор-4, идентификатор-6 или идентификатор-7 занимают одну и ту же область памяти, результат выполнения оператора INSPECT (ПРОСМОТРЕТЬ) не определен, даже если эти идентификаторы определены одной и той же статьей описания данных.

6.18.5. Примеры

В каждом из следующих примеров оператора INSPECT (ПРОСМОТРЕТЬ) предполагается, что непосредственно перед выполнением оператора значения всех данных СО-0, СО-1, СО-2, СО-3, СО-4 равны нулю. В каждом примере, кроме последнего, приводятся результаты работы двух операторов INSPECT (ПРОСМОТРЕТЬ).

Пример 1.

```
INSPECT ТЕМ TALLYING
  СО-0 FOR ALL «AB», ALL «D»
  СО-1 FOR ALL «BC»
  СО-2 FOR LEADING «EF»
  СО-3 FOR LEADING «B»
  СО-4 FOR CHARACTERS;
ПРОСМОТРЕТЬ ТЕМ СЧИТАЯ
  В СО-0, ВСЕ «AB», ВСЕ «D»
```


GOST EXPERT ГОСТ 22558-89,

В CO-1 ВСЕ «BC»
В CO-2 ВЕДУЩИЕ «EF»
В CO-3 ВЕДУЩИЕ «B»
В CO-4 ЛИТЕРЫ;

INSPECT TEM REPLACING
ALL «AB» BY «XY», «D» BY «X»
ALL «BC» BY «VW»
LEADING «EF» BY «TU»
LEADING «B» BY «S»
FIRST «G» BY «P»
CHARACTERS BY «Z»

ПРОСМОТРЕТЬ ТЕМ ЗАМЕНЯЯ
ВСЕ «AB» НА «XY», «D» НА «X»
ВСЕ «BC» НА «VW»
ВЕДУЩИЕ «EF» НА «TU»
ВЕДУЩИЕ «B» НА «S»
ПЕРВЫЙ «G» НА «P»
ЛИТЕРЫ НА «Z»

Результаты действия оператора показаны ниже.

Начальное значение данного ТЕМ	CO-0	CO-1	CO-2	CO-3	CO-4	Результирующее значение данного ТЕМ
EFABDBCGABEFGG	3	1	1	0	5	TUXYXVWRXYZZPZ
BABABC	2	0	0	1	1	SXYXYZ
BBBC	0	1	0	2	0	SSVW

Пример 2

INSPECT TEM TALLYING
CO-0 FOR CHARACTERS
CO-0 FOR ALL «A»

ПРОСМОТРЕТЬ ТЕМ СЧИТАЯ
В CO-0 ЛИТЕРЫ
В CO-1 ВСЕ «A»

INSPECT TEM REPLACING
CHARACTERS BY «Z»
ALL «A» BY «X»

ПРОСМОТРЕТЬ ТЕМ ЗАМЕНЯЯ
ЛИТЕРЫ НА «Z»
ВСЕ «A» НА «X»

Результаты работы операторов показаны ниже.

GOST EXPERT ГОСТ 22558-89,

Исходное значение данного ТЕМ	CO-0	CO-1	Результирующее значение данного ТЕМ
BBB	3	0	ZZZ
ABA	3	0	ZZZ

Пример 3

```
INSPECT TEM TALLYING
  CO-0 FOR ALL «AB» BEFORE «BC»
  CO-1 FOR LEADING «B» AFTER «D»
  CO-2 FOR CHARACTERS AFTER «A» BEFORE «C»
ПРОСМОТРЕТЬ ТЕМ СЧИТАЯ
  В CO-0 ВСЕ «AB» ДО «BC»
  В CO-1 ВЕДУЩИЕ «B» ПОСЛЕ «D»
  В CO-2 ЛИТЕРЫ ПОСЛЕ «A» ДО «C»
INSPECT TEM REPLACING
  ALL «AB» BY «XY» BEFORE «BC»
  LEADING «B» BY «W» AFTER «D»
  FIRST «E» BY «V» AFTER «D»
  CHARACTERS BY «Z» AFTER «A» BEFORE «C»
ПРОСМОТРЕТЬ ТЕМ ЗАМЕНЯЯ
  ВСЕ «AB» НА «XY» ДО «BC»
  ВЕДУЩИЕ «B» НА «W» ПОСЛЕ «D»
  ПЕРВЫЙ «E» НА «V» ПОСЛЕ «D»
  ЛИТЕРЫ НА «Z» ПОСЛЕ «A» ДО «C»
```

Результаты работы операторов показаны ниже.

Начальное значение ТЕМ	CO-0	CO-1	CO-2	Результирующее значение ТЕМ
BBEABDABABBCABEE	3	0	2	BBEXYZXYXYZCABVE
ADDDDC	0	0	4	AZZZZC
ADDDDA	0	0	5	AZZZZZ
CDDDDC	0	0	0	CDDDDC
BDBBBDB	0	3	0	BDWWWDB

Пример 4

```
INSPECT TEM TALLYING
  CO-0 FOR ALL «AB» AFTER «BA» BEFORE «BC»
ПРОСМОТРЕТЬ ТЕМ СЧИТАЯ
  В CO-0 ВСЕ «AB» ПОСЛЕ «BA» ДО «BC»
INSPECT TEM REPLACING
  ALL «AB» BY «XY» AFTER «BA» BEFORE «BC»
```


GOST EXPERT ГОСТ 22558-89,

ПРОСМОТРЕТЬ ТЕМ ЗАМЕНЯЯ
ВСЕ «АВ» НА «XY» ПОСЛЕ «ВА» ДО «ВС».
Результаты работы операторов показаны ниже.

Исходное значение данного ТЕМ	СО 0	Результирующее значение данного ТЕМ
АВАВАВАВС	1	АВАВХУАВС

Пример 5
INSPECT ТЕМ CONVERTING
«ABCD» TO «XYZX» AFTER QUOTE BEFORE «#»
ПРОСМОТРЕТЬ ТЕМ ПРЕВРАЩАЯ
«ABCD» В «XYZX» ПОСЛЕ КАВЫЧКА ДО «#»
Этот оператор эквивалентен следующему:
INSPECT ITEM REPLACING
ALL «A» BY «X» AFTER QUOTE BEFORE «#»
ALL «B» BY «Y» AFTER QUOTE BEFORE «#»
ALL «C» BY «Z» AFTER QUOTE BEFORE «#»
ALL «D» BY «X» AFTER QUOTE BEFORE «#»
ПРОСМОТРЕТЬ ТЕМ ЗАМЕНЯЯ
ВСЕ «А» НА «X» ПОСЛЕ КАВЫЧКА ДО «#»
ВСЕ «В» НА «Y» ПОСЛЕ КАВЫЧКА ДО «#»
ВСЕ «С» НА «Z» ПОСЛЕ КАВЫЧКА ДО «#»
ВСЕ «D» НА «X» ПОСЛЕ КАВЫЧКА ДО «#»
Результаты работы оператора показаны ниже.

Начальное значение ТЕМ	Результирующее значение ТЕМ
АС «АЕВDFBCD # АВ» D	АС «XEYXFYZX # АВ» D

6.19. Оператор MOVE (ПОМЕСТИТЬ)

6.19.1. Назначение

Оператор MOVE (ПОМЕСТИТЬ) пересылает данные в соответствии с правилами редактирования в одно или несколько полей данных.

6.19.2. Общий формат

Формат 1

MOVE { идентификатор-1 / литерал-1 } TO {идентификатор-2} . . .

ПОМЕСТИТЬ { идентификатор-1 / литерал-1 } В {идентификатор-2} . . .

GOST EXPERT ГОСТ 22558-89,

Формат 2

MOVE { CORRESPONDING / CORR } идентификатор-1 TO идентификатор-2

ПОМЕСТИТЬ { СООТВЕТСТВЕННО / СООТВ } идентификатор-1 В идентификатор-2

6.19.3. Синтаксические правила

(1) Литерал-1 или данное, представленное идентификатором-1, представляют пересылаемое поле. Данное, представленное идентификатором-2, представляет принимающее поле.

(2) CORR (СООТВ) является сокращением слова CORRESPONDING (СООТВЕТСТВЕННО).

(3) При использовании фразы CORRESPONDING (СООТВЕТСТВЕННО) оба идентификатора должны представлять групповые данные.

(4) Индексное данное не может появляться как операнд оператора MOVE (ПОМЕСТИТЬ).

6.19.4. Общие правила

(1) Если используется фраза CORRESPONDING (СООТВЕТСТВЕННО), то отдельные компоненты идентификатора-1 перемещаются в соответствующие компоненты идентификатора-2 согласно правилам, приведенным в п. 6.4.3 настоящей части. Результат действия фразы такой же, как при использовании отдельного оператора MOVE (ПОМЕСТИТЬ) для каждой пары соответствующих идентификаторов.

(2) Литерал-1 или содержимое данного, представленного идентификатором-1, перемещается в данное, представленное идентификатором-2 в том порядке, в каком они определены. Правила, относящиеся к идентификатору-2, применяются также и к другим принимающим полям. Длина и индексы, относящиеся к идентификатору-2, вычисляются непосредственно перед перемещением данного в соответствующее поле.

Индексы, связанные с идентификатором-1, вычисляются только один раз непосредственно перед помещением данных в первый из принимающих операндов.

GOST EXPERT ГОСТ 22558-89,

Длина данного, представленного идентификатором-1, вычисляется только один раз непосредственно перед помещением данных в первый из принимающих операндов.

На вычисление длины идентификатора-1 или идентификатора-2 может влиять вариант DEPENDING ON (В ЗАВИСИМОСТИ ОТ) фразы OCCURS (ПОВТОРЯЕТСЯ) (см. п. 5.8 настоящей части).

Результат действия оператора
MOVE A(M) TO M, P (M)
ПОМЕСТИТЬ A(M) В M, P(M)
эквивалентен операторам

MOVE A(M) TO temp	ПОМЕСТИТЬ A(M) В пром
MOVE temp TO M	ПОМЕСТИТЬ пром В M
MOVE temp TO P(M)	ПОМЕСТИТЬ пром В P(M)

где «temp» («пром») — промежуточный результат, обеспечиваемый реализацией.

(3) Перемещение, в котором принимающий операнд является элементарным данным, а пересылаемый операнд — литералом либо элементарным данным, называется элементарным перемещением. Каждое элементарное данное принадлежит одной из следующих категорий: числовой, буквенной, буквенно-цифровой, числовой редактируемой, буквенно-цифровой редактируемой (см. п. 5.9 настоящей части). Числовые литералы принадлежат к числовой категории, а нечисловые — к буквенно-цифровой категории. Стандартная константа ZERO, ZEROES (НУЛЬ, НУЛИ) при перемещении в числовое или числовое редактируемое данное относится к числовой категории.

Во всех прочих случаях она относится к буквенно-цифровой категории.

Стандартная константа SPACE, SPACES (ПРОБЕЛ, ПРОБЕЛЫ) относится к буквенной категории. Все остальные стандартные константы относятся к буквенно-цифровой категории.

К элементарному перемещению применяются следующие правила:

а) буквенно-цифровое редактируемое, стандартная константа SPACE (ПРОБЕЛ) или буквенное данное не должны перемещаться в числовое или числовое редактируемое данное;

б) числовое или числовое редактируемое данное, числовой литерал и стандартная константа ZERO (НУЛЬ) не могут перемещаться в буквенное данное;

в) числовой литерал, не представляющий целое число, или числовое данное, имеющее значение, не являющееся целым числом, не может перемещаться в буквенно-цифровое или буквенно-цифровое редактируемое данное;

GOST EXPERT ГОСТ 22558-89,

г) на уровне 1 числовое редактируемое данное не должно перемещаться в числовое или числовое редактируемое данное;

д) прочие элементарные перемещения являются допустимыми и выполняются согласно общему правилу (4).

(4) Во время выполнения допустимого элементарного перемещения производятся все необходимые преобразования данного из одной формы внутреннего представления в другую, сопровождающиеся редактированием, определенным для принимающего данного, или дередактированием, предполагаемым принимающим данным. Следующие правила применяются к допустимым элементарным перемещениям:

а) если принимающим данным является буквенно-цифровое редактируемое или буквенно-цифровое данное, производится выравнивание и любое необходимое дополнение пробелами, как это определено стандартными правилами выравнивания (см. ч. 4, п. 4.3.6). Если пересылаемое данное описано как числовое со знаком, знак числа не пересылается; если знак числа занимает отдельную позицию литеры, то эта литера не пересылается и размер пересылаемого данного на одну литеру меньше, чем его действительный размер в терминах литер стандартного формата данных (см. п. 5.12 настоящей части).

Если пересылаемый операнд описан как числовое редактируемое данное, дередактирования не происходит. Если использование пересылаемого операнда отличается от использования принимающего операнда, происходит преобразование пересылаемого операнда во внутреннее представление принимающего операнда. Если пересылаемый операнд числовой и содержит литеру шаблона P(M), считается, что все цифровые позиции, определяемые этой литерой, имеют значение нуль и учитываются при определении размера пересылаемого операнда;

б) если принимающим данным является числовое или числовое редактируемое данное, то производится выравнивание по десятичной точке и любое необходимое дополнение нулями согласно стандартным правилам выравнивания (см. ч. 4, п. 4.3.6), кроме случая, когда нули замещаются согласно требованиям редактирования.

Если пересылаемый операнд является числовым редактируемым, требуется дередактирование для установления неотредактированного числового значения операнда, возможно, со знаком, затем это неотредактированное числовое значение помещается в принимающее поле.

1) Если принимающим является числовое данное со знаком, знак пересылаемого данного помещается в принимающее данное при этом выполняется необходимое преобразование представления знака (см. п. 5.12 настоящей части). Если пересылаемое дан-

GOST EXPERT ГОСТ 22558-89,

ное описано без знака, принимающему данному присваивается положительный знак.

2) Если принимающее данное не имеет знака числа, при перемещении используется абсолютное значение пересылаемого данного.

3) Пересылаемое данное, описанное как буквенно-цифровое, перемещается так, как если бы оно было описано как числовое целое без знака;

в) если принимающее поле описано как буквенное, имеет место выравнивание и необходимое дополнение пробелами по ранее определенным правилам (см. ч. 4, п. 4.3.6).

(5) Перемещение, которое не является элементарным, рассматривается как элементарное перемещение буквенно-цифрового данного в буквенно-цифровое; однако при этом не выполняются преобразования данных из одной формы внутреннего представления в другую. При таком перемещении принимающее поле будет заполняться без рассмотрения особенностей отдельных элементарных или групповых данных, содержащихся в пересылаемом или принимающем поле, кроме случая, отмеченного в общем правиле фразы OCCURS (ПОВТОРЯЕТСЯ) (см. ч. 4, п. 5.8).

(6) Ниже иллюстрируются допустимые сочетания категорий операндов оператора MOVE (ПОМЕСТИТЬ).

Категория пересылаемого данного	Категория принимающего данного		
	Буквенная	Буквенно-цифровая, буквенно-цифровая редактируемая	Числовая (целое), числовая (нецелое), числовая редактируемая
Буквенная	Да По п. 4в	Да По п. 4а	Нет По п. 3а
Буквенно-цифровая	Да По п. 4в	Да По п. 4а	Да По п. 4б
Буквенно-цифровая редактируемая	Да По п. 4в	Да По п. 4а	Нет По п. 3а
Числовая (целое)	Нет По п. 3б	Да По п. 4а	Да По п. 4б
Числовая (нецелое)	Нет По п. 3б	Нет По п. 3в	Да По п. 4б
Числовая редактируемая	Нет По п. 3б	Да По п. 4а	Да По п. 4б

Примечание. «Да» обозначает допустимость пересылки, «нет» — запрещение. Ссылка на общее правило указывает правило, запрещающее пересылку или объясняющее порядок перемещения.

GOST EXPERT ГОСТ 22558-89,

6.20. **Оператор MULTIPLY (УМНОЖИТЬ)**

6.21.1. Назначение

Оператор MULTIPLY (УМНОЖИТЬ) позволяет перемножить числовые данные и присвоить значение произведения результату.

6.20.2. Общий формат

Формат 1

MULTIPLY { идентификатор-1 / литерал-1 } BY {идентификатор-2 [ROUNDED]} ...
[ON SIZE ERROR повелительный-оператор-1]
[NOT ON SIZE ERROR повелительный-оператор-2]
[END-MULTIPLY]

УМНОЖИТЬ { идентификатор-1 / литерал-1 } НА {идентификатор-2 [ОКРУГЛЯЯ]} ...
[ПРИ ПЕРЕПОЛНЕНИИ повелительный-оператор-1]
[БЕЗ ПЕРЕПОЛНЕНИЯ повелительный-оператор-2]
[КОНЕЦ-УМНОЖИТЬ]

Формат 2

MULTIPLY { идентификатор-1 / литерал-1 } BY { идентификатор-2 / литерал-2 }
GIVING {идентификатор-3 [ROUNDED]} ...
[ON SIZE ERROR повелительный-оператор-1]
[NOT ON SIZE ERROR повелительный-оператор-2]
[END-MULTIPLY]

УМНОЖИТЬ { идентификатор-1 / литерал-1 } НА { идентификатор-2 / литерал-2 }
ПОЛУЧАЯ {идентификатор-3 [ОКРУГЛЯЯ]} ...
[ПРИ ПЕРЕПОЛНЕНИИ повелительный-оператор-1]
[БЕЗ ПЕРЕПОЛНЕНИЯ повелительный-оператор-2]
[КОНЕЦ-УМНОЖИТЬ]

6.20.3. Синтаксические правила

(1) Каждый идентификатор должен представлять числовое эле-

ментарное данное, за исключением формата 2, в котором любой идентификатор, который появляется только справа от слова GIVING (ПОЛУЧАЯ), должен представлять элементарное числовое или элементарное числовое редактируемое данное.

(2) Каждый литерал должен быть числовым литералом.

(3) Композиция операндов, полученная в результате умножения операндов, выровненных по положению их десятичной точки, не должна содержать более 18 цифр.

6.20.4. Общие правила

(1) При использовании формата 1 литерал-1 или значение данного, представленного идентификатором-1, хранится в промежуточном данном. Затем его промежуточное данное умножается на данное, представленное идентификатором-2. Значение множителя (значение данного, представленного идентификатором-2) заменяется этим произведением; аналогично промежуточное данное умножается на каждое последующее вхождение идентификатора-2 в указанном порядке слева направо.

(2) При использовании формата 2 литерал-1 или значение данного, представленное идентификатором-1, умножается на литерал-2 или значение данного, представленного идентификатором-2, а результат хранится в данном, представленном каждым идентификатором-3.

(3) Дополнительные правила и пояснения, относящиеся к этому оператору, приводятся в соответствующих параграфах (см. ч. 4, п. 6.4.3; ч. 6, пп. 6.4.1, 6.4.2, 6.4.4—6.4.6).

6.21. Оператор PERFORM (ВЫПОЛНИТЬ)

6.21.1. Назначение

Оператор PERFORM (ВЫПОЛНИТЬ) используется для явной передачи управления одной или нескольким процедурам и неявного возврата управления после завершения этих процедур. Оператор PERFORM (ВЫПОЛНИТЬ) используется также для управления выполнением одного или более повелительных операторов, находящихся в области действия этого оператора PERFORM (ВЫПОЛНИТЬ).

6.21.2. Общий формат

Формат 1

PERFORM [имя-процедуры-1 [{ THROUGH / THRU } имя-процедуры-2]]
[повелительный-оператор-1 END-PERFORM]

ВЫПОЛНИТЬ [имя-процедуры-1 [ПО имя-процедуры-2]]
[повелительный-оператор-1 КОНЕЦ-ВЫПОЛНИТЬ]

Формат 2

PERFORM [имя-процедуры-1 [{THROUGH / THRU} имя-процедуры-2]]
{идентификатор-1 / целое-1} TIMES [повелительный-оператор-1
END-PERFORM]

ВЫПОЛНИТЬ [имя-процедуры-1 [ПО имя-процедуры-2]]
{идентификатор-1 / целое-1} {РАЗ / РАЗА}
[повелительный-оператор-1 КОНЕЦ-ВЫПОЛНИТЬ]

Формат 3

PERFORM [имя-процедуры-1 [{THROUGH / THRU} имя-процедуры-2]]
[[WITH TEST {BEFORE / AFTER}]] UNTIL условие-1
[повелительный-оператор-1 END-PERFORM]

ВЫПОЛНИТЬ [имя-процедуры-1 [ПО имя-процедуры-2]]
[[С ПРОВЕРКОЙ В {НАЧАЛЕ / КОНЦЕ}]] ДО условие-1
[повелительный-оператор-1 КОНЕЦ-ВЫПОЛНИТЬ]

Формат 4

PERFORM [имя-процедуры-1 [{THROUGH / THRU} имя-процедуры-2]]
[WITH TEST {BEFORE / AFTER}]

GOST EXPERT ГОСТ 22558-89,



6.21.3. Синтаксические правила

(1) Если опущено имя-процедуры-1, должны быть определены повелительный-оператор-1 и фраза END-PERFORM (КОНЕЦ-ВЫПОЛНИТЬ), если определено имя-процедуры-1, не должны употребляться повелительный-оператор-1 и фраза END-PERFORM (КОНЕЦ-ВЫПОЛНИТЬ).

(2) При использовании формата 4, если опущено имя-процедуры-1, не должна употребляться фраза AFTER (ЗАТЕМ).

GOST EXPERT ГОСТ 22558-89,

(3) Если не указаны ни фраза TEST BEFORE (С ПРОВЕРКОЙ В НАЧАЛЕ), ни фраза TEST AFTER (С ПРОВЕРКОЙ В КОНЦЕ), предполагается по умолчанию TEST BEFORE (С ПРОВЕРКОЙ В НАЧАЛЕ).

(4) Каждый идентификатор представляет числовое элементарное данное, описанное в разделе данных. В формате 2 идентификатор-1 представляет целое числовое элементарное данное.

(5) Каждый литерал является числовым литералом.

(6) Если указана фраза END-PERFORM (КОНЕЦ-ВЫПОЛНИТЬ), оператор PERFORM (ВЫПОЛНИТЬ) является оператором с ограничителем области действия.

(7) Если во фразе VARYING (МЕНЯЯ) или AFTER (ЗАТЕМ) указано имя-индекса, то:

а) идентификатор, относящийся к фразам FROM (ОТ) и BY (НА), должен представлять положительное целое;

б) литерал, относящийся к фразе FROM (ОТ), должен представлять положительное целое;

в) литерал, относящийся к фразе BY (НА), должен представлять целое, отличное от нуля.

(8) Если во фразе FROM (ОТ) задано имя-индекса, то:

а) идентификатор, относящийся к фразам VARYING (МЕНЯЯ) и AFTER (ЗАТЕМ), должен представлять целое;

б) идентификатор, относящийся к фразе BY (НА), должен представлять целое;

в) литерал, относящийся к фразе BY (НА), должен представлять целое.

(9) Литерал во фразе BY (НА) не должен равняться нулю.

(10) Условие-1, условие-2, ... могут быть любыми условными выражениями (см. п. 6.3 настоящей части).

(11) Если указаны и имя-процедуры-1, и имя-процедуры-2 и какое-либо из них является именем-процедуры в декларативной части программы, то оба имени-процедуры должны быть именами-процедур из одной и той же секции этой части.

(12) Допускаются до шести фраз AFTER (ЗАТЕМ) в формате 4 оператора PERFORM (ВЫПОЛНИТЬ).

6.21.4. Общие правила

(1) Значения данных, на которые ссылаются идентификатор-4 и идентификатор-7, не должны быть равными нулю.

(2) Если в фразах VARYING (МЕНЯЯ) и AFTER (ЗАТЕМ) указано имя-индекса и в соответствующей фразе FROM (ОТ) задан идентификатор, то значение данного, на которое ссылается идентификатор, должно быть положительным.

(3) Если указано имя-процедуры-1, оператор PERFORM (ВЫПОЛНИТЬ) называется оператором с вынесенной областью действия; если имя-процедуры-1 не указано, оператор PERFORM (ВЫПОЛНИТЬ) называется оператором с встроенной областью действия.

(4) Операторы, находящиеся в области действия имени-процедуры-1 (до имени-процедуры-2, если оно указано) для оператора PERFORM (ВЫПОЛНИТЬ) с вынесенной областью действия или содержащиеся внутри самого оператора PERFORM (ВЫПОЛНИТЬ) в случае оператора PERFORM (ВЫПОЛНИТЬ) с встроенной областью действия, будут называться указанным множеством операторов.

(5) Фраза END-PERFORM (КОНЕЦ-ВЫПОЛНИТЬ) ограничивает область действия оператора PERFORM (ВЫПОЛНИТЬ) со встроенной областью действия (см. ч. 4, п. 6.4.3).

(6) Оператор PERFORM (ВЫПОЛНИТЬ) со встроенной областью действия функционирует аналогично оператору PERFORM (ВЫПОЛНИТЬ) с вынесенной областью действия согласно следующим общим правилам, за исключением того факта, что операторы, содержащиеся в операторе PERFORM (ВЫПОЛНИТЬ) со встроенной областью действия, выполняются вместо операторов, принадлежащих области действия имени-процедуры-1 (до имени-процедуры-2, если оно указано). Если это специально не оговаривается, все общие правила равно применимы к оператору PERFORM (ВЫПОЛНИТЬ) со встроенной и вынесенной областью действия.

(7) При выполнении оператора PERFORM (ВЫПОЛНИТЬ) управление передается первому оператору указанного множества операторов, кроме случаев, оговоренных общими правилами 10б, 10в и 10г.

Эта передача управления выполняется один раз для каждого выполнения оператора PERFORM (ВЫПОЛНИТЬ). Для тех случаев, когда передача управления указанному множеству операторов имеет место, неявная передача управления на конец оператора PERFORM (ВЫПОЛНИТЬ) осуществляется следующим образом:

а) если имя-процедуры-1 является именем-параграфа и имя-процедуры-2 в операторе не указано, возврат производится после выполнения последнего оператора имени-процедуры-1;

б) если имя-процедуры-1 есть имя-секции и имя-процедуры-2 не указано, возврат производится после выполнения последнего оператора последнего параграфа имени-процедуры-1;

в) если указано имя-процедуры-2 и оно является именем- параграфа, возврат производится после выполнения последнего оператора этого параграфа;

GOST EXPERT ГОСТ 22558-89,

г) если указано имя-процедуры-2 и оно является именем-секции, возврат производится после выполнения последнего оператора последнего параграфа этой секции;

д) если указан оператор PERFORM (ВЫПОЛНИТЬ) с встроенной областью действия, его выполнение завершается после выполнения последнего содержащегося в нем оператора.

(8) Имя-процедуры-1 и имя-процедуры-2 не обязательно связывать какими-либо отношениями, за исключением того, что последовательность операторов должна выполняться, начиная от процедуры, названной именем-процедуры-1, и кончая процедурой, указанной именем-процедуры-2. В частности, между именем-процедуры-1 и концом имени-процедуры-2 могут встречаться операторы GO TO (ПЕРЕЙТИ) и PERFORM (ВЫПОЛНИТЬ).

Если имеется два или больше путей, ведущих к точке возврата, то имя-процедуры-2 может быть именем-параграфа, состоящего из оператора EXIT (ВЫЙТИ), к которому должны вести все эти пути.

(9) Если к указанному множеству операторов управление передается не по оператору PERFORM (ВЫПОЛНИТЬ), то от последнего оператора к следующему выполнимому оператору оно передается так, как если бы не было оператора PERFORM (ВЫПОЛНИТЬ), в котором упоминается это множество.

(10) Операторы PERFORM (ВЫПОЛНИТЬ) действуют следующим образом:

а) формат 1 является основным форматом оператора PERFORM (ВЫПОЛНИТЬ). Указанное множество операторов выполняется один раз, и затем управление передается на конец оператора PERFORM (ВЫПОЛНИТЬ);

б) формат 2 является вариантом с повторениями (вариант TIMES (РАЗ)). При его использовании указанное множество операторов выполняется столько раз, сколько указывается начальным значением данного, представленного идентификатором-1 или целым-1. Значение идентификатора-1 или целое-1 должно быть положительным. Если значение идентификатора-1 отрицательно или нуль, управление передается на конец оператора PERFORM (ВЫПОЛНИТЬ).

После выполнения указанного множества операторов соответствующее число раз управление передается на конец оператора PERFORM (ВЫПОЛНИТЬ). Обращение к идентификатору-1 не может изменить на протяжении выполнения оператора PERFORM (ВЫПОЛНИТЬ) число выполнений указанного множества операторов, заданное начальным значением данного, представленного идентификатором-1;

в) формат 3 указывает выполнение указанного множества операторов до тех пор, пока условие, указанное фразой UNTIL (ДО),

GOST EXPERT ГОСТ 22558-89,

не станет истинным. Если условие истинно, управление передается на конец оператора PERFORM (ВЫПОЛНИТЬ). Если в момент начала выполнения оператора PERFORM (ВЫПОЛНИТЬ) условие истинно и фраза TEST BEFORE (С ПРОВЕРКОЙ В НАЧАЛЕ) указана или подразумевается, указанные процедуры не выполняются и управление передается на конец оператора PERFORM (ВЫПОЛНИТЬ).

Если указана фраза TEST AFTER (С ПРОВЕРКОЙ В КОНЦЕ), оператор PERFORM (ВЫПОЛНИТЬ) выполняется так же, как и в случае, когда указана фраза TEST BEFORE (С ПРОВЕРКОЙ В НАЧАЛЕ) с тем отличием, что условие проверяется после выполнения указанного множества операторов.

Индексирование и модификация ссылок, имеющие отношение к операндам, указанным в условии-1, вычисляются при каждой проверке условия;

г) формат 4 используется для увеличения значения одного или более идентификаторов или имен-индексов в установленном порядке во время выполнения оператора PERFORM (ВЫПОЛНИТЬ). В нижеприведенных пояснениях все сказанное для значения идентификатора, указанного во фразах VARYING (МЕНЯЯ), AFTER (ЗАТЕМ), FROM (ОТ) (текущего значения), относится и к именам-индексов.

Если указаны имя-индекса-1 или имя-индекса-2, значение соответствующего индекса в начале оператора PERFORM (ВЫПОЛНИТЬ) должно быть установлено на номер вхождения элемента в таблице. Если указаны имя-индекса-2 или имя-индекса-4, значение данного, представленного идентификатором-2 или идентификатором-5, в начале оператора PERFORM (ВЫПОЛНИТЬ) должно быть равно номеру вхождения элемента в таблице, соответствующей имени-индекса-2 или имени-индекса-4. Как описано ниже, последующее изменение имени-индекса-1 или имени-индекса-3 не должно приводить к тому, что соответствующий индекс примет значение, находящееся вне допустимого для него диапазона, за исключением ситуации после завершения оператора PERFORM (ВЫПОЛНИТЬ), когда индекс, соответствующий имени-индекса-1, может содержать значение, большее или меньшее границ допустимого диапазона на одно значение шага изменения. Если идентификатор-2 или идентификатор-5 индексированы, индексы вычисляются каждый раз, когда устанавливается или изменяется значение данного, представленного идентификатором. Если индексированы идентификатор-3, идентификатор-4, идентификатор-6 или идентификатор-7, индексы вычисляются каждый раз при установке или изменении содержимого данного, представленного идентифи-

GOST EXPERT ГОСТ 22558-89,

фикатором. Индексирование и модификация ссылок, соответствующие операндам, указанным в условии-1 или условии-2, выполняются при каждой проверке условия.

Далее представлены схемы выполнения нескольких типов оператора PERFORM (ВЫПОЛНИТЬ) формата 4, которые не должны рассматриваться как предписания по их реализации.

1) Если фраза TEST BEFORE (С ПРОВЕРКОЙ В НАЧАЛЕ) указана или подразумевается.

Когда изменяется данное, представленное одним идентификатором, то в начале выполнения оператора PERFORM (ВЫПОЛНИТЬ) данное, представленное идентификатором-2, устанавливается равным литералу-1 или текущему значению данного, представленного идентификатором-3; если условие, заданное фразой UNTIL (ДО), ложно, то один раз выполняется указанное множество операторов.

Затем к значению данного, представленного идентификатором-2, прибавляется значение данного, представленного идентификатором-4 или литерала-2 (при этом оно увеличивается или уменьшается), и условие-1 опять проверяется. Цикл продолжается до тех пор, пока это условие не станет истинным; в этом случае управление передается на конец оператора PERFORM (ВЫПОЛНИТЬ).

Если условие-1 истинно в начале выполнения оператора PERFORM (ВЫПОЛНИТЬ), то управление передается на конец оператора PERFORM (ВЫПОЛНИТЬ).

На рис. 1 представлена схема алгоритма оператора PERFORM (ВЫПОЛНИТЬ) с фразой VARYING (МЕНЯЯ) и фразой TEST BEFORE (С ПРОВЕРКОЙ В НАЧАЛЕ) и одним условием.

Когда указано изменение данных, представленных двумя идентификаторами, содержимому данного, представленного идентификатором-2, присваивается значение литерала-1 или текущее значение данного, представленного идентификатором-3; содержимому данного, представленного идентификатором-5, присваивается значение литерала-3 или текущее значение данного, представленного идентификатором-6, после чего вычисляется условие-1; если оно истинно, то управление передается на конец оператора PERFORM (ВЫПОЛНИТЬ), если оно ложно, вычисляется условие-2. Если условие-2 ложно, то один раз выполняется указанное множество операторов, после чего содержимое данного, представленного идентификатором-5, увеличивается на литерал-4 или содержимое данного, представленного идентификатором-7, и опять вычисляется условие-2. Этот цикл вычислений и увеличений продолжается до тех пор, пока

GOST EXPERT ГОСТ 22558-89,

условие не станет истинным. Когда условие-2 истинно, содержимое данного, представленного идентификатором-2, увеличивается на литерал-2 или содержимое данного, представленного идентификатором-4, а содержимому данного, представленного идентификатором-5, присваивается значение литерала-3 или текущее значение данного, представленного идентификатором-6, и опять вычисляется условие-1. Оператор PERFORM (ВЫПОЛНИТЬ) завершается, если условие-1 истинно, иначе цикл продолжается, пока условие-1 не станет истинным.

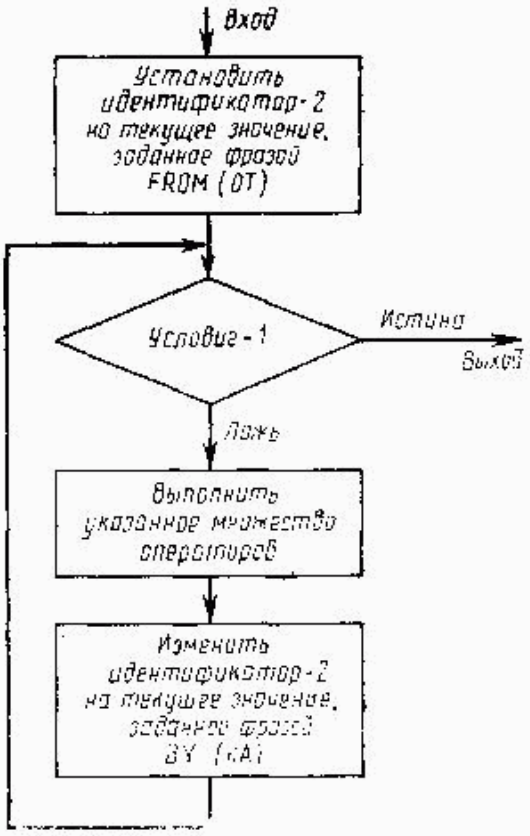


Рис. 1

На рис. 2 представлен алгоритм оператора PERFORM (ВЫПОЛНИТЬ) с фразой VARYING (МЕНЯЯ), фразой TEST BEFORE (С ПРОВЕРКОЙ В НАЧАЛЕ) и двумя условиями.

По окончании выполнения оператора PERFORM (ВЫПОЛНИТЬ) данное, представленное идентификатором-5, содержит литерал-3 или текущее значение данного, представленного идентификатором-6.

Данное, представленное идентификатором-2, содержит значение, которое отличается от последнего используемого значения на величину приращения (положительную или отрицательную), кроме случая, когда условие-1 было истинно в начале выполнения оператора PERFORM (ВЫПОЛНИТЬ); в ука-

GOST EXPERT ГОСТ 22558-89,

занном случае данное, представленное идентификатором-2, содержит литерал-1 или текущее значение данного, представленного идентификатором-3.

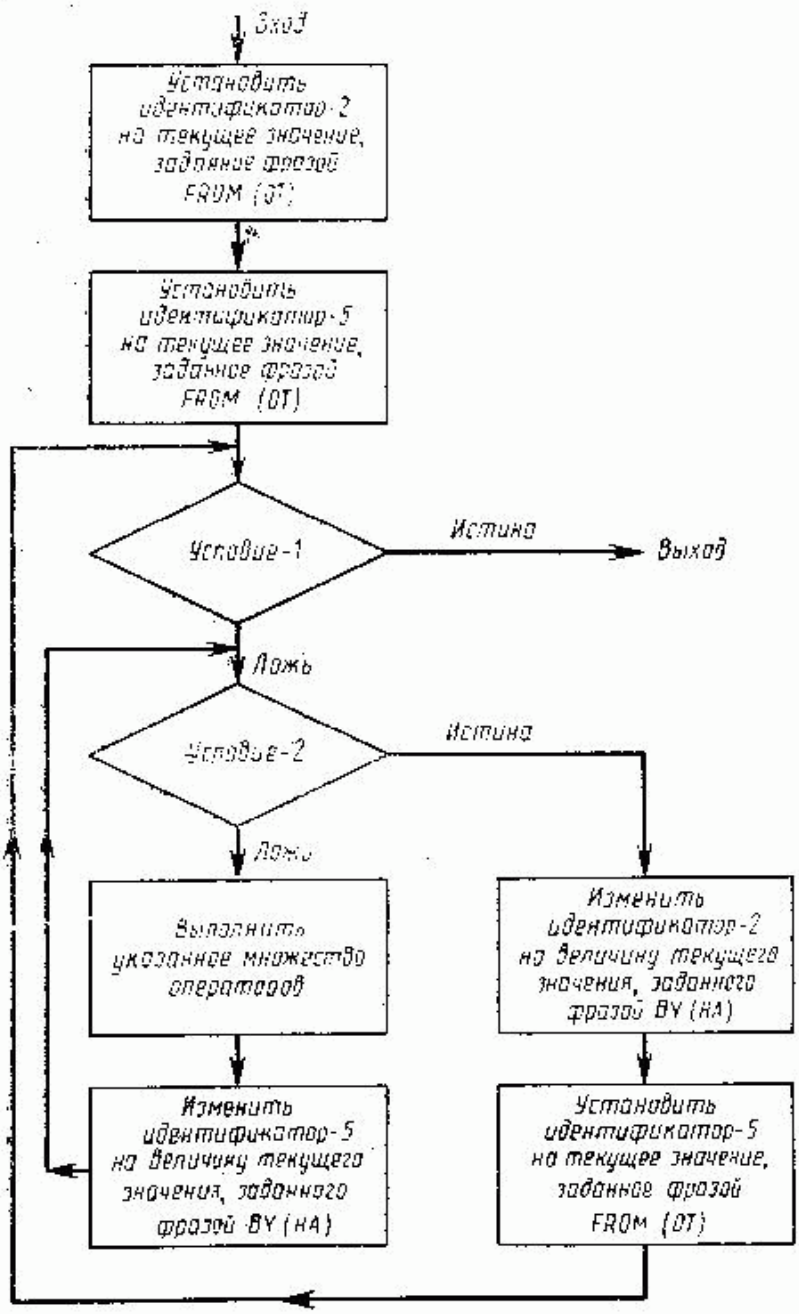


Рис. 2

2) Если фраза TEST AFTER (С ПРОВЕРКОЙ В КОНЦЕ) указана или подразумевается.

При изменении данного, представленного одним идентификатором, в начале выполнения оператора PERFORM (ВЫПОЛНИТЬ) содержимому данного, представленного идентификатором-2, присваивается значение литерала-1 или текущее зна-

GOST EXPERT ГОСТ 22558-89,

чение данного, представленного идентификатором-3; затем один раз выполняется указанное множество операторов и проверяется условие-1 из фразы UNTIL (ДО). Если условие ложно, значение данного, представленного идентификатором-1, изменяется на величину приращения (положительного или отрицательного), заданного литералом-2 или значением данного, представленного идентификатором-4; затем опять выполняется указанное множество операторов. Цикл повторяется, пока условие-1 не станет истинным, после чего управление передается на конец оператора PERFORM (ВЫПОЛНИТЬ).

На рис. 3 представлен алгоритм оператора PERFORM (ВЫПОЛНИТЬ) с фразой VARYING (МЕНЯЯ) и фразой TEST AFTER (С ПРОВЕРКОЙ В КОНЦЕ) и одним условием.

При изменении данных, представленных двумя идентификаторами, содержимому данного, представленного идентификатором-2, присваивается значение литерала-1 или текущее значение данного, представленного идентификатором-3, затем содержимому данного, представленного идентификатором-5, присваивается значение литерала-3 или текущее значение данного, представленного идентификатором-6; после этого выполняется указанное множество операторов. Затем вычисляется условие-2; если оно ложно, содержимое данного, представленного идентификатором-5, изменяется на величину литерала-4 или содержимого данного, представленного идентификатором-7; далее опять выполняется указанное множество операторов. Цикл продолжается до тех пор, пока условие-2 не окажется истинным в момент вычисления условия-2 и не потребуется вычислять условие-1. Если же оно ложно, содержимое данного, представленного идентификатором-2, изменяется на величину литерала-2 или содержимого данного, представленного идентификатором-1, а содержимому данного, представленного идентификатором-5, присваивается значение литерала-3 или текущее значение данного, представленного идентификатором-6; затем опять выполняется указанное множество операторов. Этот цикл повторяется до тех пор, пока вновь вычисленное условие-1 не станет истинным, тогда управление передается на конец оператора PERFORM (ВЫПОЛНИТЬ).

После завершения выполнения оператора PERFORM (ВЫПОЛНИТЬ) каждое данное, измененное фразами AFTER (ЗАТЕМ) и VARYING (МЕНЯЯ), содержит то значение, которое оно получило в результате последнего выполнения указанного множества операторов.

На рис. 4 представлен алгоритм оператора PERFORM (ВЫПОЛНИТЬ) с фразой VARYING (МЕНЯЯ) и фразой TEST AFTER (С ПРОВЕРКОЙ В КОНЦЕ) и двумя условиями.

GOST EXPERT ГОСТ 22558-89,

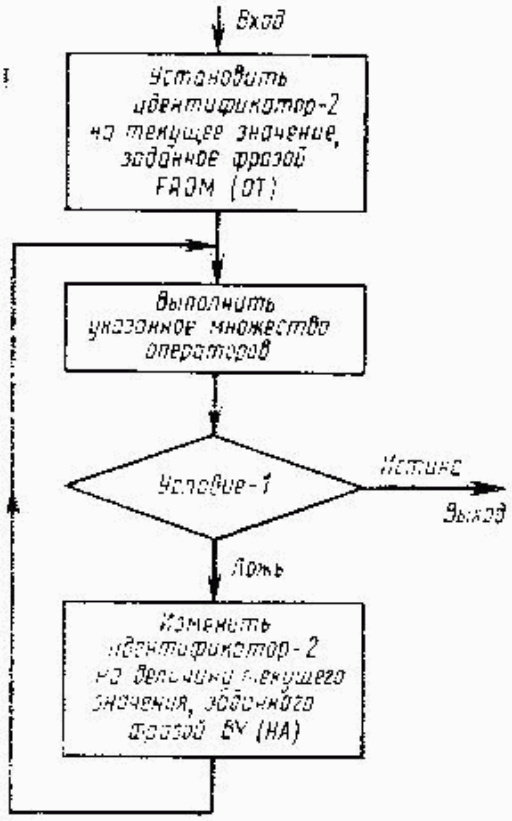


Рис. 3

Во время выполнения соответствующего оператору PERFORM (ВЫПОЛНИТЬ) указанного множества операторов следует учитывать изменение переменных, указанных во фразе VARYING (МЕНЯЯ) (данного, представленного идентификатором-2, и имени-индекса-1), во фразе BY (НА) (данного, представленного идентификатором-4), во фразе AFTER (ЗАТЕМ) (данного, представленного идентификатором-5, и имени-индекса-3), во фразе FROM (ОТ) (данного, представленного идентификатором-3, и имени-индекса-2); эти изменения будут влиять на выполнение оператора PERFORM (ВЫПОЛНИТЬ).

При изменении данных, представленных двумя идентификаторами, каждый раз, когда изменяется содержимое данного, представленного идентификатором-2, данное, представленное идентификатором-5, проходит полный цикл (с учетом фраз FROM (ОТ), BY (НА), UNTIL (ДО)).

При изменении трех или более данных, представленных идентификаторами, обработка происходит так же, как и в случае двух идентификаторов, за исключением того факта, что данное, изменяемое каждой фразой AFTER (ЗАТЕМ), проходит полный цикл всякий раз, когда добавляется приращение (положительное или отрицательное) к данному, измененному предыдущей фразой AFTER (ЗАТЕМ).

GOST EXPERT ГОСТ 22558-89,

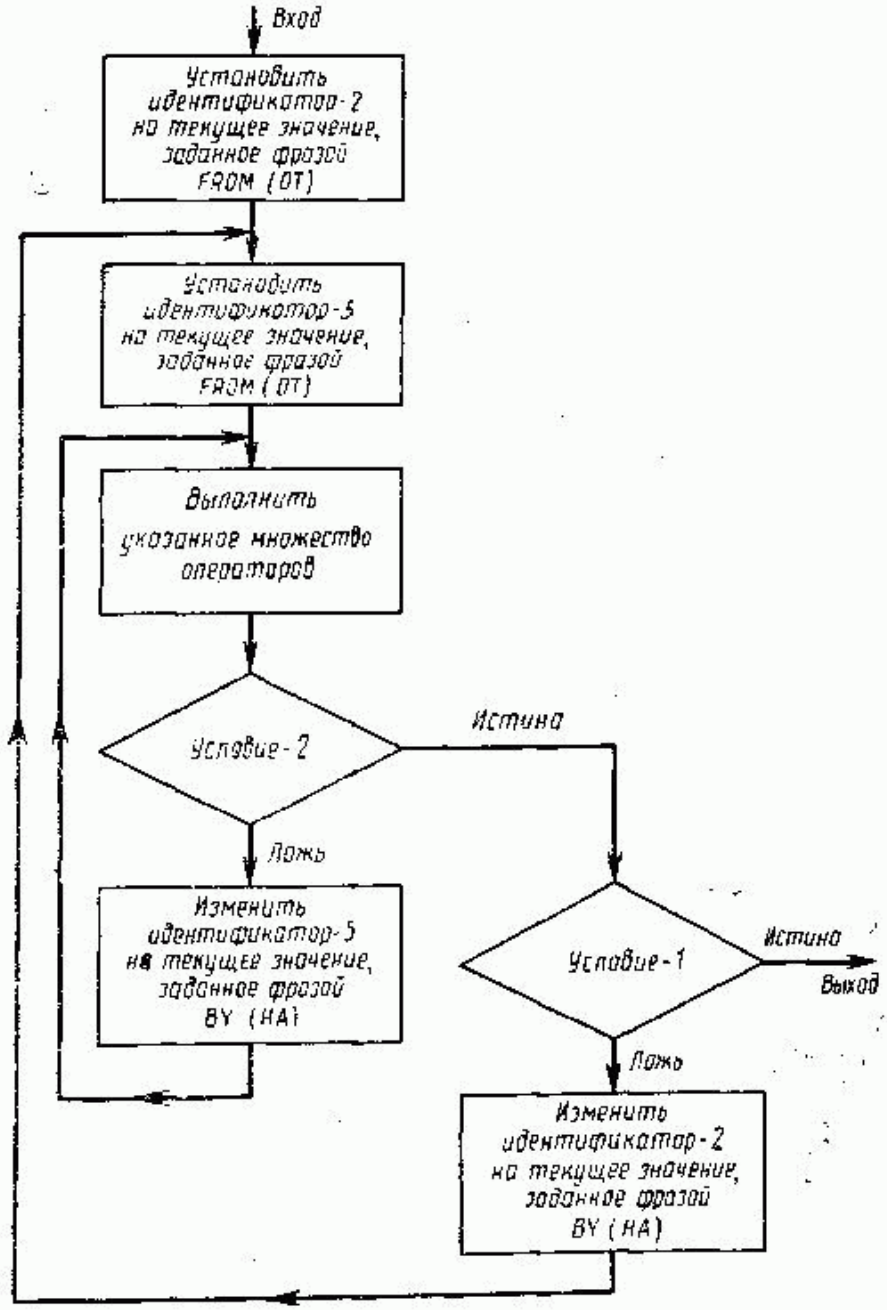


Рис. 4

GOST EXPERT ГОСТ 22558-89,

(11) Область действия оператора PERFORM (ВЫПОЛНИТЬ) логически состоит из всех тел операторов, которые выполняются в результате выполнения оператора PERFORM (ВЫПОЛНИТЬ) вплоть до выполнения неявной передачи управления на конец оператора PERFORM (ВЫПОЛНИТЬ).

Область действия включает все операторы, на которые передается управление операторами GO TO (ПЕРЕЙТИ К), EXIT (ВЫЙТИ), CALL (ВЫЗВАТЬ), PERFORM (ВЫПОЛНИТЬ), входящими в область действия оператора PERFORM (ВЫПОЛНИТЬ), а также все операторы в декларативных процедурах, которые выполняются как результат выполнения операторов, принадлежащих области действия оператора PERFORM (ВЫПОЛНИТЬ). Операторы, принадлежащие области действия оператора PERFORM (ВЫПОЛНИТЬ), не обязательно должны следовать друг за другом в тексте исходной программы.

(12) Операторы, выполняемые в результате передачи управления по оператору EXIT PROGRAM (ВЫЙТИ ИЗ ПРОГРАММЫ), не принадлежат области действия оператора PERFORM (ВЫПОЛНИТЬ) в том случае, когда:

а) оператор EXIT PROGRAM (ВЫЙТИ ИЗ ПРОГРАММЫ) определен в той же программе, что и оператор PERFORM (ВЫПОЛНИТЬ);
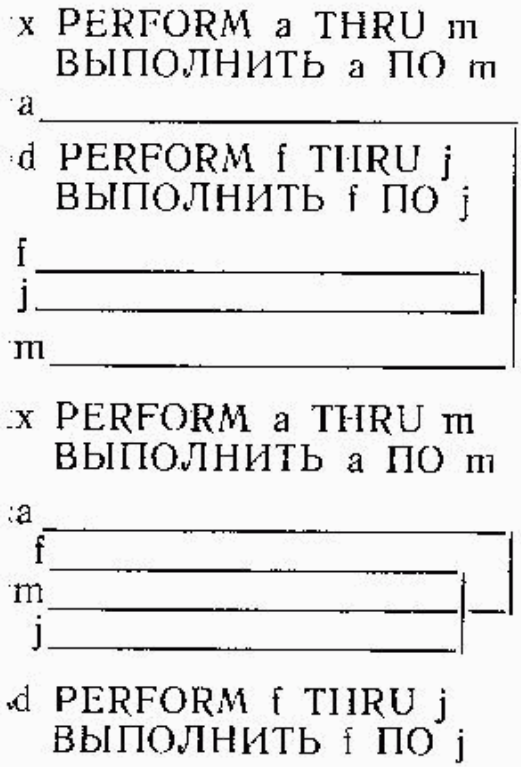
б) оператор EXIT PROGRAM (ВЫЙТИ ИЗ ПРОГРАММЫ) принадлежит области действия оператора PERFORM (ВЫПОЛНИТЬ).

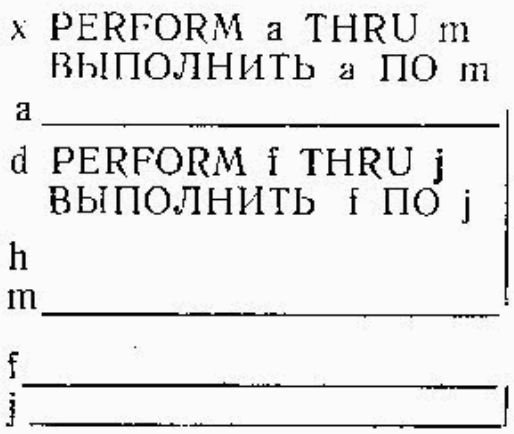
(13) Имя-процедуры-1 и имя-процедуры-2 не должны быть именами секций или параграфов в любой другой программе данной единицы исполнения независимо от того, содержит ли другая программа программу, включающую оператор PERFORM (ВЫПОЛНИТЬ), или сама содержится в ней. Операторы из других программ в единице исполнения могут быть доступны только в результате выполнения оператора PERFORM (ВЫПОЛНИТЬ), область действия которого включает операторы CALL (ВЫЗВАТЬ) и EXIT PROGRAM (ВЫЙТИ ИЗ ПРОГРАММЫ) (ч. 10, п. 1.3.8).

(14) Если область действия оператора PERFORM (ВЫПОЛНИТЬ) включает другой оператор PERFORM (ВЫПОЛНИТЬ), то последовательность процедур, связанная с внутренним оператором PERFORM (ВЫПОЛНИТЬ), должна полностью включаться в логическую последовательность, определяемую первым оператором PERFORM (ВЫПОЛНИТЬ), или полностью находиться вне этой последовательности. Таким образом, активный оператор PERFORM (ВЫПОЛНИТЬ), выполнение которого начинается внутри области действия другого активного в данный момент оператора PERFORM (ВЫПОЛНИТЬ), не должен передавать управление на точку выхода последнего; более того, несколько та-

GOST EXPERT ГОСТ 22558-89,

ких активных операторов PERFORM (ВЫПОЛНИТЬ) не могут иметь общую точку выхода. Ниже приведены примеры возможных комбинаций областей действия нескольких операторов PERFORM (ВЫПОЛНИТЬ).





(15) Оператор PERFORM (ВЫПОЛНИТЬ), который появляется в секции, не являющейся независимым сегментом, может иметь внутри своей области действия кроме декларативных секций, вызываемых из его области действия, только следующие процедуры:

а) секции и (или) параграфы, содержащиеся полностью в одном или в нескольких сегментах, не являющихся независимыми;

б) секции и (или) параграфы, полностью содержащиеся в одном независимом сегменте.

(16) Оператор PERFORM (ВЫПОЛНИТЬ), который появляется в независимом сегменте, может иметь внутри своей области действия кроме декларативных секций, вызываемых из области действия, только следующие процедуры:

а) секции и (или) параграфы, полностью содержащиеся в одном или нескольких сегментах, не являющихся независимыми;

б) секции и (или) параграфы, полностью содержащиеся в том же независимом сегменте, что и оператор PERFORM (ВЫПОЛНИТЬ).

GOST EXPERT ГОСТ 22558-89,

6.22. Оператор SEARCH (ИСКАТЬ)

6.22.1. Назначение

Оператор SEARCH (ИСКАТЬ) используется для поиска в таблице некоторого элемента таблицы, который удовлетворяет заданному условию, и установления соответствующего значения индекса, указывающего этот элемент таблицы.

6.22.2. Общий формат

Формат 1

```
SEARCH идентификатор-1 [VARYING { идентификатор-2 | имя-индекса-1 } ]
  [AT END повелительный-оператор-1]
  { WHEN условие-1 { повелительный-оператор-2 | NEXT SENTENCE } } ...
  [END-SEARCH]

ИСКАТЬ В идентификатор-1 [ МЕНЯЯ { идентификатор-2 | имя-индекса-1 } ]
  [В КОНЦЕ повелительный-оператор-1]
  { КОГДА условие-1 { повелительный-оператор-2 | СЛЕДУЮЩЕЕ ПРЕДЛОЖЕНИЕ } } ...
  [КОНЕЦ-ИСКАТЬ]
```

Формат 2

```
SEARCH ALL идентификатор-1
  [AT END повелительный-оператор-1]
  WHEN { имя-данного-1 { IS EQUAL TO | IS = } { идентификатор-3 | литерал-1 | арифметическое-выражение-1 }
       | имя-условия-1 }
```


GOST EXPERT ГОСТ 22558-89,

```
[ AND { имя-данного-2 { IS EQUAL TO / IS = } { идентификатор-4 / литерал-2 / арифметическое-выражение-2 } } ... ]
      { имя условия-2 }

{ повелительный-оператор-2 / NEXT SENTENCE }

[END-SEARCH]

ИСКАТЬ ОСОБО В идентификатор-1 [В КОНЦЕ повелительный-оператор-1]

КОГДА { имя-данного-1 { РАВНО / = } { идентификатор-3 / литерал-1 / арифметическое-выражение-1 } }
      { имя-условия-1 }

[ И { имя-данного-2 { РАВНО / = } { идентификатор-4 / литерал-2 / арифметическое-выражение-2 } } ... ]
   { имя-условия-2 }

{ повелительный-оператор-2 / СЛЕДУЮЩЕЕ ПРЕДЛОЖЕНИЕ }

[КОНЕЦ-ИСКАТЬ]
```

6.22.3. Синтаксические правила

(1) В форматах 1 и 2 идентификатор-1 не должен быть индексирован или представлять модифицированную ссылку, но его описание должно содержать фразу OCCURS (ПОВТОРЯЕТСЯ) с вариантом INDEXED (ИНДЕКСИРУЕТСЯ). Описа-

GOST EXPERT ГОСТ 22558-89,

ние идентификатора-1 в формате 2 должно содержать также вариант ASCENDING (DESCENDING) KEY (ПО ВОЗРАСТАНИЮ (УБЫВАНИЮ) КЛЮЧА) во фразе OCCURS (ПОВТОРЯЕТСЯ).

(2) Идентификатор-2 должен ссылаться на данное, описанное с фразой USAGE IS INDEX (ДЛЯ ИНДЕКСА) или как числовое элементарное данное, не имеющее позиций справа от подразумеваемой десятичной точки. Идентификатор-2 не может индексироваться первым (или единственным) именем-индекса, определенным вариантом INDEXED BY (ИНДЕКСИРУЕТСЯ) во фразе OCCURS (ПОВТОРЯЕТСЯ), соответствующей идентификатору-1.

(3) В формате 1 условие-1 может быть любым допустимым в Коболе условием (см. п. 6.3 настоящей части).

(4) В формате 2 все упоминаемые имена-условий должны быть определены как имеющие только одно значение. Имя-данного, связанное с именем-условия, должно быть указано во фразе ASCENDING (DESCENDING) KEY (ПО ВОЗРАСТАНИЮ (УБЫВАНИЮ) КЛЮЧА) идентификатора-1. Каждое имя-данного-1, имя-данного-2 может уточняться. Каждое имя-данного-1, имя-данного-2 должно быть индексировано первым именем-индекса, связанным с идентификатором-1, вместе с другими требующимися индексами и должно быть указано во фразе ASCENDING (DESCENDING) KEY (ПО ВОЗРАСТАНИЮ (УБЫВАНИЮ) КЛЮЧА) идентификатора-1. Идентификатор-3, идентификатор-4 или идентификаторы, задаваемые в арифметическом-выражении-1, арифметическом-выражении-2 не должны быть указаны во фразе ASCENDING (DESCENDING) KEY (ПО ВОЗРАСТАНИЮ (УБЫВАНИЮ) КЛЮЧА) идентификатора-1 или индексироваться первым именем-индекса, связанным с идентификатором-1.

Если в формате 2 ссылаются на некоторое имя-данного, указанное в варианте ASCENDING (DESCENDING) KEY (ПО ВОЗРАСТАНИЮ (УБЫВАНИЮ) КЛЮЧА) для данного, представленного идентификатором-1, или на связанное с ним имя-условия, то все предшествующие имена-данных, указанные во фразе ASCENDING (DESCENDING) KEY (ПО ВОЗРАСТАНИЮ (УБЫВАНИЮ) КЛЮЧА) для данного, представленного идентификатором-1, или соответствующие им имена-условий тоже должны быть указаны.

(5) Если указана фраза END-SEARCH (КОНЕЦ-ИСКАТЬ), то нельзя указывать фразу NEXT SENTENCE (СЛЕДУЮЩЕЕ ПРЕДЛОЖЕНИЕ).

6.22.4. Общие правила

(1) Оператор SEARCH (ИСКАТЬ) может завершаться:

GOST EXPERT ГОСТ 22558-89,

а) фразой END-SEARCH (КОНЕЦ-ИСКАТЬ) на соответствующем уровне вложенности;

б) разделителем точка;

в) фразами ELSE (ИНАЧЕ) и END-IF (КОНЕЦ-ЕСЛИ), соответствующими предшествующему оператору IF (ЕСЛИ) (см. ч. 4, п. 6.4.3).

(2) Если используется формат 1 оператора SEARCH (ИСКАТЬ), то имеет место последовательный тип поиска (перебор), начиная с текущей установки индекса.

а) Если в начале выполнения оператора SEARCH (ИСКАТЬ) имя-индекса, связанное с идентификатором-1, содержит значение, которое соответствует номеру вхождения, большему чем наибольший допустимый номер вхождения для идентификатора-1, то оператор SEARCH (ИСКАТЬ) сразу завершается. Наибольший допустимый номер вхождения определяется во фразе OCCURS (ПОВТОРЯЕТСЯ) (см. п. 5.8 настоящей части). Тогда, если в операторе указана фраза AT END (В КОНЦЕ), выполняется повелительный-оператор-1, иначе управление передается на конец оператора SEARCH (ИСКАТЬ).

б) Если в начале выполнения оператора SEARCH (ИСКАТЬ) имя-индекса, связанное с идентификатором-1, содержит значение, которое соответствует номеру вхождения, не превосходящему наибольший допустимый номер вхождения для идентификатора-1, то оператор SEARCH (ИСКАТЬ) вычисляет условия в том порядке, в каком они написаны, используя установленные индексы, чтобы определить вхождения проверяемых данных. Если ни одно из условий не удовлетворено, имя-индекса для идентификатора-1 увеличивается, чтобы обеспечить ссылку на следующее вхождение. Этот процесс повторяется с использованием вновь установленного имени-индекса до тех пор, пока очередное значение имени-индекса для идентификатора-1 не превзойдет наибольшего допустимого. В этом последнем случае поиск заканчивается как указывалось выше. Если одно из условий удовлетворено при его вычислении, поиск заканчивается и управление передается на повелительный оператор, связанный с этим условием, если он указан, или на следующее выполнимое предложение, если указана фраза NEXT SENTENCE (СЛЕДУЮЩЕЕ ПРЕДЛОЖЕНИЕ); имя-индекса остается установленным на тот номер вхождения, при котором удовлетворилось условие.

(3) В формате 2 оператора SEARCH (ИСКАТЬ) результат операции SEARCH ALL (ИСКАТЬ ОСОБО) определен только в следующих ситуациях:

а) данные в таблице упорядочены таким же образом, как это

GOST EXPERT ГОСТ 22558-89,

описано в варианте KEY IS (ПО ВОЗРАСТАНИЮ (УБЫВАНИЮ) КЛЮЧА), связанном с идентификатором-1;

б) значения ключа (ключей), упомянутого во фразе WHEN (КОГДА), достаточны, чтобы однозначно идентифицировать элемент таблицы.

(4) Если используется формат 2 оператора SEARCH (ИСКАТЬ), то может иметь место непоследовательный тип поиска; начальная установка имени-индекса для идентификатора-1 игнорируется, и установка имени-индекса меняется в ходе операции поиска способом, определяемым реализацией, с теми ограничениями, что никогда значение имени-индекса не превзойдет значение, которое соответствует последнему элементу в таблице, и не будет меньше значения, которое соответствует первому элементу таблицы. Длина таблицы определяется во фразе OCCURS (ПОВТОРЯЕТСЯ) (см. п. 5.8 настоящей части). Если для каждой установки индекса в разрешенном интервале какое-либо условие, задаваемое во фразе WHEN (КОГДА), не удовлетворено, то управление передается повелительному-оператору-1 фразы AT END (В КОНЦЕ), если она указана, или на конец оператора SEARCH (ИСКАТЬ), если эта фраза не задана. В любом случае конечная установка индекса не определена. Если все условия могут быть удовлетворены, то индекс устанавливается на соответствующее вхождение, которое позволяет удовлетворить условия, и управление передается повелительному-оператору-2, если он указан, или следующему выполнимому предложению, если указана фраза NEXT SENTENCE (СЛЕДУЮЩЕЕ ПРЕДЛОЖЕНИЕ).

(5) После выполнения повелительного-оператора-1 или повелительного-оператора-2, которые не кончаются оператором GO TO (ПЕРЕЙТИ), управление передается на конец оператора SEARCH (ИСКАТЬ).

(6) В формате 2 имя-индекса, которое используется в операции поиска, есть первое (или единственное) имя-индекса, указанное в варианте INDEXED BY (ИНДЕКСИРУЕТСЯ) фразы OCCURS (ПОВТОРЯЕТСЯ), связанной с идентификатором-1. Все другие имена-индексов для идентификатора-1 остаются неизменными.

(7) Если в формате 1 фраза VARYING (МЕНЯЯ) не используется, имя-индекса, которое используется в операции поиска, есть первое (или единственное) имя-индекса, указанное в варианте INDEXED BY (ИНДЕКСИРУЕТСЯ) фразы OCCURS (ПОВТОРЯЕТСЯ), связанной с идентификатором-1. Все другие имена-индексов для идентификатора-1 остаются неизменными.

GOST EXPERT ГОСТ 22558-89,

(8) Если в формате 1 задана фраза VARYING (МЕНЯЯ) имя-индекса-1 и это имя-индекса-1 указано в варианте INDEXED (ИНДЕКСИРУЕТСЯ) фразы OCCURS (ПОВТОРЯЕТСЯ), связанной с идентификатором-1, то оно используется для поиска. Если имя-индекса-1 не указано в варианте INDEXED BY (ИНДЕКСИРУЕТСЯ) идентификатора-1 или задана фраза VARYING (МЕНЯЯ) идентификатор-2, то для поиска используется первое (или единственное) имя-индекса, заданное в варианте INDEXED BY (ИНДЕКСИРУЕТСЯ) фразы OCCURS (ПОВТОРЯЕТСЯ), связанной с идентификатором-1. Кроме того, имеют место следующие операции:

а) если используется фраза VARYING (МЕНЯЯ) имя-индекса-1 и это имя-индекса-1 появляется в варианте INDEXED BY (ИНДЕКСИРУЕТСЯ) фразы OCCURS (ПОВТОРЯЕТСЯ), связанной со статьей другой таблицы, то номер вхождения, представляемый именем-индекса-1, увеличивается на такую же величину и в то же время, что и номер вхождения, представляемый именем-индекса, связанным с идентификатором-1;

б) если задана фраза VARYING (МЕНЯЯ) идентификатор-2 и идентификатор-2 есть индексное данное, то данное, представленное идентификатором-2, увеличивается на такую же величину и в то же время, что и индекс, связанный с идентификатором-1. Если идентификатор-2 не является индексным данным, то значение, представленное идентификатором-2, увеличивается на единицу в то же время, что и индекс, связанный с идентификатором-1.

(9) Область действия оператора SEARCH (ИСКАТЬ) ограничивается фразой END-SEARCH (КОНЕЦ-ИСКАТЬ) (см. ч. 4, п. 6.4.3).

(10) На рис. 1 приведена схема выполнения формата 1 оператора SEARCH (ИСКАТЬ), содержащего две фразы WHEN (КОГДА). Эта схема не должна рассматриваться как предписание по реализации оператора. На приведенном рисунке звездочками помечены следующие операции:

* — операции, включаемые только тогда, когда соответствующий вариант указан в формате оператора;

** — каждая из этих передач ведет к точке выхода оператора SEARCH (ИСКАТЬ), если только повелительный-оператор не заканчивается оператором GO TO (ПЕРЕЙТИ).

6.23. Оператор SET (УСТАНОВИТЬ)

6.23.1. Назначение

(1) Оператор SET (УСТАНОВИТЬ) учреждает точки обращений для операций обработки таблиц, устанавливая имена-индексов, соответствующие элементам таблицы.

GOST EXPERT ГОСТ 22558-89,

(2) Оператор SET (УСТАНОВИТЬ) используется для изменения состояния внешних переключателей.

(3) Оператор SET (УСТАНОВИТЬ) используется также для изменения значений условных переменных.

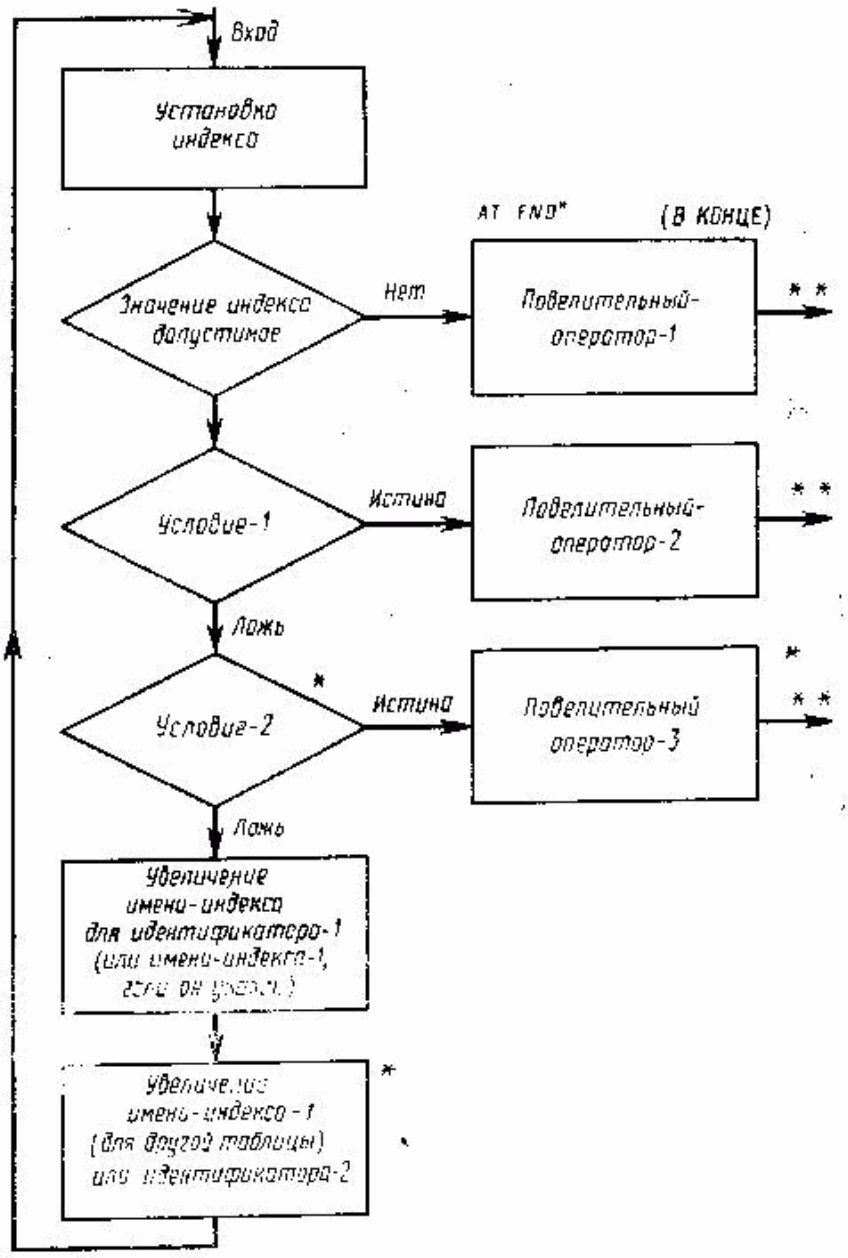


Рис. 1

6.23.2. Общий формат

Формат 1

$$\underline{\text{SET}} \left\{ \begin{array}{l} \text{имя-индекса-1} \\ \text{идентификатор-1} \end{array} \right\} \ldots \underline{\text{TO}} \left\{ \begin{array}{l} \text{идентификатор-2} \\ \text{имя-индекса-2} \\ \text{целое-1} \end{array} \right\}$$

GOST EXPERT ГОСТ 22558-89,

УСТАНОВИТЬ { идентификатор-1 / имя-индекса-1 } ... НА { идентификатор-2 / имя-индекса-2 / целое-1 }

Формат 2

SET {имя-индекса-3} ... { UP BY / DOWN BY } { идентификатор-3 / целое-2 }

УСТАНОВИТЬ {имя-индекса-3} ... { ПРИБАВЛЯЯ / ВЫЧИТАЯ }

{ идентификатор-3 / целое-2 }

Формат 3

SET {{мнемоническое-имя-1} ... ТО { ON / OFF } } ...

УСТАНОВИТЬ {{мнемоническое-имя-1} ... НА

{ ВКЛЮЧЕНО / ВЫКЛЮЧЕНО } } ...

Формат 4

SET {имя-условия-1} ... TO TRUE

УСТАНОВИТЬ {имя-условия-1} ... НА ИСТИНА

6.23.3. Синтаксические правила

(1) Все, сказанное для имени-индекса-1, идентификатора-1 и имени-индекса-3, применяется в равной мере ко всем повторениям элементов формата.

(2) Идентификатор-1 и идентификатор-2 должны представлять индексные данные или элементарные данные, описанные как целые.

(3) Идентификатор-3 должен представлять элементарное числовое целое.

(4) Целое-1 и целое-2 могут быть со знаком. Целое-1 должно быть положительным.

(5) Мнемоническое-имя-1 должно быть связано с внешним переключателем, состояние которого может быть изменено. Конкретная реализация уточняет, на какие внешние переключатели может ссылаться оператор SET (УСТАНОВИТЬ).

GOST EXPERT ГОСТ 22558-89,

(6) Имя-условия-1 должно соответствовать условной переменной.

6.23.4. Общие правила

Форматы 1 и 2

(1) Имена-индексов рассматриваются относительно конкретной таблицы и определяются заданием варианта INDEXED (ИНДЕКСИРУЕТСЯ) фразы OCCURS (ПОВТОРЯЕТСЯ) для этой таблицы.

(2) Если задано имя-индекса-1, то его значение после выполнения оператора SET (УСТАНОВИТЬ) должно соответствовать номеру вхождения элемента в таблицу, соответствующую имени-индекса-1. Значение индекса, соответствующего имени-индекса, после выполнения оператора PERFORM (ВЫПОЛНИТЬ) или оператора SEARCH (ИСКАТЬ) может быть установлено на номер вхождения, выходящий за пределы области допустимых значений для индексов соответствующей таблицы (см. п. 6.21 настоящей части). Если указано имя-индекса-2, то его значение перед выполнением оператора SET (УСТАНОВИТЬ) должно соответствовать номеру вхождения элемента в таблицу, связанную с именем-индекса-1.

Если задано имя-индекса-3, то его значение до и после выполнения оператора SET (УСТАНОВИТЬ) должно соответствовать номеру вхождения элемента в таблицу, связанную с именем-индекса-3.

(3) В формате 1 выполняются следующие действия:

а) имя-индекса-1 устанавливается на значение, указывающее на элемент таблицы, соответствующий номеру вхождения, определяемому именем-индекса-2, идентификатором-2 или целым-1. Если идентификатор-2 есть индексное данное или если имя-индекса-2 относится к той же таблице, что и имя-индекса-1, то преобразование не происходит;

б) если идентификатор-1 есть индексное данное, то он может быть установлен равным либо значению имени-индекса-2, либо идентификатору-2, где идентификатор-2 также является индексным данным; в обоих случаях преобразование не происходит;

в) если идентификатор-1 не является индексным данным, то он может быть установлен только на номер вхождения, соответствующий значению имени-индекса-2. В этом случае ни идентификатор-2, ни целое-1 не могут быть использованы;

г) процесс повторяется для каждого вхождения имени-индекса-1 и идентификатора-1, если они заданы. При этом каждый раз используется значение имени-индекса-2 или данного, представленного идентификатором-2, которое они имели в начале выполнения оператора. Всякое индексирование, связанное с идентифика-

GOST EXPERT ГОСТ 22558-89,

тором-1, выполняется непосредственно перед изменением соответствующего данного.

(4) В формате 2 значение имени-индекса-3 увеличивается или уменьшается на значение, которое соответствует числу вхождений, представляемому целым-2 или значением данного, представленного идентификатором-3, после этого процесс повторяется для каждого вхождения имени-индекса-3 и т. д. Каждый раз используется значение данного, представленного идентификатором-3, такое же, как в начале выполнения оператора.

(5) Ниже представлены допустимые комбинации операндов в операторе SET (УСТАНОВИТЬ) в формате 1. Указанные ссылки относятся к общим правилам, применяемым при пересылке для данного сочетания операндов.

Пересылаемое данное	Принимающее данное		
	Целое-данное	Имя-индекса	Индексное данное
Целое-литерал	Не разрешено 3в	Разрешено 3а	Не разрешено 3б
Целое-данное	Не разрешено 3в	Разрешено 3а	Не разрешено 3б
Имя-индекса	Разрешено 3в	Разрешено 3а	Разрешено 3б
Индексное	Не разрешено 3в	Разрешено 3а*	Разрешено 3б*

* Никакие преобразования не имеют места.

Формат 3

(6) Состояние каждого внешнего переключателя, соответствующего мнемоническому-имени-1, изменяется таким образом, что результирующее значение «истина» вычисления имени-условия, связанного с этим переключателем, будет отражать состояние «включено», если указана фраза ON (ВКЛЮЧЕНО), или состояние «выключено»; если указана фраза OFF (ВЫКЛЮЧЕНО) (см. п. 6.3.1.4 настоящей части).

Формат 4

(7) Литерал во фразе VALUE (ЗНАЧЕНИЕ), соответствующей имени-условия-1, присваивается условной переменной согласно правилам использования фразы VALUE (ЗНАЧЕНИЕ) (см. п. 5.15 настоящей части). Если указано несколько литералов во фразе VALUE (ЗНАЧЕНИЕ), условной перемен-

GOST EXPERT ГОСТ 22558-89,

ной присваивается значение первого литерала, заданного в этой фразе.

(8) Если указано несколько имен условий при использовании формата 4, результаты такие же, как и в случае записи отдельного оператора SET (УСТАНОВИТЬ) для каждого имени-условия-1 в том же порядке, в каком они указаны в операторе SET (УСТАНОВИТЬ).

6.24. Оператор STOP (ОСТАНОВИТЬ)

6.24.1. Назначение

Оператор STOP (ОСТАНОВИТЬ) прекращает полностью или временно выполнение единицы исполнения.

Указание литерала в операторе STOP (ОСТАНОВИТЬ) является устаревшим элементом в настоящем стандарте и будет удалено в следующей редакции стандарта.

6.24.2. Общий формат

$$\underline{\text{STOP}} \left\{ \begin{array}{l} \underline{\text{RUN}} \\ \underline{\text{литерал-1}} \end{array} \right\}$$

$$\underline{\text{ОСТАНОВИТЬ}} \left\{ \begin{array}{l} \underline{\text{РАБОТУ}} \\ \underline{\text{литерал-1}} \end{array} \right\}$$

6.24.3. Синтаксические правила

(1) Литерал-1 не может быть стандартной константой, начинающейся словом ALL (ВСЕ).

(2) Если оператор STOP RUN (ОСТАНОВИТЬ РАБОТУ) появляется в последовательности повелительных операторов предложения, то он должен быть последним оператором предложения.

(3) Если литерал числовой, то он должен быть целым без знака.

6.24.4. Общие правила

(1) Если указана фраза RUN (РАБОТУ), выполнение единицы исполнения прекращается и управление передается операционной системе.

(2) При выполнении оператора STOP RUN (ОСТАНОВИТЬ РАБОТУ) выполняется неявный оператор CLOSE (ЗАКРЫТЬ) без необязательных вариантов для всех открытых файлов данной единицы исполнения. Никакие процедуры USE (ИСПОЛЬЗОВАТЬ), соответствующие этим файлам, не выполняются.

(3) Если единица исполнения получала сообщения, то по оператору STOP RUN (ОСТАНОВИТЬ РАБОТУ) система управления сообщениями уничтожает в очереди сообщения, лишь частично полученные этой единицей исполнения.

Часть сообщения, передаваемая из единицы исполнения оператором SEND (ПОСЛАТЬ), но не законченная EMI (ИКС) или EGI (ИКГ), исключается из системы.

GOST EXPERT ГОСТ 22558-89,

(4) Если указан вариант STOP (ОСТАНОВИТЬ) литерал-1, выполнение единицы исполнения приостанавливается, а литерал-1 сообщается оператору ЭВМ. Продолжение функционирования единицы исполнения начинается со следующего выполнимого оператора в том случае, когда подключена зависящая от реализации процедура управления возобновлением единицы исполнения.

6.25. **Оператор STRING (СОБРАТЬ)**

6.25.1. Назначение

Оператор STRING (СОБРАТЬ) соединяет часть или полное содержимое двух или нескольких данных в одно данное.

6.25.2. Общий формат

```
STRING { { идентификатор-1 } ...
       { { литерал-1       }

  DELIMITED BY { { идентификатор-2 } } ...
               { { литерал-2       } }
               { { SIZE            } }

  INTO идентификатор-3
  [WITH POINTER идентификатор-4]
  [ON OVERFLOW повелительный-оператор-1]
  [NOT ON OVERFLOW повелительный-оператор-3]
  [END-STRING]

СОБРАТЬ { { идентификатор-1 } ... ОГРАНИЧИВАЯСЬ
        { { литерал-1       }

  { { идентификатор-2 } } ...
  { { литерал-2       } }
  { { РАЗМЕРОМ        } }

  В идентификатор-3 [УКАЗАТЕЛЬ идентификатор-4]
  [ПРИ ПЕРЕПОЛНЕНИИ повелительный-оператор-1]
[БЕЗ ПЕРЕПОЛНЕНИЯ повелительный-оператор-2]
[КОНЕЦ-СОБРАТЬ]
```

6.25.3. Синтаксические правила

(1) Литерал-1 и литерал-2 не могут быть стандартными константами, начинающимися со слова ALL (ВСЕ).

(2) Все литералы должны быть нечисловыми литералами и все идентификаторы, за исключением идентификатора-4, долж-

GOST EXPERT ГОСТ 22558-89,

ны быть определены явно или неявно с использованием DISPLAY (ДЛЯ ВЫДАЧИ).

(3) Идентификатор-3 не может быть модификацией ссылки.

(4) Идентификатор-3 не может быть редактируемым данным; он не может быть описан с фразой JUSTIFIED (СДВИНУТО).

(5) Идентификатор-4 должен быть описан как элементарное числовое целое данное, имеющее достаточный размер для того, чтобы содержать значение, равное увеличенному на единицу размеру данного, представленного идентификатором-3. Символ P (M) не может использоваться в строке-литер шаблона данного, представленного идентификатором-4.

(6) Если идентификатор-1 или идентификатор-2 представляют элементарные числовые данные, то они должны быть описаны как целые, причем соответствующие строки-литер шаблона не должны содержать литеры P (M).

6.25.4. Общие правила

(1) Идентификатор-1 и литерал-1 представляют собой пересылаемые данные, а идентификатор-3 — принимающее данное.

(2) Литерал-2 и значение данного, представленного идентификатором-2, указывают одну или несколько литер, ограничивающих перемещение. Если используется фраза SIZE (РАЗМЕРОМ), данные, определенные идентификатором-1 или литералом-1, помещаются полностью. Если в качестве ограничителя используется стандартная константа, она представляет нечисловой литерал длиной в одну литеру.

(3) Если вместо литерала-1 и литерала-2 используется стандартная константа, то она представляет данное длиной в одну литеру с использованием DISPLAY (ДЛЯ ВЫДАЧИ).

(4) Перемещение данного при выполнении оператора STRING (СОБРАТЬ) определяется следующими правилами:

а) литеры из литерала-1 или из значения данного, представляемого идентификатором-1, передаются в данное, представляемое идентификатором-3, согласно правилам перемещения буквенно-цифрового в буквенно-цифровое данное, однако дополнение пробелами не производится;

б) если задана фраза DELIMITED (ОГРАНИЧИВАЯСЬ) без фразы SIZE (РАЗМЕРОМ), значение данного, представляемого идентификатором-1, или значение литерала-1 передается в принимающее данное в последовательности, заданной оператором STRING (СОБРАТЬ), начиная с самой левой литеры и продолжая слева направо, пока не будет достигнут конец данного или не встретится литера (литеры), заданная литералом-2 или значением данного, представленного идентификатором-2. Эта литера (литеры) не пересылается;

GOST EXPERT ГОСТ 22558-89,

в) если задана фраза DELIMITED BY SIZE (ОГРАНИЧИВАЯСЬ РАЗМЕРОМ), значение, представленное литералом-1, или значение данного, представляемого идентификатором-1, передаются в последовательности, заданной оператором STRING (СОБРАТЬ), в данное, представляемое идентификатором-3, пока все данные не будут переданы или не будет достигнут конец данного, определяемого идентификатором-3.

Эти действия повторяются до исчерпания всех вхождений литерала-1 или данных, представленных идентификатором-1.

(5) Если задана фраза POINTER (УКАЗАТЕЛЬ), значение данного, представленного идентификатором-4, перед выполнением оператора STRING (СОБРАТЬ) должно быть установлено в начальное значение (не меньше 1).

(6) Если фраза POINTER (УКАЗАТЕЛЬ) не задана, оператор выполняется так, как если бы было задано начальное значение данного, представленного идентификатором-4, равное 1.

(7) Когда литеры пересылаются в принимающее данное, на которое ссылается идентификатор-3, пересылка литер в принимающее данное начинается от указанной значением данного, представленного идентификатором-4, позиции литеры этого данного (при условии, что значение данного, представленного идентификатором-4, не превышает длины данного, представленного идентификатором-3) и затем данное, представленное идентификатором-4, увеличивается на единицу перед пересылкой каждой литеры или перед завершением выполнения оператора STRING (СОБРАТЬ). Значение данного, представленного идентификатором-4, во время выполнения оператора STRING (СОБРАТЬ) изменяется только вышеописанным методом.

(8) В результате выполнения оператора STRING (СОБРАТЬ) изменяется только та часть данного, представленного идентификатором-3, к которой было обращение во время выполнения оператора STRING (СОБРАТЬ). Оставшаяся часть данного, представленного идентификатором-3, после выполнения оператора STRING (СОБРАТЬ) не изменяется.

(9) Перед каждой пересылкой литеры в данное, представленное идентификатором-3, если значение данного, представленного идентификатором-4, окажется меньшим 1 либо превзойдет число позиций литер в данном, представляемом идентификатором-3, дальнейшее перемещение в данное, представляемое идентификатором-3, прекращается и фраза NOT ON OVERFLOW (БЕЗ ПЕРЕПОЛНЕНИЯ), если она указана, игнорируется, а управление передается на конец оператора STRING (СОБРАТЬ). Если в формате оператора указана фраза ON OVERFLOW (ПРИ ПЕРЕПОЛНЕНИИ), то выполняется повелительный-оператор-1, указанный в этой фразе. Если управле-

GOST EXPERT ГОСТ 22558-89,

ние передается повелительному-оператору-1, выполнение продолжается согласно правилам для каждого оператора, указанного в повелительном-операторе-1. Если выполняется оператор ветвления процедур или условный оператор, вызывающий явную передачу управления, то управление передается согласно правилам для этих операторов; в противном случае по завершении выполнения повелительного-оператора-1 управление передается на конец оператора STRING (СОБРАТЬ).

(10) Если во время выполнения оператора STRING (СОБРАТЬ) с фразой NOT ON OVERFLOW (БЕЗ ПЕРЕПОЛНЕНИЯ) условия, описанные в общем правиле (9), не встречаются, после пересылки данных согласно другим общим правилам фраза ON OVERFLOW (ПРИ ПЕРЕПОЛНЕНИИ), если она указана, игнорируется и управление передается на конец оператора STRING (СОБРАТЬ), или, если указана фраза NOT ON OVERFLOW (БЕЗ ПЕРЕПОЛНЕНИЯ), повелительному-оператору-2. Если управление передается повелительному-оператору-2, выполнение продолжается согласно правилам для операторов, указанных в повелительном-операторе-2. Если выполняется ветвление процедуры или условный оператор, ведущий к явной передаче управления, управление передается в соответствии с правилами для этого оператора; в противном случае после завершения выполнения повелительного-оператора-2 управление передается на конец оператора STRING (СОБРАТЬ).

(11) Фраза END-STRING (КОНЕЦ-СОБРАТЬ) ограничивает область действия оператора STRING (СОБРАТЬ) (см. ч. 4, п. 6.4.3).

(12) Если идентификатор-1 и идентификатор-2 занимают ту же область памяти, что и идентификатор-3 и идентификатор-4, или идентификатор-3 и идентификатор-4 занимают одну и ту же область памяти, результат выполнения оператора STRING (СОБРАТЬ) не определен, даже если они определены одной и той же статьей описания данных (см. п. 6.4.5 настоящей части).

6.26. Оператор SUBTRACT (ОТНЯТЬ)

6.26.1. Назначение

Оператор SUBTRACT (ОТНЯТЬ) используется для того, чтобы вычесть одно числовое данное или сумму двух или более числовых данных из одного или более данных и установить значения одного или более данных равными результату.

6.26.2. Общий формат

Формат 1

SUBTRACT { идентификатор-1 / литерал-1 } ... FROM {идентификатор-2 [ROUNDED]}...

GOST EXPERT ГОСТ 22558-89,

[ON SIZE ERROR повелительный-оператор-1]
[NOT ON SIZE ERROR повелительный-оператор-2]
[END-SUBTRACT]

ОТНЯТЬ { литерал-1 / идентификатор-1 } ...
ОТ {идентификатор-2 [ОКРУГЛЯЯ]} ...
[ПРИ ПЕРЕПОЛНЕНИИ повелительный-оператор-1]
[БЕЗ ПЕРЕПОЛНЕНИЯ повелительный-оператор-2]
[КОНЕЦ-ОТНЯТЬ]

Формат 2

SUBTRACT { идентификатор-1 / литерал-1 } ... FROM { идентификатор-2 / литерал-2 }
GIVING {идентификатор-3 [ROUNDED]} ...
[ON SIZE ERROR повелительный-оператор-1]
[NOT ON SIZE ERROR повелительный-оператор-2]
[END-SUBTRACT]

ОТНЯТЬ { литерал-1 / идентификатор-1 } ... ОТ { литерал-2 / идентификатор-2 }
ПОЛУЧАЯ {идентификатор-3 [ОКРУГЛЯЯ]} ...
[ПРИ ПЕРЕПОЛНЕНИИ повелительный-оператор-1]
[БЕЗ ПЕРЕПОЛНЕНИЯ повелительный-оператор-2]
[КОНЕЦ-ОТНЯТЬ]

Формат 3

SUBTRACT { CORRESPONDING / CORR } идентификатор-1
FROM идентификатор-2 [ROUNDED]
[ON SIZE ERROR повелительный-оператор-1]
[NOT ON SIZE ERROR повелительный-оператор-2]
[END-SUBTRACT]

GOST EXPERT ГОСТ 22558-89,

ОТНЯТЬ {СООТВЕТСТВЕННО / СООТВ} идентификатор-1
ОТ идентификатор-2 [ОКРУГЛЯЯ]
[ПРИ ПЕРЕПОЛНЕНИИ повелительный-оператор-1]
[БЕЗ ПЕРЕПОЛНЕНИЯ повелительный-оператор-2]
[КОНЕЦ-ОТНЯТЬ]

6.26.3. Синтаксические правила

(1) Каждый идентификатор должен представлять числовое элементарное данное, исключая формат 2, в котором идентификатор, указанный справа от слова GIVING (ПОЛУЧАЯ), должен относиться к числовому или к числовому редактируемому данному,

и формат 3, в котором каждый идентификатор должен относиться к групповому данному.

(2) Каждый литерал должен быть числовым литералом.

(3) Композиция операндов не должна содержать более восемнадцати цифр (см. п. 6.4.4 настоящей части). При этом в формате 1 композиция операндов определяется использованием всех операндов в заданном операторе; в формате 2 композиция операндов определяется использованием всех операндов в заданном операторе, исключая идентификаторы, которые следуют за словом GIVING (ПОЛУЧАЯ); в формате 3 композиция операндов определяется отдельно для каждой пары соответствующих данных.

(4) CORR (СООТВ) есть сокращение слова CORRESPONDING (СООТВЕТСТВЕННО).

6.26.4. Общие правила

(1) В формате 1 значения операндов, предшествующие слову FROM (ОТ), складываются вместе, эта сумма сохраняется в промежуточном данном. Значение этого промежуточного данного вычитается из значения данного, представленного идентификатором-2, причем результат запоминается как новое значение данного, представленного идентификатором-2; затем этот процесс повторяется для всех последующих вхождений идентификатора-2 в порядке слева направо.

(2) В формате 2 все литералы и значения данных, представленные идентификаторами, предшествующими слову FROM (ОТ), складываются вместе и эта сумма вычитается из литерала-2 или значения данного, представленного идентификатором-2, после чего полученная разность запоминается как новое значение каждого данного, представленного идентификатором-3.

GOST EXPERT ГОСТ 22558-89,

(3) Если используется формат 3, данные из группы, представленной идентификатором-1, вычитаются от соответствующих данных из группы, представленной идентификатором-2, и запоминаются в последних.

(4) Отведение достаточного поля для выполнения вычислений без потери значащих цифр обеспечивается компилятором.

(5) Дополнительные правила и пояснения, относящиеся к этому оператору, приводятся в соответствующих параграфах (см. ч. 4, п. 6.4.3; ч. 6, пп. 6.4.1—6.4.6).

6.27. Оператор UNSTRING (РАЗОБРАТЬ)

6.27.1. Назначение

Оператор UNSTRING (РАЗОБРАТЬ) позволяет расчленить данное, находящееся в пересылаемом поле, и поместить отдельные его части в несколько принимающих полей.

6.27.2. Общий формат

UNSTRING идентификатор-1

[DELIMITED BY [ALL] {идентификатор-2 / литерал-1} [OR [ALL] {идентификатор-3 / литерал-2}] ...]

INTO {идентификатор-4 [DELIMITER IN идентификатор-5] [COUNT IN идентификатор-6]} ...

[WITH POINTER идентификатор-7]

[TALLYING IN идентификатор-8]

[ON OVERFLOW повелительный-оператор-1]

[NOT ON OVERFLOW повелительный-оператор-2]

[END-UNSTRING]

РАЗОБРАТЬ идентификатор-1 [ОГРАНИЧИВАЯСЬ [ВСЕМИ]

{идентификатор-2 / литерал-1} [[ИЛИ [ВСЕМИ]

{идентификатор-3 / литерал-2}] ...]

В {идентификатор-4 [ОГРАНИЧИТЕЛЬ В идентификатор-5]

[СЧЕТ В идентификатор-6]} ...

GOST EXPERT ГОСТ 22558-89,

[УКАЗАТЕЛЬ идентификатор-7]

[СЧИТАЯ В идентификатор-8]

[ПРИ ПЕРЕПОЛНЕНИИ повелительный-оператор-1]

[БЕЗ ПЕРЕПОЛНЕНИЯ повелительный-оператор-2]

[КОНЕЦ-РАЗОБРАТЬ]

6.27.3. Синтаксические правила

(1) Литерал-1 и литерал-2 должны быть нечисловыми литералами. Литералы могут быть стандартными константами, кроме начинающихся со слова ALL (ВСЕ).

(2) Идентификатор-1, идентификатор-2, идентификатор-3 и идентификатор-5 должны быть определены явно или неявно как буквенно-цифровые данные.

(3) Идентификатор-4 может быть задан как буквенное, буквенно-цифровое или числовое данное (причем в строке литер шаблона не должен использоваться символ P(M)), описанное явно или неявно с использованием DISPLAY (ДЛЯ ВЫДАЧИ).

(4) Идентификатор-6 и идентификатор-8 должны быть описаны как элементарные числовые целые, при этом символ P (M) не может использоваться в строке литер шаблона.

(5) Идентификатор-7 должен быть описан как элементарное числовое целое, его размер должен быть достаточен, чтобы содержать значение, равное размеру данного, представленного идентификатором-1, увеличенное на единицу. Символ P (M) не может использоваться в строке литер шаблона идентификатора-7.

(6) Фразы DELIMITER (ОГРАНИЧИТЕЛЬ) и COUNT (СЧЕТ) могут быть заданы только тогда, когда задана фраза DELIMITED (ОГРАНИЧИВАЯСЬ).

(7) Идентификатор-1 не должен быть модифицированной ссылкой.

6.27.4. Общие правила

(1) Все сказанное об идентификаторе-2 и литерале-1 относится в равной мере к идентификатору-3 и литералу-2 соответственно.

(2) Идентификатор-1 представляет собой пересылаемое поле;

(3) Идентификатор-4 представляет собой поле принимающего данного, идентификатор-5 представляет принимающее поле для ограничителя.

(4) Литерал-1 или данное, представленное идентификатором-2, задает границу пересылаемого данного, называемую ограничителем.

GOST EXPERT ГОСТ 22558-89,

(5) Данное, представленное идентификатором-6, представляет количество литер в пересылаемом данном, выделенных с помощью ограничителя для перемещения в принимающее данное, представленное идентификатором-4. В это количество не входят литеры ограничителя.

(6) Значение данного, представленного идентификатором-7, указывает относительную позицию литеры в поле, определенном идентификатором-1.

(7) К моменту завершения оператора UNSTRING (РАЗОБРАТЬ) значение данного, представленного идентификатором-8, указывает число принимающих данных, участвовавших в операции.

(8) Если стандартная константа используется как ограничитель, она представляет собой нечисловой литерал размером в одну литеру.

Если задана фраза ALL (ВСЕМИ), одно или несколько последовательных вхождений литерала-1 или значения данного, представленного идентификатором-2, обрабатываются так же, как одно вхождение, которое помещается в принимающее поле согласно общему правилу 13 г.

(9) Когда при просмотре данного, представленного идентификатором-1, обнаруживаются два последовательных вхождения ограничителя, текущее принимающее поле заполняется пробелами, если оно описано как буквенное или буквенно-цифровое, или нулями, если оно описано как числовое.

(10) Литерал-1 или значение данного, представленного идентификатором-2, может содержать любые литеры из множества литер машины.

(11) Каждый литерал-1 или данное, представленное идентификатором-2, представляет один ограничитель. Когда ограничитель содержит две или более литеры, эти литеры должны находиться в указанном ограничителем порядке в последовательных позициях пересылаемого данного, чтобы быть распознанными в качестве ограничителя.

(12) Если во фразе DELIMITED BY (ОГРАНИЧИВАЯСЬ) задано два или более ограничителя, между ними существует условие OR (ИЛИ). Каждый ограничитель поочередно сравнивается с рассматриваемыми позициями пересылаемого поля в указанной последовательности. При обнаружении совпадения рассматриваемые литеры считаются единым ограничителем. Ни одна литера в пересылаемом поле не может рассматриваться принадлежащей нескольким ограничителям.

(13) К началу работы оператора UNSTRING (РАЗОБРАТЬ) данное, определяемое идентификатором-4, является текущим принимающим полем, в которое пересылаются литеры данного,

GOST EXPERT ГОСТ 22558-89,

представленного идентификатором-1. Позиции последнего просматриваются согласно следующим правилам:

а) если задана фраза POINTER (УКАЗАТЕЛЬ), строка литер, представленная идентификатором-1, просматривается, начиная с относительной позиции литеры, указанной значением данного, представленного идентификатором-7. Если фраза POINTER (УКАЗАТЕЛЬ) не задана, проверка строки литер начинается с самой левой ее позиции;

б) если задана фраза DELIMITED BY (ОГРАНИЧИВАЯСЬ), просмотр прекращается при обнаружении вхождения ограничителя, заданного литералом-1 или значением данного, представленного идентификатором-2 (см. общее правило 11). Если фраза DELIMITED BY (ОГРАНИЧИВАЯСЬ) не задана, число просматриваемых литер равно размеру текущего принимающего данного (однако, если знак принимающего данного занимает отдельную позицию литеры, число просматриваемых литер полагается на единицу меньшим размера текущего принимающего поля данного).

Если вхождение ограничителя не обнаружено, просмотр оканчивается на последней позиции литеры данного, представленного идентификатором-1;

в) просмотренные литеры (за исключением литер, совпадающих с ограничителем, если они имеются) рассматриваются как элементарные буквенно-цифровые данные и пересылаются в текущее принимающее поле согласно правилам оператора MOVE (ПОМЕСТИТЬ) (см. п. 6.19 настоящей части);

г) если задана фраза DELIMITER (ОГРАНИЧИТЕЛЬ), обнаруженные при просмотре литеры ограничителя рассматриваются как элементарное буквенно-цифровое данное и помещаются согласно правилам оператора MOVE (ПОМЕСТИТЬ) в данное, представленное идентификатором-5.

Если вхождение ограничителя в данное, представленное идентификатором-1, не обнаружено, то данное, представленное идентификатором-5, заполняется пробелами;

д) если задана фраза COUNT IN (СЧЕТ), то количество просмотренных литер (исключая литеры ограничителя, если они имеются) помещается согласно правилам для перемещения элементарного данного в данное, представленное идентификатором-6;

е) если задана фраза DELIMITED BY (ОГРАНИЧИВАЯСЬ), то после заполнения очередного принимающего данного просмотр продолжается с первой литеры справа от ограничителя. Если фраза DELIMITED BY (ОГРАНИЧИВАЯСЬ) не задана, просмотр продолжается от первой литеры, находящейся справа от последней пересланной литеры;

GOST EXPERT ГОСТ 22558-89,

ж) после завершения пересылки в принимающее данное, представленное идентификатором-4, текущим принимающим полем становится данное, представленное следующим вхождением идентификатора-4, и все описанные выше действия повторяются до тех пор, пока не будут исчерпаны все литеры в данном, определяемом идентификатором-1, или не будут заполнены все принимающие поля.

(14) Начальная установка значения данного, связанного с фразой POINTER (УКАЗАТЕЛЬ) или TALLYING (СЧИТАЯ), осуществляется пользователем.

(15) Значение данного, представленного идентификатором-7, увеличивается на единицу при просмотре каждой литеры в данном, представленном идентификатором-1. После завершения выполнения оператора UNSTRING (РАЗОБРАТЬ) с фразой POINTER (УКАЗАТЕЛЬ) значение данного, представленного идентификатором-7, становится равным начальному значению плюс число просмотренных литер данного, представленного идентификатором-1.

(16) После завершения выполнения оператора UNSTRING (РАЗОБРАТЬ) с фразой TALLYING (СЧИТАЯ) значение данного, представленного идентификатором-8, становится равным его начальному значению плюс число принимающих полей, участвовавших в выполнении оператора.

(17) Условие переполнения возникает при следующих ситуациях:

а) если к моменту начала работы оператора UNSTRING (РАЗОБРАТЬ) значение данного, представленного идентификатором-7, меньше 1 или больше размера данного, представленного идентификатором-1;

б) если в некоторый момент выполнения оператора UNSTRING (РАЗОБРАТЬ) данное, представленное идентификатором-1, содержит непросмотренные позиции, а все принимающие поля уже использованы.

(18) При обнаружении условия переполнения выполнение оператора UNSTRING (РАЗОБРАТЬ) прекращается; если указана фраза NOT ON OVERFLOW (БЕЗ ПЕРЕПОЛНЕНИЯ), то она игнорируется, а управление передается на конец оператора UNSTRING (РАЗОБРАТЬ); если же была указана фраза ON OVERFLOW (ПРИ ПЕРЕПОЛНЕНИИ), то управление передается повелительному-оператору-1. В этом случае выполнение продолжается согласно правилам для операторов, указанных в повелительном-операторе-1. Если выполняется оператор ветвления процедур или условный оператор, вызывающий явную передачу управления, управление передается согласно правилам для соответствующих операторов; в противном случае пос-

GOST EXPERT ГОСТ 22558-89,

ле завершения выполнения повелительного-оператора-1 управление передается на конец оператора UNSTRING (РАЗОБРАТЬ).

(19) Фраза END-UNSTRING (КОНЕЦ-РАЗОБРАТЬ) ограничивает область действия оператора UNSTRING (РАЗОБРАТЬ) (см. ч. 4, п. 6.4.3).

(20) Если во время выполнения оператора UNSTRING (РАЗОБРАТЬ) описанные в пункте (17) условия не возникли, после завершения пересылки данных согласно другим общим правилам фраза OVERFLOW (ПРИ ПЕРЕПОЛНЕНИИ), если она задана, игнорируется, а управление передается на конец оператора UNSTRING (РАЗОБРАТЬ) или, если указана фраза NOT OVERFLOW (БЕЗ ПЕРЕПОЛНЕНИЯ), повелительному-оператору-2. Если управление передано повелительному-оператору-2, выполнение продолжается согласно правилам для операторов, указанных в повелительном-операторе-2. Если выполняется оператор ветвления процедур или условный оператор, подразумевающий явную передачу управления, управление передается согласно правилам для соответствующих операторов; в противном случае после завершения выполнения повелительного-оператора-2 управление передается на конец оператора UNSTRING (РАЗОБРАТЬ).

(21) Если идентификатор-1, идентификатор-2 и идентификатор-3 занимают ту же область памяти, что и идентификатор-5, идентификатор-6, идентификатор-7 или идентификатор-8, или если идентификатор-4, идентификатор-5 и идентификатор-6 занимают ту же область памяти, что и идентификатор-7 или идентификатор-8, или если идентификатор-7 и идентификатор-8 занимают одну и ту же область памяти, результат выполнения оператора UNSTRING (РАЗОБРАТЬ) не определен, даже если эти идентификаторы описаны одной и той же статьей описания данных (см. п. 6.4.5 настоящей части).

7. ОТЛАДКА В ЯДРЕ

7.1. Общее описание

Средства отладки в ядре обеспечивают пользователя отладочными строками и переключателем времени компиляции для отладочных строк.

7.2. Переключатель времени компиляции

Фраза WITH DEBUGGING MODE (В РЕЖИМЕ ОТЛАДКИ) является частью параграфа SOURCE-COMPUTER (ИСХОДНАЯ-МАШИНА) (см. п. 4.3 настоящей части). Она служит переключателем времени компиляции для отладочных строк, содержащихся в отдельно компилируемой программе.

GOST EXPERT ГОСТ 22558-89,

Если в отдельно компилируемой программе указана фраза WITH DEBUGGING MODE (В РЕЖИМЕ ОТЛАДКИ), все отладочные строки компилируются, как определено в данном представлении ядра. Если же фраза WITH DEBUGGING MODE (В РЕЖИМЕ ОТЛАДКИ) не указана, все отладочные строки компилируются так, как если бы они представляли строки комментариев.

Наличие или отсутствие фразы WITH DEBUGGING MODE (В РЕЖИМЕ ОТЛАДКИ) логически определяется после обработки всех операторов COPY (КОПИРОВАТЬ) и REPLACE (ЗАМЕНИТЬ).

7.3. Отладочные строки

Отладочной строкой называется строка с литерой D(T) в области индикатора строки. Любая отладочная строка, состоящая только из пробелов от границы A до границы R, считается пустой.

Содержимое отладочной строки должно быть таким, при котором программа является синтаксически правильной независимо от того, содержит ли она отладочные строки или они рассматриваются как комментарии.

После обработки всех операторов COPY (КОПИРОВАТЬ) и REPLACE (ЗАМЕНИТЬ) отладочная строка рассматривается как комментарий, если в параграфе SOURCE-COMPUTER (ИСХОДНАЯ-МАШИНА) не указана фраза WITH DEBUGGING MODE (В РЕЖИМЕ ОТЛАДКИ).

Допускаются несколько последовательно расположенных отладочных строк.

В отдельно компилируемой программе отладочные строки могут располагаться только после параграфа OBJECT-COMPUTER (РАБОЧАЯ-МАШИНА).

Часть 7. МОДУЛЬ ПОСЛЕДОВАТЕЛЬНОГО ВВОДА-ВЫВОДА

1. ВВЕДЕНИЕ В МОДУЛЬ ПОСЛЕДОВАТЕЛЬНОГО ВВОДА-ВЫВОДА

1.1. Назначение

Модуль последовательного ввода-вывода обеспечивает возможность доступа к записям файла в установленной последовательности. Последовательность устанавливается в результате занесения записей в файл.

1.2. Характеристика уровней

Уровень 1 последовательного ввода-вывода обеспечивает неполные возможности для статей управления файлом, статей описания файла и статей параграфа I-O-CONTROL (УПРАВЛЕНИЕ-ВВОДОМ-ВЫВОДОМ). В разделе процедур уровень 1 последовательного ввода-вывода обеспечивает ограниченные возможности операторов CLOSE (ЗАКРЫТЬ), USE (ИСПОЛЬЗОВАТЬ),

GOST EXPERT ГОСТ 22558-89,

OPEN (ОТКРЫТЬ), WRITE (ПИСАТЬ) и READ (ЧИТАТЬ). Полные возможности обеспечиваются для оператора REWRITE (ОБНОВИТЬ).

Уровень 2 последовательного ввода-вывода обеспечивает полные возможности для статей управления файлом, статей описания файла и статей параграфа I-O-CONTROL (УПРАВЛЕНИЕ-ВВОДОМ-ВЫВОДОМ). В разделе процедур уровень 2 последовательного ввода-вывода обеспечивает полные возможности операторов CLOSE (ЗАКРЫТЬ), OPEN (ОТКРЫТЬ), READ (ЧИТАТЬ), REWRITE (ОБНОВИТЬ), WRITE (ПИСАТЬ) и USE (ИСПОЛЬЗОВАТЬ).

1.3. Понятия языка

1.3.1. Организация

Последовательный файл организован так, что каждая запись, кроме последней, имеет единственную запись-преемника и каждая запись, кроме первой, имеет единственную запись-предшественника. Это отношение следования устанавливается оператором WRITE (ПИСАТЬ) при создании файла. Однажды установленное отношение следования не меняется. Исключением является случай, когда записи добавляются в конец файла.

Последовательно организованный файл массовой памяти имеет ту же логическую структуру, что и файл в последовательной запоминающей среде; однако последовательный файл массовой памяти может обновляться на месте. При использовании этого метода новые записи не могут быть добавлены в файл, а каждая заменяющая запись должна иметь тот же размер, что и исходная запись.

1.3.2. Метод доступа

При последовательном доступе порядок, в котором осуществляется доступ к записям, соответствует порядку, в котором записи первоначально записывались.

1.3.3. Указатель текущего тома

Указатель текущего тома — это логическое понятие, используемое в этом документе для облегчения точного задания текущего тома последовательного файла.

1.3.4. Указатель позиции файла

Указатель позиции в файле — это логическое понятие, используемое в этом документе для облегчения точной спецификации следующей записи, к которой должен осуществляться доступ при выполнении заданных операций ввода-вывода. На установку указателя позиции файла влияют только операторы CLOSE (ЗАКРЫТЬ), OPEN (ОТКРЫТЬ) и READ (ЧИТАТЬ). Понятие указателя позиции файла не имеет смысла для файла, открытого как выходной или как дополняемый.

GOST EXPERT ГОСТ 22558-89,

1.3.5. Состояние ввода-вывода

Состояние ввода-вывода — это логическое понятие, характеризующееся двухсимвольным значением, которое устанавливается для указания состояния операции ввода-вывода во время выполнения операторов CLOSE (ЗАКРЫТЬ), OPEN (ОТКРЫТЬ), READ (ЧИТАТЬ), REWRITE (ОБНОВИТЬ) или WRITE (ПИСАТЬ) перед выполнением любого повелительного оператора, связанного с этим оператором ввода-вывода, и перед выполнением любой применимой процедуры USE AFTER STANDARD EXCEPTION (ИСПОЛЬЗОВАТЬ ПОСЛЕ СТАНДАРТНОЙ ПРОЦЕДУРЫ ОШИБКИ). Значение состояния ввода-вывода доступно Кобол-программе посредством фразы FILE STATUS (СОСТОЯНИЕ ФАЙЛА) в статье управления файлом.

Состояние ввода-вывода определяет также, будет ли выполняться процедура USE AFTER STANDARD EXCEPTION (ИСПОЛЬЗОВАТЬ ПОСЛЕ СТАНДАРТНОЙ ПРОЦЕДУРЫ ОШИБКИ). Если возникает любое условие, отличное от тех, которые определены ниже как «успешное завершение», может выполняться указанная процедура по правилам, заданным для оператора USE (ИСПОЛЬЗОВАТЬ). Если возникает одно из условий «успешное завершение», никакая процедура такого типа не будет выполняться (п. 4.6 настоящей части).

Некоторые классы значений состояния ввода-вывода задают критические условия ошибки. Условия, которые определены в реализации как критические, должны начинаться с цифр 3, 4 и 9. Если значение состояния ввода-вывода для операции ввода-вывода задает такое условие ошибки, реализацией определяются действия, которые предпринимаются после выполнения любой применимой по оператору USE AFTER STANDARD EXCEPTION (ИСПОЛЬЗОВАТЬ ПОСЛЕ СТАНДАРТНОЙ ПРОЦЕДУРЫ ОШИБКИ) процедуры или, если ни одна такая процедура не применима, после завершения стандартной системной обработки ошибок ввода-вывода. Состояние ввода-вывода задает одно из следующих условий, возникающих после завершения операции ввода-вывода.

(1) Успешное завершение. Оператор ввода-вывода выполнился успешно.

(2) В конце. Оператор последовательного чтения был выполнен неуспешно из-за того, что возникло условие конца файла.

(3) Постоянная ошибка. Оператор ввода-вывода выполнен неуспешно в результате ошибки, которая исключает дальнейшую обработку файла. Выполняются все заданные процедуры обработки ошибочных ситуаций. Условие постоянной ошибки остается действующим на все последующие операции ввода-вывода файла до тех пор, пока не будут вызваны определенные реализацией средства для устранения условия постоянной ошибки.

GOST EXPERT ГОСТ 22558-89,

(4) Логическая ошибка. Оператор ввода-вывода выполнен неуспешно из-за недопустимой последовательности операций ввода-вывода, выполняемых над файлом, или в результате нарушения ограничений, заданных пользователем.

(5) Ошибка, определяемая реализацией. Оператор ввода-вывода выполнен неуспешно в результате возникновения условия, определенного реализацией.

Ниже приводится список значений, помещаемых в состояние ввода-вывода для перечисленных выше условий, возникающих в результате выполнения операций ввода-вывода для последовательного файла. Если применимо более одного значения, значение, которое помещается в состояние ввода-вывода, определяется реализацией.

(1) Успешное завершение

а) Состояние ввода-вывода=00. Оператор выполнился успешно и нет никакой другой доступной информации об операции ввода-вывода.

б) Состояние ввода-вывода=04. Оператор READ (ЧИТАТЬ) выполнился успешно, но длина обрабатываемой записи не соответствует фиксированным свойствам этого файла.

в) Состояние ввода-вывода=05. Оператор OPEN (ОТКРЫТЬ) успешно выполнился, но указанный в нем необязательный файл во время выполнения оператора OPEN (ОТКРЫТЬ) отсутствует. Если режим открытия I-O (ВХОДНОЙ-ВЫХОДНОЙ) или EXTEND (ДОПОЛНЯЕМЫЙ), файл будет создаваться.

г) Состояние ввода-вывода=07. Оператор ввода-вывода успешно выполнился. Однако для оператора CLOSE (ЗАКРЫТЬ) с фразами NO REWIND (БЕЗ ПЕРЕМОТКИ), REEL/UNIT (КАТУШКУ/ТОМ) или FOR REMOVAL (С УДАЛЕНИЕМ), или для оператора OPEN (ОТКРЫТЬ) с фразой NO REWIND (БЕЗ ПЕРЕМОТКИ) заданный файл расположен в среде, к которой неприменимо понятие катушки (тома).

(2) Условие в конце с неуспешным завершением

а) Состояние ввода-вывода=10. Встретился последовательный оператор READ (ЧИТАТЬ), а в файле не существует следующей логической записи из-за того, что:

1) встретился конец файла;

2) оператор READ (ЧИТАТЬ) применяется первый раз для отсутствующего необязательного входного файла.

(3) Условие постоянной ошибки с неуспешным завершением

а) Состояние ввода-вывода=30. Возникла постоянная ошибка и нет другой доступной информации об операции ввода-вывода.

GOST EXPERT ГОСТ 22558-89,

б) Состояние ввода-вывода=34. Возникла постоянная ошибка из-за нарушения границ; была совершена попытка записать запись за внешне определенные границы файла. Способ, которым задаются эти границы, определяется реализацией.

в) Состояние ввода-вывода=35. Постоянная ошибка возникла из-за того, что оператор OPEN (ОТКРЫТЬ) с фразой INPUT (ВХОДНОЙ), I-O (ВХОДНОЙ-ВЫХОДНОЙ) или EXTEND (ДОПОЛНЯЕМЫЙ) выдан для отсутствующего обязательного файла.

г) Состояние ввода-вывода=37. Постоянная ошибка возникла из-за того, что оператор OPEN (ОТКРЫТЬ) выдан для файла, который не поддерживает режим, заданный в операторе OPEN (ОТКРЫТЬ). Возможны следующие нарушения:

1) задана фраза EXTEND (ДОПОЛНЯЕМЫЙ) или OUTPUT (ВЫХОДНОЙ), а файл не допускает операции записи;

2) задана фраза I-O (ВХОДНОЙ-ВЫХОДНОЙ), а файл не допускает операции ввода и вывода, которые разрешены для последовательного файла, открываемого в режиме I-O (ВХОДНОЙ-ВЫХОДНОЙ);

3) задана фраза INPUT (ВХОДНОЙ), а файл не допускает операции чтения.

д) Состояние ввода-вывода=38. Постоянная ошибка возникла из-за того, что выдан оператор OPEN (ОТКРЫТЬ) для файла, ранее закрытого с замком.

е) Состояние ввода-вывода=39. Оператор OPEN (ОТКРЫТЬ) завершился неуспешно из-за обнаруженного для этого файла несоответствия фиксированных свойств файла и свойств, заданных в программе.

(4) Условие логической ошибки с неуспешным завершением

а) Состояние ввода-вывода=41. Оператор OPEN (ОТКРЫТЬ) выдан для открытого файла.

б) Состояние ввода-вывода=42. Оператор CLOSE (ЗАКРЫТЬ) выдан для неоткрытого файла.

в) Состояние ввода-вывода=43. В случае файла массовой памяти в режиме последовательного доступа последним оператором ввода-вывода, выполненным для соответствующего файла перед выполнением оператора REWRITE (ОБНОВИТЬ), не был успешно выполненный оператор READ (ЧИТАТЬ).

г) Состояние ввода-вывода=44. Нарушение границ возникает по следующим причинам:

GOST EXPERT ГОСТ 22558-89,

1) сделана попытка записать или обновить запись, длиннее максимально допустимой или короче минимально допустимой в соответствии с фразой RECORD IS VARYING (В ЗАПИСИ ПЕРЕМЕННОЕ ЧИСЛО) для данного имени файла;

2) сделана попытка обновить запись последовательного файла, а размер записи отличен от размера заменяемой записи.

д) Состояние ввода-вывода=46. Выдан оператор последовательного чтения для файла, открытого в режиме ввода или ввода-вывода, и не была установлена следующая запись по одной из следующих причин:

1) предыдущий оператор READ (ЧИТАТЬ) закончился неуспешно, но не вызвал условие «в конце»;

2) предыдущий оператор READ (ЧИТАТЬ) вызвал условие «в конце».

е) Состояние ввода-вывода=47. Был выдан оператор READ (ЧИТАТЬ) для файла, не открытого в режиме ввода или ввода-вывода.

ж) Состояние ввода-вывода=48. Был выдан оператор WRITE (ПИСАТЬ) для файла, не открытого в режиме вывода или дополнения.

з) Состояние ввода-вывода=49. Был выдан оператор REWRITE (ОБНОВИТЬ) для файла, не открытого в режиме ввода-вывода.

(5) Условие неуспешного завершения, определяемое реализацией

а) Состояние ввода-вывода=9x. Существуют определяемые реализацией условия. Эти условия не должны дублировать никакие условия, определенные для значений от 00 до 49 состояния ввода-вывода. Значение x определяется реализацией.

1.3.6. Условие «в конце»

Условие «в конце» может возникнуть в результате выполнения оператора READ (ЧИТАТЬ) (п. 4.4 настоящей части).

1.3.7. Условие противоречия свойств файла

Условие противоречия свойств файла может возникнуть в результате выполнения операторов OPEN (ОТКРЫТЬ), REWRITE (ОБНОВИТЬ) и WRITE (ПИСАТЬ). При возникновении условия противоречия свойств файла выполнение оператора ввода-вывода, который обнаружил это условие, считается неуспешным и файл не подвергается воздействию (пп. 4.3, 4.5, 4.7 настоящей части).

При обнаружении условия противоречия свойств файла выполняются следующие действия в указанном ниже порядке:

(1) в состояние ввода-вывода данного файла помещается значение, указывающее на условие противоречия свойств файла (см. п. 1.3.5 настоящей части);

GOST EXPERT ГОСТ 22558-89,

(2) если для данного имени файла задан оператор USE AFTER EXCEPTION/ERROR (ИСПОЛЬЗОВАТЬ ПОСЛЕ ПРОЦЕДУРЫ ОШИБКИ), выполняется указанная в нем процедура.

1.3.8. Специальный регистр LINAGE-COUNTER (СЧЕТЧИК-ВЕРСТКИ)

Зарезервированное слово LINAGE-COUNTER (СЧЕТЧИК-ВЕРСТКИ) является именем для счетчика строк, порождаемого, если в статье описания файла задана фраза LINAGE (ВЕРСТКА) (п. 3.7 настоящей части). Этот счетчик неявно описан как целое без знака, размер которого равен размеру целого-1 или данного, заданного именем-данного-1 во фразе LINAGE (ВЕРСТКА). LINAGE-COUNTER (СЧЕТЧИК-ВЕРСТКИ) может использоваться только в операторах раздела процедур, однако его значение может изменить только система управления вводом-выводом.

2. РАЗДЕЛ ОБОРУДОВАНИЯ В МОДУЛЕ ПОСЛЕДОВАТЕЛЬНОГО ВВОДА-ВЫВОДА

2.1. Секция ввода-вывода

Секция ввода-вывода располагается в разделе оборудования исходной программы. Секция ввода-вывода содержит информацию, необходимую для управления передачей и обработкой данных между внешней памятью и объектной программой. Секция ввода-вывода в разделе оборудования исходной Кобол-программы необязательна.

Общий формат секции ввода-вывода показан ниже.

INPUT-OUTPUT SECTION.

FILE-CONTROL. {статья-управления-файлом} . . .

[I-O-CONTROL. [статья-управления-вводом-выводом]]

СЕКЦИЯ ВВОДА-ВЫВОДА.

УПРАВЛЕНИЕ-ФАЙЛАМИ. {статья-управления-файлом} . . .

[УПРАВЛЕНИЕ-ВВОДОМ-ВЫВОДОМ. [статья-управления-вводом-выводом]]

2.2. Параграф FILE-CONTROL (УПРАВЛЕНИЕ-ФАЙЛАМИ)

2.2.1. Назначение

Параграф FILE-CONTROL (УПРАВЛЕНИЕ-ФАЙЛАМИ) позволяет задать относящуюся к файлу информацию.

2.2.2. Общий формат

FILE-CONTROL. {статья-управления-файлом} . . .

GOST EXPERT ГОСТ 22558-89,

УПРАВЛЕНИЕ-ФАЙЛАМИ. {статья-управления-файлом}...

2.3. Статья управления файлом

2.3.1. Назначение

Статья управления файлом объявляет существенные физические свойства последовательного файла.

2.3.2. Общий формат

SELECT [OPTIONAL] имя-файла-1

ASSIGN TO { имя-реализации-1 / литерал-1 } ...

[[RESERVE целое-1 [AREA / AREAS]]]

[[ORGANIZATION IS] SEQUENTIAL]

[[PADDING CHARACTER IS { имя-данного-1 / литерал-2 }]]
[[RECORD DELIMITER IS { STANDARD-1 / имя-реализации-2 }]]

[ACCESS MODE IS SEQUENTIAL]

[FILE STATUS IS имя-данного-2].

ДЛЯ [[НЕОБЯЗАТЕЛЬНОГО]] имя-файла-1

НАЗНАЧИТЬ { имя-реализации-1 / литерал-1 } ...

[РЕЗЕРВИРОВАТЬ целое-1 ОБЛАСТЕЙ]

[[ОРГАНИЗАЦИЯ] ПОСЛЕДОВАТЕЛЬНАЯ]

[[ЛИТЕРА ЗАПОЛНИТЕЛЬ { имя-данного-1 / литерал-2 }]]
[[ОГРАНИЧИТЕЛЬ ЗАПИСИ { СТАНДАРТ-А / имя-реализации-2 }]]

[ДОСТУП ПОСЛЕДОВАТЕЛЬНЫЙ]

[СОСТОЯНИЕ ФАЙЛА имя-данного-2].

GOST EXPERT ГОСТ 22558-89,

2.3.3. Синтаксические правила

(1) В статье управления файлом фраза SELECT (ДЛЯ) должна указываться первой. Фразы, которые следуют за фразой SELECT (ДЛЯ), могут появляться в любом порядке.

(2) Каждое имя-файла из раздела данных должно быть определено в параграфе FILE-CONTROL (УПРАВЛЕНИЕ-ФАЙЛАМИ) только один раз. Каждое имя-файла, указанное во фразе SELECT (ДЛЯ), должно иметь статью описания файла в разделе данных той же самой программы.

(3) Литерал-1 должен быть нечисловым литералом и не должен быть стандартной константой. Значение и правила для допустимого содержимого имени-реализации-1 и значения литерала-1 определяются реализацией.

2.3.4. Общие правила

(1) Если определитель файла, на который ссылается имя-файла-1, является внешним определителем файла (ч. 10, п. 4.5), все статьи управления файлом в единице исполнения, которые ссылаются на этот определитель файла, должны иметь:

а) одну и ту же спецификацию фразы OPTIONAL (НЕОБЯЗАТЕЛЬНОГО);

б) корректную спецификацию для имени-реализации-1 или литерала-1 во фразе ASSIGN (НАЗНАЧИТЬ). Правила корректности имени-реализации-1 и литерала-1 определяются реализацией;

в) корректную спецификацию для имени-реализации-2 во фразе RECORD DELIMITER (ОГРАНИЧИТЕЛЬ ЗАПИСИ). Правила корректности имени-реализации-2 определяются реализацией;

г) одно и то же значение целого-1 во фразе RESERVE (РЕЗЕРВИРОВАТЬ);

д) одну и ту же организацию;

е) один и тот же метод доступа;

ж) одну и ту же спецификацию фразы PADDING CHARACTER (ЛИТЕРА ЗАПОЛНИТЕЛЬ).

(2) Фраза OPTIONAL (НЕОБЯЗАТЕЛЬНОГО) применима только к файлам, открытым в режиме ввода, ввода-вывода или дополнения. Ее указание требуется для файлов, которые могут отсутствовать во время выполнения объектной программы.

(3) Фраза SELECT (ДЛЯ) задает связь между файлом, на который ссылается имя-файла-1, и средой памяти, на которую ссылается имя-реализации-1 или литерал-1.

GOST EXPERT ГОСТ 22558-89,

(4) Фразы ACCESS MODE (ДОСТУП), FILE STATUS (СОСТОЯНИЕ ФАЙЛА), ORGANIZATION IS SEQUENTIAL (ОРГАНИЗАЦИЯ ПОСЛЕДОВАТЕЛЬНАЯ), PADDING CHARACTER (ЛИТЕРА ЗАПОЛНИТЕЛЬ), RECORD DELIMITER (ОГРАНИЧИТЕЛЬ ЗАПИСИ) и RESERVE (РЕЗЕРВИРОВАТЬ) описываются на следующих страницах.

2.4. Фраза ACCESS MODE (ДОСТУП)

2.4.1. Назначение

Фраза ACCESS MODE (ДОСТУП) задает порядок, в котором осуществляется доступ к записям в файле.

2.4.2. Общий формат

ACCESS MODE IS SEQUENTIAL

ДОСТУП ПОСЛЕДОВАТЕЛЬНЫЙ

2.4.3. Общие правила

(1) Если фраза ACCESS MODE (ДОСТУП) не задана, предполагается последовательный доступ.

(2) Доступ к записям в файле осуществляется в последовательности, диктуемой организацией файла. Для последовательных файлов эта последовательность задается отношением предшественник—преемник, устанавливаемым выполнением операторов WRITE (ПИСАТЬ) при создании или дополнении файла.

(3) Если соответствующий определитель файла является внешним определителем файла, каждая статья управления файлом в единице исполнения, соответствующая этому определителю файла, должна задавать один и тот же метод доступа.

2.5. Фраза FILE STATUS (СОСТОЯНИЕ ФАЙЛА)

2.5.1. Назначение

Фраза FILE STATUS (СОСТОЯНИЕ ФАЙЛА) задает данное, которое содержит состояние операции ввода-вывода.

2.5.2. Общий формат

FILE STATUS IS имя-данного-1

СОСТОЯНИЕ ФАЙЛА имя-данного-1

2.5.3. Синтаксические правила

(1) Имя-данного-1 может уточняться.

(2) Имя-данного-1 должно быть определено в разделе данных как данное из двух литер буквенно-цифровой категории и не должно определяться в секции файлов, в секции отчетов или в секции коммуникаций.

2.5.4. Общие правила

(1) Если задана фраза FILE STATUS (СОСТОЯНИЕ ФАЙЛА), данное, на которое ссылается имя-данного-1, всегда обновляется, как только обновляется состояние ввода-вывода, чтобы содержать значение этого состояния ввода-вывода. Это значение ха-

GOST EXPERT ГОСТ 22558-89,

рактеризует состояние выполнения оператора (см. п. 1.3.5 настоящей части).

(2) Данное, на которое ссылается имя-данного-1 и которое обновляется при выполнении оператора ввода-вывода, определяется в статье управления файлом, связанной с именем файла, указанным в этом операторе.

2.6. Фраза ORGANIZATION IS SEQUENTIAL (ОРГАНИЗАЦИЯ ПОСЛЕДОВАТЕЛЬНАЯ)

2.6.1. Назначение

Фраза ORGANIZATION IS SEQUENTIAL (ОРГАНИЗАЦИЯ ПОСЛЕДОВАТЕЛЬНАЯ) указывает, что логической структурой файла является последовательная организация.

2.6.2. Общий формат

[ORGANIZATION IS] SEQUENTIAL

[ОРГАНИЗАЦИЯ] ПОСЛЕДОВАТЕЛЬНАЯ

2.6.3. Общие правила

(1) Фраза ORGANIZATION IS SEQUENTIAL (ОРГАНИЗАЦИЯ ПОСЛЕДОВАТЕЛЬНАЯ) указывает, что логической структурой файла является последовательная организация. Организация файла устанавливается во время его создания и впоследствии не может быть изменена.

(2) Последовательная организация является постоянной логической структурой файла, в которой запись идентифицируется отношением предшественник—преемник, которое устанавливается при помещении записей в файл.

(3) Если фраза ORGANIZATION (ОРГАНИЗАЦИЯ) не задана, предполагается последовательная организация.

2.7. Фраза PADDING CHARACTER (ЛИТЕРА ЗАПОЛНИТЕЛЬ)

2.7.1. Назначение

Фраза PADDING CHARACTER (ЛИТЕРА ЗАПОЛНИТЕЛЬ) определяет литеру, которая будет использоваться для заполнения блока в последовательных файлах.

2.7.2. Общий формат

PADDING CHARACTER IS { имя-данного-1 / литерал-1 }

ЛИТЕРА ЗАПОЛНИТЕЛЬ { имя-данного-1 / литерал-1 }

2.7.3. Синтаксические правила

(1) Литерал-1 должен быть односимвольным нечисловым литералом.

(2) Имя-данного-1 может уточняться.

GOST EXPERT ГОСТ 22558-89,

(3) Имя-данного-1 должно определяться в разделе данных как односимвольное данное буквенно-числовой категории и не должно определяться в секции коммуникаций, секции файлов или секции отчетов.

2.7.4. Общие правила

(1) Фраза PADDING CHARACTER (ЛИТЕРА ЗАПОЛНИТЕЛЬ) задает литеру, которая будет использоваться для заполнения блока в последовательных файлах. Во время выполнения операций ввода любая часть блока, находящаяся за последней логической записью и состоящая полностью из литер заполнителя, будет пропускаться. Во время выполнения операций ввода логическая запись, состоящая полностью из литер заполнителя, будет игнорироваться. При операциях вывода любая часть блока, находящаяся после последней логической записи, будет полностью заполняться литерами заполнителя.

(2) Если фраза PADDING CHARACTER (ЛИТЕРА ЗАПОЛНИТЕЛЬ) не применима для типа устройства, которое назначено файлу, создание или распознаваниє литер заполнителя не будет иметь места.

(3) Во время выполнения оператора OPEN (ОТКРЫТЬ) для файла, который будет создаваться, литерал-1 или значение данного, на которое ссылается имя-данного-1, используется как значение литеры заполнителя. Литера заполнитель является фиксированным свойством файла.

(4) Если для файла задана фраза CODE-SET (АЛФАВИТ), преобразование литеры заполнителя, заданной литералом-1 или содержимым имени-данного-1, устанавливается для файла при его открытии.

(5) Если фраза PADDING CHARACTER (ЛИТЕРА ЗАПОЛНИТЕЛЬ) не задана, значение, используемое для литеры заполнителя, определяется реализацией.

(6) Если соответствующий определитель файла является внешним определителем файла, все фразы PADDING CHARACTER (ЛИТЕРА ЗАПОЛНИТЕЛЬ) в единице исполнения, связанные с этим определителем файла, должны иметь одни и те же спецификации. Если задано имя-данного-1, оно должно ссылаться на внешнее данное.

2.8. Фраза RECORD DELIMITER (ОГРАНИЧИТЕЛЬ ЗАПИСИ)

2.8.1. Назначение

Фраза RECORD DELIMITER (ОГРАНИЧИТЕЛЬ ЗАПИСИ) указывает способ определения длины записей переменной длины во внешней среде.

GOST EXPERT ГОСТ 22558-89,

2.8.2. Общий формат

RECORD DELIMITER IS { STANDARD-1 / имя-реализации-1 }

ОГРАНИЧИТЕЛЬ ЗАПИСИ { СТАНДАРТ-А / имя-реализации-1 }

2.8.3. Синтаксические правила

(1) Фраза RECORD DELIMITER (ОГРАНИЧИТЕЛЬ ЗАПИСИ) может задаваться только для записей переменной длины.

(2) Если задана фраза STANDARD-1 (СТАНДАРТ-А), внешней средой должен быть файл на магнитной ленте.

2.8.4. Общие правила

(1) Фраза RECORD DELIMITER (ОГРАНИЧИТЕЛЬ ЗАПИСИ) используется для указания способа определения длины записей переменной длины во внешней среде. Любой используемый способ не будет отражаться на области записи или размере записи, используемом в программе.

(2) Если задана фраза STANDARD-1 (СТАНДАРТ-А), способ для определения длины записей переменной длины определяется в соответствии с Американским Национальным Стандартом X3.27—1978 «Метки магнитных лент и структура файлов для обмена информацией» и стандартом ИСО 1001 «Метки магнитных лент и структура файлов для обмена информацией».

(3) Если задано имя-реализации-1, для определения длины записей переменной длины используется метод, определяемый реализацией.

(4) Если фраза RECORD DELIMITER (ОГРАНИЧИТЕЛЬ ЗАПИСИ) не задана, способ, используемый для определения длины записей переменной длины, определяется реализацией.

(5) Во время успешного выполнения оператора OPEN (ОТКРЫТЬ) ограничителем записи является ограничитель, заданный во фразе RECORD DELIMITER (ОГРАНИЧИТЕЛЬ ЗАПИСИ) в статье управления файлом, связанной с именем-файла, указанным в операторе OPEN (ОТКРЫТЬ).

(6) Если соответствующий определитель файла является внешним определителем файла, все фразы RECORD DELIMITER (ОГРАНИЧИТЕЛЬ ЗАПИСИ) в единице исполнения, относящиеся к данному определителю файла, должны иметь одинаковые спецификации.

2.9. Фраза RESERVE (РЕЗЕРВИРОВАТЬ)

2.9.1. Назначение

Фраза RESERVE (РЕЗЕРВИРОВАТЬ) позволяет пользо-

GOST EXPERT ГОСТ 22558-89,

вателю указать количество распределяемых областей ввода-вывода.

2.9.2. Общий формат

RESERVE целое-1 [AREA / AREAS]

РЕЗЕРВИРОВАТЬ целое-1 ОБЛАСТЕЙ

2.9.3. Общие правила

(1) Фраза RESERVE (РЕЗЕРВИРОВАТЬ) позволяет пользователю задавать количество распределяемых областей ввода-вывода. Если задана фраза RESERVE (РЕЗЕРВИРОВАТЬ), количество распределяемых областей ввода-вывода равно значению целого-1. Если фраза RESERVE (РЕЗЕРВИРОВАТЬ) не задана, количество распределяемых областей ввода-вывода определяется реализацией.

2.10. Параграф I-O-CONTROL (УПРАВЛЕНИЕ-ВВОДОМ-ВЫВОДОМ)

2.10.1. Назначение

Параграф I-O-CONTROL (УПРАВЛЕНИЕ-ВВОДОМ-ВЫВОДОМ) указывает контрольные точки для перепрогона, общие области памяти, которые могут использоваться различными файлами, а также файлы, которые располагаются на одной катушке магнитной ленты.

Фраза RERUN (ПЕРЕПРОГОН) и фраза MULTIPLE FILE TAPE (НА ОДНОЙ КАТУШКЕ) параграфа I-O-CONTROL (УПРАВЛЕНИЕ-ВВОДОМ-ВЫВОДОМ) рассматриваются в настоящем стандарте как устаревшие элементы, которые будут удалены в следующей редакции стандарта.

2.10.2. Общий формат

I-O-CONTROL.

[[RERUN [ON { имя-файла-1 / имя-реализации-1 }]]
EVERY { { [END OF] { REEL / UNIT } / целое-1 RECORDS } OF имя-файла-2 / целое-2 CLOCK-UNITS / имя-условия-1 }]...
[SAME [RECORD] AREA FOR имя-файла-3 {имя-файла-4}...]...

GOST EXPERT ГОСТ 22558-89,

[MULTIPLE FILE TAPE CONTAINS {имя-файла-5 [POSITION целое-3]} ...] ... .]

УПРАВЛЕНИЕ-ВВОДОМ-ВЫВОДОМ.

[[ПЕРЕПРОГОН [НА { имя-файла-1 / имя-реализации-1 }]

{ КАЖДЫЙ КОНЕЦ { КАТУШКИ / ТОМА } имя-файла-2
КАЖДЫЕ целое-1 ЗАПИСЕЙ
КАЖДЫЕ целое-2 ЕДИНИЦ-ВРЕМЕНИ
КАЖДОЕ имя-условия-1 }]...

[ОБЩАЯ ОБЛАСТЬ [ЗАПИСИ] ДЛЯ имя-файла-3 {имя-файла-4} ...] ...

[НА ОДНОЙ КАТУШКЕ {имя-файла-5 [ПОЗИЦИЯ целое-3]} ...] ... .]

2.10.3. Синтаксические правила

(1) Порядок задания фраз не существенен.

2.10.4. Общие правила

(1) Фразы MULTIPLE FILE TAPE (НА ОДНОЙ КАТУШКЕ), SAME (ОБЩАЯ) и RERUN (ПЕРЕПРОГОН) описываются ниже в алфавитном порядке.

2.11. **Фраза MULTIPLE FILE TAPE (НА ОДНОЙ КАТУШКЕ)**

2.11.1. Назначение

Фраза MULTIPLE FILE TAPE (НА ОДНОЙ КАТУШКЕ) задает расположение файлов на катушке, содержащей несколько файлов. Фраза MULTIPLE FILE TAPE (НА ОДНОЙ КАТУШКЕ) является устаревшим элементом в настоящем стандарте и будет удалена в следующей редакции стандарта.

2.11.2. Общий формат

MULTIPLE FILE TAPE CONTAINS {имя-файла-1 [POSITION целое-1]} ...
НА ОДНОЙ КАТУШКЕ {имя-файла-1 [ПОЗИЦИЯ целое-1]} ...

2.11.3. Общие правила

(1) Фраза MULTIPLE FILE TAPE (НА ОДНОЙ КАТУШКЕ) требуется, когда более чем один файл расположен на одной

GOST EXPERT ГОСТ 22558-89,

физической катушке ленты. Независимо от общего числа файлов на одной катушке, должны описываться только те файлы, которые используются в объектной программе. Если все имена файлов перечисляются в последовательном порядке, фразу POSITION (ПОЗИЦИЯ) можно не задавать. Если некоторый файл из последовательности не указан, должна задаваться позиция относительно начала ленты. Одновременно может быть открыто не более одного файла на одной и той же катушке ленты.

2.12. Фраза RERUN (ПЕРЕГОН)

2.12.1. Назначение

Фраза RERUN (ПЕРЕПРОГОН) указывает контрольные точки для перепрогона. Фраза RERUN (ПЕРЕПРОГОН) является устаревшим элементом в настоящем стандарте и будет удалена в следующей редакции стандарта.

2.12.2. Общий формат

```
RERUN [ ON { имя-файла-1          } ]
           { имя-реализации-1     }

      { { [END OF] { REEL } }                          }
      { {          { UNIT } } OF имя-файла-2           }
EVERY { { целое-1 RECORDS   }                          }
      { целое-2 CLOCK-UNITS                            }
      { имя-условия-1                                  }

ПЕРЕПРОГОН [ НА { имя-файла-1          } ]
               { имя-реализации-1      }

      { { КАЖДЫЙ КОНЕЦ { КАТУШКИ } }                    }
      { {              { ТОМА    } } имя-файла-2       }
      { { КАЖДЫЕ целое-1 ЗАПИСЕЙ   }                   }
      { КАЖДЫЕ целое-2 ЕДИНИЦ-ВРЕМЕНИ                  }
      { КАЖДОЕ имя-условия-1                           }
```

2.12.3. Синтаксические правила

(1) Имя-файла-1 должно ссылаться на файл с последовательной организацией.

(2) Фраза END OF REEL/UNIT (КОНЕЦ КАТУШКИ/ТОМА) может использоваться, только если имя-файла-2 ссылается на файл с последовательной организацией.

(3) Если задана фраза целое-1 RECORDS (целое-1 ЗАПИСЕЙ) или фраза целое-2 CLOCK-UNITS (ЕДИНИЦ-ВРЕМЕНИ), во

GOST EXPERT ГОСТ 22558-89,

фразе RERUN (ПЕРЕПРОГОН) должно задаваться имя-реализации-1.

(4) Если для файла, на который ссылается имя-файла-2, указано более одной фразы RERUN (ПЕРЕПРОГОН), должны выполняться следующие ограничения:

а) если задано несколько фраз целое-1 RECORDS (целое-1 ЗАПИСЕЙ), никакие две из них не могут задавать одно и то же имя-файла-2;

б) если задано несколько фраз END OF REEL (КОНЕЦ КАТУШКИ) или END OF UNIT (КОНЕЦ ТОМА), никакие две из них не могут задавать одно и тоже имя-файла-2.

(5) Может быть задана только одна фраза RERUN (ПЕРЕПРОГОН), содержащая фразу CLOCK-UNITS (ЕДИНИЦ-ВРЕМЕНИ).

2.12.4. Общие правила

(1) Фраза RERUN (ПЕРЕПРОГОН) определяет, когда и где записывается информация перепрогона. Эта информация записывается следующими способами:

а) если задается имя-файла-1, информация перепрогона записывается на каждой катушке или каждом томе выходного файла и реализацией определяется, где на ленте или в файле должна быть записана информация перепрогона;

б) если задано имя-реализации-1, информация перепрогона записывается как отдельный файл на устройстве, определяемом реализацией.

(2) Существует семь вариантов фразы RERUN (ПЕРЕПРОГОН), различающихся условиями, при которых могут быть установлены точки перепрогона. Реализация должна обеспечивать хотя бы один из указанных вариантов фразы RERUN (ПЕРЕПРОГОН).

а) Если вариант END OF REEL (КОНЕЦ КАТУШКИ) или END OF UNIT (КОНЕЦ ТОМА) используется без варианта ON (НА), информация для перепрогона записывается в файл имя-файла-2, который должен быть выходным.

б) Если используется вариант END OF REEL (КОНЕЦ КАТУШКИ) или END OF UNIT (КОНЕЦ ТОМА) и за словом ON (НА указано имя-файла-1, информация для перепрогона записывается в файл имя-файла-1, который должен быть выходным файлом. Для файла имя-файла-2 выполняются обычные функции закрытия катушки или тома. Имя-файла-2 может быть именем входного или выходного файла.

в) Если используется вариант END OF REEL (КОНЕЦ КАТУШКИ) или END OF UNIT (КОНЕЦ ТОМА) и за словом ON (НА) указано имя-реализации, информация для перепрогона записывается на устройство, определенное реализацией. Имя-файла-2 может быть именем входного или выходного файла.

GOST EXPERT ГОСТ 22558-89,

г) Если используется вариант целое-1 RECORDS (ЗАПИСЕЙ), информация для перепрогона записывается на устройство, указанное именем-реализации, которое должно быть задано за словом ON (НА), после обработки каждых целое-1 записей имени-файла-2. Файл, указанный именем-файла-2, может быть входным или выходным и иметь любую организацию и доступ.

д) Если используется вариант целое-2 CLOCK-UNITS (ЕДИНИЦ-ВРЕМЕНИ), информация для перепрогона записывается на устройство, указанное именем-реализации, которое должно быть задано после слова ON (НА), всякий раз, когда истекает промежуток времени, вычисленный внутренними часами.

е) Если используется вариант имя-условия и после слова ON (НА) задано имя-реализации, информация для перепрогона записывается на устройство, указанное именем-реализации, всякий раз, когда состояние переключателя соответствует состоянию, определенному именем-условия. Переключатель должен быть определен в параграфе SPECIAL-NAMES (СПЕЦИАЛЬНЫЕ-ИМЕНА) секции конфигурации раздела оборудования. Момент проверки состояния переключателя определяется реализацией.

ж) Если используется вариант имя-условия и после слова ON (НА) указано имя-файла-1, информация для перепрогона записывается в файл имя-файла-1, который должен быть выходным файлом, всякий раз, когда состояние переключателя соответствует состоянию, определенному именем-условия. В этом случае, как и в предыдущем, переключатель должен быть определен в параграфе SPECIAL-NAMES (СПЕЦИАЛЬНЫЕ-ИМЕНА) секции конфигурации раздела оборудования. Момент проверки состояния переключателя определяется реализацией.

2.13. Фраза SAME AREA (ОБЩАЯ ОБЛАСТЬ)

2.13.1. Назначение

Фраза SAME AREA (ОБЩАЯ ОБЛАСТЬ) определяет область памяти, которая должна разделяться различными файлами.

2.13.2. Общий формат

SAME [RECORD] AREA FOR имя-файла-1 {имя-файла-2}...

ОБЩАЯ ОБЛАСТЬ [ЗАПИСИ] ДЛЯ имя-файла-1

{имя-файла-2}...

2.13.3. Синтаксические правила

(1) Имя-файла-1 и имя-файла-2 должны быть определены в параграфе FILE-CONTROL (УПРАВЛЕНИЕ-ФАЙЛАМИ) этой же программы.

(2) Имя-файла-1 и имя-файла-2 не могут ссылаться на определитель внешнего файла.

GOST EXPERT ГОСТ 22558-89,

(3) Если программа содержит более одной фразы SAME AREA (ОБЩАЯ ОБЛАСТЬ), должны выполняться следующие ограничения:

а) одно имя файла не должно появляться более чем в одной фразе SAME AREA (ОБЩАЯ ОБЛАСТЬ);

б) имя файла не должно появляться более чем в одной фразе SAME RECORD AREA (ОБЩАЯ ОБЛАСТЬ ЗАПИСИ);

в) если одно или более имен-файлов из фразы SAME AREA (ОБЩАЯ ОБЛАСТЬ) появляется во фразе SAME RECORD AREA (ОБЩАЯ ОБЛАСТЬ ЗАПИСИ), все имена файлов из этой фразы SAME AREA (ОБЩАЯ ОБЛАСТЬ) должны появиться во фразе SAME RECORD AREA (ОБЩАЯ ОБЛАСТЬ ЗАПИСИ). Однако во фразе SAME RECORD AREA (ОБЩАЯ ОБЛАСТЬ ЗАПИСИ) могут задаваться также дополнительные имена-файлов, не встречающиеся во фразе SAME AREA (ОБЩАЯ ОБЛАСТЬ). Правило, согласно которому в любой заданный момент времени может быть открыт только один из файлов, упомянутых во фразе SAME AREA (ОБЩАЯ ОБЛАСТЬ), имеет предпочтение над правилом, что все файлы, заданные во фразе SAME RECORD AREA (ОБЩАЯ ОБЛАСТЬ ЗАПИСИ), могут быть открыты в любой момент времени.

(4) Файлы, заданные во фразе SAME AREA (ОБЩАЯ ОБЛАСТЬ) или SAME RECORD AREA (ОБЩАЯ ОБЛАСТЬ ЗАПИСИ), не обязаны иметь одну и ту же организацию или метод доступа.

2.13.4. Общие правила

(1) Фраза SAME AREA (ОБЩАЯ ОБЛАСТЬ) указывает, что два или более файлов, на которые ссылаются имя-файла-1, имя-файла-2 и которые не являются сортируемыми или сливаемыми файлами, во время обработки должны использовать одну и ту же область памяти. К разделяемой области относятся все области памяти, связанные с файлами, заданными именем-файла-1, именем-файла-2, поэтому не допускается, чтобы в один и тот же момент времени был открыт более чем один из этих файлов (см. синтаксическое правило 3в).

(2) Фраза SAME RECORD AREA (ОБЩАЯ ОБЛАСТЬ ЗАПИСИ) указывает, что два или более файлов, на которые ссылаются имя-файла-1, имя-файла-2, должны использовать одну и ту же область памяти для обработки текущей логической записи. Все эти файлы могут быть открыты одновременно. Логическая запись в общей области записи рассматривается как логическая запись каждого открытого как выходной файла, имя которого встречается в этой фразе SAME RECORD AREA (ОБЩАЯ ОБЛАСТЬ ЗАПИСИ), а также последнего прочитанного

GOST EXPERT ГОСТ 22558-89,

входного файла, имя которого задано в данной фразе SAME RECORD AREA (ОБЩАЯ ОБЛАСТЬ ЗАПИСИ). Это эквивалентно неявному переопределению этой области, т. е. записи выравниваются по позиции самой левой литеры.

3. РАЗДЕЛ ДАННЫХ В МОДУЛЕ ПОСЛЕДОВАТЕЛЬНОГО ВВОДА-ВЫВОДА

3.1. Секция файлов

Секция файлов расположена в разделе данных исходной программы. Каждый файл определяется статьей описания файла и одной или более статей описания записи. Статьи описания записи задаются непосредственно за статьей описания файла.

Общий формат секции файлов в модуле последовательного ввода-вывода приводится ниже.

FILE SECTION.

[статья-описания-файла
{статья-описания-записи} . . .] . . .

СЕКЦИЯ ФАЙЛОВ.

[статья-описания-файла
{статья-описания-записи} . . .] . . .

3.1.1. Статья описания файла

В программе статья описания файла FD (ОФ) представляет высший уровень организации в секции файлов. За заголовком секции файлов следует статья описания файла, состоящая из индикатора уровня FD (ОФ), имени файла и последовательности независимых фраз. Фразы статьи описания файла указывают размер логической и физической записи, наличие или отсутствие записей меток, значения элементов меток, определяемых реализацией, имена записей данных, которые составляют файл, и число строк на логической печатной странице. Статья должна заканчиваться точкой.

3.1.2. Структура описания записи

Описание записи состоит из последовательности статей описания данных, которые описывают характеристики отдельной записи. Каждая статья описания данного состоит из номера уровня, за которым следует имя данного или фраза FILLER (ЗАПОЛНИТЕЛЬ), если они указываются, и последовательность независимых фраз. Описание записи имеет иерархическую структуру и поэтому фразы, которые используются в статье, могут значительно изменяться в зависимости от наличия подчиненных статей. Структура описания записи и элементов, допустимых в статье описания записи, описывается в ч. 4, п. 4.3.2 и ч. 6, п. 5.3. Допустимость отдельных фраз в статье описания данных зависит от уровня модуля ядра, поддерживаемого реализацией.

GOST EXPERT ГОСТ 22558-89,

3.1.3. Начальные значения

Начальное значение данного в секции файлов не определено.

3.2. **Статья описания файла**

3.2.1. Назначение

Статья описания файла обеспечивает информацию о физической структуре, идентификации и именах записей, относящихся к данному файлу.

3.2.2. Общий формат

FD имя-файла-1

[BLOCK CONTAINS [[целое-1 TO]] целое-2 { RECORDS / CHARACTERS }]

[RECORD { CONTAINS целое-3 CHARACTERS / IS VARYING IN SIZE [[FROM целое-4] [TO целое-5]] [CHARACTERS] [DEPENDING ON имя-данного-1] / CONTAINS целое-6 TO целое-7 CHARACTERS }]

[LABEL { RECORD IS / RECORDS ARE } { STANDARD / OMITTED }]

[VALUE OF {имя-реализации-1 IS { [имя-данного-2] / литерал-1 } } ...]

[DATA { RECORD IS / RECORDS ARE } {имя-данного-3} ...]

[LINAGE IS { имя-данного-4 / целое-8 } LINES [[WITH FOOTING AT { имя-данного-5 / целое-9 }] [LINES AT TOP { имя-данного-6 / целое-10 }] [LINES AT BOTTOM { имя-данного-7 / целое-11 }]]]

[CODE-SET IS имя-алфавита-1].

GOST EXPERT ГОСТ 22558-89,

GOST ГОСТ 22558-89,

[АЛФАВИТ имя-алфавита-1].

3.2.3. Синтаксические правила

(1) Индикатор уровня FD (ОФ) идентифицирует начало статьи описания файла и должен предшествовать имени-файла-1.

(2) Фразы, которые следуют за именем-файла-1, могут задаваться в любом порядке.

(3) Одна или несколько статей описания записи должны следовать за статьей описания файла.

3.2.4. Общие правила

(1) Статья описания файла связывает имя-файла-1 с определителем файла.

(2) Фразы BLOCK CONTAINS (В БЛОКЕ), CODE-SET (АЛФАВИТ), DATA RECORDS (ЗАПИСИ ДАННЫХ), LABEL RECORD (МЕТКИ), LINAGE (ВЕРСТКА), RECORD (В ЗАПИСИ) и VALUE OF (ЗНАЧЕНИЕ) описываются в алфавитном порядке.

3.3. Фраза BLOCK CONTAINS (В БЛОКЕ)

3.3.1. Назначение

Фраза BLOCK CONTAINS (В БЛОКЕ) определяет размер физической записи.

3.3.2. Общий формат

BLOCK CONTAINS [целое-1 TO] целое-2 { RECORDS / CHARACTERS }

В БЛОКЕ [ОТ целое-1 ДО] целое-2 { ЗАПИСЕЙ / ЛИТЕР }

3.3.3. Общие правила

(1) Эта фраза обязательна за исключением следующих случаев:

а) физическая запись содержит только одну полную логическую запись;

б) устройство, назначенное файлу, допускает один и только один размер физической записи;

в) количество записей, содержащихся в блоке, определяется операционной средой.

(2) Размер физической записи должен быть определен в литерах, если имеет место одна из следующих ситуаций:

а) в файле массовой памяти логическая запись больше физической записи;

б) физическая запись содержит дополнения (области, не содержащиеся в логической записи);

в) логические записи группируются таким образом, что будет предполагаться неточный размер физической записи.

GOST EXPERT ГОСТ 22558-89,

(3) Когда используется слово CHARACTERS (ЛИТЕР), размер физической записи указывается числом позиций литер, требуемых для запоминания физической записи.

(4) Если целое-1 не задано, целое-2 задает точный размер физической записи. Если указаны и целое-1, и целое-2, они представляют соответственно минимальный и максимальный размер физической записи.

(5) Если соответствующий определитель файла является внешним определителем файла, все фразы BLOCK CONTAINS (В БЛОКЕ) в одной единице исполнения, относящиеся к этому определителю файла, должны иметь одни и те же значения целого-1 и целого-2.

3.4. Фраза CODE-SET (АЛФАВИТ)

3.4.1. Назначение

Фраза CODE-SET (АЛФАВИТ) указывает соглашение для кодов литер, используемых для представления данных на носителе данных.

3.4.2. Общий формат

CODE-SET IS имя-алфавита-1

АЛФАВИТ имя-алфавита-1

3.4.3. Синтаксические правила

(1) Если для файла указана фраза CODE-SET (АЛФАВИТ), все данные в этом файле должны быть описаны как USAGE IS DISPLAY (ДЛЯ ВЫДАЧИ), а описания числовых данных со знаком должны содержать фразу SIGH SEPARATE (ЗНАК ОТДЕЛЬНО).

(2) Имя-алфавита не должно определяться в разделе оборудования литеральной фразой.

3.4.4. Общие правила

(1) Если задана фраза CODE-SET (АЛФАВИТ):

а) после успешного выполнения оператора OPEN (ОТКРЫТЬ) в качестве набора литер, используемого для представления данных на внешнем носителе, используется набор, заданный именем-алфавита-1 в статье описания файла для имени-файла, заданного в операторе OPEN (ОТКРЫТЬ) (см. ч. 6, п. 4.5);

б) она задает алгоритм преобразования набора литер, используемого для представления данных на внешнем носителе, во внутреннее представление и обратно при выполнении операций ввода или вывода.

(2) Если фраза CODE-SET (АЛФАВИТ) не задана, предполагается использование на внешнем носителе внутреннего набора литер.

(3) Если соответствующий определитель файла является внешним определителем файла, все фразы CODE-SET (АЛФАВИТ) в

GOST EXPERT ГОСТ 22558-89,

единице исполнения, относящиеся к этому определителю файла, должны задавать один и тот же набор литер.

3.5. **Фраза DATA RECORDS (ЗАПИСИ ДАННЫХ)**

3.5.1. Назначение

Фраза DATA RECORDS (ЗАПИСИ ДАННЫХ) используется только в целях документации для указания имен записей данных, связанных с файлом. Фраза DATA RECORDS (ЗАПИСИ ДАННЫХ) является устаревшим элементом в настоящем стандарте и будет удалена в следующей редакции стандарта.

3.5.2. Общий формат

DATA { RECORD IS / RECORDS ARE } {имя-данного-1} ...

ЗАПИСИ ДАННЫХ {имя-данного-1} ...

3.5.3. Синтаксическое правило

(1) Имя-данного-1 есть имя записи данных, которое должно появиться в статье описания записи на уровне 01.

3.5.4. Общие правила

(1) Наличие более одного имени-данного означает, что файл содержит более одного типа записей данных. Эти записи могут быть разных размеров, разных форматов и т. д. Порядок, в котором перечислены имена-записей, не существенен.

(2) Логически все записи данных файла помещаются в одну и ту же область. Это в одинаковой мере относится к файлам, содержащим более одного типа записей данных.

3.6. **Фраза LABEL RECORDS (МЕТКИ)**

3.6.1. Назначение

Фраза LABEL RECORDS (МЕТКИ) указывает, присутствуют ли метки.

Фраза LABEL RECORDS (МЕТКИ) является устаревшим элементом в настоящем стандарте и будет удалена в следующей редакции стандарта.

3.6.2. Общий формат

LABEL { RECORD IS / RECORDS ARE } { STANDARD / OMITTED }

МЕТКИ { СТАНДАРТНЫ / ОПУЩЕНЫ }

3.6.3. Общие правила

(1) Вариант OMITTED (ОПУЩЕНЫ) указывает, что нет явных меток для файла или устройства, назначенного файлу.

(2) Вариант STANDARD (СТАНДАРТНЫ) указывает, что существуют метки файла или устройства, назначенного файлу, и эти метки соответствуют спецификациям меток, определенным реализацией.

GOST EXPERT ГОСТ 22558-89,

(3) Если фраза LABEL RECORDS (МЕТКИ) для файла не задана, метки для этого файла должны соответствовать спецификациям меток, определенным реализацией.

(4) Если определитель файла, соответствующий этой статье описания файла, является внешним определителем файла (ч. 10, п. 4.5), все фразы LABEL RECORDS (МЕТКИ) в одной единице исполнения, связанные с этим определителем файла, должны иметь одинаковую спецификацию.

3.7. **Фраза LINAGE (ВЕРСТКА)**

3.7.1. Назначение

Фраза LINAGE (ВЕРСТКА) указывает на размер логической страницы в строках. Она также позволяет указать размер верхнего и нижнего поля на странице и номер строки тела страницы, на которой начинается область концовки.

3.7.2. Общий формат

LINAGE IS { имя-данного-1 / целое-1 } LINES [WITH FOOTING AT { имя-данного-2 / целое-2 }]

[LINES AT TOP { имя-данного-3 / целое-3 }] [LINES AT BOTTOM { имя-данного-4 / целое-4 }]

ВЕРСТКА { имя-данного-1 / целое-1 } СТРОК [КОНЦОВКА ОТ { имя-данного-2 / целое-2 }]

[ВЕРХНЕЕ ПОЛЕ { имя-данного-3 / целое-3 }] [НИЖНЕЕ ПОЛЕ { имя-данного-4 / целое-4 }]

3.7.3. Синтаксические правила

(1) Имя-данного-1, имя-данного-2, имя-данного-3, имя-данного-4 должны относиться к элементарным целым числовым данным.

(2) Имя-данного-1, имя-данного-2, имя-данного-3, имя-данного-4 могут уточняться.

(3) Значение целого-2 должно быть не больше целого-1.

GOST EXPERT ГОСТ 22558-89,

(4) Значения целого-3 и целого-4 могут быть нулем.

3.7.4. Общие правила

(1) Фраза LINAGE (ВЕРСТКА) указывает на размер логической страницы в строках. Размер логической страницы есть сумма величин, заданных в каждом варианте фразы, исключая вариант FOOTING (КОНЦОВКА). Если вариант LINES AT BOTTOM (ВЕРХНЕЕ ПОЛЕ) или LINES AT TOP (НИЖНЕЕ ПОЛЕ) не задан, размеры соответствующих полей равны нулю. Если не задан вариант FOOTING (КОНЦОВКА), условие конца страницы не возникает, независимо от того, существует условие переполнения страницы или нет.

Размер логической страницы может устанавливаться независимо от размера физической страницы.

(2) Целое-1 или значение данного, представленного именем-данного-1, указывает число строк, которые могут быть записаны и (или) пропущены на логической странице. Это значение должно быть больше нуля. Та часть логической страницы, на которой эти строки могут быть записаны и (или) пропущены, называется телом страницы.

(3) Целое-2 или значение данного, представленного именем-данного-2, указывает номер строки тела страницы, на которой начинается область концовки. Это значение должно быть больше нуля, но не должно превышать целое-1 или значение данного, представленного именем-данного-1.

Область концовки включает в себя область логической страницы между строкой, определяемой целым-2 или значением данного, представленного именем-данного-2, и строкой, определяемой целым-1 или значением данного, представленного именем-данного-1, включительно.

(4) Целое-3 или значение данного, представленного именем-данного-3, указывает число строк, содержащихся в верхнем поле логической страницы. Это значение может быть равно нулю.

(5) Целое-4 или значение данного, представленного именем-данного-4, указывает число строк, содержащихся в нижнем поле логической страницы. Это значение может быть равно нулю.

(6) Целое-1, целое-3 и целое-4, если они указаны, используются во время открытия файла при выполнении оператора OPEN (ОТКРЫТЬ) с фразой OUTPUT (ВЫХОДНОЙ) и указывают число строк, содержащееся в каждой из названных областей логической страницы. Целое-2, если оно указано, должно использоваться во время определения области концовки. Эти значения используются для всех логических страниц, которые записываются в файл, во время выполнения данной программы.

(7) Значения данных, представленных именем-данного-1,

GOST EXPERT ГОСТ 22558-89,

именем-данного-3 и именем-данного-4, если они указаны, используются следующим образом:

а) во время выполнения оператора OPEN (ОТКРЫТЬ) с фразой OUTPUT (ВЫХОДНОЙ) для определения количества строк в каждой из указанных областей для первой логической страницы;

б) во время выполнения оператора WRITE (ПИСАТЬ) с фразой ADVANCING PAGE (ПРОДВИЖЕНИЯ СТРАНИЦЫ) или при возникновении условия переполнения страницы для определения количества строк в каждой из указанных областей для следующей страницы (п. 4.7 настоящей части).

(8) Значение данного, представленного именем-данного-2, во время выполнения оператора OPEN (ОТКРЫТЬ) с фразой OUTPUT (ВЫХОДНОЙ) используется для определения области концовки первой логической страницы, а во время выполнения оператора WRITE (ПИСАТЬ) с фразой ADVANCING PAGE (ПРОДВИЖЕНИЯ СТРАНИЦЫ) или при возникновении условия переполнения страницы — для определения области концовки следующей логической страницы.

(9) LINAGE-COUNTER (СЧЕТЧИК-ВЕРСТКИ) генерируется, если задана фраза LINAGE (ВЕРСТКА). Его значение в каждый момент времени указывает номер строки тела текущей логической страницы, на которую позиционировано устройство. LINAGE-COUNTER (СЧЕТЧИК-ВЕРСТКИ) подчиняется следующим правилам:

а) для каждого файла, описанного в секции файлов с фразой LINAGE (ВЕРСТКА), генерируется отдельный LINAGE-COUNTER (СЧЕТЧИК-ВЕРСТКИ);

б) LINAGE-COUNTER (СЧЕТЧИК-ВЕРСТКИ) может использоваться в операторах раздела процедур, однако только система управления вводом-выводом может изменять его значение. Если в программе LINAGE-COUNTER (СЧЕТЧИК-ВЕРСТКИ) используется для нескольких файлов, он должен при необходимости уточняться именем файла;

в) при выполнении оператора WRITE (ПИСАТЬ) связанный с файлом LINAGE-COUNTER (СЧЕТЧИК-ВЕРСТКИ) автоматически увеличивается по следующим правилам:

1) если указан оператор WRITE (ПИСАТЬ) с фразой ADVANCING PAGE (ПРОДВИЖЕНИЯ СТРАНИЦЫ), LINAGE-COUNTER (СЧЕТЧИК-ВЕРСТКИ) автоматически устанавливается на 1. После установки счетчика верстки на 1 его значение неявно увеличивается до тех пор, пока не превысит значение, заданное целым-1 или данным, на которое ссылается имя-данного-1;

GOST EXPERT ГОСТ 22558-89,

2) если во фразе ADVANCING (ПРОДВИЖЕНИЯ) указаны идентификатор-2 или целое-1, LINAGE-COUNTER (СЧЕТЧИК-ВЕРСТКИ) увеличивается на целое-1 или значение данного, представленного идентификатором-2;

3) если фраза ADVANCING (ПРОДВИЖЕНИЯ) в операторе WRITE (ПИСАТЬ) не указана, LINAGE-COUNTER (СЧЕТЧИК-ВЕРСТКИ) увеличивается на 1 (п. 4.7 настоящей части);

4) когда устройство позиционируется на первую строку каждой следующей логической страницы, LINAGE-COUNTER (СЧЕТЧИК-ВЕРСТКИ) автоматически устанавливается на 1 (п. 4.7 настоящей части);

г) при выполнении оператора OPEN (ОТКРЫТЬ) с фразой OUTPUT (ВЫХОДНОЙ) для соответствующего файла LINAGE-COUTER (СЧЕТЧИК-ВЕРСТКИ) автоматически устанавливается на 1.

(10) Никаких дополнительных пропусков между соседними логическими страницами не делается.

(11) Если определитель файла, соответствующий этой статье описания файла, является внешним определителем файла, все статьи описания файла в пределах одной единицы исполнения, относящиеся к этому определителю файла, должны иметь:

а) фразу LINAGE (ВЕРСТКА), если хотя бы одна статья описания файла имеет фразу LINAGE (ВЕРСТКА);

б) одни и те же значения соответственно для целого-1, целого-2, целого-3 и целого-4, если они заданы;

в) одни и те же внешние данные, на которые ссылаются имя-данного-1, имя-данного-2, имя-данного-3 и имя-данного-4.

3.8. **Фраза RECORD (В ЗАПИСИ)**

3.8.1. Назначение

Фраза RECORD (В ЗАПИСИ) задает количество позиций литер в записях фиксированной длины или диапазон числа позиций литер в записях переменной длины. Если число позиций литер меняется, фраза задает его минимальное и максимальное значения.

3.8.2. Общий формат

Формат 1

RECORD CONTAINS целое-1 CHARACTERS

В ЗАПИСИ целое-1 ЛИТЕР

Формат 2

RECORD IS VARYING IN SIZE [[FROM целое-2] [TO целое-3] CHARACTERS]
[DEPENDING ON имя-данного-1]

GOST EXPERT ГОСТ 22558-89,

В ЗАПИСИ ПЕРЕМЕННОЕ ЧИСЛО [[ОТ целое-2] [ДО целое-3] ЛИТЕР]
[В ЗАВИСИМОСТИ ОТ имя-данного-1]

Формат 3
RECORD CONTAINS целое-4 ТО целое-5 CHARACTERS
В ЗАПИСИ ОТ целое-4 ДО целое-5 ЛИТЕР

3.8.2. Синтаксические правила

Формат 1

(1) Ни одна статья описания записи для файла не может определять число позиций литер, большее, чем целое-1.

Формат 2

(2) Статьи описания записей для файла не должны описывать записи, которые содержат меньшее количество позиций литер, чем задано целым-2, или записи, которые содержат количество позиций литер больше, чем задано целым-3.

(3) Целое-3 должно быть больше целого-2.

(4) Имя-данного-1 должно описывать элементарное целое без знака в секции рабочей памяти или секции связи.

3.8.4. Общие правила

Все форматы

(1) Если фраза RECORD (В ЗАПИСИ) не задана, размер каждой записи данных полностью определяется статьей описания записи.

(2) Если соответствующий определитель файла является внешним определителем файла, все статьи описания файла в пределах одной и той же единицы исполнения, которые соответствуют этому определителю файла, должны задавать одни и те же значения для целого-1 или целого-2 и целого-3. Если фраза RECORD (В ЗАПИСИ) не задана, все статьи описания записей, относящиеся к этому определителю файла, должны иметь одинаковую длину.

Формат 1

(3) Формат 1 используется для описания записей фиксированной длины. Целое-1 задает число позиций литер, содержащихся в каждой записи файла.

Формат 2

(4) Формат 2 используется для описания записей переменной длины. Целое-2 задает минимальное количество позиций литер, которое может содержаться в любой записи файла. Целое-3 задает максимальное количество позиций литер в любой записи файла.

GOST EXPERT ГОСТ 22558-89,

(5) Количество позиций литер, соответствующее статье описания записи, определяется как сумма количества позиций литер во всех элементарных данных, за исключением переопределений и переименований, плюс все неявные заполнители, обусловленные выравниванием. Если задается таблица:

а) минимальное число элементов таблицы, описанной в записи, используется в вышеупомянутом суммировании для определения минимального количества позиций литер, соответствующего описанию записи;

б) максимальное число элементов таблицы, описанной в записи, используется в вышеупомянутом суммировании для определения максимального количества литер, соответствующего описанию записи.

(6) Если целое-2 не задано, минимальное количество позиций литер, содержащееся в любой записи файла, равно наименьшему числу позиций литер, заданному для записей этого файла.

(7) Если целое-3 не задано, максимальное количество позиций литер, содержащееся в любой записи файла, равно наибольшему числу позиций литер, заданному для записей этого файла.

(8) Если задано имя-данного-1, количество позиций литер записи должно быть помещено в данное, на которое ссылается имя-данного-1, до выполнения любого оператора REWRITE (ОБНОВИТЬ), RELEASE (ПЕРЕДАТЬ) или WRITE (ПИСАТЬ) для этого файла.

(9) Если задано имя-данного-1, выполнение операторов REWRITE (ОБНОВИТЬ), RELEASE (ПЕРЕДАТЬ), WRITE (ПИСАТЬ), START (ПОДВЕСТИ) или DELETE (УДАЛИТЬ), либо неуспешное выполнение операторов RETURN (ВЕРНУТЬ) или READ (ЧИТАТЬ) не меняет содержимого данного, на которое ссылается имя-данного-1.

(10) Во время выполнения операторов REWRITE (ОБНОВИТЬ), RELEASE (ПЕРЕДАТЬ) или WRITE (ПИСАТЬ) количество позиций литер в записи определяется следующим образом:

а) если задано имя-данного-1, — содержимым данного, на которое ссылается имя-данного-1;

б) если имя-данного-1 не задано и запись не содержит данных с переменным числом вхождений, — числом позиций литер в записи;

в) если имя-данного-1 не задано и запись содержит данные с переменным числом вхождений, — суммой размеров фиксированной части и части таблицы, описанной с числом вхождений, во время выполнения оператора вывода.

GOST EXPERT ГОСТ 22558-89,

(11) Если имя-данного-1 задано, после успешного выполнения оператора RETURN (ВЕРНУТЬ) или READ (ЧИТАТЬ) для файла содержимое данного, на которое ссылается имя-данного-1, будет указывать число позиций литер только что прочитанной записи.

(12) Если в операторе RETURN (ВЕРНУТЬ) или READ (ЧИТАТЬ) задана фраза INTO (В), количество позиций литер в текущей записи, участвующей как посылаемое данное в неявном операторе MOVE (ПОМЕСТИТЬ), определяется следующим образом:

а) если задано имя-данного-1, — содержимым данного, на которое ссылается имя-данного-1;

б) если имя-данного-1 не задано, — значением, которое могло быть помещено в данное, на которое ссылается имя-данного-1, если бы оно было задано.

Формат 3

(13) Если используется формат 3 фразы RECORD (В ЗАПИСИ), целое-4 и целое-5 ссылаются на минимальное количество литер в записи данных наименьшего размера и на максимальное количество литер в записи данных наибольшего размера соответственно. Однако, в этом случае размер каждой записи данных полностью определяется в статье описания записи.

(14) Размер каждой записи данных определяется в терминах числа позиций литер, необходимых для заполнения логической записи, независимо от типов литер, используемых для представления элементов в логической записи. Размер записи определяется как сумма количества литер всех элементарных данных фиксированной длины плюс сумма максимального количества литер всех подчиненных записи элементов переменной длины. Эта сумма может отличаться от фактического размера записи (см. ч. 4, п. 4.3.4; ч. 6, пп. 5.13, 5.14).

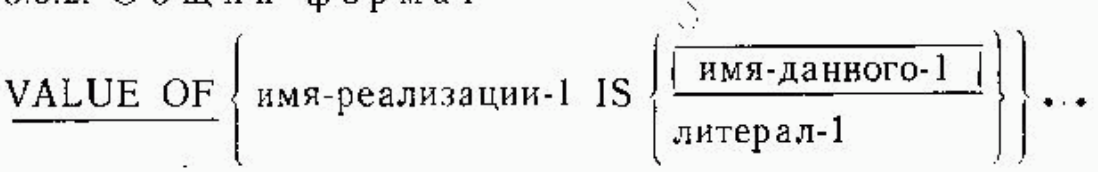
3.9. Фраза VALUE (ЗНАЧЕНИЕ)

3.9.1. Назначение

Фраза VALUE (ЗНАЧЕНИЕ) конкретизирует описание данного в записях меток, связанных с файлом.

Фраза VALUE (ЗНАЧЕНИЕ) является устаревшим элементом в настоящем стандарте и будет удалена в следующей редакции стандарта.

3.9.2. Общий формат



GOST EXPERT ГОСТ 22558-89,

{ЗНАЧЕНИЕ / ЗНАЧ} {имя-реализации-1 {имя-данного-1 / литерал-1} ...

3.9.3. Синтаксические правила

(1) Имя-данного-1 может быть уточнено, если необходимо, но не может индексироваться или быть описано с фразой USAGE IS INDEX (ДЛЯ ИНДЕКСА).

(2) Имя-данного-1 должно быть определено в секции рабочей памяти.

3.9.4. Общие правила

(1) Для входных файлов соответствующая программа обработки метки проверяет, совпадают ли значения имени-реализации-1 и литерала-1 или значения данного, на которое ссылается имя-данного-1, в зависимости от того, что из них указано.

Для выходных файлов значение имени-реализации-1 делается равным значению литерала-1 или значению данного, на которое ссылается имя-данного-1, в зависимости от того, что из них указано.

(2) Если соответствующий определитель файла является внешним определителем файла, все фразы VALUE OF (ЗНАЧЕНИЕ) в одной и той же единице исполнения, соответствующие этому определителю файла, должны быть совместимы. Правила совместимости определяются реализацией.

4. РАЗДЕЛ ПРОЦЕДУР В МОДУЛЕ ПОСЛЕДОВАТЕЛЬНОГО ВВОДА-ВЫВОДА

4.1. Общее описание

Когда в исходной Кобол-программе присутствует оператор USE (ИСПОЛЬЗОВАТЬ) модуля последовательного ввода-вывода, раздел процедур содержит декларативные процедуры. Ниже приводится общий формат раздела процедур в случае, когда присутствует оператор USE (ИСПОЛЬЗОВАТЬ).

PROCEDURE DIVISION.

DECLARATIVES.

{имя-секции SECTION.

оператор USE.

[имя-параграфа.

[предложение] ...} ...}...

GOST EXPERT ГОСТ 22558-89,

```
END DECLARATIVES.
  {имя-секции SECTION.
  [имя-параграфа.
    [предложение] ...] ...}...
РАЗДЕЛ ПРОЦЕДУР.
ДЕКЛАРАТИВЫ.
  {СЕКЦИЯ имя-секции.
          оператор ИСПОЛЬЗОВАТЬ.
  [имя-параграфа.
    [предложение] ...] ...}...
КОНЕЦ ДЕКЛАРАТИВ.
  {СЕКЦИЯ имя-секции.
  [имя-параграфа.
    [предложение] ...] ...}...
```

4.2. **Оператор CLOSE (ЗАКРЫТЬ)**

4.2.1. Назначение

Оператор CLOSE (ЗАКРЫТЬ) завершает обработку катушки, тома и файла с необязательной перемоткой и (или) удалением, если необходимо.

4.2.2. Общий формат

```
CLOSE { имя-файла-1 [ { REEL } [FOR REMOVAL] ] } ...
      {             { UNIT }                   }
      {             [ WITH { NO REWIND } ]     }
      {                    { LOCK      }       }

ЗАКРЫТЬ { { КАТУШКУ } имя-файла-1 [С УДАЛЕНИЕМ] } ...
        { { ТОМ    }               [          ] }
        { имя-файла-1 [ [ БЕЗ ПЕРЕМОТКИ ] ]     }
        {             [ [ С ЗАМКОМ      ] ]     }
```

4.2.3. Синтаксические правила

(1) Файлы, перечисленные в операторе CLOSE (ЗАКРЫТЬ), могут иметь различную организацию и доступ.

4.2.4. Общие правила

Термины «катушка» и «том» являются синонимами и полностью взаимозаменяемы в операторе CLOSE (ЗАКРЫТЬ), если не ого-

GOST EXPERT ГОСТ 22558-89,

ворено обратное. Интерпретация последовательных файлов массовой памяти логически эквивалентна интерпретации файлов на ленте или аналогичных последовательных носителях. Файл, содержащийся в многофайловой среде, логически рассматривается как последовательный однотомный (однокатушечный) файл.

(1) Оператор CLOSE (ЗАКРЫТЬ) может быть выполнен только для файла, который был открыт.

(2) Для того, чтобы показать действие различных типов оператора CLOSE (ЗАКРЫТЬ) для различных носителей данных, все файлы разделяются на следующие категории:

а) без катушек (томов). Файл, носитель которого такой, что для него понятие перемотки катушек (томов) не имеет смысла;

б) последовательный однокатушечный (однотомный). Последовательный файл, который полностью располагается на одной катушке (томе);

в) последовательный файл, который располагается на нескольких катушках (томах).

(3) Результаты выполнения каждого типа оператора CLOSE (ЗАКРЫТЬ) для каждой категории файла приведены ниже.

Формат оператора CLOSE (ЗАКРЫТЬ)	Выполняемые действия для различных категорий файла		
	Без катушек (томов)	Последовательный однокатушечный (однотомный)	Последовательный многокатушечный (многотомный)
CLOSE (ЗАКРЫТЬ)	**В**	**В, Ж**	**А, В, Ж**
CLOSE WITH LOCK (ЗАКРЫТЬ С ЗАМКОМ)	**В, Д**	**В, Ж, Д**	**А, В, Д, Ж**
CLOSE WITH NO REWIND (ЗАКРЫТЬ БЕЗ ПЕРЕМОТКИ)	**В, З**	**В, Б**	**А, Б, В**
CLOSE REEL/UNIT (ЗАКРЫТЬ КАТУШКУ/ТОМ)	**Е**	**Е, Ж**	**Е, Ж**
CLOSE REEL/UNIT FOR REMOVAL (ЗАКРЫТЬ КАТУШКУ/ТОМ С УДАЛЕНИЕМ)	**Е**	**Г, Е, Ж**	**Г, Е, Ж**

Примечание. Значения символов **А—З** приведены ниже. Там, где эти значения зависят от того, является ли файл входным, выходным или входным-выходным, приводятся дополнительные пояснения, в противном случае эти определения относятся к входным, выходным или входным-выходным файлам.

А — влияние на обработанные ранее катушки (тома).

Входные и входные-выходные файлы. Все катушки (тома) в файле, предшествующие текущей катушке (тому), закрываются,

GOST EXPERT ГОСТ 22558-89,

если только для них не выполнялся оператор CLOSE REEL (ЗАКРЫТЬ КАТУШКУ) или CLOSE UNIT (ЗАКРЫТЬ ТОМ). Если текущая катушка (том) в файле не последняя, все следующие катушки (тома) не обрабатываются.

Выходные файлы. Все катушки (тома) файла, предшествующие текущей катушке (тому), закрываются, если для них не выполнялся оператор CLOSE REEL (ЗАКРЫТЬ КАТУШКУ) или CLOSE UNIT (ЗАКРЫТЬ ТОМ).

Б — текущая катушка не перематывается.

Текущая катушка (том) остается в текущей позиции.

В — закрыть файл.

Входные и входные-выходные файлы. Если файл установлен в конце и записи меток для этого файла специфицированы, метки обрабатываются в соответствии со стандартной процедурой обработки меток, определенной реализацией. Действия оператора CLOSE (ЗАКРЫТЬ) не определены, когда записи меток специфицированы, но в файле отсутствуют, или когда записи меток не специфицированы, но присутствуют. Выполняются операции закрытия, определенные реализацией. Если файл установлен в конце и записи меток для него не специфицированы, метки не обрабатываются, но другие операции закрытия, определенные реализацией, выполняются. Если файл установлен не в конце, выполняются операции закрытия, определенные реализацией, но конечные метки не обрабатываются.

Выходные файлы. Если записи меток для файла специфицированы, метки создаются в соответствии со стандартной процедурой обработки меток, определенной реализацией. Действия оператора CLOSE (ЗАКРЫТЬ) не определены, когда записи меток специфицированы, но в файле отсутствуют, или когда они не специфицированы, но присутствуют. Выполняются операции закрытия, определенные реализацией. Если записи меток для файла не указаны, метки не обрабатываются, но другие операции закрытия, определенные реализацией, выполняются.

Г — удаление катушки (тома).

Если это применимо, производится перемотка текущей катушки или тома и логическое удаление их из единицы исполнения. Однако катушка или том может снова стать доступной в порядке расположения катушек или томов в файле, если впоследствии за оператором CLOSE (ЗАКРЫТЬ) без фразы REEL (КАТУШКУ) или UNIT (ТОМ) для этого файла будет выполнен оператор OPEN (ОТКРЫТЬ).

Д — закрыть с замком.

Файл закрывается; он не может быть открыт во время выполнения данной единицы исполнения.

GOST EXPERT ГОСТ 22558-89,

Е — закрыть катушку или закрыть том.

Входные и входные-выходные файлы.

Выполняются следующие операции:

1) если текущая катушка или том является последней или единственной для файла или если катушка располагается на носителе иной природы, смена катушки или тома не производится; указатель текущего тома остается неизменным;

2) если для файла существует другая катушка (том), производится смена катушки (тома), указатель текущей катушки (тома) указывает на следующую катушку (том), существующую в файле, выполняется стандартная процедура обработки начальных меток катушки (тома). Если на текущем томе не существует записей данных, производится смена катушки (тома).

Выходные файлы (носители в виде катушки или тома). Выполняются следующие операции:

1) стандартная процедура обработки конечных меток катушки или тома;

2) смена катушки (тома). Указатель текущего тома указывает на новую катушку (том);

3) стандартная процедура обработки начальных меток катушки или тома;

4) следующий оператор WRITE (ПИСАТЬ) записывает следующую логическую запись на следующую катушку или том данного файла.

Выходные файлы (носители не в виде катушки или тома).

Выполнение этого оператора считается успешным. Файл остается открытым; не осуществляются никакие действия, кроме указанных в общем правиле (4).

Ж — перемотка.

Текущая катушка или аналогичное устройство устанавливается на физическое начало.

З — необязательные фразы игнорируются.

Оператор CLOSE (ЗАКРЫТЬ) выполняется так, как будто нет необязательных фраз.

(4) Выполнение оператора CLOSE (ЗАКРЫТЬ) приводит к обновлению значения состояния ввода-вывода, связанного с именем-файла-1 (см. п. 1.3.5 настоящей части).

(5) Если отсутствует необязательный входной файл, для файла не выполняется обработка катушки (тома) или конца файла, а указатель позиции файла и указатель текущего тома не изменяются.

(6) После успешного завершения оператора CLOSE (ЗАКРЫТЬ) без фразы REEL (КАТУШКУ) или UNIT (ТОМ) область

GOST EXPERT ГОСТ 22558-89,

записи, связанная с именем-файла-1, становится недоступной. В случае неуспешного выполнения оператора CLOSE (ЗАКРЫТЬ) доступность области записи не определена.

(7) После успешного завершения оператора CLOSE (ЗАКРЫТЬ) без фразы REEL (КАТУШКУ) или UNIT (ТОМ) файл не входит более в число открытых файлов; он более не связан с определителем файла.

(8) Если в операторе CLOSE (ЗАКРЫТЬ) указано более одного имени-файла-1, результат выполнения этого оператора CLOSE (ЗАКРЫТЬ) является таким же, как если бы отдельный оператор CLOSE (ЗАКРЫТЬ) был написан для каждого имени-файла-1 в том же порядке, в каком эти имена файлов указаны в операторе CLOSE (ЗАКРЫТЬ).

4.3. **Оператор OPEN (ОТКРЫТЬ)**

4.2.1. Назначение

Оператор OPEN (ОТКРЫТЬ) подготавливает файл к обработке. Фраза REVERSED (РЕВЕРСНО) является устаревшим элементом в настоящем стандарте и будет удалена в следующей редакции.

4.3.2. Общий формат

```
        ┌ INPUT {имя-файла-1 [ REVERSED          ] }...                    ┐
        │                    [ WITH NO REWIND    ]                        │
        │ OUTPUT {имя-файла-2 [WITH NO REWIND] }...                        │ ...
OPEN    │ I-O {имя-файла-3}...                                             │
        └ [EXTEND {имя-файла-4}...]                                        ┘

        ┌ ВХОДНОЙ {имя-файла-1                                             ┐
        │          [ РЕВЕРСНО     ] }...                                   │
        │          [ БЕЗ ПЕРЕМОТКИ ]                                       │
        │ ВЫХОДНОЙ {имя-файла-2                                            │
ОТКРЫТЬ │          [БЕЗ ПЕРЕМОТКИ] }...                                    │ ...
        │ ВХОДНОЙ-ВЫХОДНОЙ {имя-файла-3}...                                │
        └ [ДОПОЛНЯЕМЫЙ {имя-файла-4}...]                                   ┘
```


GOST EXPERT ГОСТ 22558-89,

4.3.3. Синтаксические правила

(1) Фраза REVERSED (РЕВЕРСНО) может указываться только для последовательных файлов.

(2) Фраза EXTEND (ДОПОЛНЯЕМЫЙ) не может использоваться для файлов, находящихся на одной катушке с другими файлами.

(3) Фраза EXTEND (ДОПОЛНЯЕМЫЙ) может использоваться только для файлов, для которых не была указана фраза LINAGE (ВЕРСТКА).

(4) Файлы, перечисленные в операторе OPEN (ОТКРЫТЬ), могут иметь различную организацию или доступ.

4.3.4. Общие правила

(1) Успешное выполнение оператора OPEN (ОТКРЫТЬ) делает файл доступным для обработки.

Успешное выполнение оператора OPEN (ОТКРЫТЬ) связывает файл с именем-файла посредством определителя файла.

Файл доступен, если он физически имеется в наличии и распознан системой управления вводом-выводом. Приведенная ниже табл. 1. демонстрирует результаты открытия доступных и недоступных файлов.

Таблица 1

Фраза оператора	Файл доступен	Файл недоступен
INPUT (ВХОДНОЙ)	Нормальное открытие	Открытие неуспешное
INPUT (ВХОДНОЙ) (Необязательный файл)	Нормальное открытие	Нормальное открытие; при первом чтении возникает условие конца
I-O (ВХОДНОЙ-ВЫХОДНОЙ)	Нормальное открытие	Открытие неуспешное
I-O (ВХОДНОЙ-ВЫХОДНОЙ (Необязательный файл)	Нормальное открытие	Открытие приводит к созданию файла
OUTPUT (ВЫХОДНОЙ)	Нормальное открытие; файл не содержит записей	Открытие приводит к созданию файла
EXTEND (ДОПОЛНЯЕМЫЙ)	Нормальное открытие	Открытие неуспешное
EXTEND (ДОПОЛНЯЕМЫЙ) (Необязательный файл)	Нормальное открытие	Открытие приводит к созданию файла

(2) Успешное выполнение оператора OPEN (ОТКРЫТЬ) делает область записи доступной программе. Если определитель фай-

GOST EXPERT ГОСТ 22558-89,

ла, связанный с именем файла, является внешним, то существует единственная область записи, связанная с определителем файла для единицы исполнения.

(3) Если файл не открыт, не может быть выполнен ни один оператор, явно или неявно относящийся к файлу, за исключением оператора MERGE (СЛИТЬ) с фразами USING (ИСПОЛЬЗУЯ) и GIVING (ПОЛУЧАЯ), оператора OPEN (ОТКРЫТЬ) или оператора SORT (СОРТИРОВАТЬ) с фразами USING (ИСПОЛЬЗУЯ) и GIVING (ПОЛУЧАЯ).

(4) Оператор OPEN (ОТКРЫТЬ) должен быть успешно выполнен перед выполнением любого другого допустимого оператора ввода-вывода.

В табл. 2 приведены допустимые операторы ввода-вывода.

Таблица 2

Оператор	Допустимость сочетаний операторов ввода-вывода для вариантов оператора OPEN (ОТКРЫТЬ)			
	INPUT (ВХОДНОЙ)	OUTPUT (ВЫХОДНОЙ)	I-O (ВХОДНОЙ-ВЫХОДНОЙ)	EXTEND (ДОПОЛНЯЕМЫЙ)
READ (ЧИТАТЬ)	X	—	X	—
WRITE (ПИСАТЬ)	—	X	—	X
REWRITE (ОБНОВИТЬ)	—	—	X	—

Примечание. Знак «X» означает допустимые сочетания оператора ввода-вывода и варианта OPEN (ОТКРЫТЬ), «—» — недопустимые.

(5) Файл может быть открыт как INPUT (ВХОДНОЙ), OUTPUT (ВЫХОДНОЙ), EXTEND (ДОПОЛНЯЕМЫЙ) и I-O (ВХОДНОЙ-ВЫХОДНОЙ) в одной и той же программе. Каждому последующему оператору OPEN (ОТКРЫТЬ) для одного и того же файла должно предшествовать выполнение оператора CLOSE (ЗАКРЫТЬ) без фраз REEL (КАТУШКУ), UNIT (ТОМ) или LOCK (С ЗАМКОМ) для данного файла.

(6) Выполнение оператора OPEN (ОТКРЫТЬ) не извлекает и не записывает первую запись данных.

(7) Если указаны записи меток для файла, начальные метки обрабатываются следующим образом:

а) когда указана фраза INPUT (ВХОДНОЙ), оператор OPEN (ОТКРЫТЬ) осуществляет проверку меток в соответствии с процедурами, определенными реализацией для проверки входных меток;

GOST EXPERT ГОСТ 22558-89,

б) когда указана фраза OUTPUT (ВЫХОДНОЙ), выполнение оператора OPEN (ОТКРЫТЬ) вызывает запись меток в соответствии с процедурами, определенными реализацией для записи выходных меток.

Действия оператора OPEN (ОТКРЫТЬ) не определены, когда записи меток специфицированы, но в файле отсутствуют, или не специфицированы, но присутствуют.

(8) Если во время выполнения оператора OPEN (ОТКРЫТЬ) возникает условие противоречия свойств файла, выполнение оператора OPEN (ОТКРЫТЬ) считается неуспешным (см. п. 1.3.7 настоящей части).

(9) Фразы NO REWIND (БЕЗ ПЕРЕМОТКИ) и REVERSED (РЕВЕРСНО) должны использоваться:

а) только для последовательных однокатушечных (однотомных) файлов (см. п. 4.2 настоящей части);

б) только для последовательных файлов, целиком содержащихся на одной катушке ленты, в среде многофайловых лент (см. п. 2.11 настоящей части).

(10) Фразы REVERSED (РЕВЕРСНО) и NO REWIND (БЕЗ ПЕРЕМОТКИ) игнорируются, если они не применимы к внешнему носителю, на котором располагается файл.

(11) Если внешний носитель для файла допускает перемотку, то выполняются следующие действия:

а) если ни одна из фраз REVERSED (РЕВЕРСНО), EXTEND (ДОПОЛНЯЕМЫЙ), NO REWIND (БЕЗ ПЕРЕМОТКИ) не указана, при выполнении оператора OPEN (ОТКРЫТЬ) файл устанавливается в начало;

б) если задана фраза NO REWIND (БЕЗ ПЕРЕМОТКИ), при выполнении оператора OPEN (ОТКРЫТЬ) переустановка файла не выполняется; файл уже должен быть установлен в начало перед выполнением оператора OPEN (ОТКРЫТЬ);

в) если задана фраза REVERSED (РЕВЕРСНО), при выполнении оператора OPEN (ОТКРЫТЬ) файл устанавливается в конец.

(12) Если задана фраза REVERSED (РЕВЕРСНО), последующий оператор READ (ЧИТАТЬ) делает записи файла доступными в обратном порядке, начиная с последней записи.

(13) Если файл, открытый с фразой INPUT (ВХОДНОЙ), является необязательным файлом, не имеющимся в наличии, оператор OPEN (ОТКРЫТЬ) устанавливает указатель позиции файла для указания того, что необязательный входной файл отсутствует.

GOST EXPERT ГОСТ 22558-89,

(14) Если файл открыт с фразами INPUT (ВХОДНОЙ) или I-O (ВХОДНОЙ-ВЫХОДНОЙ), указатель позиции файла устанавливается на единицу.

(15) Если задана фраза EXTEND (ДОПОЛНЯЕМЫЙ), при выполнении оператора OPEN (ОТКРЫТЬ) файл устанавливается непосредственно за последней логической записью файла. Последней логической записью последовательного файла является последняя записанная в файл запись.

(16) Если задана фраза EXTEND (ДОПОЛНЯЕМЫЙ) и фраза LABEL RECORD (МЕТКИ) указывает, что записи меток присутствуют, выполнение оператора OPEN (ОТКРЫТЬ) включает следующие действия:

а) начальные метки файла обрабатываются только для одно-катушечных или однотомных файлов;

б) начальные метки катушки (тома) обрабатываются на последней катушке (томе), как если бы файл открывался как INPUT (ВХОДНОЙ);

в) имеющиеся конечные метки файла обрабатываются, как если бы файл открывался как INPUT (ВХОДНОЙ). Затем эти метки удаляются;

г) затем обработка продолжается, как если бы файл был открыт как OUTPUT (ВЫХОДНОЙ).

(17) Оператор OPEN (ОТКРЫТЬ) с фразой I-O (ВХОДНОЙ-ВЫХОДНОЙ) должен относиться к файлу, поддерживающему операции ввода и вывода, допустимые для последовательного файла, открытого как I-O (ВХОДНОЙ-ВЫХОДНОЙ). Выполнение оператора OPEN (ОТКРЫТЬ) с фразой I-O (ВХОДНОЙ-ВЫХОДНОЙ) подготавливает файл, на который он ссылается, как для операций ввода, так и для операций вывода.

(18) Если указана фраза I-O (ВХОДНОЙ-ВЫХОДНОЙ) и во фразе LABEL RECORD (МЕТКИ) указано, что записи меток присутствуют, выполнение оператора OPEN (ОТКРЫТЬ) включает следующие шаги:

а) проверку меток в соответствии с процедурами, определенными реализацией для проверки входных-выходных меток;

б) запись новых меток в соответствии с процедурами, определенными реализацией для записи входных-выходных меток.

(19) Файл, содержащийся в среде многофайловой ленты, логически эквивалентен последовательному файлу, содержащемуся в однофайловой ленточной среде.

(20) Если совокупность файлов размещена на одной катушке ленты и на один из этих файлов ссылаются в операторе OPEN (ОТКРЫТЬ), то применяются следующие правила:

а) одновременно в открытом состоянии может находиться не более одного файла;

GOST EXPERT ГОСТ 22558-89,

б) нет ограничений на порядок, в котором файлы могут быть открыты для ввода;

в) если один из файлов, на который ссылается имя файла, является субъектом оператора OPEN (ОТКРЫТЬ) с фразой OUTPUT (ВЫХОДНОЙ), во время выполнения оператора OPEN (ОТКРЫТЬ) на соответствующей катушке должны уже существовать все файлы, номер позиций которых меньше, чем номер позиции данного файла. Кроме того, в это время на катушке не могут существовать файлы, номер позиции которых больше, чем номер позиции данного файла;

г) каждый файл должен быть последовательным.

(21) Для необязательного файла, являющегося недоступным, успешное выполнение оператора OPEN (ОТКРЫТЬ) с фразами I-O (ВХОДНОЙ-ВЫХОДНОЙ) или EXTEND (ДОПОЛНЯЕМЫЙ) приводит к созданию файла. Это создание происходит так, как если бы в указанном порядке выполнялись следующие операторы:

OPEN OUTPUT имя-файла.
CLOSE имя-файла.
ОТКРЫТЬ ВЫХОДНОЙ имя-файла.
ЗАКРЫТЬ имя-файла.

За этими операторами следует выполнение оператора OPEN (ОТКРЫТЬ), указанного в исходной программе.

Успешное выполнение оператора OPEN (ОТКРЫТЬ) с фразой OUTPUT (ВЫХОДНОЙ) приводит к созданию файла. После успешного создания такой файл не содержит записей данных.

(22) После успешного выполнения оператора OPEN (ОТКРЫТЬ) указатель текущего тома устанавливается на:

а) первую или единственную катушку (том) для доступного входного или входного-выходного файла;

б) катушку (том), содержащую последнюю логическую запись дополняемого файла;

в) новую катушку (том) для недоступного выходного, входного-выходного или дополняемого файла.

(23) Во время выполнения оператора OPEN (ОТКРЫТЬ) обновляется значение состояния ввода-вывода, связанного с именем-файла.

(24) Если в операторе OPEN (ОТКРЫТЬ) указано более чем одно имя-файла, результат выполнения этого оператора OPEN (ОТКРЫТЬ) такой, как если бы отдельный оператор OPEN (ОТКРЫТЬ) был написан для каждого имени-файла в том порядке, как они указаны в операторе OPEN (ОТКРЫТЬ).

(25) Минимальный и максимальный размеры записей файла

GOST EXPERT ГОСТ 22558-89,

устанавливаются во время создания файла и не должны изменяться впоследствии.

4.4. Оператор READ (ЧИТАТЬ)

4.4.1. Назначение

Оператор READ (ЧИТАТЬ) делает доступной следующую логическую запись файла.

4.4.2. Общий формат

READ имя-файла-1 [NEXT] RECORD [INTO идентификатор-1]
[AT END повелительный-оператор-1]
[NOT AT END повелительный-оператор-2]
[END-READ]

ЧИТАТЬ [СЛЕДУЮЩУЮ] ЗАПИСЬ имя-файла-1
[В идентификатор-1]
[В КОНЦЕ повелительный-оператор-1]
[НЕ В КОНЦЕ повелительный-оператор-2]
[КОНЕЦ-ЧИТАТЬ]

4.4.3. Синтаксические правила

(1) Область памяти, связанная с идентификатором-1, и область записи, связанная с именем-файла-1, не должны быть одной и той же областью памяти.

(2) Фраза AT END (В КОНЦЕ) должна быть указана, если для имени-файла-1 не определена соответствующая процедура USE AFTER STANDARD EXCEPTION (ИСПОЛЬЗОВАТЬ ПОСЛЕ СТАНДАРТНОЙ ПРОЦЕДУРЫ ОШИБКИ).

4.4.4. Общие правила

(1) Во время выполнения оператора READ (ЧИТАТЬ) файл, на который ссылается имя-файла-1, должен быть открыт как входной или входной-выходной (см. п. 4.3 настоящей части).

(2) Фраза NEXT (СЛЕДУЮЩУЮ) не обязательна и не влияет на выполнение оператора OPEN (ОТКРЫТЬ).

(3) Выполнение оператора READ (ЧИТАТЬ) вызывает обновление значения состояния ввода-вывода, связанного с именем-файла-1.

(4) В начале выполнения оператора READ (ЧИТАТЬ) установка указателя позиции файла используется для определения доступной записи согласно следующим правилам. Сравнения для записей в последовательных файлах относятся к номерам записей.

GOST EXPERT ГОСТ 22558-89,

а) Если указатель позиции файла указывает, что нет правильной следующей записи, то выполнение оператора READ (ЧИТАТЬ) неуспешно.

б) Если указатель позиции файла указывает, что отсутствует необязательный входной файл, выполнение происходит так, как указано в общем правиле (10).

в) Если указатель позиции файла был установлен предыдущим оператором OPEN (ОТКРЫТЬ), выбирается первая существующая запись файла, номер записи которой больше или равен указателю позиции файла.

г) Если указатель позиции файла установлен предыдущим оператором READ (ЧИТАТЬ), выбирается первая существующая запись файла, номер записи которой больше указателя позиции файла.

Если обнаружена запись, удовлетворяющая приведенным выше правилам, она становится доступной в области записи, соответствующей имени-файла-1.

Если запись, удовлетворяющая приведенным выше правилам, не найдена, то указатель позиции файла устанавливается для указания того, что не существует следующей логической записи, и выполнение продолжается согласно общему правилу (10).

Если запись становится доступной, то указатель позиции файла устанавливается на номер записи, ставшей доступной.

(5) Независимо от используемого способа совмещения времени выборки с временем обработки, концепция оператора READ (ЧИТАТЬ) остается неизменной в том, что запись становится доступной объектной программе до выполнения любого оператора, следующего за этим оператором READ (ЧИТАТЬ), если повелительный-оператор-2 не указан, или до выполнения повелительного-оператора-2, если он указан.

(6) Когда логические записи описаны более чем одной статьей описания записи, эти записи автоматически используют одну и ту же область памяти; это эквивалентно неявному переопределению области. По завершении оператора READ (ЧИТАТЬ) значения всех данных, находящихся вне диапазона текущей записи данных, не определены.

(7) Фраза INTO (В) может быть указана в операторе READ (ЧИТАТЬ), если:

а) только одно описание записи подчиняется статье описания файла;

б) все имена-записей, связанные с именем-файла-1, и данное, на которое ссылается идентификатор-1, описывают групповое данное или элементарное буквенно-цифровое данное.

GOST EXPERT ГОСТ 22558-89,

(8) Результат выполнения оператора READ (ЧИТАТЬ) с фразой INTO (В) эквивалентен применению следующих правил в указанном порядке:

а) выполняется тот же оператор READ (ЧИТАТЬ) без фразы INTO (В);

б) текущая запись пересылается из области записи в область, указываемую идентификатором-2, в соответствии с правилами для оператора MOVE (ПОМЕСТИТЬ) без фразы CORRESPONDING (СООТВЕТСТВЕННО). Размер текущей записи определяется правилами, указанными для фразы RECORD (В ЗАПИСИ). Если статья описания файла содержит фразу RECORD VARYING (В ЗАПИСИ ПЕРЕМЕННОЕ ЧИСЛО ЛИТЕР), неявная пересылка является групповой. Неявный оператор MOVE (ПОМЕСТИТЬ) не выполняется, если выполнение оператора READ (ЧИТАТЬ) было неуспешным. Индексы, связанные с идентификатором-1, вычисляются после того, как запись была прочитана и непосредственно перед ее пересылкой в область идентификатора-1. Запись доступна в области записи и в данном, на которое ссылается идентификатор-1.

(9) Если при выполнении оператора READ (ЧИТАТЬ) обнаруживается конец катушки или тома или катушка (том) не содержит логических записей, а логический конец файла не достигнут, выполняются следующие операции:

а) стандартная процедура конечных меток катушки (тома);

б) смена катушек (томов). Значение указателя тома указывает на следующую существующую для файла катушку (том);

в) стандартная процедура начальных меток катушки (тома).

(10) Если указатель позиции файла указывает, что не существует следующей логической записи или что нет в наличии необязательного входного файла, выполняются следующие действия в указанном порядке:

а) значение, полученное в результате установки указателя позиции файла, присваивается состоянию ввода-вывода, связанному с именем-файла-1, для обозначения условия конца (см. п. 1.3.5 настоящей части);

б) если фраза AT END (В КОНЦЕ) указана в операторе, вызвавшем это условие, управление передается повелительному-оператору-1 во фразе AT END (В КОНЦЕ). Никакие процедуры USE AFTER STANDARD EXCEPTION (ИСПОЛЬЗОВАТЬ ПОСЛЕ СТАНДАРТНОЙ ПРОЦЕДУРЫ ОШИБКИ), связанные с именем-файла-1, не выполняются;

в) если фраза AT END (В КОНЦЕ) не указана, с именем-файла-1 должна быть связана процедура USE AFTER STANDARD EXCEPTION (ИСПОЛЬЗОВАТЬ ПОСЛЕ СТАНДАРТНОЙ ПРО-

GOST EXPERT ГОСТ 22558-89,

ЦЕДУРЫ ОШИБКИ), которая в этом случае и выполняется. Возврат из этой процедуры осуществляется к следующему после оператора READ (ЧИТАТЬ) выполнимому оператору.

Если возникает условие конца, выполнение оператора READ (ЧИТАТЬ) является неуспешным.

(11) Если во время выполнения оператора READ (ЧИТАТЬ) условие конца не возникает, фраза AT END (В КОНЦЕ) игнорируется, если она указана, и выполняются следующие действия:

а) устанавливается значение указателя позиции файла и изменяется значение состояния ввода-вывода, связанного с именем-файла-1;

б) если существует условие особой ситуации, но не условие конца, управление передается процедуре USE AFTER EXCEPTION (ИСПОЛЬЗОВАТЬ ПОСЛЕ ПРОЦЕДУРЫ ОШИБКИ) согласно правилам оператора USE (ИСПОЛЬЗОВАТЬ), применимого к имени-файла-1 (п. 4.6 настоящей части);

в) если условие особой ситуации не существует, запись становится доступной в области записи и выполняются любые неявные пересылки, предопределенные фразой INTO (В). Управление передается в точку выхода оператора READ(ЧИТАТЬ) или повелительному-оператору-2, если он указан. В последнем случае выполнение продолжается согласно правилам для операторов, указанных в повелительном-операторе-2. Если выполняется оператор ветвления или условный оператор, вызывающие явную передачу управления, оно передается в соответствии с правилами для этих операторов, в противном случае после завершения выполнения повелительного-оператора-2 управление передается в точку выхода оператора READ (ЧИТАТЬ).

(12) После неуспешного завершения выполнения оператора READ (ЧИТАТЬ) содержимое соответствующей области записи не определено; указателю позиции файла присвоено значение, указывающее, что правильная следующая запись не установлена.

(13) Если число позиций литер прочитанной записи меньше чем минимальный размер, указанный в статьях описания записи для имени-файла-1, то часть области записи, находящаяся справа от последней прочитанной литеры, является неопределенной. Если число позиций литер в читаемой записи больше, чем максимальный размер, указанный в статьях описания записи для имени-файла-1, то запись усекается справа до максимального размера. В любом из этих случаев оператор READ (ЧИТАТЬ) выполняется успешно и состояние ввода-вывода устанавливается для указания, что произошло нарушение длины записи (см. п. 1.3 настоящей части).

(14) Фраза END-READ (КОНЕЦ-ЧИТАТЬ) ограничивает область действия оператора READ (ЧИТАТЬ) (см. ч. 4, п. 4.6.3).

GOST EXPERT ГОСТ 22558-89,

4.5. Оператор REWRITE (ОБНОВИТЬ)

4.5.1. Назначение

Оператор REWRITE (ОБНОВИТЬ) логически заменяет запись в файле на устройстве массовой памяти.

4.5.2. Общий формат

REWRITE имя-записи-1 [FROM идентификатор-1]

ОБНОВИТЬ имя-записи-1 [ИЗ ПОЛЯ идентификатор-1]

4.5.3. Синтаксические правила

(1) Имя-записи-1 и идентификатор-1 не должны относиться к одной и той же области памяти.

(2) Имя-записи-1 — это имя логической записи в секции файлов раздела данных. Оно может быть уточнено.

4.5.4. Общие правила

(1) Во время выполнения этого оператора файл, связанный с именем-записи-1, должен быть файлом массовой памяти и должен быть открыт как входной-выходной (см. п. 4.3 настоящей части).

(2) Последним оператором ввода-вывода для соответствующего файла перед выполнением оператора REWRITE (ОБНОВИТЬ) должен быть успешно выполненный оператор READ (ЧИТАТЬ). Система управления массовой памятью (СУМП) логически заменяет запись, которая была извлечена оператором READ (ЧИТАТЬ).

(3) Логическая запись, включенная в файл при успешном выполнении оператора REWRITE (ОБНОВИТЬ), становится недоступной в области записи, за исключением случая, когда имя-файла, связанное с именем-записи-1, описано во фразе SAME RECORD AREA (ОБЩАЯ ОБЛАСТЬ ЗАПИСИ). Логическая запись доступна программе и как запись файла, связанного с именем-записи-1, и как запись других файлов, указанных в той же фразе SAME RECORD AREA (ОБЩАЯ ОБЛАСТЬ ЗАПИСИ), что и соответствующий выходной файл.

(4) Выполнение оператора REWRITE (ОБНОВИТЬ) с фразой FROM (ИЗ ПОЛЯ) эквивалентно выполнению следующих операторов в указанном порядке:

а) Оператор

MOVE идентификатор-1 TO имя-записи-1

(ПОМЕСТИТЬ идентификатор-1 В имя-записи-1) соответственно правилам, описанным для оператора MOVE (ПОМЕСТИТЬ);

б) тот же оператор REWRITE (ОБНОВИТЬ) без фразы FROM (ИЗ ПОЛЯ).

(5) После завершения выполнения оператора REWRITE (ОБНОВИТЬ) информация в области, указанной идентификатором-1, остается доступной, даже если информация в области, указанной именем-записи-1, не является доступной кроме случая, опреде-

GOST EXPERT ГОСТ 22558-89,

ленного фразой SAME RECORD AREA (ОБЩАЯ ОБЛАСТЬ ЗАПИСИ).

(6) Выполнение оператора REWRITE (ОБНОВИТЬ) не влияет на указатель позиции файла.

(7) Выполнение оператора REWRITE (ОБНОВИТЬ) вызывает обновление состояния ввода-вывода для файла, связанного с именем-записи-1 (см. п. 1.3.5 настоящей части).

(8) При выполнении оператора REWRITE (ОБНОВИТЬ) логическая запись передается операционной системе.

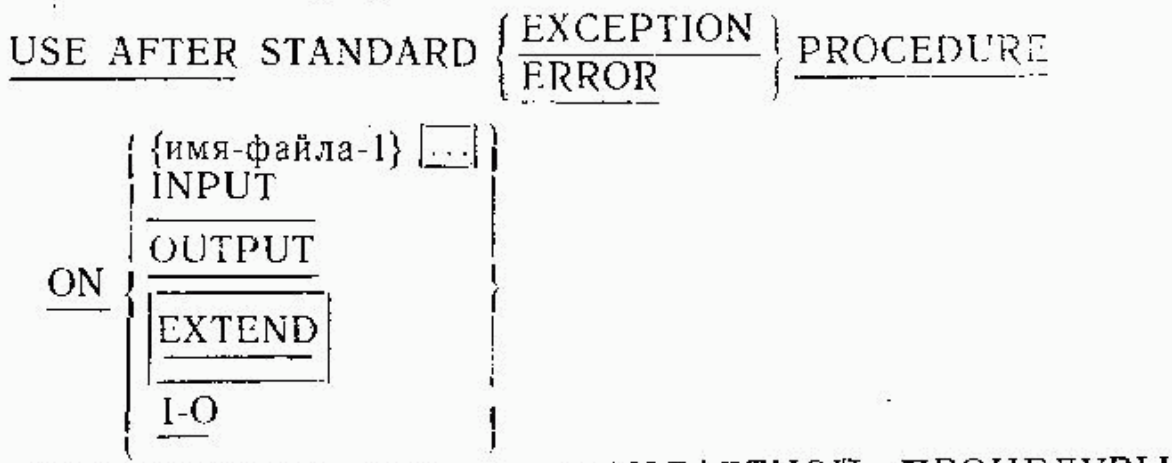
(9) Если число позиций литер, описанное в записи, указанной именем-записи-1, не равно числу позиций литер в заменяемой записи, то выполнение оператора REWRITE (ОБНОВИТЬ) будет неуспешным, операции обновления не произойдет, содержимое области записи не изменится и состояние ввода-вывода файла, связанного с именем-записи-1, примет значение, указывающее причину условия (особого состояния) (см. п. 1.3.5 настоящей части).

4.6. **Оператор USE (ИСПОЛЬЗОВАТЬ)**

4.6.1. Назначение

Оператор USE (ИСПОЛЬЗОВАТЬ) определяет процедуры обработки ошибок ввода-вывода дополнительно к стандартным процедурам, предоставляемым системой управления вводом-выводом.

4.6.2. Общий формат



GOST EXPERT ГОСТ 22558-89,

4.6.3. Синтаксические правила

(1) Оператор USE (ИСПОЛЬЗОВАТЬ) должен непосредственно следовать за заголовком секции декларативной части раздела процедур и должен быть единственным в предложении. Остальная часть декларативной секции должна состоять из одного или более процедурных параграфов, определяющих процедуры, которые должны использоваться.

(2) Сам оператор USE (ИСПОЛЬЗОВАТЬ) никогда не выполняется; он только определяет условия, вызывающие выполнение указанных после него процедур.

(3) Появление имени-файла-1 в операторе USE (ИСПОЛЬЗОВАТЬ) не должно требовать одновременного выполнения более чем одной декларативной секции.

(4) Слова ERROR и EXCEPTION обозначают особую ситуацию и являются синонимами.

(5) Файлы, к которым явно или неявно обращаются в операторе USE (ИСПОЛЬЗОВАТЬ), могут иметь различную организацию или доступ.

(6) Каждая из фраз INPUT (ВХОДНЫХ), OUTPUT (ВЫХОДНЫХ), I-O (ВХОДНЫХ-ВЫХОДНЫХ) и EXTEND (ДОПОЛНЯЕМЫХ) может указываться лишь раз в декларативной части раздела процедур.

4.6.4. Общие правила

(1) Декларативные процедуры могут быть включены в любую исходную Кобол-программу, независимо от того, содержит ли эта программа другую программу или сама содержится в другой программе. Декларатива вызывается тогда, когда во время выполнения программы выполняются условия, описанные в операторе USE (ИСПОЛЬЗОВАТЬ), предшествующем декларативе. Только одна декларатива внутри отдельно скомпилированной программы, содержащей оператор, который вызвал уточняющее условие, вызывается тогда, когда выполняется какое-либо из условий, описанных в операторе USE (ИСПОЛЬЗОВАТЬ), предшествующем декларативе, во время выполнения программы. Если не существует уточняющей декларативы в отдельно скомпилированной программе, то декларатива не выполняется.

(2) Внутри декларативной процедуры не должно быть обращений к каким-либо процедурам вне декларативной части раздела процедур.

(3) К именам процедур, связанных с оператором USE (ИСПОЛЬЗОВАТЬ), могут быть обращения в другой декларативной секции или в недекларативной процедуре только оператором PERFORM (ВЫПОЛНИТЬ).

GOST EXPERT ГОСТ 22558-89,

(4) Когда имя-файла-1 описано явно, то к имени-файла-1 не применяется никаких других операторов USE (ИСПОЛЬЗОВАТЬ).

(5) Процедуры, связанные с оператором USE (ИСПОЛЬЗОВАТЬ), выполняются системой управления вводом-выводом после завершения стандартной программы ошибки ввода-вывода при неудачном выполнении операции ввода-вывода, если только не сработает фраза AT END (В КОНЦЕ). При выполнении процедур соблюдаются следующие правила:

а) если указано имя-файла-1, то соответствующая процедура выполняется при выполнении условия, описанного в операторе USE (ИСПОЛЬЗОВАТЬ);

б) если указано INPUT (ВХОДНЫХ), то соответствующая процедура выполняется при выполнении условия, описанного в операторе USE (ИСПОЛЬЗОВАТЬ), для какого-либо файла, открытого для ввода, или же в процессе открытия для ввода, за исключением файлов, указываемых именем-файла-1 в другом операторе USE (ИСПОЛЬЗОВАТЬ), описывающем такое же условие;

в) если указано OUTPUT (ВЫХОДНЫХ), то соответствующая процедура выполняется при выполнении условия, описанного в операторе USE (ИСПОЛЬЗОВАТЬ), для какого-либо файла, открытого для вывода, или же в процессе открытия для вывода, за исключением файлов, указываемых именем-файла-1 в другом операторе USE (ИСПОЛЬЗОВАТЬ), описывающем такое же условие.

г) если указано I-O (ВХОДНЫХ-ВЫХОДНЫХ), то соответствующая процедура выполняется при выполнении условия, описанного в операторе USE (ИСПОЛЬЗОВАТЬ), для какого-либо файла, открытого для ввода-вывода или в процессе открытия для ввода-вывода, за исключением файлов, указанных именем-файла-1 в другом операторе USE (ИСПОЛЬЗОВАТЬ), описывающем такое же условие;

д) если указано EXTEND (ДОПОЛНЯЕМЫХ), то соответствующая процедура выполняется при выполнении условия, описанного в операторе USE (ИСПОЛЬЗОВАТЬ), для какого-либо файла, открытого для дополнения или процессе открытия для дополнения, за исключением файлов, указанных именем-файла-1 в другом операторе USE (ИСПОЛЬЗОВАТЬ), описывающем такое же условие.

(6) После выполнения процедуры, связанной с оператором USE (ИСПОЛЬЗОВАТЬ), управление передается вызывающей программе в системе управления вводом-выводом. Если значение состояния ввода-вывода не указывает на критическую ошибку ввода-вывода, то система управления вводом-выводом возвращает управление оператору, следующему за оператором ввода-вывода, выполнение которого вызвало ошибку. Если значение состояния ввo-

GOST EXPERT ГОСТ 22558-89,

да-вывода указывает на критическую ошибку, то действие определяется реализацией.

(7) В процедуре, связанной с оператором USE (ИСПОЛЬЗОВАТЬ), не должны выполняться никакие операторы, которые могут потребовать выполнения процедуры, связанной с другим оператором USE (ИСПОЛЬЗОВАТЬ), вызванной ранее и еще не вернувшей управление вызвавшей ее программе.

4.7. **Оператор WRITE (ПИСАТЬ)**

4.7.1. Назначение

Оператор WRITE (ПИСАТЬ) включает логическую запись в выходной файл. Он используется также для вертикального позиционирования строк на логической странице.

WRITE имя-записи-1 [FROM идентификатор-1]

[{ BEFORE / AFTER } ADVANCING { { идентификатор-2 / целое-1 } LINE / LINES | { мнемоническое-имя-1 / PAGE } }]

[AT { END-OF-PAGE / EOP } повелительный оператор-1]

[NOT AT { END-OF-PAGE / EOP } повелительный оператор-2]

[END-WRITE]

ПИСАТЬ имя-записи-1 [ИЗ ПОЛЯ идентификатор-1]

[{ ПОСЛЕ / ДО } ПРОДВИЖЕНИЯ { { идентификатор-2 / целое-1 } СТРОК | { мнемоническое-имя-1 / СТРАНИЦЫ } }]

[В КОНЦЕ СТРАНИЦЫ повелительный-оператор-1]

[НЕ В КОНЦЕ СТРАНИЦЫ повелительный-оператор-2]

[КОНЕЦ-ПИСАТЬ]

GOST EXPERT ГОСТ 22558-89,

4.7.3. Синтаксические правила

(1) Имя-записи-1 и идентификатор-1 не должны относиться к одной и той же области памяти.

(2) Имя-записи-1 является именем логической записи в секции файлов раздела данных и может быть уточнено.

(3) Фраза ADVINCING (ПРОДВИЖЕНИЯ) мнемоническос-имя-1 не может быть указана для файла, который связан со статьей описания файла, содержащей фразу LINAGE (ВЕРСТКА).

(4) Идентификатор-2 должен относиться к элементарному целому данному.

(5) Целое-1 может быть положительным числом или нулем, но не должно быть отрицательным.

(6) Когда специфицируется мнемоническое-имя-1, то имя связывается с особым свойством, описанным реализацией. Мнемоническое-имя-1 определяется в параграфе SPECIAL-NAMES (СПЕЦИАЛЬНЫЕ-ИМЕНА) раздела оборудования.

(7) Фразы ADVANCING (ПРОДВИЖЕНИЯ СТРАНИЦЫ) и END-OF-PAGE (В КОНЦЕ СТРАНИЦЫ) не должны быть специфицированы в одном операторе WRITE (ПИСАТЬ).

(8) Если указана фраза END-OF-PAGE (В КОНЦЕ СТРАНИЦЫ) или NOT END-OF-PAGE (НЕ В КОНЦЕ СТРАНИЦЫ), то фраза LINAGE (ВЕРСТКА) должна быть указана в статье описания соответствующего файла.

(9) Слова END-OF-PAGE и EOP являются эквивалентами.

4.7.4. Общие правила

(1) Файл, указанный именем-файла, связанным с именем-записи-1, должен быть открыт для вывода или дополнения ко времени выполнения этого оператора (см. п. 4.3.1 настоящей части).

(2) Логическая запись, включаемая в файл при успешном выполнении оператора WRITE (ПИСАТЬ), становится недоступной в области записи. Исключение представляют случаи, когда имя файла, связанное с именем-записи-1, указано в фразе SAME RECORD AREA (ОБЩАЯ ОБЛАСТЬ ЗАПИСИ). Если имя относится к имени-файла, указанному во фразе SAME RECORD AREA (ОБЩАЯ ОБЛАСТЬ ЗАПИСИ), логическая запись доступна программе и как запись файла, связанного в именем-записи-1, и как запись других файлов, указанных в той же фразе SAME RECORD AREA (ОБЩАЯ ОБЛАСТЬ ЗАПИСИ), что и соответствующий выходной файл.

GOST EXPERT ГОСТ 22558-89,

ЯЗЫК ПРОГРАММИРОВАНИЯ КОБОЛ

ГОСТ 22558—89
(СТ СЭВ 6184—88, ИСО 1989—85)

Части 1—7

Редактор *В. П. Огурцов*
Технический редактор *О. Н. Никитина*
Корректор *Н. Д. Чехотина*

Сдано в наб. 30.01.90 Подп. в печ. 29.04.91 28,0 усл. п. л. 28,25 усл. кр.-отт. 29,60 уч.-изд. л.
Тир. 11000 Цена 5 р. 90 к.

Ордена «Знак Почета» Издательство стандартов, 123557, Москва, ГСП,
Новопресненский пер., 3.
Калужская типография стандартов, ул. Московская, 256. Зак. 254

ГОСУДАРСТВЕННЫЙ СТАНДАРТ
СОЮЗА ССР

ЯЗЫК ПРОГРАММИРОВАНИЯ КОБОЛ

ГОСТ 22558—89
(СТ СЭВ 6184—88, ИСО 1989—85)

ЧАСТИ 8—17

Издание официальное

10 р. 70 к.

КОМИТЕТ СТАНДАРТИЗАЦИИ И МЕТРОЛОГИИ СССР
Москва

GOST EXPERT ГОСТ 22558-89,

(3) Результаты выполнения оператора WRITE (ПИСАТЬ) с фразой FROM (ИЗ ПОЛЯ) эквивалентны выполнению следующих операторов в указанном порядке:

а) оператор MOVE идентификатор-1 TO имя-записи-1 (ПОМЕСТИТЬ идентификатор-1 В имя-записи-1) соответственно правилам, специфицированным в операторе MOVE (ПОМЕСТИТЬ);

б) тот же оператор WRITE (ПИСАТЬ) без фразы FROM (ИЗ ПОЛЯ).

(4) После завершения выполнения оператора WRITE (ПИСАТЬ) информация в области, указанной идентификатором-1, остается доступной, даже если информация в области, указанной именем-записи-1, не является доступной, за исключением случая, определяемого фразой SAME RECORD AREA (ОБЩАЯ ОБЛАСТЬ ЗАПИСИ).

(5) Выполнение оператора WRITE (ПИСАТЬ) не влияет на указатель позиции файла.

(6) Выполнение оператора WRITE (ПИСАТЬ) вызывает обновление состояния ввода-вывода имени-файла, связанного с именем-записи-1 (см. п. 1.3.5 настоящей части).

(7) При выполнении оператора WRITE (ПИСАТЬ) логическая запись передается операционной системе.

(8) Количество позиций литер в записи, указанной именем-записи-1, не должно быть больше наибольшего или меньше наименьшего числа-литер, допустимого фразой RECORD IS VARYING (В ЗАПИСИ ПЕРЕМЕННОЕ ЧИСЛО), связанной с именем-файла, связанного с именем-записи-1. В любом случае выполнение оператора WRITE (ПИСАТЬ) неуспешно, операция записи не происходит, содержимое области записи не меняется, и состояние ввода-вывода файла, связанного с именем-записи-1, принимает значение, указывающее на причину условия (см. п. 1.3.5 настоящей части).

(9) Если во время выполнения оператора WRITE (ПИСАТЬ) с фразой NOT END-OF-PAGE (НЕ В КОНЦЕ СТРАНИЦЫ) не наступает условие конца страницы, то управление передается повелительному-оператору-2 следующим образом:

а) если выполнение оператора WRITE (ПИСАТЬ) успешно, то управление передается после того, как запись записана, и после изменения состояния ввода-вывода имени-файла, связанного с именем-записи-1;

б) если выполнение оператора WRITE (ПИСАТЬ) закончилось неуспешно, то управление передается после изменения состояния ввода-вывода имени-файла, связанного с именем-записи-1, и после выполнения какой-либо процедуры, специфициро-

GOST EXPERT ГОСТ 22558-89,

ванной оператором USE AFTER STANDARD EXCEPTION (ИСПОЛЬЗОВАТЬ ПОСЛЕ СТАНДАРТНОЙ ПРОЦЕДУРЫ ОШИБКИ), применимым к имени-файла, связанного с именем-записи-1.

(10) Фраза END-WRITE (КОНЕЦ-ПИСАТЬ) ограничивает область действия оператора WRITE (ПИСАТЬ) (см. ч. 4, п. 6.4.3).

(11) Отношение следования последовательного файла устанавливается порядком выполнения операторов WRITE (ПИСАТЬ) при создании файла. Отношение не изменяется, за исключением случая, когда записи добавляются в конец файла.

(12) Когда последовательный файл открыт как EXTEND (ДОПОЛНЯЕМЫЙ), то в результате выполнения оператора WRITE (ПИСАТЬ) записи будут добавляться в конец файла так, как если бы файл был открыт как OUTPUT (ВЫХОДНОЙ). Если в файле есть записи, то первая запись, записанная после выполнения оператора OPEN (ОТКРЫТЬ) с фразой EXTEND (ДОПОЛНЯЕМЫЙ), является следующей после последней записи в файле.

(13) Когда делается попытка записать запись за внешне определенными границами последовательного файла, то возникает условие ошибки и содержимое области записи остается неизменным. Происходят следующие действия:

а) значение состояния ввода-вывода для имени-файла, связанного с именем-записи-1, устанавливается в значение, указывающее на нарушение границ;

б) если для имени-файла, связанного с именем-записи-1, явно или неявно специфицирована декларатива USE AFTER STANDARD EXCEPTION (ИСПОЛЬЗОВАТЬ ПОСЛЕ СТАНДАРТНОЙ ПРОЦЕДУРЫ ОШИБКИ), то будет выполняться эта декларативная процедура;

в) если декларатива USE AFTER STANDARD EXCEPTION (ИСПОЛЬЗОВАТЬ ПОСЛЕ СТАНДАРТНОЙ ПРОЦЕДУРЫ ОШИБКИ) не специфицирована явно или неявно для имени файла, связанного с именем-записи-1, то результат будет неопределенным.

(14) Если распознается конец катушки/тома и внешне определенные границы файла не превышены, то выполняются следующие операции:

а) стандартная процедура конечных меток катушки/тома;

б) смена катушки/тома. Указатель текущего тома изменяется для указания на следующую катушку/том, существующую для файла;

GOST EXPERT ГОСТ 22558-89,

в) стандартная процедура начальных меток катушки/тома.

(15) Фразы ADVANCING (ПРОДВИЖЕНИЯ) и END-OF-PAGE (В КОНЦЕ СТРАНИЦЫ) позволяют управлять вертикальным позиционированием строки на печатаемой странице. Если фраза ADVANCING (ПРОДВИЖЕНИЯ) не используется, реализацией будет обеспечиваться автоматическое продвижение, как если бы пользователь указал фразу AFTER ADVANCING 1 LINE (ПОСЛЕ ПРОДВИЖЕНИЯ 1 СТРОК). Если фраза ADVANCING (ПРОДВИЖЕНИЯ) используется, продвижение обеспечивается следующим образом:

а) если целое-1 или данное, указанное идентификатором-2, положительно, то печатаемая страница продвигается на число строк, равное этому значению;

б) если значение данного, указанного идентификатором-2, отрицательно, то результаты будут неопределенными;

в) если целое-1 или значение данного, указанного идентификатором-2, равно нулю, то перемещение печатаемой страницы не происходит;

г) если указано мнемоническое-имя-1, печатаемая страница продвигается в соответствии с правилами, установленными реализацией для данного устройства;

д) если используется фраза BEFORE (ДО), строка выводится до продвижения печатаемой страницы в соответствии с приведенными выше правилами;

е) если используется фраза AFTER (ПОСЛЕ), строка выводится после продвижения печатаемой страницы в соответствии с приведенными выше правилами;

ж) если во фразе ADVANCING (ПРОДВИЖЕНИЯ) указано слово PAGE (СТРАНИЦЫ) и в статье описания файла, связанного с выводимой записью, указана фраза LINAGE (ВЕРСТКА), то запись выводится до или после (в зависимости от используемой фразы) позиционирования устройства на следующую логическую страницу. Позиционирование устройства производится на первую строку, которая может быть записана на следующей логической странице в соответствии со спецификацией фразы LINAGE (ВЕРСТКА);

з) если во фразе ADVANCING (ПРОДВИЖЕНИЯ) указано слово PAGE (СТРАНИЦЫ) и в статье описания файла, указанного с выводимой записью, не указана фраза LINAGE (ВЕРСТКА), запись выводится до или после (в зависимости от используемой фразы) позиционирования устройства на следующую физическую страницу. Позиционирование на следующую физическую

GOST EXPERT ГОСТ 22558-89,

страницу выполняется в соответствии с правилами, определенными реализацией. Если понятие физической страницы не применимо к указанному устройству, реализацией обеспечивается такое продвижение, как если бы пользователь указал фразу BEFORE (ДО) или AFTER ADVANCING 1 LINE (ПОСЛЕ ПРОДВИЖЕНИЯ 1 СТРОК).

(16) Если во время выполнения оператора WRITE (ПИСАТЬ) с фразой END-OF-PAGE (В КОНЦЕ СТРАНИЦЫ) достигается логический конец печатаемой страницы, выполняется повелительный-оператор-1, указанный во фразе END-OF-PAGE (В КОНЦЕ СТРАНИЦЫ). Логический конец специфицируется фразой LINAGE (ВЕРСТКА), связанной с именем-записи.

(17) Условие конца страницы возникает всякий раз, когда выполнение оператора WRITE (ПИСАТЬ) с фразой END-OF-PAGE (В КОНЦЕ СТРАНИЦЫ) вызывает печать или протяжку в области концовки тела страницы. Это происходит тогда, когда при выполнении оператора WRITE (ПИСАТЬ) LINAGE-COUNTER (СЧЕТЧИК-ВЕРСТКИ) становится равным или большим, чем значение целого-2 или данного, представленного именем-данного-2 во фразе LINAGE (ВЕРСТКА), если они указаны. В этом случае выполняется оператор WRITE (ПИСАТЬ), а затем повелительный-оператор-1, указанный во фразе END-OF-PAGE (В КОНЦЕ СТРАНИЦЫ).

Всякий раз, когда выполнение оператора WRITE (ПИСАТЬ) (с фразой END-OF-PAGE (В КОНЦЕ СТРАНИЦЫ) или без нее) не может быть удовлетворено полностью в границах тела текущей страницы, возникает условие переполнения страницы. Это происходит тогда, когда оператор WRITE (ПИСАТЬ), если бы он был выполнен, вызвал бы установку LINAGE-COUNTER (СЧЕТЧИК-ВЕРСТКИ) на значение, превышающее значение целого-1 или данного, представленного именем-данного-1, во фразе LINAGE (ВЕРСТКА). В этом случае запись выводится на логической странице до или после (в зависимости от используемой фразы) позиционирования устройства на первую строку, которая может быть записана на следующей логической странице, в соответствии со спецификацией фразы LINAGE (ВЕРСТКА). После вывода записи и позиционирования устройства выполняется повелительный-оператор-1 фразы END-OF-PAGE (В КОНЦЕ СТРАНИЦЫ), если она указана.

Условие переполнения страницы возникает в том случае, когда в результате выполнения данного оператора WRITE (ПИСАТЬ) значение LINAGE-COUNTER (СЧЕТЧИК-ВЕРСТКИ) одновременно превысит значение целого-2 или данного, указанного именем-данного-2 фразы LINAGE (ВЕРСТКА), и целого-1 или данного, указанного именем-данного-1 фразы LINAGE (ВЕРСТКА).

GOST EXPERT ГОСТ 22558-89,

Часть 8. МОДУЛЬ ОТНОСИТЕЛЬНОГО ВВОДА-ВЫВОДА

1. ВВЕДЕНИЕ В МОДУЛЬ ОТНОСИТЕЛЬНОГО ВВОДА-ВЫВОДА

1.1. Назначение

Модуль относительного ввода-вывода обеспечивает возможность произвольного или последовательного доступа к записям файла массовой памяти. Каждая запись относительного файла однозначно идентифицируется целой положительной величиной, указывающей относительную позицию логической записи в файле.

1.2. Характеристика уровней

Уровень 1 относительного ввода—вывода обеспечивает ограниченные возможности для статьи управления файлом, статьи описания файла и статей в параграфе I-O-CONTROL (УПРАВЛЕНИЕ-ВВОДОМ-ВЫВОДОМ). В разделе процедур уровень 1 относительного ввода-вывода обеспечивает ограниченные возможности операторов CLOSE (ЗАКРЫТЬ), OPEN (ОТКРЫТЬ), READ (ЧИТАТЬ), REWRITE (ОБНОВИТЬ), USE (ИСПОЛЬЗОВАТЬ), WRITE (ПИСАТЬ) и полные возможности оператора DELETE (УДАЛИТЬ).

Уровень 2 относительного ввода-вывода обеспечивает полные возможности для статьи управления файлом и статей в параграфе I-O-CONTROL (УПРАВЛЕНИЕ-ВВОДОМ-ВЫВОДОМ). В разделе процедур уровень 2 относительного ввода-вывода обеспечивает полные возможности операторов CLOSE (ЗАКРЫТЬ), DELETE (УДАЛИТЬ), OPEN (ОТКРЫТЬ), READ (ЧИТАТЬ), REWRITE (ОБНОВИТЬ), START (ПОДВЕСТИ), USE (ИСПОЛЬЗОВАТЬ) и WRITE (ПИСАТЬ).

1.3. Понятия языка

1.3.1. Организация

Файл с относительной организацией является файлом массовой памяти, любая запись которого может быть запомнена или извлечена по значению относительного номера записи.

Концептуально файл с относительной организацией состоит из последовательной цепочки областей, каждая из которых может содержать логическую запись. Каждой из этих областей назначается относительный номер записи. Каждая логическая запись относительного файла идентифицируется относительным номером записи ее области памяти. Например, десятая запись — это запись, адресуемая относительным номером записи 10 и находящаяся в десятой области записи, независимо от того, были заполнены первые девять областей или нет.

Для получения более эффективного доступа к записям в относительном файле число позиций литер, резервируемое в запомина-

GOST EXPERT ГОСТ 22558-89,

ющей среде для хранения логической записи, может отличаться от числа позиций литер в описании этой записи в программе.

1.3.2. Методы доступа

При относительной организации последовательный доступ осуществляется в возрастающем порядке значений относительного номера записи. Доступными являются только записи, находящиеся в данное время в файле. Для установки исходной точки для последовательного извлечения записей может быть использован оператор START (ПОДВЕСТИ).

При произвольном доступе к файлу операторы ввода-вывода применяются для доступа к записям в порядке, указанном программистом. В случае файла с относительной организацией программист указывает требуемую запись, помещая ее относительный номер в данное, определенное как относительный ключ.

При динамическом доступе программист может произвольно переходить от последовательного доступа к произвольному и наоборот, применяя соответствующие формы операторов ввода-вывода.

1.3.3. Указатель позиции файла

Указатель позиции файла — это логическое понятие, используемое в этом документе для облегчения точной спецификации следующей записи, к которой должен осуществляться доступ при выполнении заданных операций ввода-вывода. На установку указателя позиции файла влияют только операторы CLOSE (ЗАКРЫТЬ), OPEN (ОТКРЫТЬ), READ (ЧИТАТЬ) и START (ПОДВЕСТИ). Понятие указателя позиции файла не имеет смысла для файла, открытого для вывода или дополнения.

1.3.4. Состояние ввода-вывода

Состояние ввода-вывода — это логическое понятие, характеризующееся двухсимвольным значением, которое устанавливается для указания состояния операции ввода-вывода во время выполнения операторов CLOSE (ЗАКРЫТЬ), DELETE (УДАЛИТЬ), OPEN (ОТКРЫТЬ), READ (ЧИТАТЬ), REWRITE (ОБНОВИТЬ), START (ПОДВЕСТИ) или WRITE (ПИСАТЬ) перед выполнением любого повелительного оператора, связанного с этим оператором ввода-вывода, и перед выполнением любой применимой процедуры USE AFTER EXCEPTION (ИСПОЛЬЗОВАТЬ ПОСЛЕ СТАНДАРТНОЙ ПРОЦЕДУРЫ ОШИБКИ). Значение состояния ввода-вывода доступно Кобол-программе посредством фразы FILE STATUS (СОСТОЯНИЕ ФАЙЛА) в статье управления файлом.

GOST EXPERT ГОСТ 22558-89,

Состояние ввода-вывода определяет также, будет ли выполняться процедура USE AFTER STANDARD EXCEPTION (ИСПОЛЬЗОВАТЬ ПОСЛЕ СТАНДАРТНОЙ ПРОЦЕДУРЫ ОШИБКИ). Если возникает любое условие, отличное от тех, которые определены ниже как «успешное завершение», может выполняться указанная процедура по правилам, заданным для оператора USE (ИСПОЛЬЗОВАТЬ). Если возникает одно из условий «успешное завершение», никакая процедура такого типа не будет выполняться (п. 4.8. настоящей части).

Некоторые классы значений состояния ввода-вывода задают критические условия ошибки. К таким значениям относятся значения, которые начинаются с цифры 3 или 4, а также значения, начинающиеся с цифры 9, которые определяются реализацией как критические. Если значение состояния ввода—вывода для операции ввода—вывода задает такое условие ошибки, реализацией определяются действия, которые предпринимаются после выполнения применимой процедуры USE AFTER STANDARD EXCEPTION (ИСПОЛЬЗОВАТЬ ПОСЛЕ СТАНДАРТНОЙ ПРОЦЕДУРЫ ОШИБКИ), или, если ни одна такая процедура не применима, после завершения стандартной системной процедуры обработки ошибок ввода-вывода. Состояние ввода-вывода задает одно из следующих условий, возникающих после завершения операции ввода-вывода:

(1) успешное завершение. Оператор ввода-вывода выполнился успешно;

(2) в конце. Оператор последовательного чтения был выполнен неуспешно из-за того, что возникло условие конца файла;

(3) ошибка ключа. Оператор ввода-вывода выполнен неуспешно из-за того, что возникло условие ошибки ключа;

(4) постоянная ошибка. Оператор ввода-вывода выполнен неуспешно в результате ошибки, которая исключает дальнейшую обработку файла. Выполняются все заданные процедуры обработки ошибочных ситуаций. Условие постоянной ошибки остается действующим на все последующие операции ввода-вывода файла до тех пор, пока не будут вызваны определенные реализацией средства для устранения условия постоянной ошибки;

(5) логическая ошибка. Оператор ввода-вывода выполнен неуспешно из-за недопустимой последовательности операций ввода-вывода, выполняемых над файлом, или в результате нарушения ограничений, заданных пользователем.

(6) ошибка, определяемая реализацией. Оператор ввода-вывода выполнен неуспешно в результате возникновения условия, определенного реализацией.

Ниже приводится список значений, помещаемых в состояние ввода-вывода для перечисленных выше условий, возникающих в результате выполнения операций ввода-вывода для относительно-

GOST EXPERT ГОСТ 22558-89,

го файла. Если применимо более одного значения, значение, которое помещается в состояние ввода-вывода, определяется реализацией.

(1) Успешное завершение

а) Состояние ввода-вывода=00. Оператор ввода-вывода выполнился успешно и нет никакой другой доступной информации об операции ввода-вывода.

б) Состояние ввода-вывода=04. Оператор READ (ЧИТАТЬ) выполнился успешно, но длина обрабатываемой записи не соответствует фиксированным свойствам этого файла.

в) Состояние ввода-вывода=05. Оператор OPEN (ОТКРЫТЬ) успешно выполнился, но указанный в нем необязательный файл во время выполнения оператора OPEN (ОТКРЫТЬ) отсутствует. Если режим открытия I-O (ВХОДНОЙ-ВЫХОДНОЙ) или EXTEND (ДОПОЛНЯЕМЫЙ), файл будет создаваться.

(2) Условие «в конце» с неуспешным завершением

а) Состояние ввода-вывода=10. Делается попытка выполнить последовательный оператор READ (ЧИТАТЬ), а в файле не существует следующей логической записи из-за того, что:

1) достигнут конец файла;

2) делается попытка выполнить последовательный оператор READ (ЧИТАТЬ) в первый раз для отсутствующего необязательного входного файла.

б) Состояние ввода-вывода—14.

Делается попытка выполнить последовательный оператор READ (ЧИТАТЬ) для относительного файла, и число значащих цифр в относительном номере записи больше, чем размер относительного ключа, описанного для файла.

(3) Условие ошибки ключа с неуспешным завершением

а) Состояние ввода-вывода=22. Сделана попытка записать запись, в результате которой был бы создан дублирующий ключ в относительном файле.

б) Состояние ввода-вывода=23. Это условие возникает, если:

1) сделана попытка произвольного доступа к записи, которой нет в файле, или

2) сделана попытка выполнить оператор START (ПОДВЕСТИ) или READ (ЧИТАТЬ) с произвольным доступом для необязательного входного файла, который отсутствует.

в) Состояние ввода-вывода=24. Сделана попытка занесения записей в относительный файл вне его границ, определенных внешним образом. Способ определения границ описывается реализацией. Возможно, был применен оператор последовательной запи-

GOST EXPERT ГОСТ 22558-89,

си в относительный файл и число значащих цифр в относительном номере записи больше, чем размер данного относительный ключ, определенного для файла.

(4) Условие постоянной ошибки с неуспешным завершением

а) Состояние ввода-вывода=30. Возникла постоянная ошибка и нет никакой другой доступной информации об операции ввода-вывода.

б) Состояние ввода-вывода=35. Постоянная ошибка возникла из-за того, что делается попытка выполнить оператор OPEN (ОТКРЫТЬ) с фразой INPUT (ВХОДНОЙ), I-O (ВХОДНОЙ-ВЫХОДНОЙ) или EXTEND (ДОПОЛНЯЕМЫЙ) для файла, который обязательно должен присутствовать, но не присутствует.

в) Состояние ввода-вывода 37. Постоянная ошибка возникла из-за того, что оператор OPEN (ОТКРЫТЬ) выдан для файла, который не поддерживает режим, заданный в операторе OPEN (ОТКРЫТЬ). Возможны следующие нарушения:

1) задана фраза EXTEND (ДОПОЛНЯЕМЫЙ) или OUTPUT (ВЫХОДНОЙ), а файл не допускает операции записи;

2) задана фраза I-O (ВХОДНОЙ-ВЫХОДНОЙ), а файл не допускает операции ввода и вывода, которые разрешены для относительного файла, открываемого в режиме ввода-вывода;

3) задана фраза INPUT (ВХОДНОЙ), а файл не допускает операции чтения.

г) Состояние ввода-вывода=38. Постоянная ошибка возникла из-за того, что выдан оператор OPEN (ОТКРЫТЬ) для файла, ранее закрытого с замком.

д) Состояние ввода-вывода=39. Оператор OPEN (ОТКРЫТЬ) завершился неуспешно из-за обнаруженного для этого файла несоответствия фиксированных свойств файла и свойств, заданных в программе.

(5) Условие логической ошибки с неуспешным завершением

а) Состояние ввода-вывода=41. Оператор OPEN (ОТКРЫТЬ) выдан для открытого файла.

б) Состояние ввода-вывода=42. Оператор CLOSE (ЗАКРЫТЬ) выдан для неоткрытого файла.

в) Состояние ввода-вывода=43. При последовательном методе доступа последний оператор ввода-вывода, выполненный для файла до выполнения оператора DELETE (УДАЛИТЬ) или REWRITE (ОБНОВИТЬ), не является успешно выполненным оператором READ (ЧИТАТЬ).

г) Состояние ввода-вывода=44. Нарушение границ возникает по следующим причинам:

GOST EXPERT ГОСТ 22558-89,

1) совершена попытка записать или обновить запись, которая длиннее максимально допустимой или короче минимально допустимой в соответствии с фразой RECORD IS VARYING (В ЗАПИСИ ПЕРЕМЕННОЕ ЧИСЛО), связанной с именем файла;

2) на уровне 1 сделана попытка обновить запись относительного файла, а размер заменяющей записи отличен от размера заменяемой записи.

д) Состояние ввода-вывода=46. Выдан оператор последовательного чтения для файла, открытого в режиме ввода или ввода-вывода, и не была установлена правильная следующая запись по одной из следующих причин:

1) предыдущий оператор START (ПОДВЕСТИ) закончился неуспешно, или

2) предыдущий оператор READ (ЧИТАТЬ) закончился неуспешно, но не вызвал условие «в конце», или

3) предыдущий оператор READ (ЧИТАТЬ) вызвал условие «в конце».

е) Состояние ввода-вывода=47. Был выдан оператор READ (ЧИТАТЬ) или START (ПОДВЕСТИ) для файла, не открытого в режиме ввода или ввода-вывода.

ж) Состояние ввода-вывода=48. Был выдан оператор WRITE (ПИСАТЬ) для файла, не открытого в режиме ввода-вывода, вывода или дополнения.

з) Состояние ввода-вывода=49. Был выдан оператор DELETE (УДАЛИТЬ) или REWRITE (ОБНОВИТЬ) для файла, не открытого в режиме ввода-вывода.

(б) Определяемое реализацией условие с неуспешным завершением

а) Состояние ввода-вывода=9x. Существуют определяемые реализацией условия. Эти условия не должны дублировать никакие условия, определенные для значений от 00 до 49 состояния ввода-вывода. Значения *x* определяются реализацией.

1.3.5. Условие ошибки ключа

Условие ошибки ключа может возникнуть в результате выполнения операторов DELETE (УДАЛИТЬ), READ (ЧИТАТЬ), REWRITE (ОБНОВИТЬ), START (ПОДВЕСТИ) или WRITE (ПИСАТЬ). Когда возникает условие ошибки ключа, оператор ввода-вывода, вызвавший эту ситуацию, является неуспешным и файл не изменяется (пп. 4.3, 4.5—4.7, 4.9 настоящей части).

GOST EXPERT ГОСТ 22558-89,

Если условие ошибки ключа возникает после выполнения операции ввода-вывода, определенной в операторе ввода-вывода, то происходит следующие действия в указанном порядке:

(1) состояние ввода-вывода определителя файла, связанного с оператором, устанавливается в значение, определяющее условие ошибки ключа (см. п. 1.3.4 настоящей части);

(2) если в операторе ввода-вывода указана фраза INVALID KEY (ПРИ ОШИБКЕ КЛЮЧА), то никакая процедура USE AFTER EXCEPTION (ИСПОЛЬЗОВАТЬ ПОСЛЕ ПРОЦЕДУРЫ ОШИБКИ), связанная с определителем файла, не выполняется, и управление передается повелительному оператору, указанному во фразе INVALID KEY (ПРИ ОШИБКЕ КЛЮЧА). Выполнение продолжается в соответствии с правилами для каждого оператора, указанного в этом повелительном операторе. Если выполняется оператор ветвления процедуры или условный оператор, который вызывает явную передачу управления, то управление передается в соответствии с правилами для этого оператора; в противном случае после завершения выполнения повелительного оператора, указанного во фразе INVALID KEY (ПРИ ОШИБКЕ КЛЮЧА), управление передается в конец оператора ввода-вывода, а фраза NOT INVALID KEY (БЕЗ ОШИБКИ КЛЮЧА), если она указана, игнорируется (п. 4.8 настоящей части);

(3) если в операторе ввода-вывода фраза INVALID KEY (ПРИ ОШИБКЕ КЛЮЧА) не указана, то с определителем файла должна быть связана процедура USE AFTER EXCEPTION (ИСПОЛЬЗОВАТЬ ПОСЛЕ ПРОЦЕДУРЫ ОШИБКИ), и эта процедура выполняется, а управление передается в соответствии с правилами оператора USE (ИСПОЛЬЗОВАТЬ). Если указана фраза NOT INVALID KEY (БЕЗ ОШИБКИ КЛЮЧА), то она игнорируется (п. 4.8 настоящей части).

Если после выполнения операции ввода-вывода, указанной в операторе ввода-вывода, условия ошибки ключа нет, то фраза INVALID KEY (ПРИ ОШИБКЕ КЛЮЧА), если она указана, игнорируется. Состояние ввода-вывода определителя файла, связанного с оператором, обновляется, и выполняются следующие действия:

(1) если возникает условие ошибки, которое не является условием ошибки ключа, то управление передается в соответствии с правилами оператора USE (ИСПОЛЬЗОВАТЬ), а затем выполняется процедура USE AFTER EXCEPTION (ИСПОЛЬЗОВАТЬ ПОСЛЕ ПРОЦЕДУРЫ ОШИБКИ), связанная с определителем файла (п. 4.8 настоящей части);

(2) если условие ошибки не возникает, то управление передается в конец оператора ввода-вывода или повелительному оператору, указанному во фразе NOT INVALID KEY (БЕЗ ОШИБКИ КЛЮЧА). В последнем случае выполнение продолжается в соответствии

GOST EXPERT ГОСТ 22558-89,

с правилами для каждого оператора, указанного в повелительном операторе. Если выполняется оператор ветвления процедуры или условный оператор, который вызывает явную передачу управления, то управление передается в соответствии с правилами для этого оператора; в противном случае, после завершения выполнения повелительного оператора, указанного во фразе NOT INVALID KEY (БЕЗ ОШИБКИ КЛЮЧА), управление передается в конец оператора ввода-вывода.

1.3.6. Условие «в конце»

Условие «в конце» может возникнуть в результате выполнения оператора READ (ЧИТАТЬ) (п. 4.5 настоящей части).

1.3.7. Условие противоречия свойств файла

Условие противоречия свойств файла может возникнуть в результате выполнения операторов OPEN (ОТКРЫТЬ), REWRITE (ОБНОВИТЬ) и WRITE (ПИСАТЬ). При возникновении условия противоречия свойств файла выполнение оператора ввода-вывода, который обнаружил это условие, считается неуспешным и файл не изменяется (пп. 4.4, 4.7, 4.9 настоящей части).

При обнаружении условия противоречия свойств файла выполняются следующие действия в указанном ниже порядке:

(1) в состояние ввода-вывода, связанное с именем файла, помещается значение, указывающее на условие противоречия свойств файла (см. п. 1.3.5 настоящей части);

(2) если для данного имени-файла задана процедура USE AFTER EXCEPTION (ИСПОЛЬЗОВАТЬ ПОСЛЕ ПРОЦЕДУРЫ ОШИБКИ), выполняется указанная процедура.

2. РАЗДЕЛ ОБОРУДОВАНИЯ В МОДУЛЕ ОТНОСИТЕЛЬНОГО ВВОДА-ВЫВОДА

2.1. **Секция ввода-вывода**

Информация, относящаяся к секции ввода-вывода, находится в ч. 7, п. 2.1.

2.2. **Параграф FILE-CONTROL (УПРАВЛЕНИЕ-ФАЙЛАМИ)**

Информация о параграфе FILE-CONTROL (УПРАВЛЕНИЕ-ФАЙЛАМИ) находится в ч. 7, п. 2.2.

2.3. **Статья управления файлом**

2.3.1. Назначение

Статья управления файлом объявляет существенные физические свойства относительного файла.

2.3.2. Общий формат

SELECT [OPTIONAL] имя-файла-1

ASSIGN TO { имя-реализации-1 / литерал-1 } ...

GOST EXPERT ГОСТ 22558-89,

```
[[RESERVE целое-1 [AREA / AREAS]]]
[ORGANIZATION IS] RELATIVE
[ ACCESS MODE IS { SEQUENTIAL [RELATIVE KEY IS имя-данного-1] / RANDOM / [DYNAMIC] } [RELATIVE KEY IS имя-данного-1] ]
[FILE STATUS IS имя-данного-2]
ДЛЯ [[НЕОБЯЗАТЕЛЬНОГО]] имя-файла-1
НАЗНАЧИТЬ {имя-реализации-1 / литерал-1} ...
[[РЕЗЕРВИРОВАТЬ целое-1 ОБЛАСТЕЙ]]
[ОРГАНИЗАЦИЯ] ОТНОСИТЕЛЬНАЯ
[ ДОСТУП { ПОСЛЕДОВАТЕЛЬНЫЙ / [ОТНОСИТЕЛЬНЫЙ КЛЮЧ имя-данного-1] / ПРОИЗВОЛЬНЫЙ / [ДИНАМИЧЕСКИЙ] } ОТНОСИТЕЛЬНЫЙ КЛЮЧ имя-данного-1 ]
[СОСТОЯНИЕ ФАЙЛА имя-данного-2].
```

2.3.3. Синтаксические правила

(1) В статье управления файлом фраза SELECT (ДЛЯ) должна указываться первой. Фразы, которые следуют за фразой SELECT (ДЛЯ), могут появляться в любом порядке.

(2) Каждое имя-файла из раздела данных должно быть определено в параграфе FILE-CONTROL (УПРАВЛЕНИЕ-ФАЙЛАМИ) только один раз. Каждое имя-файла, указанное в фразе SELECT (ДЛЯ), должно иметь статью описания файла в разделе данных той же самой программы.

(3) Литерал-1 должен быть нечисловым литералом и не должен быть стандартной константой. Значение и правила для допустимого содержимого имени-реализации-1 и значения литерала-1 определяются реализацией.

GOST EXPERT ГОСТ 22558-89,

2.3.4. Общие правила

(1) Если определитель файла, на который ссылается имя-файла-1, является внешним определителем файла (ч. 10, п. 4.5), все статьи управления файлом в единице исполнения, которые ссылаются на этот определитель файла, должны иметь:

а) одну и ту же спецификацию фразы OPTIONAL (НЕОБЯЗАТЕЛЬНОГО);

б) корректную спецификацию для имени-реализации-1 или литерала-1 во фразе ASSIGN (НАЗНАЧИТЬ). Правила корректности имени-реализации-1 и литерала-1 определяются реализацией;

в) одно и то же значение целого-1 во фразе RESERVE (РЕЗЕРВИРОВАТЬ);

г) одну и ту же организацию;

д) один и тот же метод доступа;

е) одно и то же внешнее данное для имени-данного-1 во фразе RELATIVE KEY (ОТНОСИТЕЛЬНЫЙ КЛЮЧ).

(2) Для данных внешней среды принимается внутренний набор литер.

(3) Фраза OPTIONAL (НЕОБЯЗАТЕЛЬНОГО) применима только к файлам, открытым в режиме ввода, ввода-вывода или дополнения. Ее указание требуется для файлов, которые могут отсутствовать во время выполнения объектной программы.

(4) Фраза ASSIGN (НАЗНАЧИТЬ) задает связь между файлом, на который ссылается имя-файла-1, и запоминающей средой, на которую ссылается имя-реализации-1 или литерал-1.

(5) Фраза RESERVE (РЕЗЕРВИРОВАТЬ) для модуля относительного ввода вывода та же, что и для модуля последовательного ввода-вывода. Описание фразы RESERVE (РЕЗЕРВИРОВАТЬ) находится в ч. 7, п. 2.9.

(6) Фраза FILE STATUS (СОСТОЯНИЕ ФАЙЛА) для модуля относительного ввода-вывода та же, что и для модуля последовательного ввода-вывода. Описание фразы FILE STATUS (СОСТОЯНИЕ ФАЙЛА) находится в ч. 7, п. 2.5. Содержимое данного, связанного с фразой FILE STATUS (СОСТОЯНИЕ ФАЙЛА) относительного файла, определяется в п. 1.3.4 настоящей части.

(7) Фразы ACCESS MODE (ДОСТУП) и ORGANIZATION IS RELATIVE (ОРГАНИЗАЦИЯ ОТНОСИТЕЛЬНАЯ) описываются ниже.

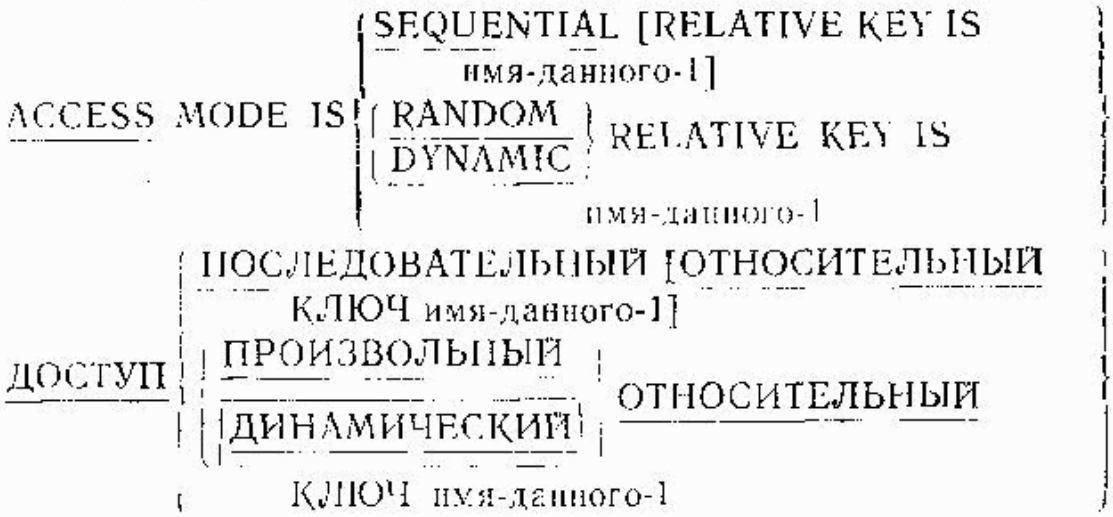
GOST EXPERT ГОСТ 22558-89,

2.4. **Фраза ACCESS MODE (ДОСТУП)**

2.4.1. Назначение

Фраза ACCESS MODE (ДОСТУП) задает порядок, в котором осуществляется доступ к записям в файле.

2.4.2. Общий формат



2.4.3. Синтаксические правила

(1) Имя-данного-1 может быть уточнено.

(2) Имя-данного-1 должно ссылаться на данное, описанное как целое без знака, не содержащее в описании шаблона литеры P(M).

(3) Имя-данного-1 не должно быть определено в статье описания записи, связанной с этим именем-файла.

(4) Фраза ACCESS MODE IS RANDOM (ДОСТУП ПРОИЗВОЛЬНЫЙ) не должна быть указана для имен-файлов, указанных в фразах USING (ИСПОЛЬЗУЯ) или GIVING (ПОЛУЧАЯ) операторов SORT (СОРТИРОВАТЬ) или MERGE (СЛИТЬ).

(5) Если на относительный файл ссылаются в операторе START (ПОДВЕСТИ), для этого файла должна быть указана фраза RELATIVE KEY (ОТНОСИТЕЛЬНЫЙ КЛЮЧ) во фразе ACCESS MODE (ДОСТУП).

2.4.4. Общие правила

(1) Если фраза ACCESS MODE (ДОСТУП) не задана, предполагается последовательный доступ.

(2) Если доступ последовательный, записи становятся доступными в последовательности, диктуемой организацией файла. Эта последовательность есть возрастающая последовательность относительных номеров записей, находящихся в файле.

(3) Если доступ произвольный, значение данного, определенного как относительный ключ, указывает на запись, к которой должен быть осуществлен доступ.

GOST EXPERT ГОСТ 22558-89,

(4) Если доступ динамический, к записям файла можно обращаться методом последовательного и (или) произвольного доступа.

(5) Записи в относительном файле однозначно идентифицируются относительными номерами записей. Относительный номер записи указывает порядковую позицию логической записи в файле. Первая логическая запись имеет относительный номер записи 1, следующие логические записи имеют относительные номера 2, 3, 4 и так далее.

(6) Данное, представленное именем-данного-1, используется для передачи относительного номера записи между пользователем и СУМП.

(7) Данное, представляющее относительный ключ, связанный с выполнением оператора ввода-вывода, является данным, на которое ссылается имя-данного-1 во фразе ACCESS MODE (ДОСТУП).

(8) Если соответствующий определитель файла является внешним определителем файла, то каждая статья управления файлом в единице исполнения, связанная с этим определителем файла, должна указывать один и тот же метод доступа. Кроме того, имя-данного-1 должно ссылаться на внешнее имя-данного, и фраза RELATIVE KEY (ОТНОСИТЕЛЬНЫЙ КЛЮЧ) в каждой соответствующей статье управления файлом должна в каждом случае ссылаться на одно и то же внешнее имя-данного.

2.5. Фраза **ORGANIZATION IS RELATIVE (ОРГАНИЗАЦИЯ ОТНОСИТЕЛЬНАЯ)**

2.5.1. Назначение

Фраза ORGANIZATION IS RELATIVE (ОРГАНИЗАЦИЯ ОТНОСИТЕЛЬНАЯ) определяет относительную организацию как логическую структуру файла.

2.5.2. Общий формат

[ORGANIZATION IS] RELATIVE

[ОРГАНИЗАЦИЯ] ОТНОСИТЕЛЬНАЯ

2.5.3. Общие правила

(1) Фраза ORGANIZATION IS RELATIVE (ОРГАНИЗАЦИЯ ОТНОСИТЕЛЬНАЯ) определяет относительную организацию как логическую структуру файла. Организация файла устанавливается при создании файла и впоследствии не может быть изменена.

(2) Относительная организация является постоянной логической структурой файла, в которой каждая запись однозначно определяется целым значением больше нуля, которое указывает порядковую позицию записи в файле.

GOST EXPERT ГОСТ 22558-89,

2.6. Параграф I-O-CONTROL (УПРАВЛЕНИЕ-ВВОДОМ-ВЫВОДОМ)

2.6.1. Назначение

Параграф I-O-CONTROL (УПРАВЛЕНИЕ-ВВОДОМ-ВЫВОДОМ) указывает контрольные точки для перепрогона, а также общие области памяти, которые могут совместно использоваться различными файлами. Фраза RERUN (ПЕРЕПРОГОН) параграфа I-O-CONTROL (УПРАВЛЕНИЕ-ВВОДОМ-ВЫВОДОМ) рассматривается в настоящем стандарте как устаревший элемент и будет удалена в следующей редакции стандарта.

2.6.2. Общий формат

I-O-CONTROL.

```
[ RERUN ON { имя-файла-1          }
           { имя-реализации-1     }
                                                                  ]...
[          { целое-1 RECORDS OF имя-файла-2 }                     ]
[    EVERY { целое-2 CLOCK-UNITS            }                     ]
[          { имя-условия-1                  }                     ]

[SAME [RECORD] AREA FOR имя-файла-3
                {имя-файла-4} ... ] ... ].
```

УПРАВЛЕНИЕ-ВВОДОМ-ВЫВОДОМ.

```
[[ ПЕРЕПРОГОН НА { имя-файла-1      }                              ]...
                { имя-реализации-1 }                              ]
[               { КАЖДЫЕ целое-1 ЗАПИСЕЙ имя-файла-2 }            ]
[               { КАЖДОЕ целое-2 ЕДИНИЦ-ВРЕМЕНИ       }          ]
[               { имя-условия-1                       }          ]

[ОБЩАЯ ОБЛАСТЬ [ЗАПИСИ] ДЛЯ имя-файла-3
                {имя-файла 4} ... ] ... ].
```

2.6.3. Общие правила

(1) Фраза RERUN (ПЕРЕПРОГОН) для модуля относительного ввода-вывода является подмножеством фразы RERUN (ПЕРЕПРОГОН) модуля последовательного ввода-вывода. Описание фразы RERUN (ПЕРЕПРОГОН) см. в ч. 7, п. 2.12.

(2) Фраза SAME (ОБЩАЯ) для модуля относительного ввода-вывода та же, что и для модуля последовательного ввода-вывода. Описание фразы SAME (ОБЩАЯ) см. в ч. 7, п. 2.13.

GOST EXPERT ГОСТ 22558-89,

3. РАЗДЕЛ ДАННЫХ В МОДУЛЕ ОТНОСИТЕЛЬНОГО ВВОДА ВЫВОДА

3.1. Секция файлов

Информацию о секции файлов см. в ч. 7, п. 3.1.

3.2. Статья описания файла

3.2.1. Назначение

Статья описания файла содержит применимую к относительному файлу информацию о физической структуре, идентификации и именах записей.

3.2.2. Общий формат

FD имя-файла-1

[BLOCK CONTAINS [[целое-1 TO] целое-2 {RECORDS / CHARACTERS}]

[RECORD {CONTAINS целое-3 CHARACTERS / [IS VARYING IN SIZE [[FROM целое-4] [TO целое-5] CHARACTERS] DEPENDING ON имя-данного-1]} / CONTAINS целое-6 TO целое-7 CHARACTERS]

[LABEL {RECORD IS / RECORDS ARE} {STANDARD / OMITTED}]

[VALUE OF {имя-реализации-1 IS {имя-данного-2 / литерал-1}} ...]

[DATA {RECORD IS / RECORDS ARE} {имя-данного-3} ...]

ОФ имя-файла-1

[В БЛОКЕ [ОТ целое-1 ДО] целое-2 {ЗАПИСЕЙ / ЛИТЕР}]

[В ЗАПИСИ {целое-3 ЛИТЕР / [ПЕРЕМЕННОЕ ЧИСЛО [[ОТ целое-4] [ДО целое-5] ЛИТЕР] [В ЗАВИСИМОСТИ ОТ имя-данного-1]] / ОТ целое-6 ДО целое-7 ЛИТЕР}]

GOST EXPERT ГОСТ 22558-89,

[МЕТКИ { СТАНДАРТНЫ / ОПУЩЕНЫ }]

[{ ЗНАЧЕНИЕ / ЗНАЧ } { имя-реализацик-1 { имя-данного-2 / литерал-1 } } ...]

[ЗАПИСИ ДАННЫХ {имя-данного-3} . . .].

3.2.3. Синтаксические правила

(1) Индикатор уровня FD (ОФ) идентифицирует начало статьи описания файла и должен предшествовать имени-файла-1.

(2) Фразы, которые следуют за именем-файла-1, могут задаваться в любом порядке.

(3) Одна или несколько статей описания записи должны следовать за статьей описания файла.

3.2.4. Общие правила

(1) Статья описания файла связывает имя-файла-1 с определителем файла.

(2) Фраза BLOCK CONTAINS (В БЛОКЕ) для модуля относительного ввода-вывода та же, что и для модуля последовательного ввода-вывода. Описание фразы BLOCK CONTAINS (В БЛОКЕ) находится в ч. 7, п. 3.3.

(3) Фраза DATA RECORDS (ЗАПИСИ ДАННЫХ) для модуля относительного ввода-вывода такая же, как и для модуля последовательного ввода-вывода. Описание фразы DATA RECORDS (ЗАПИСИ ДАННЫХ) находится в ч. 7, п. 3.5. Фраза DATA RECORDS (ЗАПИСИ ДАННЫХ) рассматривается в настоящем стандарте как устаревший элемент и будет удалена в следующей редакции стандарта.

(4) Фраза LABEL RECORD (МЕТКИ) для модуля относительного ввода-вывода такая же, как и для модуля последовательного ввода-вывода. Описание фразы LABEL RECORD (МЕТКИ) находится в ч. 7, п. 3.6. Фраза LABEL RECORD (МЕТКИ) рассматривается в настоящем стандарте как устаревший элемент и будет удалена в следующей редакции стандарта.

(5) Фраза RECORD (В ЗАПИСИ) для модуля относительного ввода-вывода такая же, как и для модуля последовательного ввода-вывода. Описание фразы RECORD (В ЗАПИСИ) находится в ч. 7, п. 3.8.

(6) Фраза VALUE OF (ЗНАЧЕНИЕ) для модуля относительного ввода-вывода такая же, как и для модуля последовательного ввода-вывода. Описание фразы VALUE OF (ЗНАЧЕНИЕ) находится в ч. 7, п. 3.9. Фраза VALUE OF (ЗНАЧЕНИЕ) рассматривается в настоящем стандарте как устаревший элемент и будет удалена в следующей редакции стандарта.

GOST EXPERT ГОСТ 22558-89,

4. РАЗДЕЛ ПРОЦЕДУР В МОДУЛЕ ОТНОСИТЕЛЬНОГО ВВОДА-ВЫВОДА

4.1. Общее описание

Если оператор USE (ИСПОЛЬЗОВАТЬ) модуля относительного ввода-вывода имеется в исходной Кобол-программе, раздел процедур содержит декларативные процедуры. Ниже приводится общий формат раздела процедур для случая, когда оператор USE (ИСПОЛЬЗОВАТЬ) указан.

PROCEDURE DIVISION.

DECLARATIVES.

{имя-секции SECTION.

оператор USE.

[имя-параграфа.

[предложение] . . .] . . . } . . .

END DECLARATIVES.

{имя-секции SECTION.

[имя-параграфа.

[предложение] . . .] . . . } . . .

РАЗДЕЛ ПРОЦЕДУР.

ДЕКЛАРАТИВЫ.

{СЕКЦИЯ имя-секции.

оператор ИСПОЛЬЗОВАТЬ.

[имя-параграфа.

[предложение] . . .] . . . } . . .

КОНЕЦ ДЕКЛАРАТИВ.

{СЕКЦИЯ имя-секции.

[имя-параграфа.

[предложение] . . .] . . . }

4.2. Оператор CLOSE (ЗАКРЫТЬ)

4.2.1. Назначение

Оператор CLOSE (ЗАКРЫТЬ) завершает обработку файла, возможно, с замком.

4.2.2. Общий формат

CLOSE {имя-файла-1 [WITH LOCK]} . . .

ЗАКРЫТЬ {имя-файла-1 [С ЗАМКОМ]} . . .

4.2.3. Синтаксическое правило

Файлы, перечисленные в операторе CLOSE (ЗАКРЫТЬ), могут иметь различную организацию и доступ.

GOST EXPERT ГОСТ 22558-89,

4.2.4. Общие правила

(1) Оператор CLOSE (ЗАКРЫТЬ) может быть использован только для файла, который был открыт.

(2) Относительные файлы классифицируются как принадлежащие к категории непоследовательных однотомных или многотомных файлов. Результаты выполнения оператора CLOSE (ЗАКРЫТЬ) для этой категории файлов приведены ниже.

Формат оператора GLOSE (ЗАКРЫТЬ)	Действия операторов GLOSE (ЗАКРЫТЬ) для непоследовательного однотомного (многотомного) файла
CLOSE (ЗАКРЫТЬ)	А
CLOSE WITH LOCK (ЗАКРЫТЬ С ЗАМКОМ)	А, Б

Определения символов А и Б приведены ниже.

Там, где эти определения зависят от того, является ли файл входным, выходным или входным-выходным, приводятся дополнительные пояснения; в противном случае эти определения относятся к входным, выходным и входным-выходным файлам.

А — закрыть файл.

Входные и входные-выходные файлы (доступ последовательный). Если файл установлен в конце и указаны записи меток для этого файла, метки обрабатываются в соответствии со стандартной процедурой обработки меток, определенной реализацией. Действия оператора CLOSE (ЗАКРЫТЬ) не определены, когда записи меток специфицированы, но в файле отсутствуют, или когда записи меток не специфицированы, но присутствуют. Выполняются операции закрытия, определенные реализацией. Если файл установлен в конце и записи меток для него не специфицированы, метки не обрабатываются, но другие операции закрытия, определенные реализацией, выполняются. Если файл установлен не в конце, операции закрытия, определенные реализацией, выполняются, но конечные метки не обрабатываются.

Входные и входные-выходные файлы (доступ произвольный или динамический). Выходные файлы (доступ произвольный, динамический или последовательный). Если записи меток для файла специфицированы, метки обрабатываются в соответствии со стандартной процедурой обработки меток, определенной реализацией. Действия оператора CLOSE (ЗАКРЫТЬ) не определены, когда записи меток специфицированы, но в файле отсутствуют, или когда они не специфицированы, но присутствуют. Выполняются операции закрытия, определенные реализацией. Если записи

GOST EXPERT ГОСТ 22558-89,

меток для файла не указаны, метки не обрабатываются, но другие операции закрытия, определенные реализацией, выполняются.

Б — закрыть с замком.
Файл закрыт и не может быть опять открыт во время выполнения этой единицы исполнения.

(3) Выполнение оператора CLOSE (ЗАКРЫТЬ) приводит к изменению значения состояния ввода-вывода, относящегося к имени-файла-1 (см. п. 1.3.4 настоящей части).

(4) Если не присутствует необязательный входной файл, для файла не производится обработка конца файла, и указатель позиции файла не меняется.

(5) После успешного завершения оператора CLOSE (ЗАКРЫТЬ) область записи, связанная с именем-файла, становится недоступной. В случае неуспешного выполнения оператора CLOSE (ЗАКРЫТЬ) доступность области записи является неопределенной.

(6) После успешного завершения оператора CLOSE (ЗАКРЫТЬ) файл перестает быть открытым, он больше не связан ни с каким определителем файла.

(7) Если в операторе CLOSE (ЗАКРЫТЬ) указаны несколько имен-файлов, результат выполнения этого оператора CLOSE (ЗАКРЫТЬ) такой же, как если бы отдельный оператор CLOSE (ЗАКРЫТЬ) был написан для каждого имени файла в том порядке, как они указаны в этом операторе CLOSE (ЗАКРЫТЬ).

4.3. Оператор DELETE (УДАЛИТЬ)

4.3.1. Назначение

Оператор DELETE (УДАЛИТЬ) логически удаляет запись из файла массовой памяти.

4.3.2. Общий формат

DELETE имя-файла-1 RECORD
[INVALID KEY повелительный-оператор-1]
[NOT INVALID KEY повелительный-оператор-2]
[END-DELETE]

УДАЛИТЬ ЗАПИСЬ имя-файла
[ПРИ ОШИБКЕ КЛЮЧА повелительный-оператор-1]
[БЕЗ ОШИБКИ КЛЮЧА повелительный-оператор-2]
[КОНЕЦ-УДАЛИТЬ]

4.3.3. Синтаксические правила

(1) Фраза INVALID KEY (ПРИ ОШИБКЕ КЛЮЧА) не должна указываться для оператора DELETE (УДАЛИТЬ), который ссылается на файл с последовательным доступом.

GOST EXPERT ГОСТ 22558-89,

(2) Фраза INVALID KEY (ПРИ ОШИБКЕ КЛЮЧА) должна быть указана в операторе DELETE (УДАЛИТЬ), который ссылается на файл не с последовательным доступом и для которого не определена процедура USE AFTER STANDARD EXCEPTION (ИСПОЛЬЗОВАТЬ ПОСЛЕ СТАНДАРТНОЙ ПРОЦЕДУРЫ ОШИБКИ).

4.3.4. Общие правила

(1) Файл, представленный именем-файла-1, должен быть файлом массовой памяти. Он должен быть открыт для ввода-вывода ко времени выполнения этого оператора (п. 4.4 настоящей части).

(2) Для файлов с последовательным доступом последним оператором ввода-вывода, выполняемым для имени-файла-1 перед выполнением оператора DELETE (УДАЛИТЬ), должен быть успешно выполненный оператор READ (ЧИТАТЬ). Система управления массовой памятью логически удаляет из файла запись, которая была извлечена по оператору READ (ЧИТАТЬ).

(3) Для файла с произвольным или динамическим доступом логически удаляется из файла запись, идентифицируемая значением данного, определенного как относительный ключ, связанный с именем-файла-1. Если файл не содержит записи с указанным ключом, возникает условие ошибки ключа (см. п. 1.3.5 настоящей части).

(4) После успешного выполнения оператора DELETE (УДАЛИТЬ) идентифицированная запись логически удаляется из файла и становится недоступной.

(5) Выполнение оператора DELETE (УДАЛИТЬ) не влияет на содержимое области записи или на содержимое данного, представленного именем-данного, указанного в варианте DEPENDING ON (В ЗАВИСИМОСТИ ОТ) фразы RECORD (В ЗАПИСИ), относящейся к имени-файла-1.

(6) Выполнение оператора DELETE (УДАЛИТЬ) не влияет на указатель позиции файла.

(7) При выполнении оператора DELETE (УДАЛИТЬ) обновляется значение состояния ввода-вывода, связанного с именем-файла-1 (см. п. 1.3.4 настоящей части).

(8) Передача управления после успешного или неуспешного выполнения оператора DELETE (УДАЛИТЬ) зависит от наличия или отсутствия в операторе DELETE (УДАЛИТЬ) необязательных фраз INVALID KEY (ПРИ ОШИБКЕ КЛЮЧА) и NOT INVALID KEY (БЕЗ ОШИБКИ КЛЮЧА) (см. п. 1.3.5 настоящей части).

(9) Фраза END-DELETE (КОНЕЦ-УДАЛИТЬ) ограничивает область действия оператора DELETE (УДАЛИТЬ) (см. ч. 4, п. 6.4.3).

4.4. **Оператор OPEN (ОТКРЫТЬ)**

4.4.1. Назначение

GOST EXPERT ГОСТ 22558-89,

Оператор OPEN (ОТКРЫТЬ) подготавливает файл к обработке.

4.4.2. Общий формат

```
        ┌ INPUT {имя-файла-1} ...      ┐
        │ OUTPUT {имя-файла-2} ...     │
OPEN    │ I-O {имя-файла-3} ...        │ ...
        └ EXTEND {имя-файла-4} ...     ┘

        ┌ ВХОДНОЙ {имя-файла-1} ...                  ┐
        │ ВЫХОДНОЙ {имя-файла-2} ...                 │
ОТКРЫТЬ │ ВХОДНОЙ-ВЫХОДНОЙ {имя-файла-3} ...         │ ...
        └ ДОПОЛНЯЕМЫЙ {имя-файла-4} ...              ┘
```

4.4.3. Синтаксические правила

(1) Фраза EXTEND (ДОПОЛНЯЕМЫЙ) должна использоваться только для файлов с последовательным доступом.

(2) Файлы, перечисленные в операторе OPEN (ОТКРЫТЬ), могут иметь различную организацию и доступ.

4.4.4. Общие правила

(1) Успешное выполнение оператора OPEN (ОТКРЫТЬ) делает файл доступным для обработки и переводит файл в режим открытия.

Успешное выполнение оператора OPEN (ОТКРЫТЬ) связывает файл с именем-файла посредством определителя файла.

Файл доступен, если он физически имеется в наличии и распознан системой управления вводом-выводом. Приведенная ниже табл. 1 демонстрирует результаты открытия доступных и недоступных файлов.

Таблица 1

Фраза оператора	Файл доступен	Файл недоступен
INPUT (ВХОДНОЙ)	Нормальное открытие	Открытие неуспешное
INPUT (ВХОДНОЙ) (необязательный файл)	Нормальное открытие	Нормальное открытие; при первом чтении возникает условие конца или условие ошибки ключа
I-O (ВХОДНОЙ-ВЫХОДНОЙ)	Нормальное открытие	Открытие неуспешное

GOST EXPERT ГОСТ 22558-89,

Продолжение табл. 1

Фраза оператора	Файл доступен	Файл недоступен
I-O (ВХОДНОЙ-ВЫХОДНОЙ) (необязательный файл)	Нормальное открытие	Открытие приводит к созданию файла
OUTPUT (ВЫХОДНОЙ)	Нормальное открытие; файл не содержит записей	Открытие приводит к созданию файла
EXTEND (ДОПОЛНЯЕМЫЙ)	Нормальное открытие	Открытие неуспешное
EXTEND (ДОПОЛНЯЕМЫЙ) (необязательный файл)	Нормальное открытие	Открытие приводит к созданию файла

(2) Успешное выполнение оператора OPEN (ОТКРЫТЬ) делает область записи доступной программе. Если определитель файла, связанный с именем файла, является внешним, то существует единственная область записи, связанная с определителем, для единицы исполнения.

(3) Если файл не открыт, не может быть выполнен ни один оператор, явно или неявно относящийся к файлу, за исключением оператора MERGE (СЛИТЬ) с фразами USING (ИСПОЛЬЗУЯ) и GIVING (ПОЛУЧАЯ), оператора OPEN (ОТКРЫТЬ) или оператора SORT (СОРТИРОВАТЬ) с фразами USING (ИСПОЛЬЗУЯ) и GIVING (ПОЛУЧАЯ).

(4) Оператор OPEN (ОТКРЫТЬ) должен быть успешно выполнен перед выполнением любого другого допустимого оператора ввода-вывода.

В табл. 2 X означает, что указанный оператор, используемый при указанном в строке методе доступа, может использоваться в режиме открытия, заданном в заголовке столбца.

(5) Файл может быть открыт с фразами INPUT (ВХОДНОЙ), OUTPUT (ВЫХОДНОЙ), EXTEND (ДОПОЛНЯЕМЫЙ) и I-O (ВХОДНОЙ-ВЫХОДНОЙ) в одной и той же единице исполнения. После первоначального выполнения оператора OPEN (ОТКРЫТЬ) для файла каждому последующему выполнению оператора OPEN (ОТКРЫТЬ) для этого же файла должно предшествовать выполнение для него оператора CLOSE (ЗАКРЫТЬ) без фразы LOCK (С ЗАМКОМ).

(6) Выполнение оператора OPEN (ОТКРЫТЬ) не извлекает и не записывает первую запись данных.

(7) Если указаны записи меток для файла, начальные метки обрабатываются следующим образом:

GOST EXPERT ГОСТ 22558-89,

а) когда указана фраза INPUT (ВХОДНОЙ), оператор OPEN (ОТКРЫТЬ) осуществляет проверку меток в соответствии с процедурами, определенными реализацией для проверки входных меток;

б) когда указана фраза OUTPUT (ВЫХОДНОЙ), выполнение оператора OPEN (ОТКРЫТЬ) вызывает запись меток в соответствии с процедурами, определенными реализацией для записи выходных меток.

Таблица 2

Метод доступа	Оператор	Режим открытия			
		для ввода	для вывода	для ввода-вывода	для дополнения
Последовательный	READ (ЧИТАТЬ)	X		X	
	WRITE (ПИСАТЬ)		X		X
	REWRITE (ОБНОВИТЬ)			X	
	START (ПОДВЕСТИ)	X		X	
	DELETE (УДАЛИТЬ)			X	
Произвольный	READ (ЧИТАТЬ)	X		X	
	WRITE (ПИСАТЬ)		X	X	
	REWRITE (ОБНОВИТЬ)			X	
	START (ПОДВЕСТИ)				
	DELETE (УДАЛИТЬ)			X	
Динамический	READ (ЧИТАТЬ)	X		X	
	WRITE (ПИСАТЬ)		X	X	
	REWRITE (ОБНОВИТЬ)			X	
	START (ПОДВЕСТИ)	X		X	
	DELETE (УДАЛИТЬ)			X	

Действия оператора OPEN (ОТКРЫТЬ) не определены, когда записи меток специфицированы, но в файле отсутствуют или не специфицированы, но присутствуют.

(8) Если во время выполнения оператора OPEN (ОТКРЫТЬ) возникает условие противоречия свойств файла, выполнение оператора OPEN (ОТКРЫТЬ) считается неуспешным (см. п. 1.3.7 настоящей части).

(9) Если файл, открытый с фразой INPUT (ВХОДНОЙ), является необязательным файлом, не имеющимся в наличии, оператор OPEN (ОТКРЫТЬ) устанавливает указатель пози-

GOST EXPERT ГОСТ 22558-89,

ции файла для указания того, что необязательный входной файл отсутствует.

(10) Если файл открыт с фразами INPUT (ВХОДНОЙ) или I-O (ВХОДНОЙ-ВЫХОДНОЙ), указатель позиции файла устанавливается на единицу.

(11) Если указана фраза EXTEND (ДОПОЛНЯЕМЫЙ), оператор OPEN (ОТКРЫТЬ) устанавливает файл непосредственно после его последней логической записи. Последней логической записью относительного файла является существующая в данный момент запись с наибольшим относительным номером записи.

(12) Если задана фраза EXTEND (ДОПОЛНЯЕМЫЙ) и фраза LABEL RECORD (МЕТКИ) указывает, что записи меток присутствуют, выполнение оператора OPEN (ОТКРЫТЬ) включает следующие действия:

а) начальные метки файла обрабатываются только для однокатушечных или однотомных файлов;

б) начальные метки катушки (тома) обрабатываются на последней катушке (томе), как если бы файл открывался как INPUT (ВХОДНОЙ);

в) имеющиеся конечные метки файла обрабатываются, как если бы файл открывался как INPUT (ВХОДНОЙ). Затем эти метки удаляются;

г) затем обработка продолжается, как если бы файл был открыт как OUTPUT (ВЫХОДНОЙ).

(13) Оператор OPEN (ОТКРЫТЬ) с фразой I-O (ВХОДНОЙ-ВЫХОДНОЙ) должен относиться к файлу, поддерживающему операции ввода и вывода, допустимые для относительного файла, открытого для ввода-вывода. Выполнение оператора OPEN (ОТКРЫТЬ) с фразой I-O (ВХОДНОЙ-ВЫХОДНОЙ) подготавливает файл, на который он ссылается, как для операций ввода, так и для операций вывода.

(14) Если указана фраза I-O (ВХОДНОЙ-ВЫХОДНОЙ) и во фразе LABEL RECORD (МЕТКИ) указано, что записи меток присутствуют, выполнение оператора OPEN (ОТКРЫТЬ) включает следующие шаги:

а) проверку меток в соответствии с процедурами, определенными реализацией для проверки входных-выходных меток;

б) Запись новых меток в соответствии с процедурами, определенными реализацией для записи входных-выходных меток.

(15) Для необязательного файла, являющегося недоступным, успешное выполнение оператора OPEN (ОТКРЫТЬ) с фразами

GOST EXPERT ГОСТ 22558-89,

I-O (ВХОДНОЙ-ВЫХОДНОЙ) [или EXTEND (ДОПОЛНЯЕМЫЙ)] приводит к созданию файла. Это создание происходит так, как если бы в указанном порядке выполнялись следующие операторы:

OPEN OUTPUT имя-файла.
CLOSE имя-файла.
ОТКРЫТЬ ВЫХОДНОЙ имя-файла.
ЗАКРЫТЬ имя-файла.

За этими операторами следует выполнение оператора OPEN (ОТКРЫТЬ), указанного в исходной программе.

Успешное выполнение оператора OPEN (ОТКРЫТЬ) с фразой OUTPUT (ВЫХОДНОЙ) приводит к созданию файла, после которого этот файл не содержит записей.

(16) Во время выполнения оператора OPEN (ОТКРЫТЬ) обновляется значение состояния ввода-вывода, связанного с именем-файла (см. п. 1.3.4 настоящей части).

(17) Если в операторе OPEN (ОТКРЫТЬ) указано более, чем одно имя-файла, результат выполнения этого оператора OPEN (ОТКРЫТЬ) такой, как если бы отдельный оператор OPEN (ОТКРЫТЬ) был написан для каждого имени-файла в том порядке, как они указаны в операторе OPEN (ОТКРЫТЬ).

(18) Минимальный и максимальный размеры записей файла устанавливаются во время создания файла и не должны изменяться впоследствии.

4.5. Оператор READ (ЧИТАТЬ)

4.5.1. Назначение

При последовательном доступе оператор READ (ЧИТАТЬ) делает доступной следующую логическую запись файла. При произвольном доступе оператор READ (ЧИТАТЬ) делает доступной указанную запись файла на устройстве массовой памяти.

Формат 1

READ имя-файла-1 [NEXT] RECORD [INTO идентификатор-1]
[AT END повелительный-оператор-1]
[NOT AT END повелительный-оператор-2]
[END-READ]

ЧИТАТЬ [СЛЕДУЮЩУЮ] ЗАПИСЬ имя-файла-1
[В идентификатор-1]
[В КОНЦЕ повелительный-оператор-1]
[НЕ В КОНЦЕ повелительный-оператор-2]
[КОНЕЦ-ЧИТАТЬ]

GOST EXPERT ГОСТ 22558-89,

Формат 2

READ имя-файла-1 RECORD [INTO идентификатор-1]
[INVALID KEY повелительный-оператор-3]
[NOT INVALID KEY повелительный-оператор-4]
[END-READ]

ЧИТАТЬ ЗАПИСЬ имя-файла-1 [В идентификатор-1]
[ПРИ ОШИБКЕ КЛЮЧА повелительный-оператор-3]
[БЕЗ ОШИБКИ КЛЮЧА повелительный-оператор-4]
[КОНЕЦ-ЧИТАТЬ]

4.5.3. Синтаксические правила

(1) Область памяти, связанная с идентификатором-1, и область записи, связанная с именем-файла-1, не должны быть одной и той же областью памяти.

(2) Формат 1 должен использоваться для всех файлов с последовательным доступом.

(3) Фраза NEXT (СЛЕДУЮЩУЮ) должна быть указана для файлов с динамическим доступом, если записи файла должны извлекаться последовательно.

(4) Формат 2 используется для файлов с произвольным доступом или для файлов с динамическим доступом, если записи должны извлекаться произвольно.

(5) Фраза INVALID KEY (ПРИ ОШИБКЕ КЛЮЧА) или фраза AT END (В КОНЦЕ) должна быть указана, если для имени-файла-1 не указана никакая применимая процедура USE AFTER STANDARD EXCEPTION (ИСПОЛЬЗОВАТЬ ПОСЛЕ СТАНДАРТНОЙ ПРОЦЕДУРЫ ОШИБКИ).

4.5.4. Общие правила

(1) Во время выполнения оператора READ (ЧИТАТЬ) файл, на который ссылается имя-файла-1, должен быть открыт как входной или входной-выходной (см. п. 4.4 настоящей части).

(2) Для файлов с последовательным доступом фраза NEXT (СЛЕДУЮЩУЮ) является необязательной и не оказывает влияния на выполнение оператора READ (ЧИТАТЬ).

(3) При выполнении оператора READ (ЧИТАТЬ) обновляется значение состояния ввода-вывода, связанного с именем-файла-1 (см. п. 1.3.4 настоящей части).

(4) Установка указателя позиции файла в начале выполнения оператора READ (ЧИТАТЬ) формата 1 используется для определения записи, которая может быть доступной согласно следующим правилам. Сравнения для записей в относительных файлах относятся к относительному номеру ключа.

GOST EXPERT ГОСТ 22558-89,

а) Если указатель позиции файла указывает, что не установлена следующая запись, выполнение оператора READ (ЧИТАТЬ) является неуспешным.

б) Если указатель позиции файла указывает, что необязательного входного файла нет, оператор выполняется согласно общему правилу (10).

в) Если указатель позиции файла был установлен предыдущими операторами OPEN (ОТКРЫТЬ) или START (ПОДВЕСТИ), выбирается первая существующая запись файла, относительный номер которой больше или равен указателю позиции файла.

г) Если указатель позиции файла установлен предыдущим оператором READ (ЧИТАТЬ), выбирается первая существующая запись файла, номер записи которой больше указателя позиции файла.

Если найдена запись, удовлетворяющая вышеприведенным правилам, она становится доступной в области записи, связанной с именем-файла-1, если только не указана фраза RELATIVE KEY (ОТНОСИТЕЛЬНЫЙ КЛЮЧ) для имени-файла-1 и число значащих цифр в относительном номере выбранной записи не больше, чем размер самого относительного ключа. В последнем случае указатель позиции файла устанавливается для индикации этого условия, а оператор выполняется согласно общему правилу (10).

Если запись, удовлетворяющая приведенным выше правилам, не найдена, то указатель позиции файла устанавливается для указания того, что не существует следующей логической записи, и выполнение продолжается согласно общему правилу (10).

Если запись доступна, указатель позиции файла устанавливается на относительный номер доступной записи.

(5) Функционирование оператора READ (ЧИТАТЬ) не зависит от метода, используемого для согласования времени доступа со временем обработки; запись доступна объектной программе до выполнения повелительного-оператора-2 или повелительного-оператора-4, если они указаны, или до выполнения любого оператора, следующего за оператором READ (ЧИТАТЬ), если ни повелительный-оператор-2, ни повелительный-оператор-4 не указаны.

(6) Когда логические записи описаны более чем одной статьей описания записи, эти записи автоматически используют одну и ту же область памяти; это эквивалентно неявному переопределению области. По завершении оператора READ (ЧИТАТЬ) значения всех данных, находящихся вне диапазона текущей записи данных, не определены.

(7) Фраза INTO (В) может быть указана в операторе READ (ЧИТАТЬ), если:

а) только одно описание записи подчиняется статье описания файла;

GOST EXPERT ГОСТ 22558-89,

б) все имена-записей, связанные с именем-файла-1, и данное, на которое ссылается идентификатор-1, описывают групповое данное или элементарное буквенно-цифровое данное.

(8) Результат выполнения оператора READ (ЧИТАТЬ) с фразой INTO (В) эквивалентен применению следующих правил в указанном порядке:

а) выполняется тот же оператор READ (ЧИТАТЬ) без фразы INTO (В);

б) текущая запись пересылается из области записи в область, указываемую идентификатором-1, в соответствии с правилами для оператора MOVE (ПОМЕСТИТЬ) без фразы CORRESPONDING (СООТВЕТСТВЕННО). Размер текущей записи определяется правилами, указанными для фразы RECORD (В ЗАПИСИ). Если статья описания файла содержит фразу RECORD IS VARYING (В ЗАПИСИ ПЕРЕМЕННОЕ ЧИСЛО ЛИТЕР), неявная пересылка является групповой. Неявный оператор MOVE (ПОМЕСТИТЬ) не выполняется, если выполнение оператора READ (ЧИТАТЬ) было неуспешным. Индексы, связанные с идентификатором-1, вычисляются после того, как запись была прочитана и непосредственно перед ее пересылкой в данное. Запись доступна в области записи и в данном, на которое ссылается идентификатор-1.

(9) Если во время выполнения оператора READ (ЧИТАТЬ) формата 2 указатель позиции файла указывает, что нет необязательного входного файла, возникает условие ошибки ключа, а выполнение оператора READ (ЧИТАТЬ) является неуспешным (см. п. 1.3.5 настоящей части).

(10) Для оператора READ (ЧИТАТЬ) формата 1, если указатель позиции файла указывает, что не существует следующей логической записи или что число значащих цифр в относительном номере записи больше, чем размер самого относительного ключа, или что нет необязательного файла, выполняются следующие действия в указанном порядке:

а) значение, полученное в результате установки указателя позиции файла, присваивается состоянию ввода-вывода, связанному с именем-файла-1, для обозначения условия конца (см. п. 1.3.4 настоящей части);

б) если фраза AT END (В КОНЦЕ) указана в операторе, вызвавшем это условие, управление передается повелительному-оператору-1 во фразе AT END (В КОНЦЕ). Никакие процедуры USE AFTER STANDARD EXCEPTION (ИСПОЛЬЗОВАТЬ ПОСЛЕ СТАНДАРТНОЙ ПРОЦЕДУРЫ ОШИБКИ), связанные с именем-файла-1, не выполняются;

в) если фраза AT END (В КОНЦЕ) не указана, с именем-файла-1 должна быть связана процедура USE AFTER STANDARD

GOST EXPERT ГОСТ 22558-89,

EXCEPTION (ИСПОЛЬЗОВАТЬ ПОСЛЕ СТАНДАРТНОЙ ПРОЦЕДУРЫ ОШИБКИ), которая в этом случае и выполняется. Возврат из этой процедуры осуществляется к следующему после оператора READ (ЧИТАТЬ) выполнимому оператору.

Если возникает условие конца, выполнение оператора READ (ЧИТАТЬ) является неуспешным.

(11) Если во время выполнения оператора READ (ЧИТАТЬ) не возникают ни условие конца, ни условие ошибки ключа, фразы AT END (В КОНЦЕ) и INVALID KEY (ПРИ ОШИБКЕ КЛЮЧА) игнорируются, если они указаны, и выполняются следующие действия:

а) устанавливается значение указателя позиции файла и обновляется значение состояния ввода-вывода, связанного с именем-файла-1;

б) если возникает условие особой ситуации, не являющееся ни условием конца, ни условием ошибки ключа, управление передается процедуре USE AFTER EXCEPTION (ИСПОЛЬЗОВАТЬ ПОСЛЕ СТАНДАРТНОЙ ПРОЦЕДУРЫ ОШИБКИ) согласно правилам для оператора USE (ИСПОЛЬЗОВАТЬ), применимого к имени-файла-1 (п. 4.8 настоящей части);

в) если условие особой ситуации не существует, запись становится доступной в области записи и выполняются любые неявные пересылки, предопределенные фразой INTO (В). Управление передается в точку выхода оператора READ (ЧИТАТЬ) или повелительному-оператору-2, если он указан. В последнем случае выполнение продолжается согласно правилам для операторов, указанных в повелительном-операторе-2. Если выполняется ветвление процедуры или условный оператор, вызывающий явную передачу управления, оно передается в соответствии с правилами для этих операторов, в противном случае после завершения выполнения повелительного-оператора-2 управление передается в точку выхода оператора READ (ЧИТАТЬ).

(12) После неуспешного завершения выполнения оператора READ (ЧИТАТЬ) содержимое соответствующей области записи не определено; указателю позиции файла присвоено значение, указывающее, что правильная следующая запись не установлена.

(13) Для относительного файла, для которого указан динамический метод доступа, формат 1 оператора READ (ЧИТАТЬ) с фразой NEXT (СЛЕДУЮЩУЮ) указывает, что из файла будет извлекаться следующая логическая запись.

(14) Для относительного файла, если фраза RELATIVE KEY (ОТНОСИТЕЛЬНЫЙ КЛЮЧ) указана для имени-файла-1, при выполнении оператора READ (ЧИТАТЬ) формата 1 относительный номер записи, которая становится доступной, помещается в

GOST EXPERT ГОСТ 22558-89,

данное относительный ключ согласно правилам для оператора MOVE (ПОМЕСТИТЬ) (см. ч. 6, п. 6.19).

(15) Для относительного файла при выполнении оператора READ (ЧИТАТЬ) формата 2 указатель позиции файла устанавливается равным значению данного, на которое ссылается фраза RELATIVE KEY (ОТНОСИТЕЛЬНЫЙ КЛЮЧ) для файла, а запись, относительный номер записи которой равен указателю позиции файла, становится доступной в области записи, связанной с именем-файла-1. Если в файле нет такой записи, возникает условие ошибки ключа, и выполнение оператора READ (ЧИТАТЬ) является неуспешным (см. п. 1.3.5 настоящей части).

(16) Если число позиций литер в прочитанной записи меньше минимального размера, указанного статьями описания записей для имени-файла-1, участок области записи, находящийся справа от последней прочитанной литеры, не определен. Если число позиций литер в прочитанной записи больше максимального размера, указанного статьями описания записей для имени-файла-1, запись усекается справа до максимального размера. В обоих случаях выполнение оператора READ (ЧИТАТЬ) считается успешным, а состояние ввода-вывода указывает, что возникло несоответствие длины записи (см. п. 1.3.4 настоящей части).

(17) Фраза END-READ (КОНЕЦ-ЧИТАТЬ) ограничивает область действия оператора READ (ЧИТАТЬ) (см. ч. 4, п. 6.4.3).

4.6. Оператор REWRITE (ОБНОВИТЬ)

4.6.1. Назначение

Оператор REWRITE (ОБНОВИТЬ) логически заменяет запись в файле на устройстве массовой памяти.

4.6.2. Общий формат

REWRITE имя-записи-1 [FROM идентификатор-1]
[INVALID KEY повелительный-оператор-1]
[NOT INVALID KEY повелительный-оператор-2]
[END-REWRITE]

ОБНОВИТЬ имя-записи-1 [ИЗ ПОЛЯ идентификатор-1]
[ПРИ ОШИБКЕ КЛЮЧА повелительный-оператор-1]
[БЕЗ ОШИБКИ КЛЮЧА повелительный-оператор-2]
[КОНЕЦ-ОБНОВИТЬ]

4.6.3. Синтаксические правила

(1) Имя-записи-1 и идентификатор-1 не должны относиться к одной и той же области памяти.

(2) Имя-записи-1 — это имя логической записи в секции файлов раздела данных. Оно может быть уточнено.

GOST EXPERT ГОСТ 22558-89,

(3) Фраза INVALID KEY (ПРИ ОШИБКЕ КЛЮЧА) и NOT INVALID KEY (БЕЗ ОШИБКИ КЛЮЧА) не должна указываться в операторе REWRITE (ОБНОВИТЬ) для файлов с последовательным доступом.

(4) Фраза INVALID KEY (ПРИ ОШИБКЕ КЛЮЧА) должна быть указана в операторе REWRITE (ОБНОВИТЬ) для файлов с произвольным или динамическим доступом, для которых не определена соответствующая процедура USE AFTER EXCEPTION (ИСПОЛЬЗОВАТЬ ПОСЛЕ СТАНДАРТНОЙ ПРОЦЕДУРЫ ОШИБКИ).

4.6.4. Общие правила

(1) Во время выполнения этого оператора файл, связанный с именем-записи-1, должен быть файлом массовой памяти и должен быть открыт как входной-выходной (см. ч. 7, п. 4.3).

(2) Для файлов с последовательным доступом последним оператором ввода-вывода для соответствующего файла перед выполнением оператора REWRITE (ОБНОВИТЬ) должен быть успешно выполненный оператор READ (ЧИТАТЬ). СУМП логически заменяет запись, которая была извлечена оператором READ (ЧИТАТЬ).

(3) На уровне 1 число позиций литер в записи, представленной именем-записи-1, должно быть равно числу позиций литер в обновляемой записи. На уровне 2 число позиций литер в записи, представленной именем-записи-1, может совпадать, а может и не совпадать с числом позиций литер в обновляемой записи.

(4) Логическая запись, включенная в файл при успешном выполнении оператора REWRITE (ОБНОВИТЬ), становится недоступной в области записи, за исключением случая, когда имя-файла, связанное с именем-записи-1, описано во фразе SAME RECORD AREA (ОБЩАЯ ОБЛАСТЬ ЗАПИСИ). Логическая запись доступна программе и как запись файла, связанного с именем-записи-1, и как запись других файлов, указанных в той же фразе SAME RECORD AREA (ОБЩАЯ ОБЛАСТЬ ЗАПИСИ), что и соответствующий выходной файл.

(5) Выполнение оператора REWRITE (ОБНОВИТЬ) с фразой FROM (ИЗ ПОЛЯ) эквивалентно выполнению следующих операторов в указанном порядке:

а) оператор

MOVE идентификатор-1 INTO имя-записи-1

ПОМЕСТИТЬ идентификатор-1 В имя-записи-1

соответственно правилам, описанным для оператора MOVE (ПОМЕСТИТЬ);

б) тот же оператор REWRITE (ОБНОВИТЬ) без фразы FROM (ИЗ ПОЛЯ).

GOST EXPERT ГОСТ 22558-89,

(6) После завершения выполнения оператора REWRITE (ОБНОВИТЬ) информация в области, указанной идентификатором-1, остается доступной, даже если информация в области, указанной именем-записи-1, не является доступной, кроме случая, определяемого фразой SAME RECORD AREA (ОБЩАЯ ОБЛАСТЬ ЗАПИСИ).

(7) Выполнение оператора REWRITE (ОБНОВИТЬ) не влияет на указатель позиции файла.

(8) Выполнение оператора REWRITE (ОБНОВИТЬ) вызывает обновление состояния ввода-вывода для файла, связанного с именем-записи-1 (см. п. 1.3.4 настоящей части).

(9) При выполнении оператора REWRITE (ОБНОВИТЬ) логическая запись передается операционной системе.

(10) Передача управления после успешного или неуспешного выполнения оператора REWRITE (ОБНОВИТЬ) зависит от наличия или отсутствия в операторе REWRITE (ОБНОВИТЬ) необязательных фраз INVALID KEY (ПРИ ОШИБКЕ КЛЮЧА) и NOT INVALID KEY (БЕЗ ОШИБКИ КЛЮЧА) (см. п. 1.3.5 настоящей части).

(11) Фраза END-REWRITE (КОНЕЦ-ОБНОВИТЬ) ограничивает область действия оператора REWRITE (ОБНОВИТЬ) (см. ч. 4, п. 6.4.3).

(12) Число позиций литер в записи, представленной именем-записи-1, должно быть не больше, чем наибольшее, и не меньше, чем наименьшее число позиций литер, определенное фразой RECORD IS VARYING (В ЗАПИСИ ПЕРЕМЕННОЕ ЧИСЛО ЛИТЕР), относящейся к имени-файла, связанного с именем-записи-1. В обоих случаях выполнение оператора REWRITE (ОБНОВИТЬ) является неуспешным, обновление не происходит, содержимое области записи не изменяется, а состояние ввода-вывода файла, связанного с именем-записи-1, становится равным значению, указывающему на причину возникновения ситуации (см. п. 1.3.4 настоящей части).

(13) Для файла с произвольным или динамическим доступом система управления массовой памятью логически заменяет запись, которая указывается значением данного относительный ключ имени-файла, связанного с именем-записи-1. Если в файле нет указанной ключом записи, возникает условие ошибки ключа. Если опознано условие ошибки ключа, выполнение оператора REWRITE (ОБНОВИТЬ) является неуспешным, обновление не происходит, содержимое области записи не изменяется, а состояние ввода-вывода, относящееся к имени-файла, связанному с именем-записи-1, становится равным значению, указывающему на причину возникновения ситуаций (см. п. 1.3.4 настоящей части).

GOST EXPERT ГОСТ 22558-89,

4.7. Оператор START (ПОДВЕСТИ)

4.7.1. Назначение

Оператор START (ПОДВЕСТИ) предоставляет возможность логического позиционирования относительного файла для дальнейшего последовательного извлечения записей.

4.7.2. Общий формат

START имя-файла-1

[KEY { IS EQUAL TO
IS =
IS GREATER THAN
IS >>
IS NOT LESS THAN
IS NOT <
IS GREATER THAN OR EQUAL TO
IS >> = } имя-данного-1]

[INVALID KEY повелительный-оператор-1]

[NOT INVALID KEY повелительный-оператор-2]

[END-START]

ПОДВЕСТИ ЗАПИСЬ имя-файла-1

[КЛЮЧ { РАВЕН
=
БОЛЬШЕ
>>
НЕ МЕНЬШЕ
НЕ <
БОЛЬШЕ ИЛИ РАВЕН
>> = } имя-данного-1]

[ПРИ ОШИБКЕ КЛЮЧА повелительный-оператор-1]

[БЕЗ ОШИБКИ КЛЮЧА повелительный-оператор-2]

[КОНЕЦ-ПОДВЕСТИ]

4.7.3. Синтаксические правила

(1) Имя-файла-1 должно быть именем файла с последовательным или динамическим доступом.

(2) Имя-данного-1 может быть уточнено

(3) Если для имени-файла-1 не определена соответствующая процедура USE AFTER STANDARD EXCEPTION (ИСПОЛЬЗОВАТЬ ПОСЛЕ СТАНДАРТНОЙ ПРОЦЕДУРЫ ОШИБКИ),

GOST EXPERT ГОСТ 22558-89,

должна быть указана фраза INVALID KEY (ПРИ ОШИБКЕ КЛЮЧА).

(4) Имя-данного-1, если оно задано, должно относиться к данному, указанному во фразе RELATIVE KEY (ОТНОСИТЕЛЬНЫЙ КЛЮЧ) фразы ACCESS MODE (ДОСТУП) соответствующей статьи управления файлом.

4.7.4. Общие правила

(1) Файл, представленный именем-файла-1, ко времени выполнения оператора START (ПОДВЕСТИ) должен быть открыт для ввода или ввода-вывода (см. п. 4.4 настоящей части).

(2) Если фраза KEY (КЛЮЧ) не указана, подразумевается знак отношения EQUAL (РАВЕН).

(3) Выполнение оператора START (ПОДВЕСТИ) не изменяет ни содержимое области записи, ни содержимое данного, представленного именем-данного, указанным во фразе DEPENDING ON (В ЗАВИСИМОСТИ ОТ) фразы RECORD (В ЗАПИСИ), относящейся к имени файла-1.

(4) Сравнение, определяемое знаком отношения во фразе KEY (КЛЮЧ), проводится между ключом записи файла, представленного именем-файла-1, и данным, как указано в общем правиле (10). Применяются правила числового сравнения (см. ч. 6, п. 6.3.1.1.1).

а) Указатель позиции файла устанавливается на относительный номер записи первой логической записи файла, ключ которой удовлетворяет сравнению.

б) Если сравнение не удовлетворяется ни для одной записи файла, возникает условие ошибки ключа, и выполнение оператора START (ПОДВЕСТИ) считается неуспешным.

(5) При выполнении оператора START (ПОДВЕСТИ) обновляется значение состояния ввода-вывода, относящегося к имени-файла-1 (см. п. 1.3.4 настоящей части).

(6) Если во время выполнения оператора START (ПОДВЕСТИ) указатель позиции файла определяет, что необязательный входной файл отсутствует, возникает условие ошибки ключа, и выполнение оператора START (ПОДВЕСТИ) является неуспешным.

(7) Передача управления после успешного или неуспешного выполнения оператора START (ПОДВЕСТИ) зависит от наличия или отсутствия необязательных фраз INVALID KEY (ПРИ ОШИБКЕ КЛЮЧА) и NOT INVALID KEY (БЕЗ ОШИБКИ КЛЮЧА) в операторе START (ПОДВЕСТИ) (см. п. 1.3.5 настоящей части).

(8) После неуспешного выполнения оператора START (ПОДВЕСТИ) указатель позиции файла указывает, что правильная следующая запись не установлена.

GOST EXPERT ГОСТ 22558-89,

(9) Фраза END-START (КОНЕЦ-ПОДВЕСТИ) ограничивает область действия оператора START (ПОДВЕСТИ) (см. ч. 4, п. 6.4.3).

(10) Сравнение, описанное в общем правиле (4), использует данное, представленное фразой RELATIVE KEY (ОТНОСИТЕЛЬНЫЙ КЛЮЧ) фразы ACCESS MODE (ДОСТУП), относящейся к имени-файла-1.

4.8. Оператор USE (ИСПОЛЬЗОВАТЬ)

4.8.1. Назначение

Оператор USE (ИСПОЛЬЗОВАТЬ) определяет процедуры обработки ошибок ввода-вывода дополнительно к стандартным процедурам, предоставляемым системой управления вводом-выводом.

4.8.2. Общий формат

```
USE AFTER STANDARD [EXCEPTION | ERROR] PROCEDURE
ON { {имя-файла-1} [...] | INPUT | OUTPUT | I-O | EXTEND }

ИСПОЛЬЗОВАТЬ ПОСЛЕ СТАНДАРТНОЙ ПРОЦЕДУРЫ ОШИБКИ
ДЛЯ { {имя-файла-1} [...] | ВХОДНЫХ | ВЫХОДНЫХ | ВХОДНЫХ-ВЫХОДНЫХ | ДОПОЛНЯЕМЫХ }
```

4.8.3. Синтаксические правила

(1) Оператор USE (ИСПОЛЬЗОВАТЬ) должен непосредственно следовать за заголовком секции декларативной части раздела процедур и должен быть единственным в предложении. Остальная часть декларативной секции должна состоять из одного или более процедурных параграфов, определяющих процедуры, которые должны использоваться.

(2) Сам оператор USE (ИСПОЛЬЗОВАТЬ) никогда не выполняется; он только определяет условия, вызывающие выполнение указанных после него процедур.

GOST EXPERT ГОСТ 22558-89,

(3) Появление имени-файла-1 в операторе USE (ИСПОЛЬЗОВАТЬ) не должно требовать одновременного выполнения более чем одной декларативной секции.

(4) Слова ERROR и EXCEPTION являются синонимами и взаимозаменяемы.

(5) Файлы, к которым явно или неявно обращаются в операторе USE (ИСПОЛЬЗОВАТЬ), могут иметь различную организацию или доступ.

(6) Каждая из фраз INPUT (ВХОДНЫХ), OUTPUT (ВЫХОДНЫХ), I-O (ВХОДНЫХ-ВЫХОДНЫХ) и EXTEND (ДОПОЛНЯЕМЫХ) может указываться лишь раз в декларативной части раздела процедур.

4.8.4. Общие правила

(1) Декларативные процедуры могут быть включены в любую исходную Кобол-программу, независимо от того, содержит ли эта программа другую программу, или сама содержится в другой программе. Декларатива вызывается тогда, когда во время выполнения программы выполняются условия, описанные в операторе USE (ИСПОЛЬЗОВАТЬ), предшествующем декларативе. Только одна декларатива внутри отдельно скомпилированной программы, содержащей оператор, который вызвал уточняющее условие, вызывается тогда, когда выполняется какое-либо из условий, описанных в операторе USE (ИСПОЛЬЗОВАТЬ), предшествующем декларативе, во время выполнения программы. Если не существует уточняющей декларативы в отдельно скомпилированной программе, то декларатива не выполняется.

(2) Внутри декларативной процедуры не должно быть обращений к каким-либо процедурам в недекларативной части раздела процедур.

(3) К именам процедур, связанных с оператором USE (ИСПОЛЬЗОВАТЬ), могут быть обращения в другой декларативной секции или в недекларативной процедуре только оператором PERFORM (ВЫПОЛНИТЬ).

(4) Когда имя-файла-1 описано явно, то к имени-файла-1 не применяется никаких других операторов USE (ИСПОЛЬЗОВАТЬ).

(5) Процедуры, связанные с оператором USE (ИСПОЛЬЗОВАТЬ), выполняются системой управления вводом-выводом после завершения стандартной программы ошибки ввода-вывода при неуспешном выполнении, если только не сработает фраза AT END (В КОНЦЕ) или INVALID KEY (ПРИ ОШИБКЕ КЛЮЧА). При выполнении процедур соблюдаются следующие правила:

а) если указано имя-файла-1, то соответствующая процедура выполняется при выполнении условия, описанного в операторе USE (ИСПОЛЬЗОВАТЬ);

GOST EXPERT ГОСТ 22558-89,

б) если указано INPUT (ВХОДНЫХ), то соответствующая процедура выполняется при выполнении условия, описанного в операторе USE (ИСПОЛЬЗОВАТЬ), для какого-либо файла, открытого для ввода или в процесс открытия для ввода, за исключением файлов, указанных именем-файла-1 в другом операторе USE (ИСПОЛЬЗОВАТЬ), описывающем такое же условие;

в) если указано OUTPUT (ВЫХОДНЫХ), то соответствующая процедура выполняется при выполнении условия, описанного в операторе USE (ИСПОЛЬЗОВАТЬ), для какого-либо файла, открытого для вывода или же в процессе открытия для вывода, за исключением файлов, указанных именем-файла-1 в другом операторе USE (ИСПОЛЬЗОВАТЬ), описывающем такое же условие;

г) если указано I-O (ВХОДНЫХ-ВЫХОДНЫХ), то соответствующая процедура выполняется при выполнении условия, описанного в операторе USE (ИСПОЛЬЗОВАТЬ), для какого-либо файла, открытого для ввода-вывода или в процессе открытия для ввода-вывода, за исключением файлов, указанных именем-файла-1 в другом операторе USE (ИСПОЛЬЗОВАТЬ), описывающем такое же условие;

д) если указано EXTEND (ДОПОЛНЯЕМЫХ), то соответствующая процедура выполняется при выполнении условия, описанного в операторе USE (ИСПОЛЬЗОВАТЬ), для какого-либо файла, открытого для дополнения или в процессе открытия для дополнения, за исключением файлов, указанных именем-файла-1 в другом операторе USE (ИСПОЛЬЗОВАТЬ), описывающем такое же условие.

(6) После выполнения процедуры, связанной с оператором USE (ИСПОЛЬЗОВАТЬ), управление передается вызывающей программе в системе управления вводом-выводом. Если значение состояния ввода-вывода не указывает на критическую ошибку ввода-вывода, то система управления вводом-выводом возвращает управление оператору, следующему за оператором ввода-вывода, выполнение которого вызвало ошибку. Если значение состояния ввода-вывода указывает на критическую ошибку, то действие определяется реализацией (см. п. 1.3.4 настоящей части).

(7) В процедуре, связанной с оператором USE (ИСПОЛЬЗОВАТЬ), не должны выполняться никакие операторы, которые могут потребовать выполнения процедуры, связанной с другим оператором USE (ИСПОЛЬЗОВАТЬ), вызванной ранее и еще не вернувшей управление вызвавшей ее программе.

4.9. Оператор WRITE (ПИСАТЬ)

4.9.1. Назначение

Оператор WRITE (ПИСАТЬ) включает логическую запись в выходной или входной-выходной файл.

GOST EXPERT ГОСТ 22558-89,

4.9.2. Общий формат

WRITE имя-записи-1 [FROM идентификатор-1]
[INVALID KEY повелительный-оператор-1]
[NOT INVALID KEY повелительный-оператор-2]
[END-WRITE]

ПИСАТЬ имя-записи-1 [ИЗ ПОЛЯ идентификатор-1]
[ПРИ ОШИБКЕ КЛЮЧА повелительный-оператор-1]
[БЕЗ ОШИБКИ КЛЮЧА повелительный-оператор-2]
[КОНЕЦ-ПИСАТЬ]

4.9.3. Синтаксические правила

(1) Имя-записи-1 и идентификатор-1 не должны относиться к одной и той же области памяти.

(2) Имя-записи-1 является именем логической записи в секции файлов раздела данных и может быть уточнено.

(3) Фраза INVALID KEY (ПРИ ОШИБКЕ КЛЮЧА) должна указываться в операторе WRITE (ПИСАТЬ) для файлов, для которых не определена соответствующая процедура USE AFTER STANDARD EXCEPTION (ИСПОЛЬЗОВАТЬ ПОСЛЕ СТАНДАРТНОЙ ПРОЦЕДУРЫ ОШИБКИ).

4.9.4. Общие правила

(1) Файл, указанный именем-файла, связанным с именем-записи-1, должен быть открыт как OUTPUT (ВЫХОДНОЙ) или EXTEND (ДОПОЛНЯЕМЫЙ) ко времени выполнения этого оператора (см. п. 4.4 настоящей части).

(2) Логическая запись, включаемая в файл при успешном выполнении оператора WRITE (ПИСАТЬ), становится недоступной в области записи. Исключение представляют случаи, когда имя-файла, связанное с именем-записи-1, указано во фразе SAME RECORD AREA (ОБЩАЯ ОБЛАСТЬ ЗАПИСИ). Если имя относится к имени-файла, указанному во фразе SAME RECORD AREA (ОБЩАЯ ОБЛАСТЬ ЗАПИСИ), логическая запись доступна программе и как запись файла, связанного с именем-записи-1, и как запись других файлов, указанных в той же фразе SAME RECORD AREA (ОБЩАЯ ОБЛАСТЬ ЗАПИСИ), что и соответствующий выходной файл.

(3) Результат выполнения оператора WRITE (ПИСАТЬ) с фразой FROM (ИЗ ПОЛЯ) эквивалентен выполнению следующих операторов в указанном порядке:

а) оператор

MOVE идентификатор-1 TO имя-записи-1

ПОМЕСТИТЬ идентификатор-1 В имя-записи-1

соответственно правилам, специфицированным в операторе MOVE (ПОМЕСТИТЬ);

GOST EXPERT ГОСТ 22558-89,

б) тот же оператор WRITE (ПИСАТЬ) без фразы FROM (ИЗ ПОЛЯ).

(4) После завершения выполнения оператора WRITE (ПИСАТЬ) информация в области, указанной идентификатором-1, остается доступной, даже если информация в области, указанной именем-записи-1, не является доступной, за исключением случая, определяемого фразой SAME RECORD AREA (ОБЩАЯ ОБЛАСТЬ ЗАПИСИ).

(5) Выполнение оператора WRITE (ПИСАТЬ) не влияет на указатель позиции файла.

(6) Выполнение оператора WRITE (ПИСАТЬ) вызывает обновление состояния ввода-вывода имени-файла, связанного с именем-записи-1 (см. п. 1.3.4 настоящей части).

(7) При выполнении оператора WRITE (ПИСАТЬ) логическая запись передается операционной системе.

(8) Количество позиций литер в записи, указанной именем-записи-1, не должно быть больше наибольшего или меньше наименьшего числа литер, допустимого фразой RECORD IS VARYING (В ЗАПИСИ ПЕРЕМЕННОЕ ЧИСЛО ЛИТЕР), связанной с именем-файла, связанного с именем-записи-1. В любом случае выполнение оператора WRITE (ПИСАТЬ) неуспешно, операция записи не производится, содержимое области записи не меняется, и состояние ввода-вывода файла, связанного с именем-записи-1, принимает значение, указывающее на причину возникновения условия (см. п. 1.3.4 настоящей части).

(9) Если во время выполнения оператора WRITE (ПИСАТЬ) с фразой NOT INVALID KEY (БЕЗ ОШИБКИ КЛЮЧА) не наступает условие ошибки ключа, то управление передается повелительному-оператору-2 следующим образом:

а) если выполнение оператора WRITE (ПИСАТЬ) успешно, то управление передается после того, как запись записана, и после изменения состояния ввода-вывода имени-файла, связанного с именем-записи-1;

б) если выполнение оператора WRITE (ПИСАТЬ) неуспешно не из-за ошибки ключа, то управление передается после обновления состояния ввода-вывода для имени-файла, связанного с именем-записи-1, и после выполнения процедуры, определенной оператором USE AFTER STANDARD EXCEPTION (ИСПОЛЬЗОВАТЬ ПОСЛЕ СТАНДАРТНОЙ ПРОЦЕДУРЫ ОШИБКИ), применимой к имени-файла, связанного с именем-записи-1, если таковая указана.

(10) Фраза END-WRITE (КОНЕЦ-ПИСАТЬ) ограничивает область действия оператора WRITE (ПИСАТЬ) (см. ч. 4, п. 6.4.3).

GOST EXPERT ГОСТ 22558-89,

(11) Если относительный файл открывается как выходной, записи могут быть помещены в файл одним из следующих способов:

а) если доступ последовательный, оператор WRITE (ПИСАТЬ) передает запись системе управления массовой памятью. Первая запись будет иметь относительный номер записи 1, а последующие включаемые записи будут иметь номера записей 2, 3, 4, ... Если фраза RELATIVE KEY (ОТНОСИТЕЛЬНЫЙ КЛЮЧ) была указана для имени-файла, связанного с именем-записи-1, во время выполнения оператора WRITE (ПИСАТЬ) относительный номер переданной записи будет помещен СУМП в данное, определенное как относительный ключ, согласно правилам для оператора MOVE (ПОМЕСТИТЬ) (см. ч. 6, п. 6.1.9);

б) если доступ произвольный | или динамический, | относительный номер, который должен быть связан с записью, находящейся в области записи, должен быть помещен перед выполнением оператора WRITE (ПИСАТЬ) в данное, определенное как относительный ключ. При выполнении оператора WRITE (ПИСАТЬ) эта запись передается СУМП.

(12) Если относительный файл открыт как дополняемый, записи включаются в этот файл. Первая запись, переданная системой управления массовой памятью, имеет относительный номер записи на единицу больший, чем наибольший относительный номер записи, существующей в файле. Следующие записи, переданные СУМП, имеют соответственно большие относительные номера записи. Если для файла, связанного с именем-записи-1, указана фраза RELATIVE KEY (ОТНОСИТЕЛЬНЫЙ КЛЮЧ), во время выполнения оператора WRITE (ПИСАТЬ) относительный номер включаемой записи помещается СУМП в данное, являющееся относительным ключом, согласно правилам для оператора MOVE (ПОМЕСТИТЬ) (см. ч. 6, п. 6.19).

(13) Если файл открывается как входной-выходной и доступ произвольный | или динамический, | по оператору WRITE (ПИСАТЬ) запись вставляется в этот файл. Относительный номер, который должен быть связан с записью, находящейся в области записи, должен быть помещен перед выполнением оператора WRITE (ПИСАТЬ) в данное, определенное как относительный ключ. При выполнении оператора WRITE (ПИСАТЬ) эта запись передается СУМП.

(14) Условие ошибки ключа возникает в следующих случаях:

а) когда при произвольном | или динамическом | доступе относительный ключ определяет запись, которая уже имеется в файле;

б) когда сделана попытка писать запись вне границ, определенных для файла.

GOST EXPERT ГОСТ 22558-89,

(15) При обнаружении условия ошибки ключа выполнение оператора WRITE (ПИСАТЬ) считается неуспешным. Содержимое области записи не изменяется, а состояние ввода-вывода имени файла, связанного с именем-записи-1, устанавливается на значение, указывающее причину этого условия. Выполнение программы продолжается в соответствии с правилами для условия ошибки ключа (см. пп. 1.3.4, 1.3.5 настоящей части).

Часть 9. МОДУЛЬ ИНДЕКСНОГО ВВОДА-ВЫВОДА

1. ВВЕДЕНИЕ В МОДУЛЬ ИНДЕКСНОГО ВВОДА-ВЫВОДА

1.1. **Назначение**

Модуль индексного ввода-вывода обеспечивает возможность произвольного или последовательного доступа к записям файлов массовой памяти. Каждая запись индексного файла однозначно идентифицируется значением одного или более ключей внутри записи.

1.2. **Характеристика уровней**

Уровень 1 индексного ввода-вывода обеспечивает неполные возможности для статьи управления файлом, статьи описания файла и статей параграфа I-O-CONTROL (УПРАВЛЕНИЕ-ВВОДОМ-ВЫВОДОМ). В разделе процедур уровень 1 индексного ввода-вывода обеспечивает ограниченные возможности операторов CLOSE (ЗАКРЫТЬ), OPEN (ОТКРЫТЬ), READ (ЧИТАТЬ), USE (ИСПОЛЬЗОВАТЬ) и WRITE (ПИСАТЬ) и полные возможности оператора DELETE (УДАЛИТЬ).

Уровень 2 индексного ввода-вывода обеспечивает полные возможности для статьи управления файлом, статьи описания файла и статей параграфа I-O-CONTROL (УПРАВЛЕНИЕ-ВВОДОМ-ВЫВОДОМ). В разделе процедур уровень 2 индексного ввода-вывода обеспечивает полные возможности операторов CLOSE (ЗАКРЫТЬ), DELETE (УДАЛИТЬ), OPEN (ОТКРЫТЬ), READ (ЧИТАТЬ), START (ПОДВЕСТИ), USE (ИСПОЛЬЗОВАТЬ) и WRITE (ПИСАТЬ).

1.3. **Понятия языка**

1.3.1. Организация

Файл с индексной организацией — это файл массовой памяти, доступ к записям которого может осуществляться с помощью указанного в записи ключа.

Для каждого данного, являющегося ключом, определенным для записей файла, поддерживается индекс. Каждый индекс представляет собой множество значений соответствующего ключевого данного в каждой записи. Каждый индекс, следовательно, является механизмом, обеспечивающим доступ к любой записи файла.

GOST EXPERT ГОСТ 22558-89,

Каждый индексный файл имеет основной индекс, представляющий основной ключ каждой записи файла. Каждая запись включается в файл, изменяется или удаляется из файла в зависимости от значения основного ключа записи. Каждый основной ключ записи должен быть уникальным и неизменяемым при изменении записи.

Основной ключ записи описывается во фразе RECORD KEY (КЛЮЧ ЗАПИСИ) статьи управления файлом для данного файла.

Дополнительные ключи записи обеспечивают дополнительные средства доступа к записям файла. Эти ключи именуются во фразе ALTERNATE RECORD KEY (ДОПОЛНИТЕЛЬНЫЙ КЛЮЧ ЗАПИСИ) статьи управления файлом. Значение дополнительного ключа записи может дублироваться. Когда эти значения не уникальны, во фразе ALTERNATE RECORD KEY (ДОПОЛНИТЕЛЬНЫЙ КЛЮЧ ЗАПИСИ) указывается фраза DUPLICATES (С ДУБЛИРОВАНИЕМ).

1.3.2. Методы доступа

При индексной организации порядок последовательного доступа является возрастающим на основе значения ключа ссылки согласно основной последовательности файла. Любой ключ, соответствующий файлу, может во время обработки файла устанавливаться как ключ ссылки.

Порядок доступа к множеству записей, имеющих неуникальные значения ключей ссылки, является первоначальным порядком помещения записей в множество. Для установки в индексном файле начальной точки для последовательности последовательных обращений может использоваться оператор START (ПОДВЕСТИ).

При произвольном доступе к файлу операторы ввода-вывода применяются для доступа к записям в порядке, указанном программистом.

При индексной организации программист указывает требуемую запись, помещая значение одного из ее ключей записи в данное, представляющее ключ записи или дополнительный ключ записи.

При динамическом доступе программист может произвольно переходить от последовательного доступа к произвольному и наоборот, применяя соответствующие формы операторов ввода-вывода.

1.3.3. Указатель позиции файла

Указатель позиции файла — это логическое понятие, используемое в этом документе для облегчения точной спецификации следующей записи, к которой должен осуществляться доступ при выполнении заданной последовательности операций ввода-вывода.

GOST EXPERT ГОСТ 22558-89,

На установку указателя позиции файла влияют только операторы CLOSE (ЗАКРЫТЬ), OPEN (ОТКРЫТЬ), READ (ЧИТАТЬ) и START (ПОДВЕСТИ). Понятие указателя позиции файла не имеет смысла для файла, открытого как выходной или как дополняемый.

1.3.4. Состояние ввода-вывода

Состояние ввода-вывода — это логическое понятие, характеризующееся двухсимвольным значением, которое устанавливается для указания состояния операции ввода-вывода во время выполнения операторов CLOSE (ЗАКРЫТЬ), OPEN (ОТКРЫТЬ), READ (ЧИТАТЬ), REWRITE (ОБНОВИТЬ), START (ПОДВЕСТИ) или WRITE (ПИСАТЬ) перед выполнением любого повелительного оператора, связанного с этим оператором ввода-вывода, и перед выполнением любой применимой процедуры USE AFTER STANDARD EXCEPTION (ИСПОЛЬЗОВАТЬ ПОСЛЕ СТАНДАРТНОЙ ПРОЦЕДУРЫ ОШИБКИ). Значение состояния ввода-вывода доступно Кобол-программе посредством фразы FILE STATUS (СОСТОЯНИЕ ФАЙЛА) в статье управления файлом.

Состояние ввода-вывода определяет также, будет ли выполняться процедура USE AFTER STANDARD EXCEPTION (ИСПОЛЬЗОВАТЬ ПОСЛЕ СТАНДАРТНОЙ ПРОЦЕДУРЫ ОШИБКИ). Если возникает любое условие, отличное от тех, которые определены ниже как «успешное завершение», может выполняться указанная процедура по правилам, заданным для оператора USE (ИСПОЛЬЗОВАТЬ). Если возникает одно из условий «успешное завершение», никакая процедура такого типа не будет выполняться (*п. 4.8 настоящей части*).

Некоторые классы значений состояния ввода-вывода задают критические условия ошибки. К таким значениям относятся значения, которые начинаются с цифры 3 или 4, а также значения, начинающиеся с цифры 9, которые определяются как критические реализацией. Если значение состояния ввода-вывода для операции ввода-вывода задаст такое условие ошибки, реализацией определяются действия, которые предпринимаются после выполнения применимой процедуры USE AFTER STANDARD EXCEPTION (ИСПОЛЬЗОВАТЬ ПОСЛЕ СТАНДАРТНОЙ ПРОЦЕДУРЫ ОШИБКИ), или, если ни одна такая процедура не применима, после завершения стандартной системной процедуры обработки ошибок ввода-вывода.

Состояние ввода-вывода задает одно из следующих условий, возникающих после завершения операции ввода-вывода:

(1) Успешное завершение. Оператор ввода-вывода выполнился успешно;

GOST EXPERT ГОСТ 22558-89,

(2) В конце. Оператор последовательного чтения был выполнен неуспешно из-за того, что возникло условие конца файла;

(3) Ошибка ключа. Оператор ввода-вывода выполнен неуспешно из-за того, что возникло условие ошибки ключа;

(4) Постоянная ошибка. Оператор ввода-вывода выполнен неуспешно в результате ошибки, которая исключает дальнейшую обработку файла. Выполняются все заданные процедуры обработки ошибочных ситуаций. Условие постоянной ошибки остается действующим на все последующие операции ввода-вывода файла до тех пор, пока не будут вызваны определенные реализацией средства для устранения условия постоянной ошибки;

(5) Логическая ошибка. Оператор ввода-вывода выполнен неуспешно из-за недопустимой последовательности операций ввода-вывода, выполняемых над файлом, или в результате нарушения ограничений, заданных пользователем;

(6) Ошибка, определяемая реализацией. Оператор ввода-вывода выполнен неуспешно в результате возникновения условия, определенного реализацией.

Ниже приводится список значений, помещаемых в состояние ввода-вывода для перечисленных выше условий, возникающих в результате выполнения операций ввода-вывода для индексного файла. Если применимо более одного значения, значение, которое помещается в состояние ввода-вывода, определяется реализацией.

(1) Успешное завершение

а) Состояние ввода-вывода—00. Оператор ввода-вывода выполнился успешно и нет никакой другой доступной информации об операции ввода-вывода.

б) Состояние ввода-вывода=02. Оператор ввода-вывода успешно выполнился, но обнаружено дублирование ключа.

1) Для оператора READ (ЧИТАТЬ) значение ключа для текущего ключа ссылки равно значению такого же ключа в следующей записи в текущем ключе ссылки.

2) Для оператора REWRITE (ОБНОВИТЬ) или WRITE (ПИСАТЬ) только что записанная запись создает дублирующееся значение ключа хотя бы для одного дополнительного ключа записи, для которого дублирование разрешено.

в) Состояние ввода-вывода=04. Оператор READ (ЧИТАТЬ) выполнился успешно, но длина обрабатываемой записи не соответствует фиксированным свойствам этого файла.

г) Состояние ввода-вывода=05. Оператор OPEN (ОТКРЫТЬ) успешно выполнился, но указанный в нем необязательный файл во время выполнения оператора OPEN (ОТКРЫТЬ)

GOST EXPERT ГОСТ 22558-89,

отсутствует. Если режим открытия для ввода-вывода или дополнения, файл будет создаваться.

(2) Условие «в конце» с неуспешным завершением

а) Состояние ввода-вывода=10. Делается попытка выполнить последовательный оператор READ (ЧИТАТЬ), а в файле не существует следующей логической записи из-за того, что:

1) достигнут конец файла;

2) делается попытка выполнить последовательный оператор READ (ЧИТАТЬ) в первый раз для отсутствующего необязательного входного файла.

(3) Условие ошибки ключа при неуспешном завершении

а) Состояние ввода-вывода=21. Ошибка в последовательности при последовательном доступе к индексному файлу. Нарушено требование возрастающей последовательности значений ключа записи или значение основного ключа записи изменилось между успешным выполнением оператора READ (ЧИТАТЬ) и выполнением следующего оператора REWRITE (ОБНОВИТЬ) для этого файла (п. 4.9 настоящей части).

б) Состояние ввода-вывода=22. Сделана попытка обновить или записать запись, которая создает дублирующийся основной ключ записи или дублирующийся дополнительный ключ записи без фразы DUPLICATES (С ДУБЛИРОВАНИЕМ) в индексном файле.

в) Состояние ввода-вывода=23. Это условие возникает, если:

1) сделана попытка произвольного доступа к записи, которой нет в файле; или

2) сделана попытка выполнить оператор START (ПОДВЕСТИ) или READ (ЧИТАТЬ) с произвольным доступом для необязательного входного файла, который отсутствует.

г) Состояние ввода-вывода=24. Сделана попытка занесения записей в индексный файл вне его границ, определенных внешним образом. Способ определения границ указывается реализацией.

(1) Условие постоянной ошибки с неуспешным завершением

а) Состояние ввода-вывода=30. Возникла постоянная ошибка и нет никакой другой доступной информации об операции ввода-вывода.

б) Состояние ввода-вывода=35. Постоянная ошибка возникла из-за того, что делается попытка выполнить оператор OPEN (ОТКРЫТЬ) с фразой INPUT (ВХОДНОЙ), I-O (ВХОДНОЙ-ВЫХОДНОЙ) или EXTEND (ДОПОЛНЯЕМЫЙ) для файла, который обязательно должен присутствовать, но не присутствует.

GOST EXPERT ГОСТ 22558-89,

в) Состояние ввода-вывода=37. Постоянная ошибка возникла из-за того, что оператор OPEN (ОТКРЫТЬ) выдан для файла, который не поддерживает режим открытия, заданный в операторе OPEN (ОТКРЫТЬ). Возможны следующие нарушения:

1) задана фраза EXTEND (ДОПОЛНЯЕМЫЙ) или OUTPUT (ВЫХОДНОЙ), а файл не допускает операций записи;

2) задана фраза I-O (ВХОДНОЙ-ВЫХОДНОЙ), а файл не допускает операции ввода и вывода, которые разрешены для индексного файла, открываемого в режиме ввода-вывода;

3) задана фраза INPUT (ВХОДНОЙ), а файл не допускает операции чтения.

г) Состояние ввода-вывода=38. Постоянная ошибка возникла из-за того, что выдан оператор OPEN (ОТКРЫТЬ) для файла, ранее закрытого с замком.

д) Состояние ввода-вывода=39. Оператор OPEN (ОТКРЫТЬ) завершился неуспешно из-за обнаруженного для этого файла несоответствия фиксированных свойств файла и свойств, заданных в программе.

(5) Условие логической ошибки с неуспешным завершением

а) Состояние ввода-вывода=41. Оператор OPEN (ОТКРЫТЬ) выдан для открытого файла.

б) Состояние ввода-вывода=42. Оператор CLOSE (ЗАКРЫТЬ) выдан для неоткрытого файла.

в) Состояние ввода-вывода=43. При последовательном доступе последний оператор ввода-вывода, выполненный для файла до выполнения оператора DELETE (УДАЛИТЬ) или REWRITE (ОБНОВИТЬ), не является успешно выполненным оператором READ (ЧИТАТЬ).

г) Состояние ввода-вывода=44. Нарушение границ возникает по следующим причинам:

1) совершена попытка записать или обновить запись, которая длиннее максимально допустимой или короче минимально допустимой в соответствии с фразой RECORD IS VARYING (В ЗАПИСИ ПЕРЕМЕННОЕ ЧИСЛО), связанной с именем файла;

2) на уровне 1 сделана попытка обновить запись индексного файла, а размер заменяющей записи отличен от размера заменяемой записи.

д) Состояние ввода-вывода=46. Выдан оператор последовательного чтения для файла, открытого в режиме ввода или ввода-вывода и не была установлена правильная следующая запись по одной из следующих причин:

GOST EXPERT ГОСТ 22558-89,

1) предыдущий оператор START (ПОДВЕСТИ) закончился неуспешно, или

2) предыдущий оператор READ (ЧИТАТЬ) закончился неуспешно, но не вызвал условие «в конце»;

3) предыдущий оператор READ (ЧИТАТЬ) вызвал условие «в конце».

е) Состояние ввода-вывода=47. Был выдан оператор READ (ЧИТАТЬ) или START (ПОДВЕСТИ) для файла, не открытого в режиме ввода или ввода-вывода.

ж) Состояние ввода-вывода=48. Был выдан оператор WRITE (ПИСАТЬ) для файла, не открытого в режиме ввода-вывода, вывода или дополнения.

з) Состояние ввода-вывода=49. Был выдан оператор DELETE (УДАЛИТЬ) или REWRITE (ОБНОВИТЬ) для файла, не открытого в режиме ввода-вывода.

(6) Определяемое реализацией условие с неуспешным завершением

а) Состояние ввода-вывода=9х. Существуют определяемые реализацией условия. Эти условия не должны дублировать никакие условия, определенные для значений от 00 до 49 состояния ввода-вывода. Значение х определяется реализацией.

1.3.5. Условие ошибки ключа

Условие ошибки ключа может возникнуть в результате выполнения операторов DELETE (УДАЛИТЬ), READ (ЧИТАТЬ), REWRITE (ОБНОВИТЬ), START (ПОДВЕСТИ) или WRITE (ПИСАТЬ). Когда возникает условие ошибки ключа, оператор ввода-вывода, вызвавший эту ситуацию, является неуспешным и файл не изменяется (пп. 4.3, 4.5—4.7, 4.9 настоящей части).

Если условие ошибки ключа возникает после выполнения операции ввода-вывода, определенной в операторе ввода-вывода, то происходят следующие действия в указанном порядке:

(1) состояние ввода-вывода определителя файла, связанного с оператором, устанавливается в значение, определяющее условие ошибки ключа (см. п. 1.3.4 настоящей части);

(2) если в операторе ввода-вывода указана фраза INVALID KEY (ПРИ ОШИБКЕ КЛЮЧА), то никакая процедура USE AFTER EXCEPTION (ИСПОЛЬЗОВАТЬ ПОСЛЕ ПРОЦЕДУРЫ ОШИБКИ), связанная с определителем файла, не выполняется, и управление передается повелительному оператору, указанному во фразе INVALID KEY (ПРИ ОШИБКЕ КЛЮЧА). Выполнение продолжается в соответствии с правилами для каждого оператора, указанного в этом повелительном операторе. Если выполняется оператор ветвления процедуры или условный оператор, который вызывает явную передачу управления, то управление передается в

GOST EXPERT ГОСТ 22558-89,

соответствии с правилами для этого оператора; в противном случае после завершения выполнения повелительного оператора, указанного во фразе INVALID KEY (ПРИ ОШИБКЕ КЛЮЧА), управление передается в конец оператора ввода-вывода, а фраза NOT INVALID KEY (БЕЗ ОШИБКИ КЛЮЧА), если она указана, игнорируется;

(3) если в операторе ввода-вывода фраза INVALID KEY (ПРИ ОШИБКЕ КЛЮЧА) не указана, то с определителем файла должна быть связана процедура USE AFTER EXCEPTION (ИСПОЛЬЗОВАТЬ ПОСЛЕ ПРОЦЕДУРЫ ОШИБКИ), и эта процедура выполняется, а управление передается в соответствии с правилами оператора USE (ИСПОЛЬЗОВАТЬ). Если указана фраза NOT NIVALID KEY (БЕЗ ОШИБКИ КЛЮЧА), то она игнорируется (п. 4.8 настоящей части).

Если после выполнения операции ввода-вывода, указанной в операторе ввода-вывода, условия ошибки ключа нет, то фраза INVALID KEY (ПРИ ОШИБКЕ КЛЮЧА), если она указана, игнорируется. Состояние ввода-вывода определителя файла, связанного с оператором, обновляется, и выполняются следующие действия:

(1) если возникает условие ошибки, которое не является условием ошибки ключа, то управление передается в соответствии с правилами оператора USE (ИСПОЛЬЗОВАТЬ), а затем выполняется процедура USE AFTER EXCEPTION (ИСПОЛЬЗОВАТЬ ПОСЛЕ ПРОЦЕДУРЫ ОШИБКИ), связанная с определителем файла (п. 4.8 настоящей части);

(2) если условие ошибки не возникает, то управление передается в конец оператора ввода-вывода или повелительному оператору, указанному во фразе NOT INVALID KEY (БЕЗ ОШИБКИ КЛЮЧА). В последнем случае выполнение продолжается в соответствии с правилами для каждого оператора, указанного в повелительном операторе. Если это оператор ветвления процедуры или условный оператор, который вызывает явную передачу управления, то управление передается в соответствии с правилами для этого оператора; в противном случае, после завершения выполнения повелительного оператора, указанного во фразе NOT INVALID KEY (БЕЗ ОШИБКИ КЛЮЧА), управление передается в конец оператора ввода-вывода.

1.3.6. Условие «в конце»

Условие «в конце» может возникнуть в результате выполнения оператора READ (ЧИТАТЬ) (п. 4.5 настоящей части).

1.3.7. Условие противоречия свойств файла

Условие противоречия свойств файла может возникнуть в результате выполнения операторов OPEN (ОТКРЫТЬ), REWRITE (ОБНОВИТЬ) или WRITE (ПИСАТЬ). При возникновении условия противоречия свойств файла выполнение оператора ввода-вы-

GOST EXPERT ГОСТ 22558-89,

вода, который обнаружил это условие, считается неуспешным и файл не изменяется (пп. 4.4, 4.6, 4.9 настоящей части).

При обнаружении условия противоречия свойств файла выполняются следующие действия в указанном ниже порядке:

(1) в состояние ввода-вывода, связанное с именем-файла, помещается значение, указывающее на условие противоречия свойств файла (см. п. 1.3.4 настоящей части);

(2) если для данного имени файла задана процедура USE AFTER EXCEPTION (ИСПОЛЬЗОВАТЬ ПОСЛЕ ПРОЦЕДУРЫ ОШИБКИ), выполняется указанная процедура.

2. РАЗДЕЛ ОБОРУДОВАНИЯ В МОДУЛЕ ИНДЕКСНОГО ВВОДА-ВЫВОДА

2.1. **Секция ввода-вывода**

Информация, относящаяся к секции ввода-вывода, находится в ч. 7, п. 2.1.

2.2. **Параграф FILE-CONTROL (УПРАВЛЕНИЕ-ФАЙЛАМИ)**

Информация о параграфе FILE-CONTROL (УПРАВЛЕНИЕ-ФАЙЛАМИ) находится в ч. 7, п. 2.2.

2.3. **Статья управления файлом**

2.3.1. Назначение

Статья управления файлом объявляет существенные физические свойства индексного файла.

2.3.2. Общий формат

SELECT [OPTIONAL] имя-файла-1

ASSIGN TO { имя-реализации-1 / литерал-1 } ...

[RESERVE целое [AREA / AREAS]]

[ORGANIZATION IS] INDEXED

[ACCESS MODE IS { SEQUENTIAL / RANDOM / [DYNAMIC] }]

RECORD KEY IS имя-данного-1

[ALTERNATE RECORD KEY IS имя-данного-2 [WITH DUPLICATES]] ...

[FILE STATUS IS имя-данного-3].

GOST EXPERT ГОСТ 22558-89,

ДЛЯ [НЕОБЯЗАТЕЛЬНОГО] имя-файла-1

НАЗНАЧИТЬ { имя-реализации-1 / литерал-1 } ...

[РЕЗЕРВИРОВАТЬ целое-1 ОБЛАСТЕЙ]

[ОРГАНИЗАЦИЯ] ИНДЕКСНАЯ

[ДОСТУП { ПОСЛЕДОВАТЕЛЬНЫЙ / ПРОИЗВОЛЬНЫЙ / ДИНАМИЧЕСКИЙ }]

КЛЮЧ ЗАПИСИ имя-данного-1

[ДОПОЛНИТЕЛЬНЫЙ КЛЮЧ ЗАПИСИ имя-данного-2 [С ДУБЛИРОВАНИЕМ]] ...

[СОСТОЯНИЕ ФАЙЛА имя-данного-3].

2.3.3. Синтаксические правила

(1) В статье управления файлом фраза SELECT (ДЛЯ) должна указываться первой. Фразы, которые следуют за фразой SELECT (ДЛЯ), могут появляться в любом порядке.

(2) Каждое имя-файла из раздела данных должно быть определено в параграфе FILE-CONTROL (УПРАВЛЕНИЕ-ФАЙЛАМИ) только один раз. Каждое имя-файла, указанное в фразе SELECT (ДЛЯ), должно иметь статью описания файла в разделе данных той же самой программы.

(3) Литерал-1 должен быть нечисловым литералом и не должен быть стандартной константой. Значение и правила для допустимого содержимого имени-реализации-1 и значения литерала-1 определяются реализацией.

2.3.4. Общие правила

(1) Если определитель файла, на который ссылается имя-файла-1, является внешним определителем файла (ч. 10, п. 4.5), все статьи управления файлом в единице исполнения, которые ссылаются на этот определитель файла, должны иметь:

а) одну и ту же спецификацию фразы OPTIONAL (НЕОБЯЗАТЕЛЬНОГО);

б) корректную спецификацию для имени-реализации-1 или литерала-1 во фразе ASSIGN (НАЗНАЧИТЬ). Правила корректности имени-реализации-1 и литерала-1 определяются реализацией;

GOST EXPERT ГОСТ 22558-89,

в) одно и то же значение целого-1 во фразе RESERVE (РЕЗЕРВИРОВАТЬ);

г) одну и ту же организацию;

д) один и тот же метод доступа.

е) ту же статью описания данных для имени-данного-1 с тем же относительным размещением в соответствующей записи;

ж) ту же статью описания данных для имени-данного-2, то же относительное размещение в соответствующей записи, то же число дополнительных ключей записи и ту же фразу DUPLICATES (С ДУБЛИРОВАНИЕМ).

(2) Для данных внешней среды принимается внутренний набор литер.

(3) Для индексного файла предполагается программный алфавит, связанный с внутренним набором литер. Он определяет последовательность значений заданного ключа ссылки, используемого для последовательной обработки файла.

(4) Фраза OPTIONAL (НЕОБЯЗАТЕЛЬНОГО) применима только к файлам, открытым в режиме ввода, ввода-вывода или дополнения. Ее указание требуется для файлов, которые могут отсутствовать во время выполнения объектной программы.

(5) Фраза ASSIGN (НАЗНАЧИТЬ) задает связь между файлом, на который ссылается имя-файла-1, и запоминающей средой, на которую ссылается имя-реализации-1 или литерал-1.

(6) Фраза RESERVE (РЕЗЕРВИРОВАТЬ) для модуля индексного ввода-вывода та же, что и для модуля последовательного ввода-вывода. Описание фразы RESERVE (РЕЗЕРВИРОВАТЬ) находится в ч. 7, п. 2.9).

(7) Фраза FILE STATUS (СОСТОЯНИЕ ФАЙЛА) для модуля индексного ввода-вывода та же, что и для модуля последовательного ввода-вывода. Описание фразы FILE STATUS (СОСТОЯНИЕ ФАЙЛА) находится в ч. 7, п. 2.5. Содержимое данного, связанного с фразой FILE STATUS (СОСТОЯНИЕ ФАЙЛА) индексного файла, определяется в п. 1.3.4 настоящей части.

(8) Далее представлены фразы ACCESS MODE (ДОСТУП), ALTERNATE RECORD KEY (ДОПОЛНИТЕЛЬНЫЙ КЛЮЧ ЗАПИСИ), ORGANIZATION IS INDEXED (ОРГАНИЗАЦИЯ ИНДЕКСНАЯ), RECORD KEY (КЛЮЧ ЗАПИСИ),

2.4. Фраза ACCESS MODE (ДОСТУП)

2.4.1. Назначение

Фраза ACCESS MODE (ДОСТУП) задает порядок, в котором осуществляется доступ к записям в файле.

GOST EXPERT ГОСТ 22558-89,

2.4.2. Общий формат

ACCESS MODE IS { SEQUENTIAL / RANDOM / DYNAMIC }

ДОСТУП { ПОСЛЕДОВАТЕЛЬНЫЙ / ПРОИЗВОЛЬНЫЙ / ДИНАМИЧЕСКИЙ }

2.4.3. Синтаксические правила

(1) Фраза ACCESS MODE IS RANDOM (ДОСТУП ПРОИЗВОЛЬНЫЙ) не должна употребляться для имен-файлов, указанных во фразах USING (ИСПОЛЬЗУЯ) или GIVING (ПОЛУЧАЯ) операторов SORT (СОРТИРОВАТЬ) и MERGE (СЛИТЬ).

2.4.4. Общие правила

(1) Если фраза ACCESS MODE (ДОСТУП) не задана, предполагается последовательный доступ.

(2) Если доступ последовательный, записи становятся доступными в последовательности, диктуемой организацией файла. Для индексного файла эта последовательность является возрастающей по заданному ключу ссылки соответственно алфавитной упорядоченности файла.

(3) Если доступ произвольный, значение ключа записи для индексного файла указывает требуемую запись.

(4) Если доступ динамический, доступ к записям файла может быть последовательным и (или) произвольным.

(5) Если соответствующий определитель файла является внешним определителем файла, то каждая статья управления файлом в единице исполнения, связанная с этим определителем файла, должна указывать один и тот же метод доступа.

2.5. Фраза ALTERNATE RECORD KEY (ДОПОЛНИТЕЛЬНЫЙ КЛЮЧ ЗАПИСИ)

2.5.1. Назначение

Фраза ALTERNATE RECORD KEY (ДОПОЛНИТЕЛЬНЫЙ КЛЮЧ ЗАПИСИ) указывает путь доступа к записям индексного файла по дополнительному ключу записи.

2.5.2. Общий формат

ALTERNATE RECORD KEY IS имя-данного-1 [WITH DUPLICATES]

GOST EXPERT ГОСТ 22558-89,

ДОПОЛНИТЕЛЬНЫЙ КЛЮЧ ЗАПИСИ имя-данного-1 [С ДУБЛИРОВАНИЕМ]

2.5.3. Синтаксические правила

(1) Имя-данного-1 может уточняться.

(2) Имя-данного-1 должно определяться как буквенно-цифровое данное в статье описание записи, соответствующей имени-файла, которой подчинена фраза ALTERNATE RECORD KEY (ДОПОЛНИТЕЛЬНЫЙ КЛЮЧ ЗАПИСИ).

(3) Имя-данного-1 не должно относиться к групповому данному, содержащему переменно повторяющееся данное.

(4) Имя-данного-1 не должно относиться к данному, самая левая позиция литеры которого соответствует самой левой позиции литеры основного ключа записи или других дополнительных ключей записи, связанных с этим файлом.

(5) Если индексный файл содержит записи переменной длины, каждый дополнительный ключ записи должен содержаться в первых x позициях литер записи, где x равно минимальному размеру записи, определенному для файла (см. ч. 7, п. 3.8).

2.5.4. Общие правила

(1) Фраза ALTERNATE RECORD KEY (ДОПОЛНИТЕЛЬНЫЙ КЛЮЧ ЗАПИСИ) указывает дополнительный ключ записи для файла, к которому относится эта фраза.

(2) Описания данных имя-данного-1, а также их относительное размещение внутри записи должны быть такими же, как и при создании файла. Число дополнительных ключей для файла должно быть таким же, как и при создании файла.

(3) Фраза WITH DUPLICATES (С ДУБЛИРОВАНИЕМ) указывает, что значение дополнительного ключа записи может дублироваться в некоторых записях файла. Если фраза WITH DUPLICATES (С ДУБЛИРОВАНИЕМ) не указана, значение дополнительного ключа записи не может дублироваться в записях файла.

(4) Если файл имеет несколько статей описания записи, достаточно описать имя-данного-1 в одной из этих статей описания записи. На идентичные позиции литер, представленные именем-данного-1 в какой-либо статье описания записи, неявно ссылаются ключи для остальных статей описания записи этого файла.

(5) Если соответствующий определитель файла является внешним определителем файла, каждая статья управления файлом в единице исполнения, соответствующая этому определителю файла, должна определять ту же статью описания данных для имени-данного-1, то же относительное размещение внутри соответствующей записи, то же количество дополнительных ключей записи и ту же фразу WITH DUPLICATES (С ДУБЛИРОВАНИЕМ).

GOST EXPERT ГОСТ 22558-89,

2.6. **Фраза ORGANIZATION IS INDEXED (ОРГАНИЗАЦИЯ ИНДЕКСНАЯ)**

2.6.1. Назначение

Фраза ORGANIZATION IS INDEXED (ОРГАНИЗАЦИЯ ИНДЕКСНАЯ) указывает, что логической структурой файла является индексная организация.

2.6.2. Общий формат

[ORGANIZATION IS] INDEXED

[ОРГАНИЗАЦИЯ] ИНДЕКСНАЯ

2.6.3. Общие правила

(1) Фраза ORGANIZATION IS INDEXED (ОРГАНИЗАЦИЯ ИНДЕКСНАЯ) указывает, что логической структурой файла является индексная организация. Организация файла устанавливается во время создания файла и не может быть изменена в дальнейшем.

(2) Индексная организация это постоянная логическая структура файла, в которой каждая запись идентифицируется значением одного или более ключей в этой записи.

2.7. **Фраза RECORD KEY (КЛЮЧ ЗАПИСИ)**

2.7.1. Назначение

Фраза RECORD KEY (КЛЮЧ ЗАПИСИ) указывает путь доступа по основному ключу записи к записям в индексном файле.

2.7.2. Общий формат

RECORD KEY IS имя-данного-1

КЛЮЧ ЗАПИСИ имя-данного-1

2.7.3. Синтаксические правила

(1) Имя-данного-1 может уточняться.

(2) Имя-данного-1 должно ссылаться на данное буквенно-цифровой категории в статье описания записи, связанной с именем файла, которому подчинена фраза RECORD KEY (КЛЮЧ ЗАПИСИ).

(3) Имя-данного-1 не должно относиться к групповому данному, содержащему переменно повторяющееся данное.

(4) Если индексный файл содержит записи переменной длины, основной ключ записи должен содержаться в первых x позициях литеры записи, где x равняется минимальному размеру записи, указанному для файла (см. ч. 7, п. 3.8).

2.7.4. Общие правила

(1) Фраза RECORD KEY (КЛЮЧ ЗАПИСИ) указывает основной ключ записи для файла, к которому относится фраза. Значения основного ключа записи должны быть уникальными в записях файла.

GOST EXPERT ГОСТ 22558-89,

(2) Описание данного, соотнесенного имени-данного-1, так же как и его относительное местонахождение в записи, должно быть таким же, какое использовалось при создании файла.

(3) Если для файла имеется более одной статьи описания записи, имя-данного-1 может быть только в одной из этих статей описания записи. Идентичные позиции литер, соотнесенные имени-данного-1, в любой другой статье описания записи неявно считаются ключами для всех остальных статей описания записи этого файла.

(4) Если соответствующий файлу определитель файла является внешним определителем, все статьи описания файла в единице исполнения, связанные с этим определителем файла, должны указывать одну и ту же статью описания данного для имени-данного-1 с одним и тем же относительным местоположением в соответствующей записи.

2.8. **Параграф I-O-CONTROL (УПРАВЛЕНИЕ-ВВОДОМ-ВЫВОДОМ)**

2.8.1. Назначение

Параграф I-O-CONTROL (УПРАВЛЕНИЕ-ВВОДОМ-ВЫВОДОМ) указывает контрольные точки для перепрогона, а также общие области памяти, которые могут совместно использоваться различными файлами. Фраза RERUN (ПЕРЕПРОГОН) параграфа I-O-CONTROL (УПРАВЛЕНИЕ-ВВОДОМ-ВЫВОДОМ) рассматривается в настоящем стандарте как устаревший элемент и будет удалена в следующей редакции стандарта.

2.8.2. Общий формат

I-O-CONTROL.

```
[ [ RERUN ON { имя-файла-1      } EVERY
             { имя-реализации-1 }
        { целое-1 RECORDS OF имя-файла-2 } ] ]...
        { целое-2 CLOCK-UNITS            }
        { имя-условия-1                  }
  [SAME [RECORD] AREA FOR имя-файла-3
                          {имя-файла-4}...]...]
```

УПРАВЛЕНИЕ-ВВОДОМ-ВЫВОДОМ.

```
[ [ПЕРЕПРОГОН
  НА { имя-файла-1      }
     { имя-реализации-1 }
        { КАЖДЫЕ целое-1 ЗАПИСЕЙ имя-файла-2 } ] ]...
        { КАЖДЫЕ целое-2 ЕДИНИЦ-ВРЕМЕНИ      }
        { КАЖДОЕ имя-условия-1                }
```


GOST EXPERT ГОСТ 22558-89,

[ОБЩАЯ ОБЛАСТЬ [ЗАПИСИ] ДЛЯ имя-файла-3
{имя-файла-4} . . .] . . . .]

2.8.3. Общие правила

(1) Фраза RERUN (ПЕРЕПРОГОН) для модуля индексного ввода-вывода является подмножеством фразы RERUN (ПЕРЕПРОГОН) для модуля последовательного ввода-вывода. Поэтому спецификации фразы RERUN (ПЕРЕПРОГОН) см. ч. 7, п. 2.12.

(2) Фраза SAME (ОБЩАЯ ОБЛАСТЬ) для модуля индексного ввода-вывода такая же, как и фраза SAME (ОБЩАЯ ОБЛАСТЬ) для модуля последовательного ввода-вывода. Поэтому соответствующие спецификации см. ч. 7, п. 2.13.

3. РАЗДЕЛ ДАННЫХ В МОДУЛЕ ИНДЕКСНОГО ВВОДА-ВЫВОДА

3.1. **Секция файлов**

Информацию о секции файлов см. в ч. 7, п. 3.1.

3.2. **Статья описания файла**

3.2.1. Назначение

Статья описания файла содержит применимую к индексному файлу информацию о физической структуре, идентификации и именах записей.

3.2.2. Общий формат

FD имя-файла-1

[BLOCK CONTAINS [[целое-1 TO]] целое-2 { RECORDS / CHARACTERS }]

[RECORD { CONTAINS целое-3 CHARACTERS / [IS VARYING IN SIZE [[FROM целое-4] [TO целое-5] CHARACTERS] [DEPENDING ON имя-данного-1]] / CONTAINS целое-6 TO целое-7 CHARACTERS }]

[LABEL { RECORD IS / RECORDS ARE } { STANDARD / OMITTED }]

[VALUE OF {имя-реализации-1 IS { имя-данного-2 / литерал-1 }} . . .]

[DATA { RECORD IS / RECORDS ARE } {имя-данного-3} . . .].

GOST EXPERT ГОСТ 22558-89,

ОФ имя-файла-1

[В БЛОКЕ [[ОТ целое-1 ДО] целое-2 {ЗАПИСЕЙ / ЛИТЕР}]

[В ЗАПИСИ {целое-3 ЛИТЕР / [ПЕРЕМЕННОЕ ЧИСЛО [[ОТ целое-4] [ДО целое-5] ЛИТЕР] [В ЗАВИСИМОСТИ ОТ имя-данного-1]] / ОТ целое-6 ДО целое-7 ЛИТЕР}]

[МЕТКИ {СТАНДАРТНЫ / ОПУЩЕНЫ}]

[{ЗНАЧЕНИЕ / ЗНАЧ} {имя-реализации-1 {имя-данного-2 / литерал-1}} ...]

[ЗАПИСИ ДАННЫХ {имя-данного-3} ...].

3.2.3. Синтаксические правила

(1) Индикатор уровня FD (ОФ) идентифицирует начало статьи описания файла и должен предшествовать имени-файла-1.

(2) Фразы, которые следуют за именем-файла-1, могут задаваться в любом порядке.

(3) Одна или несколько статей описания записи должны следовать за статьей описания файла.

3.2.4. Общие правила

(1) Статья описания файла связывает имя-файла-1 с определителем файла.

(2) Фраза BLOCK CONTAINS (В БЛОКЕ) для модуля индексного ввода-вывода такая же, как и фраза BLOCK CONTAINS (В БЛОКЕ) для модуля последовательного ввода-вывода. Поэтому спецификации фразы BLOCK CONTAINS (В БЛОКЕ) см. ч. 7, п. 3.3.

(3) Фраза DATA RECORDS (ЗАПИСИ ДАННЫХ) для модуля индексного ввода-вывода такая же, как и для модуля последовательного ввода-вывода. Описание фразы DATA RECORDS (ЗАПИСИ ДАННЫХ) находится в ч. 7, п. 3.5. Фраза DATA RECORDS (ЗАПИСИ ДАННЫХ) рассматривается в настоящем стандарте как устаревший элемент и будет удалена в следующей редакции стандарта.

(4) Фраза LABEL RECORDS (МЕТКИ) для модуля индексного ввода-вывода такая же, как и для модуля последовательного ввода-вывода. Описание фразы LABEL RECORDS (МЕТКИ) находится в ч. 7, п. 3.6. Фраза LABEL RECORDS (МЕТКИ) рассмат-

GOST EXPERT ГОСТ 22558-89,

ривается в настоящем стандарте как устаревший элемент языка и будет удалена в следующей редакции стандарта.

(5) Фраза RECORD (В ЗАПИСИ) для модуля индексного ввода-вывода такая же, как и для модуля последовательного ввода-вывода. Описание фразы RECORD (В ЗАПИСИ) находится в ч. 7, п. 3.8.

(6) Фраза VALUE OF (ЗНАЧЕНИЕ) для модуля индексного ввода-вывода такая же, как и для модуля последовательного ввода-вывода. Описание фразы VALUE OF (ЗНАЧЕНИЕ) находится в ч. 7, п. 3.9. Фраза VALUE OF (ЗНАЧЕНИЕ) рассматривается в настоящем стандарте как устаревший элемент языка и будет удалена в следующей редакции стандарта.

4. РАЗДЕЛ ПРОЦЕДУР В МОДУЛЕ ИНДЕКСНОГО ВВОДА-ВЫВОДА

4.1. Общее описание

Если в исходной Кобол-программе имеется оператор USE (ИСПОЛЬЗОВАТЬ) модуля индексного ввода-вывода, раздел процедур содержит декларативные процедуры. Ниже приводится общий формат раздела процедур для случая, когда оператор USE (ИСПОЛЬЗОВАТЬ) указан.

```
PROCEDURE DIVISION.
DECLARATIVES.
  {имя-секции SECTION.
  оператор USE.
  [имя-параграфа.
    [предложение] ... ] ... }...
END DECLARATIVES.
  {имя-секции SECTION.
  [имя-параграфа.
    [предложение] ... ] ... }...
РАЗДЕЛ ПРОЦЕДУР.
ДЕКЛАРАТИВЫ.
  {СЕКЦИЯ имя-секции.
  оператор ИСПОЛЬЗОВАТЬ.
  [имя-параграфа.
    [предложение] ... } ... }...
КОНЕЦ ДЕКЛАРАТИВ.
  {СЕКЦИЯ имя-секции.
  [имя-параграфа.
    [предложение] ... ] ... }...
```


GOST EXPERT ГОСТ 22558-89,

4.2. Оператор CLOSE (ЗАКРЫТЬ)

4.2.1. Назначение

Оператор CLOSE (ЗАКРЫТЬ) завершает обработку файла, возможно с замком.

4.2.2. Общий формат

CLOSE {имя-файла-1 [WITH LOCK]}...

ЗАКРЫТЬ {имя-файла-1 [С ЗАМКОМ]}...

4.2.3. Синтаксическое правило

Файлы, перечисленные в операторе CLOSE (ЗАКРЫТЬ), могут иметь различную организацию и доступ.

4.2.4. Общие правила

(1) Оператор CLOSE (ЗАКРЫТЬ) может быть использован только для файла, который был открыт.

(2) Индексные файлы классифицируются как принадлежащие к категории непоследовательных однотомных или многотомных файлов. Результаты выполнения оператора CLOSE (ЗАКРЫТЬ) для этой категории файлов приведены ниже.

Формат оператора CLOSE (ЗАКРЫТЬ)	Категория файла
	Непоследовательный однотомный (многотомный) файл
CLOSE (ЗАКРЫТЬ)	А
CLOSE WITH LOCK (ЗАКРЫТЬ С ЗАМКОМ)	А, Б

Определения символов А и Б следуют далее.

Там, где эти определения зависят то того, как открыт файл — как входной, выходной или входной-выходной, приводятся дополнительные пояснения; в противном случае эти определения относятся к файлам, открытым как входные, выходные или входные-выходные.

А — закрыть файл.

Входные и входные-выходные файлы (доступ последовательный).

Если файл установлен в конце и указаны записи меток для этого файла, метки обрабатываются в соответствии со стандртной процедурой обработки меток, определенной реализацией. Действия оператора CLOSE (ЗАКРЫТЬ) не определены, когда записи меток специфицированы, но в файле отсутствуют, или когда записи меток не специфицированы, но присутствуют. Выполняются операции закрытия, определенные реализацией. Если файл установлен

GOST EXPERT ГОСТ 22558-89,

в конце и записи меток для него не специфицированы, метки не обрабатываются, но другие операции закрытия, определенные реализацией, выполняются. Если файл установлен не в конце, операции закрытия, определенные реализацией, выполняются, но конечные метки не обрабатываются.

Входные и входные-выходные файлы (доступ произвольный или динамический). Выходные файлы (доступ произвольный, динамический или последовательный).

Если записи меток для файла специфицированы, метки обрабатываются в соответствии со стандартной процедурой обработки меток, определенной реализацией. Действия оператора CLOSE (ЗАКРЫТЬ) не определены, когда записи меток специфицированы, но в файле отсутствуют или когда они не специфицированы, но присутствуют. Выполняются операции закрытия, определенные реализацией. Если записи меток для файла не указаны, метки не обрабатываются, но другие операции закрытия, определенные реализацией, выполняются.

Б — закрыть с замком.

Файл закрыт и не может быть опять открыт во время выполнения *этой единицы исполнения*.

(3) Выполнение оператора CLOSE (ЗАКРЫТЬ) приводит к изменению значения состояния ввода-вывода, относящегося к имени-файла-1 (см. п. 1.3.4 настоящей части).

(4) Если не присутствует необязательный входной файл, для файла не производится обработка конца файла, и указатель позиции файла не меняется.

(5) После успешного завершения оператора CLOSE (ЗАКРЫТЬ) область записи, связанная с именем-файла, становится недоступной. В случае неуспешного выполнения оператора CLOSE (ЗАКРЫТЬ) доступность области записи является неопределенной.

(6) После успешного завершения оператора CLOSE (ЗАКРЫТЬ) файл перестает быть открытым, он больше не связан ни с каким определителем файла.

(7) Если в операторе CLOSE (ЗАКРЫТЬ) указаны несколько имен-файлов, результат выполнения этого оператора CLOSE (ЗАКРЫТЬ) такой же, как если бы отдельный оператор CLOSE (ЗАКРЫТЬ) был написан для каждого имени файла в том порядке, как они указаны в этом операторе CLOSE (ЗАКРЫТЬ).

GOST ГОСТ 22558-89,

4.3. **Оператор DELETE (УДАЛИТЬ)**

4.3.1. Назначение

Оператор DELETE (УДАЛИТЬ) логически удаляет запись из файла массовой памяти.

4.3.2. Общий формат

DELETE имя-файла-1 RECORD

[INVALID KEY повелительный-оператор-1]

[NOT INVALID KEY повелительный-оператор-2]

[END-DELETE]

УДАЛИТЬ ЗАПИСЬ имя-файла-1

[ПРИ ОШИБКЕ КЛЮЧА повелительный-оператор-1]

[БЕЗ ОШИБКИ КЛЮЧА повелительный-оператор-2]

[КОНЕЦ-УДАЛИТЬ]

4.3.3. Синтаксические правила

(1) Фразы INVALID KEY (ПРИ ОШИБКЕ КЛЮЧА) и NOT INVALID KEY (БЕЗ ОШИБКИ КЛЮЧА) не должны указываться для оператора DELETE (УДАЛИТЬ), который ссылается на файл с последовательным доступом.

(2) Фраза INVALID KEY (ПРИ ОШИБКЕ КЛЮЧА) должна быть указана в операторе DELETE (УДАЛИТЬ), который ссылается на файл не с последовательным доступом и для которого не определена процедура USE AFTER STANDARD EXCEPTION (ИСПОЛЬЗОВАТЬ ПОСЛЕ СТАНДАРТНОЙ ПРОЦЕДУРЫ ОШИБКИ).

4.3.4. Общие правила

(1) Файл, представленный именем-файла-1, должен быть файлом массовой памяти. Он должен быть открыт в режиме ввода-вывода ко времени выполнения этого оператора (п. 4.4 настоящей части).

(2) Для файлов с последовательным доступом последним оператором ввода-вывода, выполняемым для имени-файла-1 перед выполнением оператора DELETE (УДАЛИТЬ), должен быть успешно выполненный оператор READ (ЧИТАТЬ). Система управления массовой памятью логически удаляет из файла запись, которая была извлечена по оператору READ (ЧИТАТЬ).

(3) Для индексного файла с произвольным | или динамическим | доступом СУМП логически удаляют из файла запись, идентифицируемую значением данного, определенного как основной ключ, связанный с именем-файла-1. Если файл не содержит записи с указанным ключом, возникает условие ошибки ключа (см. п. 1.3.5 настоящей части).

(4) После успешного выполнения оператора DELETE (УДАЛИТЬ) идентифицированная запись логически удаляется из файла и становится недоступной.

GOST EXPERT ГОСТ 22558-89,

(5) Выполнение оператора DELETE (УДАЛИТЬ) не влияет на содержимое области записи или на содержимое данного, представленного именем-данного, указанного в варианте DEPENDING ON (В ЗАВИСИМОСТИ ОТ) фразы RECORD (В ЗАПИСИ), относящейся к имени-файла-1.

(6) Выполнение оператора DELETE (УДАЛИТЬ) не влияет на указатель позиции файла.

(7) При выполнении оператора DELETE (УДАЛИТЬ) обновляется значение состояния ввода-вывода, связанного с именем-файла-1 (см. п. 1.3.4 настоящей части).

(8) Передача управления после успешного или неуспешного выполнения оператора DELETE (УДАЛИТЬ) зависит от наличия или отсутствия в операторе DELETE (УДАЛИТЬ) необязательных фраз INVALID KEY (ПРИ ОШИБКЕ КЛЮЧА) и NOT INVALID KEY (БЕЗ ОШИБКИ КЛЮЧА) (см. п. 1.3.5 настоящей части).

(9) Фраза END-DELETE (КОНЕЦ-УДАЛИТЬ) ограничивает область действия оператора DELETE (УДАЛИТЬ) (см. ч. 4, п. 6.4.3).

4.4. Оператор OPEN (ОТКРЫТЬ)

4.4.1. Назначение

Оператор OPEN (ОТКРЫТЬ) подготавливает файл к обработке.

4.4.1. Общий формат

```
OPEN { INPUT {имя-файла-1}...          }
     { OUTPUT {имя-файла-2}...         }
     { I-O {имя-файла-3}...            } ...
     { EXTEND {имя-файла-4}...         }

ОТКРЫТЬ { ВХОДНОЙ {имя-файла-1}...                  }
        { ВЫХОДНОЙ {имя-файла-2}...                 }
        { ВХОДНОЙ-ВЫХОДНОЙ {имя-файла-3}...         } ...
        { ДОПОЛНЯЕМЫЙ {имя-файла-4}...              }
```

4.4.3. Синтаксические правила

(1) Фраза EXTEND (ДОПОЛНЯЕМЫЙ) должна использоваться только для файлов с последовательным доступом.

(2) Файлы, перечисленные в операторе OPEN (ОТКРЫТЬ), могут иметь различную организацию и доступ.

4.4.4. Общие правила

(1) Успешное выполнение оператора OPEN (ОТКРЫТЬ) делает файл доступным для обработки и файл находится в режиме от-

GOST EXPERT ГОСТ 22558-89,

крытия. Успешное выполнение оператора OPEN (ОТКРЫТЬ) связывает файл с именем-файла посредством определителя файла.

Файл доступен, если он физически имеется в наличии и распознан системой управления вводом-выводом.

В табл. 1 показаны результаты открытия доступных и недоступных файлов.

Таблица 1

Фраза оператора	Файл доступен	Файл недоступен
INPUT (ВХОДНОЙ)	Нормальное открытие	Открытие неуспешное
INPUT (ВХОДНОЙ) (необязательный файл)	Нормальное открытие	Нормальное открытие; при первом чтении возникает условие конца или условие ошибки ключа
I-O (ВХОДНОЙ-ВЫХОДНОЙ)	Нормальное открытие	Открытие неуспешное
I-O (ВХОДНОЙ-ВЫХОДНОЙ) (необязательный файл)	Нормальное открытие	Открытие приводит к созданию файла
OUTPUT (ВЫХОДНОЙ)	Нормальное открытие; файл не содержит записей	Открытие приводит к созданию файла
EXTEND (ДОПОЛНЯЕМЫЙ)	Нормальное открытие	Открытие неуспешное
EXTEND (ДОПОЛНЯЕМЫЙ) (необязательный файл)	Нормальное открытие	Открытие приводит к созданию файла

(2) Успешное выполнение оператора OPEN (ОТКРЫТЬ) делает область записи доступной программе. Если определитель файла, связанный с именем файла, является внешним, то существует единственная область записи, связанная с определителем файла для единицы исполнения.

(1) Если файл не открыт, не может быть выполнен ни один оператор, явно или неявно относящийся к файлу, за исключением оператора MERGE (СЛИТЬ) с фразами USING (ИСПОЛЬЗУЯ) и GIVING (ПОЛУЧАЯ), оператора OPEN (ОТКРЫТЬ) или оператора SORT (СОРТИРОВАТЬ) с фразами USING (ИСПОЛЬЗУЯ) и GIVING (ПОЛУЧАЯ).

(4) Оператор OPEN (ОТКРЫТЬ) должен быть успешно выполнен перед выполнением любого другого допустимого оператора ввода-вывода. В табл. 2 «X» означает, что указанный оператор,

GOST EXPERT ГОСТ 22558-89,

используемый при указанном в строке методе доступа, может использоваться в режиме открытия, заданном в заголовке столбца.

(5) Файл может быть открыт с фразами INPUT (ВХОДНОЙ), OUTPUT (ВЫХОДНОЙ), EXTEND (ДОПОЛНЯЕМЫЙ) и I-O (ВХОДНОЙ-ВЫХОДНОЙ) в одной и той же единице исполнения. После первоначального выполнения оператора OPEN (ОТКРЫТЬ) для файла каждому последующему выполнению оператора OPEN (ОТКРЫТЬ) для этого же файла должно предшествовать выполнение для него оператора CLOSE (ЗАКРЫТЬ) без фразы LOCK (С ЗАМКОМ).

(6) Выполнение оператора OPEN (ОТКРЫТЬ) не извлекает и не записывает первую запись данных.

Таблица 2

Метод доступа файла	Оператор	Режим открытия			
		для ввода	для вывода	для ввода-вывода	для дополнения
Последовательный	READ (ЧИТАТЬ)	X		X	
	WRITE (ПИСАТЬ)		X		X
	REWRITE (ОБНОВИТЬ)			X	
	START (ПОДВЕСТИ)	X		X	
	DELETE (УДАЛИТЬ)			X	
Произвольный	READ (ЧИТАТЬ)	X		X	
	WRITE (ПИСАТЬ)		X	X	
	REWRITE (ОБНОВИТЬ)			X	
	START (ПОДВЕСТИ)				
	DELETE (УДАЛИТЬ)			X	
Динамический	READ (ЧИТАТЬ)	X		X	
	WRITE (ПИСАТЬ)		X	X	
	REWRITE (ОБНОВИТЬ)			X	
	START (ПОДВЕСТИ)			X	
	DELETE (УДАЛИТЬ)			X	

(7) Если указаны записи меток для файла, начальные метки обрабатываются следующим образом:

а) когда указана фраза INPUT (ВХОДНОЙ), оператор OPEN

GOST EXPERT ГОСТ 22558-89,

(ОТКРЫТЬ) осуществляет проверку меток в соответствии с процедурами, определенными реализацией для проверки входных меток;

б) когда указана фраза OUTPUT (ВЫХОДНОЙ), выполнение оператора OPEN (ОТКРЫТЬ) вызывает запись меток в соответствии с процедурами, определенными реализацией для записи выходных меток.

Действия оператора OPEN (ОТКРЫТЬ) не определены, когда записи меток специфицированы, но в файле отсутствуют, или не специфицированы, но присутствуют.

(8) Если во время выполнения оператора OPEN (ОТКРЫТЬ) возникает условие противоречия свойств файла, выполнение оператора OPEN (ОТКРЫТЬ) считается неуспешным (см. п. 1.3.7 настоящей части).

(9) Если файл, открытый с фразой INPUT (ВХОДНОЙ), является необязательным файлом, не имеющимся в наличии, оператор OPEN (ОТКРЫТЬ) устанавливает указатель позиции файла для указания того, что необязательный входной файл отсутствует.

(10) Когда файлы открыты как INPUT (ВХОДНОЙ) или I-O (ВХОДНОЙ-ВЫХОДНОЙ), указатель позиции файла устанавливается в литеры, имеющие наименьшую порядковую позицию в основной последовательности, связанной с файлом, и основной ключ записи устанавливается как ключ ссылки.

(11) Если указана фраза EXTEND (ДОПОЛНЯЕМЫЙ), оператор OPEN (ОТКРЫТЬ) устанавливает файл непосредственно после его последней логической записи.

Последней логической записью для индексного файла является существующая в данный момент запись с наибольшим значением основного ключа.

(12) Если задана фраза EXTEND (ДОПОЛНЯЕМЫЙ) и фраза LABEL (МЕТКИ) указывает, что записи меток присутствуют, выполнение оператора OPEN (ОТКРЫТЬ) включает следующие действия:

а) начальные метки файла обрабатываются только для однокатушечных или однотомных файлов;

б) начальные метки катушки (тома) обрабатываются на последней катушке (томе), как если бы файл открывался как INPUT (ВХОДНОЙ);

в) имеющиеся конечные метки файла обрабатываются, как если бы файл открывался как INPUT (ВХОДНОЙ). Затем эти метки удаляются;

г) затем обработка продолжается, как если бы файл был открыт как OUTPUT (ВЫХОДНОЙ).

GOST EXPERT ГОСТ 22558-89,

(13) Оператор OPEN (ОТКРЫТЬ) с фразой I-O (ВХОДНОЙ-ВЫХОДНОЙ) должен относиться к файлу, поддерживающему операции ввода и вывода, допустимые для индексного файла, открытого для ввода-вывода. Выполнение оператора OPEN (ОТКРЫТЬ) с фразой I-O (ВХОДНОЙ-ВЫХОДНОЙ) подготавливает файл, на который он ссылается, как для операций ввода, так и для операций вывода.

(14) Если указана фраза I-O (ВХОДНОЙ-ВЫХОДНОЙ) и во фразе LABEL RECORDS (МЕТКИ) указано, что записи меток присутствуют, выполнение оператора OPEN (ОТКРЫТЬ) включает следующие шаги:

а) проверку меток в соответствии с процедурами, определенными реализацией для проверки входных-выходных меток;

б) запись новых меток в соответствии с процедурами, определенными реализацией для записи входных-выходных меток.

(15) Для необязательного файла, являющегося недоступным, успешное выполнение оператора OPEN (ОТКРЫТЬ) с фразами I-O (ВХОДНОЙ-ВЫХОДНОЙ) | или EXTEND (ДОПОЛНЯЕМЫЙ) | приводит к созданию файла. Это создание происходит так, как если бы в указанном порядке выполнялись следующие операторы:

OPEN OUTPUT имя-файла.

CLOSE имя-файла.

ОТКРЫТЬ ВЫХОДНОЙ имя-файла.

ЗАКРЫТЬ имя-файла.

За этими операторами следует выполнение оператора OPEN (ОТКРЫТЬ), указанного в исходной программе.

Успешное выполнение оператора OPEN (ОТКРЫТЬ) с фразой OUTPUT (ВЫХОДНОЙ) приводит к созданию файла, после которого этот файл не содержит записей.

(16) Во время выполнения оператора OPEN (ОТКРЫТЬ) обновляется значение состояния ввода-вывода, связанного с именем-файла (см. п. 1.3.4 настоящей части).

(17) Если в операторе OPEN (ОТКРЫТЬ) указано более, чем одно имя-файла, результат выполнения этого оператора OPEN (ОТКРЫТЬ) такой, как если бы отдельный оператор OPEN (ОТКРЫТЬ) был написан для каждого имени-файла в том порядке, как они указаны в операторе OPEN (ОТКРЫТЬ).

(18) Минимальный и максимальный размеры записей файла устанавливаются во время создания файла и не должны изменяться впоследствии.

4.5. Оператор READ (ЧИТАТЬ)

4.5.1. Назначение

При последовательном доступе оператор READ (ЧИТАТЬ) делает доступной следующую логическую запись файла. При произ-

GOST EXPERT ГОСТ 22558-89,

вольном доступе оператор READ (ЧИТАТЬ) делает доступной указанную запись файла на устройстве массовой памяти.

4.5.2. Общий формат

Формат 1

READ имя-файла-1 [NEXT] RECORD [INTO идентификатор-1]
[AT END повелительный-оператор-1]
[NOT AT END повелительный-оператор-2]
[END-READ]

ЧИТАТЬ [СЛЕДУЮЩУЮ] ЗАПИСЬ имя-файла-1
[В идентификатор-1]
[В КОНЦЕ повелительный-оператор-1]
[НЕ В КОНЦЕ повелительный-оператор-2]
[КОНЕЦ-ЧИТАТЬ]

Формат 2

READ имя-файла-1 RECORD [INTO идентификатор-1
[KEY IS имя-данного-1]
[INVALID KEY повелительный-оператор-3]
[NOT INVALID KEY повелительный-оператор-4]
[END-READ]

ЧИТАТЬ ЗАПИСЬ имя-файла-1 [В идентификатор-1]
[КЛЮЧ имя-данного-1]
[ПРИ ОШИБКЕ КЛЮЧА повелительный-оператор-3]
[БЕЗ ОШИБКИ КЛЮЧА повелительный-оператор-4]
[КОНЕЦ-ЧИТАТЬ]

4.5.3. Синтаксические правила

(1) Область памяти, связанная с идентификатором-1, и область записи, связанная с именем-файла-1, не должны быть одной и той же областью памяти.

(2) Имя-данного-1 должно быть именем данного, определенного в качестве ключа записи, связанной с именем-файла-1.

(3) Имя-данного-1 может уточняться.

(4) Формат 1 должен использоваться для всех файлов с последовательным доступом.

GOST EXPERT ГОСТ 22558-89,

(5) Фраза NEXT (СЛЕДУЮЩУЮ) должна быть указана для файлов с динамическим доступом, если записи файла должны извлекаться последовательно.

(6) Формат 2 используется для файлов с произвольным доступом | или для файлов с динамическим доступом |, если записи должны извлекаться произвольно.

(7) Фраза INVALID KEY (ПРИ ОШИБКЕ КЛЮЧА) или фраза AT END (В КОНЦЕ) должна быть указана, если для имени-файла-1 не указана никакая применимая процедура USE AFTER STANDARD EXCEPTION (ИСПОЛЬЗОВАТЬ ПОСЛЕ СТАНДАРТНОЙ ПРОЦЕДУРЫ ОШИБКИ).

4.5.4. Общие правила

(1) Во время выполнения оператора READ (ЧИТАТЬ) файл, на который ссылается имя-файла-1, должен быть открыт как входной или входной-выходной (см. п. 4.4 настоящей части).

(2) Для файлов последовательного доступа фраза NEXT (СЛЕДУЮЩУЮ) является необязательной и не влияет на выполнение оператора READ (ЧИТАТЬ).

(3) При выполнении оператора READ (ЧИТАТЬ) обновляется значение состояния ввода-вывода, связанного с именем-файла-1 (см. п. 1.3.4 настоящей части).

(4) Установка указателя позиции файла в начале выполнения оператора READ (ЧИТАТЬ) формата 1 используется для определения записи, которая может быть доступной согласно следующим правилам.

В индексных файлах сравнения записей производится по значениям текущего ключа ссылки. Для индексных файлов сравнения производятся в соответствии с основной последовательностью для файла.

а) Если указатель позиции файла указывает, что не установлена следующая запись, выполнение оператора READ (ЧИТАТЬ) является неуспешным.

б) Если указатель позиции файла указывает, что необязательного входного файла нет, оператор выполняется согласно общему правилу (10).

в) Если указатель позиции файла был установлен предыдущими операторами OPEN (ОТКРЫТЬ) | или START (ПОДВЕСТИ) |, выбирается первая существующая запись файла, значение ключа которой больше или равно указателю позиции файла.

г) Если указатель позиции файла был установлен предыдущим оператором READ (ЧИТАТЬ) | и текущий ключ ссылки не допус-

GOST EXPERT ГОСТ 22558-89,

кает дублирование, выбирается первая имеющаяся запись в этом файле, значение ключа которой больше или равно указателю позиции файла.

д. Если указатель позиции файла был установлен предыдущим оператором READ (ЧИТАТЬ) и текущий ключ ссылки допускает дублирование, выбирается первая запись в файле, значение ключа которой либо равно указателю позиции файла и логическая позиция которой во множестве дубликатов следует сразу после записи, которая стала доступной благодаря предыдущему оператору READ (ЧИТАТЬ), либо значение ключа которой больше чем указатель позиции файла.

Если найдена запись, удовлетворяющая вышеприведенным правилам, она становится доступной в области записи, связанной с именем-файла-1.

Если запись, удовлетворяющая приведенным выше правилам, не найдена, то указатель позиции файла устанавливается для указания того, что не существует следующей логической записи, и выполнение продолжается согласно общему правилу (10).

Если запись сделалась доступной, указателю позиции файла приписывается значение текущего ключа ссылки записи, ставшей доступной.

(5) Функционирование оператора READ (ЧИТАТЬ) не зависит от метода, используемого для согласования времени доступа со временем обработки; запись доступна объектной программе до выполнения повелительного-оператора-2 или повелительного-оператора-4, если они указаны, или до выполнения любого оператора, следующего за оператором READ (ЧИТАТЬ), если ни повелительный-оператор-2, ни повелительный-оператор-4 не указаны.

(6) Когда логические записи описаны более чем одной статьей описания записи, эти записи автоматически используют одну и ту же область памяти; это эквивалентно неявному переопределению области. По завершении оператора READ (ЧИТАТЬ) значения всех данных, находящихся вне диапазона текущей записи данных, не определены.

(7) Фраза INTO (В) может быть указана в операторе READ (ЧИТАТЬ), если:

а) только одно описание записи подчиняется статье описания файла;

б) все имена-записей, связанные с именем-файла-1, и данное, на которое ссылается идентификатор-1, описывают групповое данное или элементарное буквенно-цифровое данное.

(8) Результат выполнения оператора READ (ЧИТАТЬ) с фразой INTO (В) эквивалентен применению следующих правил в указанном порядке:

GOST EXPERT ГОСТ 22558-89,

а) выполняется тот же оператор READ (ЧИТАТЬ) без фразы INTO (В);

б) текущая запись пересылается из области записи в область, указываемую идентификатором-1, в соответствии с правилами для оператора MOVE (ПОМЕСТИТЬ) без фразы CORRESPONDING (СООТВЕТСТВЕННО). Размер текущей записи определяется правилами, указанными для фразы RECORD (В ЗАПИСИ).

Если статья описания файла содержит фразу RECORD IS VARYING (В ЗАПИСИ ПЕРЕМЕННОЕ ЧИСЛО ЛИТЕР), неявная пересылка является групповой. Неявный оператор MOVE (ПОМЕСТИТЬ) не выполняется, если выполнение оператора READ (ЧИТАТЬ) было неуспешным. Индексы, связанные с идентификатором-1, вычисляются после того, как запись была прочитана и непосредственно перед ее пересылкой в данное. Запись доступна в области записи и в данном, на которое ссылается идентификатор-1.

(9) Если во время выполнения оператора READ (ЧИТАТЬ) формата 2 указатель позиции файла указывает, что нет необязательного входного файла, возникает условие ошибки ключа, а выполнение оператора READ (ЧИТАТЬ) является неуспешным (см. ч. 8, п. 1.3.5).

(10) При использовании формата 1 оператора READ (ЧИТАТЬ), если указатель позиции файла показывает, что не существует последующей логической записи, или что необязательный входной файл отсутствует, имеет место в соответствии с указанной последовательностью следующее:

а) значение, полученное по указателю позиции файла, помещается в состояние ввода-вывода, соответствующее имени-файла-1, и указывает на условие «в конце» (см. п. 1.3.4 настоящей части);

б) если в операторе, вызвавшем условие, задана фраза AT END (В КОНЦЕ), управление передается повелительному-оператору-1 фразы AT END (В КОНЦЕ).

Никакая процедура USE AFTER STANDARD EXCEPTION (ИСПОЛЬЗОВАТЬ ПОСЛЕ СТАНДАРТНОЙ ПРОЦЕДУРЫ ОШИБКИ), связанная с именем-файла-1 не выполняется;

в) если фраза AT END (В КОНЦЕ) не задана, процедура USE AFTER STANDARD EXCEPTION (ИСПОЛЬЗОВАТЬ ПОСЛЕ СТАНДАРТНОЙ ПРОЦЕДУРЫ ОШИБКИ), должна быть связана с именем-файла-1, и эта процедура выполняется. Возврат из этой процедуры осуществляется к следующему после оператора READ (ЧИТАТЬ) выполнимому оператору.

Если возникает условие «в конце», выполнение оператора READ (ЧИТАТЬ) неуспешно.

GOST EXPERT ГОСТ 22558-89,

(11) Если во время выполнения оператора READ (ЧИТАТЬ) не возникают ни условие конца, ни условие ошибки ключа, фразы AT END (В КОНЦЕ) и INVALID KEY (ПРИ ОШИБКЕ КЛЮЧА) игнорируются, если они указаны, и выполняются следующие действия:

а) устанавливается значение указателя позиции файла и обновляется значение состояния ввода-вывода, связанного с именем-файла-1;

б) если возникает условие особой ситуации, не являющееся ни условием конца, ни условием ошибки ключа, управление передается процедуре USE AFTER EXCEPTION (ИСПОЛЬЗОВАТЬ ПОСЛЕ ПРОЦЕДУРЫ ОШИБКИ) согласно правилам для оператора USE (ИСПОЛЬЗОВАТЬ), применимого к имени-файла-1 (п. 4.8 настоящей части);

в) если условие особой ситуации не существует, запись становится доступной в области записи и выполняются любые неявные пересылки, предопределенные фразой INTO (В). Управление передается в точку выхода оператора READ (ЧИТАТЬ) или повелительному-оператору-2, если он указан. В последнем случае выполнение продолжается согласно правилам для операторов, указанных в повелительном-операторе-2. Если выполняется ветвление процедуры или условный оператор, вызывающие явную передачу управления, оно передается в соответствии с правилами для этих операторов; в противном случае после завершения выполнения повелительного-оператора-2 управление передается в точку выхода оператора READ (ЧИТАТЬ).

(12) После неуспешного выполнения оператора READ (ЧИТАТЬ) содержимое соответствующей области записи не определено, ключ ссылки для индексного файла не определен, а указатель позиции файла показывает, что правильная следующая запись не установлена.

(13) Для индексного файла, для которого задан динамический доступ, оператор READ (ЧИТАТЬ) формата 1 вызывает считывание следующей логической записи из этого файла.

(14) Для индексного файла с последовательным доступом записи, имеющие одно и то же дублирующееся значение дополнительного ключа, являющегося ключом ссылки, становятся доступными в том порядке, в котором они передавались при выполнении операторов WRITE (ПИСАТЬ) или REWRITE (ОБНОВИТЬ), которые привели к возникновению этих дублирующихся значений.

(15) Для индексного файла, если фраза KEY (КЛЮЧ) задана в формате 2 оператора READ (ЧИТАТЬ), в качестве ключа ссылки при извлечении записи учреждается имя-данного-1, указанное во фразе KEY (КЛЮЧ). Для файла с динамическим дос-

GOST EXPERT ГОСТ 22558-89,

тупом этот же ключ ссылки используется для всех последующих выполнений операторов READ (ЧИТАТЬ) формата 1 до тех пор, пока для этого файла не будет учрежден другой ключ ссылки.

(16) Для индексного файла, если в операторе READ (ЧИТАТЬ) формата 2 не указан вариант KEY (КЛЮЧ), в качестве ключа ссылки при извлечении записи учреждается основной ключ записи. Для файла с динамическим доступом этот же ключ ссылки используется для всех последующих выполнений оператора READ (ЧИТАТЬ) формата 1 до тех пор, пока для этого файла не будет учрежден другой ключ ссылки.

(17) Выполнение оператора READ формата 2 устанавливает указатель позиции файла равным значению ключа ссылки. Это значение сравнивается со значением соответствующего данного в записях файла до тех пор, пока не будет найдена первая запись с равным значением.

При использовании дополнительного ключа в случае дублирующихся значений первой найденной записью будет первая запись в последовательности дубликатов, которые были переданы системе управления массовой памятью. Найденная таким образом запись становится доступной в области записи, соответствующей имени-файла-1. Если таким образом не удастся найти ни одной записи, возникает условие ошибки ключа, и выполнение оператора READ (ЧИТАТЬ) считается неуспешным (см. п. 1.3.5 настоящей части).

(18) Если число позиций литер в прочитанной записи меньше минимального размера, указанного статьями описания записей для имени-файла-1, участок области записи, находящийся справа от последней прочитанной литеры, не определен. Если число позиций литер в прочитанной записи больше максимального размера, указанного статьями описания записей для имени-файла-1, запись усекается справа до максимального размера. В обоих случаях выполнение оператора READ (ЧИТАТЬ) считается успешным, а состояние ввода-вывода указывает, что возникло несоответствие длины записи (см. п. 1.3.4 настоящей части).

(19) Фраза END-READ (КОНЕЦ-ЧИТАТЬ) ограничивает область действия оператора READ (ЧИТАТЬ) (см. ч. 4, п. 6.4.3).

4.6. **Оператор REWRITE (ОБНОВИТЬ)**

4.6.1. Назначение

Оператор REWRITE (ОБНОВИТЬ) логически заменяет запись в файле на устройстве массовой памяти.

4.6.2. Общий формат

REWRITE имя-записи-1 [FROM идентификатор-1]

GOST EXPERT ГОСТ 22558-89,

[INVALID KEY повелительный-оператор-1]
[NOT INVALID KEY повелительный-оператор-2]
[END-REWRITE]

ОБНОВИТЬ имя-записи-1 [ИЗ ПОЛЯ идентификатор-1]
[ПРИ ОШИБКЕ КЛЮЧА повелительный-оператор-1]
[БЕЗ ОШИБКИ КЛЮЧА повелительный-оператор-2]
[КОНЕЦ-ОБНОВИТЬ]

4.6.3. Синтаксические правила

(1) Имя-записи-1 и идентификатор-1 не должны относиться к одной и той же области памяти.

(2) Имя-записи-1 — это имя логической записи в секции файлов раздела данных. Оно может быть уточнено.

(3) Если для соответствующего имени-файла не указана процедура USE AFTER STANDARD EXCEPTION (ИСПОЛЬЗОВАТЬ ПОСЛЕ СТАНДАРТНОЙ ПРОЦЕДУРЫ ОШИБКИ), фразы INVALID KEY (ПРИ ОШИБКЕ КЛЮЧА) и NOT INVALID KEY (БЕЗ ОШИБКИ КЛЮЧА) должны указываться.

4.6.4. Общие правила

(1) Файл, связанный с именем-записи-1, должен быть файлом массовой памяти и должен быть открыт как входной-выходной во время выполнения этого оператора (см. п. 4.4 настоящей части).

(2) Для файлов с последовательным доступом последним оператором ввода-вывода для соответствующего файла перед выполнением оператора REWRITE (ОБНОВИТЬ) должен быть успешно выполненный оператор READ (ЧИТАТЬ). СУМП логически заменяет запись, которая была извлечена оператором READ (ЧИТАТЬ).

(3) На уровне 1 число позиций литер в записи, представленной именем-записи-1, должно быть равно числу позиций литер в обновляемой записи. На уровне 2 число позиций литер в записи, представленной именем-записи-1, может совпадать, а может и не совпадать с числом позиций литер в обновляемой записи.

(4) Логическая запись, включенная в файл при успешном выполнении оператора REWRITE (ОБНОВИТЬ), становится недоступной в области записи, за исключением случая, когда имя-файла, связанное с именем-записи-1, описано во фразе SAME RECORD AREA (ОБЩАЯ ОБЛАСТЬ ЗАПИСИ). Логическая запись доступна программе и как запись файла, связанного с именем-записи-1, и как запись других файлов, указанных в той же фразе SAME RECORD AREA (ОБЩАЯ ОБЛАСТЬ ЗАПИСИ), что и соответствующий выходной файл.

(5) Выполнение оператора REWRITE (ОБНОВИТЬ) с фразой FROM (ИЗ ПОЛЯ) эквивалентно выполнению следующих операторов в указанном порядке:

GOST EXPERT ГОСТ 22558-89,

а) оператор
MOVE идентификатор-1 TO имя-записи-1.
ПОМЕСТИТЬ идентификатор-1 В имя-записи-1 соответственно правилам, описанным для оператора MOVE (ПОМЕСТИТЬ);

б) тот же оператор REWRITE (ОБНОВИТЬ) без фразы FROM (ИЗ ПОЛЯ).

(6) После завершения выполнения оператора REWRITE (ОБНОВИТЬ) информация в области, указанной идентификатором-1, остается доступной, даже если информация в области, указанной именем-записи-1, не является доступной, за исключением случая, определенного фразой SAME RECORD AREA (ОБЩАЯ ОБЛАСТЬ ЗАПИСИ).

(7) Выполнение оператора REWRITE (ОБНОВИТЬ) не влияет на указатель позиции файла.

(8) Выполнение оператора REWRITE (ОБНОВИТЬ) вызывает обновление состояния ввода-вывода для файла, связанного с именем-записи-1 (см. п. 1.3.4 настоящей части).

(9) При выполнении оператора REWRITE (ОБНОВИТЬ) логическая запись передается операционной системе.

(10) Передача управления после успешного или неуспешного выполнения оператора REWRITE (ОБНОВИТЬ) зависит от наличия или отсутствия в операторе REWRITE (ОБНОВИТЬ) необязательных фраз INVALID KEY (ПРИ ОШИБКЕ КЛЮЧА) и NOT INVALID KEY (БЕЗ ОШИБКИ КЛЮЧА) (см. п. 1.3.5 настоящей части).

(11) Фраза END-REWRITE (КОНЕЦ-ОБНОВИТЬ) ограничивает область действия оператора REWRITE (ОБНОВИТЬ) (см. ч. 4, п. 6.4.3).

(12) Число позиций литер в записи, представленной именем-записи-1, должно быть не больше, чем наибольшее, и не меньше, чем наименьшее число позиций литер, определенное фразой RECORD IS VARYING (В ЗАПИСИ ПЕРЕМЕННОЕ ЧИСЛО ЛИТЕР), относящейся к имени-файла, связанного с именем-записи-1. В обоих случаях выполнение оператора REWRITE (ОБНОВИТЬ) является неуспешным, обновление не происходит, содержимое области записи не изменяется, а состояние ввода-вывода файла, связанного с именем-записи-1, становится равным значению, указывающему на причину возникновения ситуации (см. п. 1.3.4 настоящей части).

(13) Для файла с последовательным доступом запись, которая должна быть заменена, определяется значением основного ключа. При выполнении оператора REWRITE (ОБНОВИТЬ), значение основного ключа заменяющей записи должно быть равно значению основного ключа последней прочитанной из этого файла записи.

GOST EXPERT ГОСТ 22558-89,

(14) Для файла с произвольным или динамическим доступом запись, подлежащая замене, указывается значением основного ключа записи.

(15) Выполнение оператора REWRITE (ОБНОВИТЬ) для записи, которая имеет дополнительный ключ, производится следующим образом:

а) если значение конкретного дополнительного ключа не изменилось, порядок чтения в случае, когда этот ключ является ключом ссылки, остается неизменным;

б) если значение конкретного дополнительного ключа изменилось, последующий порядок считывания этой записи может измениться, когда этот конкретный дополнительный ключ является ключом ссылки. Если допускается дублирование значений ключа, запись логически размещается последней среди множества записей дубликатов, содержащих такое же значение дополнительного ключа записи, которое было помещено в запись.

(16) Условие ошибки ключа возникает в следующих случаях:

а) при последовательном доступе значение, содержащееся в основном ключе заменяющей записи, не равно значению основного ключа последней прочитанной из файла записи;

б) при динамическом или произвольном доступе значение основного ключа заменяемой записи не равно значению основного ключа ни для одной существующей записи файла;

в) в файле, для которого не допускаются дубликаты, уже есть запись с указанным значением дополнительного ключа.

(17) Если возникает условие ошибки ключа, выполнение оператора REWRITE (ОБНОВИТЬ) неуспешное, операция обновления не производится, данные в области записи не изменяются, а состоянию ввода-вывода для имени-файла, соответствующего имени-записи-1, присваивается значение, указывающее на причину возникновения ситуации (см. п. 1.3.4 настоящей части)

4.7. Оператор START (ПОДВЕСТИ)

4.7.1. Назначение

Оператор START (ПОДВЕСТИ) предоставляет возможность логического позиционирования индексного файла для дальнейшего последовательного извлечения записей.

GOST EXPERT ГОСТ 22558-89,

4.7.2. Общий формат

START имя-файла-1

[KEY { IS EQUAL TO / IS = / IS GREATER THAN / IS > / IS NOT LESS THAN / IS NOT < / IS GREATER THAN OR EQUAL TO / IS > = } имя-данного-1]

[INVALID KEY повелительный-оператор-1]
[NOT INVALID KEY повелительный-оператор-2]
[END-START]

ПОДВЕСТИ ЗАПИСЬ имя-файла-1

[КЛЮЧ { РАВЕН / = / БОЛЬШЕ / > / НЕ МЕНЬШЕ / НЕ < / БОЛЬШЕ ИЛИ РАВЕН / > = } имя-данного-1]

[ПРИ ОШИБКЕ КЛЮЧА повелительный-оператор-1]
[БЕЗ ОШИБКИ КЛЮЧА повелительный-оператор-2]
[КОНЕЦ-ПОДВЕСТИ]

4.7.3. Синтаксические правила

(1) Имя-файла-1 должно быть именем файла с последовательным или динамическим доступом.

(2) Имя-данного-1 может уточняться.

(3) Если для имени-файла-1 не определена соответствующая процедура USE AFTER STANDARD EXCEPTION (ИСПОЛЬЗОВАТЬ ПОСЛЕ СТАНДАРТНОЙ ПРОЦЕДУРЫ ОШИБКИ), должна быть указана фраза INVALID KEY (ПРИ ОШИБКЕ КЛЮЧА).

(4) Если задана фраза KEY (КЛЮЧ), имя-данного-1 должно ссылаться:

GOST EXPERT ГОСТ 22558-89,

а) либо на данное, описанное как ключ записи имени-файла-1 (см. пп. 2.5, 2.7 настоящей части);

б) либо на любое буквенно-цифровое данное, самая левая литера которого в записи файла соответствует самой левой литере ключа записи имени-файла-1 и длина которого не больше чем длина этого ключа.

4.7.4. Общие правила

(1) Файл, представленный именем-файла-1, ко времени выполнения оператора START (ПОДВЕСТИ) должен быть открыт для ввода или ввода-вывода (см. п. 4.4 настоящей части).

(2) Если фраза KEY (КЛЮЧ) не указана, подразумевается знак отношения EQUAL (РАВЕН).

(3) Выполнение оператора START (ПОДВЕСТИ) не изменяет ни содержимое области записи, ни содержимое данного, представленного именем-данного, указанным во фразе DEPENDING ON (В ЗАВИСИМОСТИ ОТ) фразы RECORD (В ЗАПИСИ), относящейся к имени-файла-1.

(4) Сравнение, определяемое знаком отношения во фразе KEY (КЛЮЧ), проводится между ключом записи файла, представленного именем-файла-1, и данным, как указано в общих правилах (11) и (12).

Сравнение производится относительно возрастания ключей ссылки в соответствии с основной последовательностью для файла. Если операнды сравнения неодинакового размера, сравнение производится так, как если бы более длинный операнд усекался справа до размера более короткого. Все другие правила нечислового сравнения остаются в силе (см. ч. 4, п. 6.3.1.1.2).

а) Указателю позиции файла присваивается значение ключа ссылки первой из логических записей, ключ которых удовлетворяет сравнению.

б) Если сравнение не удовлетворяется ни для одной записи файла, возникает условие ошибки ключа, и выполнение оператора START (ПОДВЕСТИ) считается неуспешным.

(5) При выполнении оператора START (ПОДВЕСТИ) обновляется значение состояния ввода-вывода, относящегося к имени-файла-1 (см. п. 1.3.4 настоящей части).

(6) Если во время выполнения оператора START (ПОДВЕСТИ) указатель позиции файла определяет, что необязательный входной файл отсутствует, возникает условие ошибки ключа, и выполнение оператора START (ПОДВЕСТИ) является неуспешным.

(7) Передача управления после успешного или неуспешного выполнения оператора START (ПОДВЕСТИ) зависит от наличия или отсутствия необязательных фраз INVALID KEY (ПРИ ОШИБКЕ КЛЮЧА) и NOT INVALID KEY (БЕЗ ОШИБКИ

GOST EXPERT ГОСТ 22558-89,

КЛЮЧА) в операторе START (ПОДВЕСТИ) (см. п. 1.3.5 настоящей части).

(8) После неуспешного выполнения оператора START (ПОДВЕСТИ) указатель позиции файла указывает, что правильная следующая запись не установлена. Кроме того для индексных файлов ключ ссылки становится неопределенным.

(9) Фраза END-START (КОНЕЦ-ПОДВЕСТИ) ограничивает область действия оператора START (ПОДВЕСТИ) (см. ч. 4, п. 6.4.3).

(10) Ключ ссылки устанавливается следующим образом:

а) если фраза KEY (КЛЮЧ) не задана, основной ключ, заданный для имени-файла-1, становится ключом ссылки;

б) если фраза KEY (КЛЮЧ) задана и имя-данного-1 задано в качестве ключа для имени-файла-1, этот ключ записи становится ключом ссылки;

в) если фраза KEY (КЛЮЧ) задана, а имя-данного-1 не задано в качестве ключа для имени-файла-1, ключом ссылки становится ключ записи, самая левая литера которого соответствует самой левой литере данного, заданного именем-данного-1.

Этот ключ ссылки используется для установления упорядочивания записей для этого оператора START (ПОДВЕСТИ), см. общее правило (4); и если выполнение оператора START (ПОДВЕСТИ) успешное, ключ ссылки используется для последующих операторов READ (ЧИТАТЬ) последовательного чтения (см. п. 4.5 настоящей части).

(11) Если фраза KEY (КЛЮЧ) задана, сравнение, описанное в общем правиле (4), использует данное, заданное именем-данного-1.

(12) Если фраза KEY (КЛЮЧ) не задана, сравнение, описанное в общем правиле (4), использует данное, заданное во фразе RECORD KEY (КЛЮЧ ЗАПИСИ), соответствующей имени-файла-1.

4.8. Оператор USE (ИСПОЛЬЗОВАТЬ)

4.8.1. Назначение

Оператор USE (ИСПОЛЬЗОВАТЬ) определяет процедуры обработки ошибок ввода-вывода дополнительно к стандартным процедурам, предоставляемым системой управления вводом-выводом.

GOST EXPERT ГОСТ 22558-89,

4.8.2. Общий формат

USE AFTER STANDARD { EXCEPTION / ERROR } PROCEDURE ON

{ {имя-файла-1} [...] / INPUT / OUTPUT / I-O / EXTEND }

ИСПОЛЬЗОВАТЬ ПОСЛЕ СТАНДАРТНОЙ ПРОЦЕДУРЫ ОШИБКИ

ДЛЯ { {имя-файла-1} [...] / ВХОДНЫХ / ВЫХОДНЫХ / ВХОДНЫХ-ВЫХОДНЫХ / ДОПОЛНЯЕМЫХ }

4.8.3. Синтаксические правила

(1) Оператор USE (ИСПОЛЬЗОВАТЬ) должен непосредственно следовать за заголовком секции декларативной части раздела процедур и должен быть единственным в предложении. Остальная часть декларативной секции должна состоять из одного или более процедурных параграфов, определяющих процедуры, которые должны использоваться.

(2) Сам оператор USE (ИСПОЛЬЗОВАТЬ) никогда не выполняется; он только определяет условия, вызывающие выполнение указанных после него процедур.

(3) Появление имени-файла-1 в операторе USE (ИСПОЛЬЗОВАТЬ) не должно требовать одновременного выполнения более чем одной декларативной секции.

(4) Слова ERROR и EXCEPTION являются синонимами и взаимозаменяемы.

(5) Файлы, к которым явно или неявно обращаются в операторе USE (ИСПОЛЬЗОВАТЬ), могут иметь различную организацию или доступ.

(6) Каждая из фраз INPUT (ВХОДНЫХ), OUTPUT (ВЫХОДНЫХ), I-O (ВХОДНЫХ-ВЫХОДНЫХ) и EXTEND (ДОПОЛНЯЕМЫХ) может указываться лишь раз в декларативной части раздела процедур.

4.8.4. Общие правила

(1) Декларативные процедуры могут быть включены в любую

GOST EXPERT ГОСТ 22558-89,

исходную Кобол-программу, независимо от того, содержит ли эта программа другую программу или сама содержится в другой программе. Декларатива вызывается тогда, когда во время выполнения программы выполняются условия, описанные в операторе USE (ИСПОЛЬЗОВАТЬ), предшествующем декларативе. Только одна декларатива внутри отдельно скомпилированной программы, содержащей оператор, который вызвал уточняющее условие, вызывается тогда, когда выполняется какое-либо из условий, описанных в операторе USE (ИСПОЛЬЗОВАТЬ), предшествующем декларативе, во время выполнения программы. Если не существует уточняющей декларативы в отдельно скомпилированной программе, то декларатива не выполняется.

(2) Внутри декларативной процедуры не должно быть обращений к каким-либо процедурам в недекларативной части раздела процедур.

(3) К именам процедур, связанных с оператором USE (ИСПОЛЬЗОВАТЬ), могут быть обращения в другой декларативной секции или в недекларативной процедуре только оператором PERFORM (ВЫПОЛНИТЬ).

(4) Когда имя-файла-1 описано явно, то к имени-файла-1 не применяется никаких других операторов USE (ИСПОЛЬЗОВАТЬ).

(5) Процедуры, связанные с оператором USE (ИСПОЛЬЗОВАТЬ), выполняются системой управления вводом-выводом после завершения стандартной программы ошибки ввода-вывода при неуспешном выполнении операции ввода-вывода, если не сработают фразы AT END (В КОНЦЕ) или INVALID KEY (ПРИ ОШИБКЕ КЛЮЧА).

При выполнении процедуры соблюдаются следующие правила:

а) если указано имя-файла-1, то соответствующая процедура выполняется при выполнении условия, описанного в операторе USE (ИСПОЛЬЗОВАТЬ);

б) если указано INPUT (ВХОДНЫХ), то соответствующая процедура выполняется при выполнении условия, описанного в операторе USE (ИСПОЛЬЗОВАТЬ), для какого-либо файла, открытого для ввода, или в процессе открытия для ввода, за исключением файлов, указанных именем-файла-1 в другом операторе USE (ИСПОЛЬЗОВАТЬ), описывающем такое же условие;

в) если указано OUTPUT (ВЫХОДНЫХ), то соответствующая процедура выполняется при выполнении условия, описанного в операторе USE (ИСПОЛЬЗОВАТЬ), для какого-либо файла, открытого для вывода, или в процессе открытия для вывода за исключением файлов, указываемых именем-файла-1 в другом операторе USE (ИСПОЛЬЗОВАТЬ), описывающем такое же условие;

г) если указано I-O (ВХОДНЫХ-ВЫХОДНЫХ), то соответствующая процедура выполняется при выполнении условия, описан-

GOST EXPERT ГОСТ 22558-89,

ного в операторе USE (ИСПОЛЬЗОВАТЬ), для какого-либо файла, открытого для ввода-вывода, или в процессе открытия для ввода-вывода, за исключением файлов, указанных именем-файла-1 в другом операторе USE (ИСПОЛЬЗОВАТЬ), описывающем такое же условие;

д) если указано EXTEND (ДОПОЛНЯЕМЫХ), то соответствующая процедура выполняется при выполнении условия, описанного в операторе USE (ИСПОЛЬЗОВАТЬ), для какого-либо файла, открытого для дополнения, за исключением файлов, указанных именем-файла-1 в другом операторе USE (ИСПОЛЬЗОВАТЬ), описывающем такое же условие.

(6) После выполнения процедуры, связанной с оператором USE (ИСПОЛЬЗОВАТЬ), управление передается вызывающей программе в системе управления вводом-выводом. Если значение состояния ввода-вывода не указывает на критическую ошибку ввода-вывода, то система управления вводом-выводом возвращает управление оператору, следующему за оператором ввода-вывода, выполнение которого вызвало ошибку. Если значение состояния ввода-вывода указывает на критическую ошибку, то действие определяется реализацией (см. п. 1.3.4 настоящей части).

(7) В процедуре, связанной с оператором USE (ИСПОЛЬЗОВАТЬ), не должны выполняться никакие операторы, которые могут потребовать выполнения процедуры USE (ИСПОЛЬЗОВАТЬ), вызванной ранее и еще не вернувшей управление вызвавшей ее программе.

4.9. Оператор WRITE (ПИСАТЬ)

4.9.1. Назначение

Оператор WRITE (ПИСАТЬ) включает логическую запись в выходной или входной-выходной файл.

4.9.2. Общий формат

WRITE имя-записи-1 [FROM идентификатор-1]
[INVALID KEY повелительный-оператор-1]
[NOT INVALID KEY повелительный-оператор-2]
[END-WRITE]

ПИСАТЬ имя-записи-1 [ИЗ ПОЛЯ идентификатор-1]
[ПРИ ОШИБКЕ КЛЮЧА повелительный-оператор-1]
[БЕЗ ОШИБКИ КЛЮЧА повелительный-оператор-2]
[КОНЕЦ-ПИСАТЬ]

4.9.3. Синтаксические правила

(1) Имя-записи-1 и идентификатор-1 не должны относиться к одной и той же области памяти.

(2) Имя-записи-1 является именем логической записи в секции файлов раздела данных и может быть уточнено.

GOST EXPERT ГОСТ 22558-89,

(3) Фраза INVALID KEY (ПРИ ОШИБКЕ КЛЮЧА) должна указываться в операторе WRITE (ПИСАТЬ) для файлов, для которых не определена соответствующая процедура USE AFTER STANDARD EXCEPTION (ИСПОЛЬЗОВАТЬ ПОСЛЕ СТАНДАРТНОЙ ПРОЦЕДУРЫ ОШИБКИ).

4.9.4. Общие правила

(1) Файл, указанный именем-файла, связанным с именем-записи-1, должен быть открыт как выходной, входной-выходной | или дополняемый | ко времени выполнения этого оператора (см. п. 4.4 настоящей части).

(2) Логическая запись, включаемая в файл при успешном выполнении оператора WRITE (ПИСАТЬ), становится недоступной в области записи. | Исключение представляют случаи, когда имя-файла, связанное с именем-записи-1, указано в фразе SAME RECORD AREA (ОБЩАЯ ОБЛАСТЬ ЗАПИСИ). Если имя относится к имени-файла, указанному во фразе SAME RECORD AREA (ОБЩАЯ ОБЛАСТЬ ЗАПИСИ), логическая запись доступна программе и как запись файла, связанного с именем-записи-1, и как запись других файлов, указанных в той же фразе SAME RECORD AREA (ОБЩАЯ ОБЛАСТЬ ЗАПИСИ), что и соответствующий выходной файл.

(3) Результат выполнения оператора WRITE (ПИСАТЬ) с фразой FROM (ИЗ ПОЛЯ) эквивалентен выполнению следующих операторов в указанном порядке:

а) оператор

MOVE идентификатор-1 TO имя-записи-1

ПОМЕСТИТЬ идентификатор-1 В имя-записи-1 соответственно правилам, специфицированным в операторе MOVE (ПОМЕСТИТЬ);

б) тот же оператор WRITE (ПИСАТЬ) без фразы FROM (ИЗ ПОЛЯ).

(4) После завершения выполнения оператора WRITE (ПИСАТЬ) информация в области, указанной идентификатором-1, остается доступной, даже если информация в области, указанной именем-записи-1, не является доступной | , за исключением случая, специфицированного фразой SAME RECORD AREA (ОБЩАЯ ОБЛАСТЬ ЗАПИСИ) | .

(5) Выполнение оператора WRITE (ПИСАТЬ) не влияет на указатель позиции файла.

(6) Выполнение оператора WRITE (ПИСАТЬ) вызывает обновление состояния ввода-вывода имени-файла, связанного с именем-записи-1 (см. п. 1.3.4 настоящей части).

GOST EXPERT ГОСТ 22558-89,

(7) При выполнении оператора WRITE (ПИСАТЬ) логическая запись передается операционной системе.

(8) Количество позиций литер в записи, указанной именем-записи-1, не должно быть больше наибольшего или меньше наименьшего числа литер, допустимого фразой RECORD IS VARYING (В ЗАПИСИ ПЕРЕМЕННОЕ ЧИСЛО ЛИТЕР), связанной с именем-файла, связанного с именем-записи-1. В любом случае выполнение оператора WRITE (ПИСАТЬ) неуспешно, операция записи не происходит, содержимое области записи не меняется, и состояние ввода-вывода файла, связанного с именем-записи-1, принимает значение, указывающее на причину возникновения условия (см. п. 1.3.4 настоящей части).

(9) Если при выполнении оператора WRITE (ПИСАТЬ) с фразой NOT INVALID KEY (БЕЗ ОШИБКИ КЛЮЧА) условие ошибки ключа не возникает, то управление передается повелительному-оператору-2 следующим образом:

а) если выполнение оператора WRITE (ПИСАТЬ) успешно, то управление передается после того, как запись записана, и после изменения состояния ввода-вывода имени-файла, связанного с именем-записи-1;

б) если выполнение оператора WRITE (ПИСАТЬ) неуспешно не из-за ошибки ключа, то управление передается после обновления состояния ввода-вывода для имени файла, связанного с именем-записи-1, и после выполнения процедуры, определенной оператором USE AFTER STANDARD EXCEPTION PROCEDURE (ИСПОЛЬЗОВАТЬ ПОСЛЕ СТАНДАРТНОЙ ПРОЦЕДУРЫ ОШИБКИ), применимой к имени-файла, связанного с именем-записи-1, если таковая указана.

(10) Фраза END-WRITE (КОНЕЦ-ПИСАТЬ) ограничивает область действия оператора WRITE (ПИСАТЬ) (см. ч. 4, п. 6.4.3).

(11) Выполнение оператора WRITE (ПИСАТЬ) вызывает передачу содержимого области записи. Система управления массовой памятью использует содержимое ключей записи таким образом, чтобы последующий доступ к записи мог базироваться на любом из этих заданных ключей записи.

(12) Значение основного ключа должно быть уникальным для записей файла.

(13) Данному, заданному в качестве основного ключа, требуемое значение должно присваиваться до выполнения оператора WRITE (ПИСАТЬ).

(14) Если файл открыт для последовательного доступа, записи должны передаваться системе управления массовой памятью в порядке возрастания значений основного ключа в соответствии с основной последовательностью для файла.

GOST EXPERT ГОСТ 22558-89,

Если файл открыт для дополнения, первая запись, передаваемая системе управления массовой памятью, должна иметь значение основного ключа большее, чем максимальное значение основных ключей существующих записей файла.

(15) Если файл открыт для произвольного или динамического доступа, записи могут передаваться системе управления массовой памятью в любом заданном в программе порядке.

(16) Если в статье описания индексного файла задана фраза ALTERNATE RECORD KEY (ДОПОЛНИТЕЛЬНЫЙ КЛЮЧ ЗАПИСИ), значения дополнительного ключа записи могут быть не уникальными, только если для этого данного задана фраза DUPLICATES (С ДУБЛИРОВАНИЕМ). В этом случае система управления массовой памятью обеспечивает помещение записей таким образом, что, когда записи просматриваются последовательно, порядок считывания этих записей тот же, в котором они передавались системе управления массовой памятью.

(17) Условие ошибки ключа возникает в следующих случаях:

а) если при последовательном доступе файл открыт для вывода или дополнения и значение основного ключа записи не больше чем значение основного ключа предыдущей записи или

б) если файл открыт для вывода или ввода-вывода и значение основного ключа записи равно значению основного ключа записи, уже существующей в файле, или

в) если файл открыт для вывода, дополнения или ввода-вывода и значение дополнительного ключа записи, для которого не допускается дублирование, равно соответствующему данному из записи, уже существующей в файле;

г) когда сделана попытка писать запись вне границ, определенных для файла.

(18) При обнаружении условия ошибки ключа выполнение оператора WRITE (ПИСАТЬ) считается неуспешным. Содержимое области записи не изменяется, а состояние ввода-вывода для имени файла, связанного с именем-записи-1, устанавливается на значение, указывающее причину этого условия. Выполнение программы продолжается в соответствии с правилами для условия ошибки ключа (см. пп. 1.3.4, 1.3.5 настоящей части).

GOST EXPERT ГОСТ 22558-89,

Часть 10. МОДУЛЬ МЕЖПРОГРАММНЫХ СВЯЗЕЙ

1. ВВЕДЕНИЕ В МОДУЛЬ МЕЖПРОГРАММНЫХ СВЯЗЕЙ

1.1. Назначение

Модуль межпрограммных связей предоставляет средства, с помощью которых программа может связываться с одной или несколькими программами. Эта связь обеспечивается: (а) возможностью передачи управления от одной программы к другой внутри единицы исполнения и (б) возможностью передачи параметров между программами, чтобы сделать определенные значения данных доступными для вызываемой программы. Кроме того, модуль межпрограммных связей обеспечивает связь между двумя программами через использование общих данных и файлов.

1.2. Характеристика уровней

Уровень 1 модуля межпрограммных связей предоставляет средства для передачи управления одной или нескольким программам, имена которых известны во время компиляции, а также средства для использования общих данных такими программами.

Уровень 2 модуля межпрограммных связей предусматривает средства для передачи управления одной или нескольким программам, имена которых не известны во время компиляции, а также возможность определить во время выполнения наличие памяти для программы, которой передается управление. Кроме того, уровень 2 обеспечивает внешние атрибуты, глобальные имена и вложение исходных программ.

1.3. Понятия языка

1.3.1. Вложенные исходные программы

Исходная программа Кобола — это синтаксически правильный набор операторов Кобола. Исходная программа Кобола может содержать другие исходные программы Кобола и эти содержащиеся программы могут обращаться к некоторым ресурсам программы, в которой они содержатся.

Программа В, содержащаяся внутри другой программы А, может быть прямо содержащейся или косвенно содержащейся. Программа В прямо содержится в программе А, если в программе А нет другой программы, которая тоже содержит программу В. Программа В косвенно содержится в программе А, если существует программа, содержащаяся в программе А, которая содержит программу В.

1.3.2. Определитель файла

Определитель файла — это область памяти, которая содержит информацию о файле и используется для установления соответст-

GOST EXPERT ГОСТ 22558-89,

вия между именем файла и физическим файлом, а также между именем файла и связанной с ним областью записи.

1.3.3. Глобальные и локальные имена

Имя данного именует даннос. Имя файла именует определитель файла. Эти имена классифицируются как глобальные или локальные.

Глобальное имя может использоваться для обращения к объекту, с которым оно связано, из программы, в которой объявлено это глобальное имя, или из любой другой программы, которая содержится в программе, где объявляется это глобальное имя.

Локальное имя можно использовать для обращения к объекту, с которым оно связано, только из программы, в которой это локальное имя объявлено.

Некоторые имена всегда являются глобальными, некоторые имена всегда локальные; есть имена, которые могут быть локальными или глобальными в зависимости от спецификаций в программе, где эти имена объявлены.

Имя записи является глобальным, если в статье описания записи, в которой объявляется имя записи, указана фраза GLOBAL (ГЛОБАЛЬНОЕ) или в случае статьи описания записи в секции файлов фраза GLOBAL (ГЛОБАЛЬНОЕ) указана в статье описания файла для имени файла, связанного со статьей описания записи. Имя данного является глобальным, если фраза GLOBAL (ГЛОБАЛЬНОЕ) указана в статье описания данного, которой объявляется имя данного, или в другой статье, которой эта статья описания данного подчинена. Имя условия, объявленное в статье описания данного, является глобальным, если эта статья подчиняется другой статье, в которой определена фраза GLOBAL (ГЛОБАЛЬНОЕ). Однако иногда специальные правила запрещают задавать фразу GLOBAL (ГЛОБАЛЬНОЕ) для определенных статей описания данных, описания файлов или описания записей.

Имя файла является глобальным, если фраза GLOBAL (ГЛОБАЛЬНОЕ) определена в его статье описания файла.

Если имя данного, имя файла или имя условия, объявленное в статье описания данного, не является глобальным, оно является локальным.

Глобальные имена являются транзитивными для программ, содержащихся в других программах.

1.3.4. Внешние и внутренние объекты

Обеспечение доступности данных обычно требует, чтобы в памяти хранились определенные представления данных. Определи-

GOST EXPERT ГОСТ 22558-89,

тели файлов обычно требуют сохранения в памяти некоторой информации о файлах. Память, соответствующая данному или определителю файла, может быть внешней или внутренней по отношению к программе, в которой данный объект объявлен.

Данное или определитель файла является внешним, если память, связанная с этим объектом, соотнесена единице исполнения, а не какой-то конкретной программе внутри единицы исполнения. К внешнему объекту может обращаться любая программа в единице исполнения, которая описывает объект. Обращение ко внешнему объекту из разных программ, использующих отдельные описания объектов, — это всегда обращение к одному и тому же объекту. В единице исполнения имеется только одно представление внешнего объекта.

Объект является внутренним, если память, связанная с этим объектом, соотнесена только программе, которая описывает этот объект. Внешние и внутренние объекты могут иметь глобальные или локальные имена.

Записи данных, описанной в секции рабочей памяти, атрибут «внешнее» присваивается путем определения фразы EXTERNAL (ВНЕШНЕЕ) в ее статье описания данного. Любое данное, описанное статьей описания данного, которая подчиняется статье, описывающей внешнюю запись, тоже получает атрибут «внешнее». Если запись или данное не имеет атрибута «внешнее», она является частью внутренних данных программы, в которой она описана.

Определителю файла атрибут «внешний» присваивается фразой EXTERNAL (ВНЕШНЕЕ) в соответствующей статье описания файла. Если определитель файла не имеет атрибута «внешний», он является внутренним по отношению к программе, в которой описывается соответствующее имя-файла.

Записи данных, описания которых подчиняются статье описания файла, в которой нет фразы EXTERNAL (ВНЕШНЕЕ), или статье описания сортируемого-сливаемого файла, а также любые данные, описания которых подчиняются статьям описания данных для таких записей, всегда являются внутренними по отношению к программе, описывающей имя-файла. Если фраза EXTERNAL (ВНЕШНЕЕ) включена в статью описания файла, записи данных и данные получают атрибут «внешнее».

Записи данных, подчиненные данные и различная связанная с ними управляющая информация, которые описаны в секциях связи, коммуникаций и отчетов программы, всегда считаются внутрен-

GOST EXPERT ГОСТ 22558-89,

ними по отношению к программе, описывающей эти данные. Для данных, описанных в секции связи, существуют некоторые особенности, касающиеся установления соответствия между описанными записями данных и другими данными, доступными для других программ.

1.3.5. Общие и начальные программы

Все программы, образующие часть единицы исполнения, могут иметь один или оба из следующих атрибутов: COMMON (ОБЩАЯ) и INITIAL (НАЧАЛЬНАЯ), или они могут не иметь этих атрибутов вообще.

Общей программой называется программа, которая несмотря на то, что она является прямо содержащейся в другой программе, может быть вызвана программой, прямо или косвенно содержащейся в этой другой программе. Атрибут «общая» присваивается путем указания фразы COMMON (ОБЩАЯ) в разделе идентификации программы. Фраза COMMON (ОБЩАЯ) облегчает написание подпрограмм, которые будут использоваться всеми программами, содержащимися в данной программе.

Начальная программа — это программа, состояние которой инициируется во время вызова. Таким образом при каждом вызове начальной программы ее состояние такое же, каким оно было при первом вызове этой программы в единице исполнения. В процессе инициации начальной программы инициируются внутренние данные программы; таким образом, данное из внутренних данных программы, описание которого содержит фразу VALUE (ЗНАЧЕНИЕ), инициируется на это указанное значение, а даннос, описание которого не содержит фразу VALUE (ЗНАЧЕНИЕ), инициируется на неопределенное значение. Файлы с внутренними определителями файлов, которые связаны с программой, не находятся в открытом состоянии. Управляющие механизмы для всех операторов PERFORM (ВЫПОЛНИТЬ), находящихся в программе, устанавливаются в их начальное состояние. Атрибут «начальная» присваивается путем указания фразы INITIAL (НАЧАЛЬНАЯ) в разделе идентификации программы.

1.3.6. Общие данные

Две программы в единице исполнения могут обращаться к общим данным в следующих случаях:

(1) к содержимому внешней записи данных можно обращаться из любой программы при условии, что запись данных описана в этой программе;

(2) если программа содержится в другой программе, обе программы могут обращаться к данным, имеющим атрибут «гло-

GOST EXPERT ГОСТ 22558-89,

бальное» в содержащей программе или в любой программе, которая прямо или косвенно содержит содержащую программу;

(3) механизм, с помощью которого значение параметра передается ссылкой из вызывающей программы в вызываемую программу, создает общее данное; вызываемая программа, которая может использовать другой идентификатор, может обращаться к данному в вызывающей программе.

1.3.7. Общие файлы

Две программы в единице исполнения могут обращаться к общим определителям файлов в следующих случаях:

(1) к внешнему определителю файла может обращаться любая программа, которая описывает этот определитель файла;

(2) если программа содержится в другой программе, обе программы могут обращаться к общему определителю файла, указывая соответствующее глобальное имя-файла в содержащей программе или в любой программе, которая прямо или косвенно содержит содержащую программу.

1.3.8. Область действия имен

Если программы прямо или косвенно содержатся в других программах, каждая программа может использовать одинаковые слова пользователя для именования объектов независимо от использования этих определенных пользователем слов другими программами (см. ч. 4, п. 4.2.2.1.1). Когда существуют одноименные объекты, ссылка в программе на такое имя, даже когда это другой тип слова пользователя, является ссылкой на объект, описываемый этой программой, а не на объект, имеющий такое же имя и описанный в другой программе.

Ссылки на следующие типы слов пользователя могут использоваться в операторах и статьях только той программы, в которой эти слова пользователя объявлены:

1) имя-коммуникации;
2) имя-параграфа;
3) имя-секции.

Ссылки на следующие типы слов пользователя могут использоваться в любой программе на Коболе при условии, что система компиляции поддерживает соответствующие библиотеки или другие системы и объекты, к которым обращаются, известны этой системе:

1) имя-библиотеки;
2) имя-текста.

Ссылки на следующие типы слов пользователя, когда они объявлены в секции коммуникаций, могут использоваться в операто-

GOST EXPERT ГОСТ 22558-89,

рах и объектах только той программы, которая содержит эту секцию:

1) имя-условия;
2) имя-данного;
3) имя-записи.

Ссылки на следующие типы имен, когда они объявлены в секции конфигурации, могут использоваться в операторах и статьях только той программы, которая содержит секцию конфигурации, или любой программы, содержащейся в этой программе:

1) имя-алфавита;
2) имя-класса;
3) имя-условия;
4) мнемоническое-имя;
5) символическая-литера.

К следующим типам слов пользователя применяются специальные соглашения, касающиеся объявлений и ссылок, когда перечисленные выше случаи не применимы:

1) имя-условия;
2) имя-данного;
3) имя-файла;
4) имя-индекса;
5) имя-программы;
6) имя-записи;
7) имя-отчета.

1.3.8.1. *Соглашения для имен программ*

Имя программы объявляется в параграфе PROGRAM-ID (ПРОГРАММА) раздела идентификации программы. На имя программы можно ссылаться только в операторах CALL (ВЫЗВАТЬ), CANCEL (ОСВОБОДИТЬ) и заголовке конца программы END-PROGRAM (КОНЕЦ ПРОГРАММЫ). Имена, присваиваемые программам, которые образуют единицу исполнения, не обязаны быть уникальными, однако, когда две программы в единице исполнения имеют одинаковые имена, хотя бы одна из этих двух программ должна прямо или косвенно содержаться в другой отдельно компилируемой программе, которая не содержит вторую из этих двух программ.

Область действия имени программы определяется следующими правилами:

(1) если имя представляет имя программы, которая не имеет атрибут «общая» и которая прямо содержится в другой программе, оно может быть указано только в операторах, входящих в содержащую программу;

(2) если имя представляет имя программы, которая имеет атрибут «общая» и прямо содержится в другой программе, на

GOST EXPERT ГОСТ 22558-89,

имя программы можно ссылаться только в операторах, входящих в эту содержащую программу или в любые программы, прямо или косвенно содержащиеся в этой содержащей программе, за исключением программы с атрибутом «общая» и всех программ, содержащихся в ней;

(3) если имя представляет имя отдельно компилируемой программы, на имя программы можно ссылаться в операторах, входящих в любую программу в единице исполнения, кроме программ, которые она прямо или косвенно содержит.

1.3.8.2. Соглашения для имен условий, имен данных, имен файлов, имен записей и имен отчетов.

Если имена условий, имена данных, имена файлов, имена записей и имена отчетов объявлены в исходной программе, на эти имена можно ссылаться только в этой программе, за исключением случая, когда одно или несколько из этих имен являются глобальными и программа содержит другие программы.

Требования к уникальности имен, определенных в одной программе в качестве имен условий, имен данных, имен файлов, имен записей и имен отчетов объясняются в других параграфах этих спецификаций (см. ч. 4, п. 4.2.2.1.1).

Программа не может ссылаться на имена условий, имена данных, имена файлов, имена записей и имена отчетов, объявленные в программе, которую она содержит.

К глобальному имени можно обращаться в программе, в которой оно объявлено, или в любых программах, которые прямо или косвенно содержатся в этой программе.

Когда программа B1 прямо содержится в другой программе, программе A, обе программы могут определять имя условия, имя данного, имя файла, имя записи или имя отчета, используя одно и то же слово пользователя. Когда к такому дублируемому имени обращаются в программе B1, для установления объекта, к которому обращаются, используются следующие правила:

(1) набор имен, используемый для установления объекта, к которому обращаются, включает все имена, которые определены в программе B1, и все глобальные имена, которые определены в программе A и в программах, содержащих прямо или косвенно программу A. Обычные правила уточнения и все другие правила установления уникальности ссылки применяются для этого набора имен до тех пор, пока не будет идентифицирован один или несколько объектов;

(2) если идентифицируется только один объект, он является объектом, к которому обращаются;

(3) если идентифицируется несколько объектов, максимум один из них может иметь имя, локальное по отношению к прог-

GOST EXPERT ГОСТ 22558-89,

рамме В1. Если ни один или один из объектов имеет имя, локальное по отношению к В1, применяются следующие правила:

а) если имя объявлено в программе В1, объект в программе В1 является объектом, к которому обращаются;

б) в остальных случаях, если программа А содержится в другой программе, объектом, к которому обращаются, является:

1) объект в программе А, если имя объявлено в программе А;

2) объект в содержащей программе, если имя объявлено не в программе А, а в программе, содержащей программу А.

Это правило применяется для последующих содержащих программ до тех пор, пока не будет найдено единственное правильное имя.

1.3.8.3. *Соглашения для имен индексов*

Если данное, имеющее один из атрибутов «внешнее» или «глобальное» или оба эти атрибута, включает таблицу, доступ к которой осуществляется по индексу, этот индекс тоже имеет соответствующий атрибут или оба атрибута. Следовательно, область действия имени-индекса такая же, как область действия имени-данного, именующего таблицу, индекс которой именуется именем-индекса, и используются те же правила именования, что и для имен-данных. Имена-индексов не могут уточняться.

2. ВЛОЖЕННЫЕ ИСХОДНЫЕ ПРОГРАММЫ

2.1. **Общее описание**

Исходная программа Кобола — это синтаксически правильный набор операторов Кобола. Исходная программа Кобола может содержать другие исходные программы Кобола, и эти содержащиеся программы могут обращаться к некоторым ресурсам программы, в которой они содержатся.

2.2. **Организация**

За исключением операторов COPY (КОПИРОВАТЬ), REPLACE (ЗАМЕНИТЬ) и заголовка конца программы, операторы, статьи, параграфы и секции исходной программы Кобола группируются в четыре раздела, которые располагаются в следующей последовательности.

1. Раздел идентификации.
2. Раздел оборудования.
3. Раздел данных.
4. Раздел процедур.

Конец исходной программы Кобола указывается заголовком конца программы или отсутствием дополнительных строк в исходной программе.

GOST EXPERT ГОСТ 22558-89,

2.3. **Структура**

Ниже приводится общий формат и последовательность представления статей и операторов, которые образуют исходную программу Кобола. Обобщенные термины раздел-идентификации, раздел-оборудования, раздел-данных, исходная-программа и заголовок-конца — программы обозначают раздел идентификации Кобола, раздел оборудования Кобола, раздел данных Кобола, раздел процедур Кобола, исходную программу Кобола и заголовок конца программы Кобола, соответственно.

2.3.1. Общий формат

Раздел-идентификации
[раздел-оборудования]
[раздел-данных]
[раздел-процедур]
[исходная-программа] ...
[заголовок-конца-программы]

2.3.2. Синтаксические правила

(1) Заголовок-конца-программы должен быть указан, если:

а) исходная программа Кобола содержит одну или несколько других исходных программ Кобола, или

б) исходная программа Кобола в другой исходной программе Кобола.

2.3.3. Общие правила

(1) Начало раздела в программе указывается соответствующим заголовком раздела. Конец раздела указывается одним из следующих способов:

а) заголовком следующего раздела в этой программе;

б) заголовком раздела идентификации, который указывает на начало другой исходной программы;

в) заголовком конца программы;

г) физической позицией, после которой больше не появляются строки исходной программы.

(2) Исходная программа Кобола, которая прямо или косвенно содержится в другой программе, рассматривается в этих спецификациях как отдельная программа, которая может дополнительно обращаться к некоторым ресурсам, определенным в содержащей программе.

(3) Объектный код, получаемый в результате компиляции исходной программы, содержащейся в другой программе, рассматривается в этих спецификациях как неотъемлемая часть объектного кода, получающегося в результате компиляции содержащей программы.

2.4. **Начальное состояние программы**

Начальное состояние программы — это состояние программы в момент ее первого вызова в единице исполнения.

GOST EXPERT ГОСТ 22558-89,

2.4.1. Характеристики программы

(1) Инициируются внутренние данные программы, находящиеся в секции рабочей памяти и секции коммуникаций. Если фраза VALUE (ЗНАЧЕНИЕ) используется в описании данного, это данное инициируется на указанное значение.

Если данному не соответствует фраза VALUE (ЗНАЧЕНИЕ), начальное значение данного не определено.

(2) Файлы с внутренними определителями файлов, связанные с программой, не находятся в открытом состоянии.

(3) Управляющие механизмы для всех операторов PERFORM (ВЫПОЛНИТЬ), находящихся в программе, устанавливаются в свое начальное состояние.

(4) Оператор GO TO (ПЕРЕЙТИ К), на который ссылается оператор ALTER (ИЗМЕНИТЬ), находящийся в этой же программе, устанавливается в свое начальное состояние.

2.4.2. Программы в начальном состоянии

Программа находится в начальном состоянии:

(1) при первом вызове программы в единице исполнения;

(2) при первом вызове программы после выполнения оператора CANCEL (ОСВОБОДИТЬ) для этой программы или программы, которая прямо или косвенно содержит эту программу;

(3) при каждом вызове программы, если она имеет атрибут «начальная»;

(4) при первом вызове программы после выполнения оператора CALL (ВЫЗВАТЬ) для программы, которая имеет атрибут «начальная» и которая прямо или косвенно содержит данную программу.

2.5. **Заголовок конца программы**

2.5.1. Назначение

Заголовок конца программы указывает на конец названной исходной программы Кобола.

2.5.2. Общий формат

END PROGRAM имя-программы.

КОНЕЦ-ПРОГРАММЫ имя-программы.

2.5.3. Синтаксические правила

(1) Имя-программы должно подчиняться правилам образования слов, определяемых пользователем.

(2) Имя-программы должно быть таким же, как имя-программы, объявленное в предыдущем параграфе PROGRAM-ID (ПРОГРАММА) (п. 3.1 настоящей части).

(3) Если параграф PROGRAM-ID (ПРОГРАММА), объявляющий некоторое имя-программы, помещается между параграфом PROGRAM-ID (ПРОГРАММА) и заголовком конца прог-

GOST ГОСТ 22558-89,

раммы, соответственно объявляющим и ссылающимся на другое имя-программы, заголовок конца программы, указывающий первое имя-программы, должен предшествовать заголовку, в котором указывается второе имя-программы.

2.5.4. Общие правила

(1) Заголовок конца программы должен присутствовать в каждой программе, которая либо содержит другую программу, либо содержится в другой программе.

(2) Заголовок конца программы обозначает конец указанной исходной программы Кобола.

(3) Если программа, которая заканчивается заголовком конца программы, содержится в другой программе, следующий оператор должен быть заголовком раздела идентификации или другим заголовком конца программы, который завершает содержащую программу.

(4) Если программа, заканчивающаяся заголовком конца программы, не содержится в другой программе и если следующий исходный оператор является оператором Кобола, он должен быть заголовком раздела идентификации программы, которая должна компилироваться отдельно от программы, заканчивающейся заголовком конца программы.

3. РАЗДЕЛ ИДЕНТИФИКАЦИИ В МОДУЛЕ МЕЖПРОГРАММНЫХ СВЯЗЕЙ

3.1. Параграф PROGRAM-ID (ПРОГРАММА) и вложенные исходные программы

3.1.1. Назначение

Параграф PROGRAM-ID (ПРОГРАММА) определяет имя, по которому идентифицируется программа, и присваивает этой программе выбранные атрибуты.

3.1.2. Общий формат

PROGRAM-ID. имя-программы [IS { COMMON / INITIAL } PROGRAM].

ПРОГРАММА. имя-программы [[ОБЩАЯ / НАЧАЛЬНАЯ]].

GOST EXPERT ГОСТ 22558-89,

3.1.3. Синтаксические правила

(1) Имя-программы должно соответствовать правилам образования слов, определяемых пользователем.

(2) Программе, содержащейся в другой программе, нельзя присваивать имя, которое уже имеет некоторая другая программа, содержащаяся в отдельно компилируемой программе, которая содержит эту программу.

(3) Необязательная фраза COMMON (ОБЩАЯ) может использоваться только тогда, когда программа содержится в другой программе.

3.1.4. Общие правила

(1) Имя-программы идентифицирует исходную программу, объектную программу и все листинги, относящиеся к конкретной программе.

(2) Фраза COMMON (ОБЩАЯ) указывает, что программа является общей. Общая программа содержится в другой программе и может быть вызвана не только из программы, в которой она содержится, но и из других программ (см. п. 1.3.8 настоящей части).

(3) Фраза INITIAL (НАЧАЛЬНАЯ) указывает, что программа является начальной. При вызове начальной программы она и все содержащиеся в ней программы устанавливаются в их начальное состояние (см. п. 2.4 настоящей части).

4. РАЗДЕЛ ДАННЫХ В МОДУЛЕ МЕЖПРОГРАММНЫХ СВЯЗЕЙ

4.1. Секция связи

Секция связи находится в разделе данных исходной программы. Секция связи появляется в вызываемой программе и описывает данные, к которым будет обращаться и вызывающая и вызываемая программа.

Секция связи в программе необходима тогда и только тогда, когда объектная программа должна работать под управлением оператора CALL (ВЫЗВАТЬ) и оператор CALL (ВЫЗВАТЬ) в вызывающей программе содержит фразу USING (ИСПОЛЬЗУЯ). Секция связи используется для описания данных, которые доступны через вызывающую программу, но к которым можно обращаться как в вызывающей, так и в вызываемой программе. Механизм, с помощью которого устанавливается соответствие между данными, описанными в секции связи вызываемой программы, и данными, описанными в вызывающей программе, рассматривается в пп. 5.1 и 5.2 настоящей части. Для имен индексов такое соответствие не устанавливается, и имена индексов в вызываемой и вызывающей программах всегда ссылаются на отдельные индексы.

GOST EXPERT ГОСТ 22558-89,

Секция связи имеет такую же структуру, как и ранее описанная секция рабочей памяти. Сначала записывается заголовок секции, за ним следуют статьи описания несвязанных данных и (или) статьи описания записей.

Ниже приведен общий формат секции связи.

LINKAGE SECTION.

[статья-описания-уровня-77]
[статья-описания-записи] ...

СЕКЦИЯ СВЯЗИ.

[статья-описания-уровня-77]
[статья-описания-записи] ...

Если к данному секции связи обращаются в программе, которая не является вызываемой, результат не определен.

4.1.1. Несвязанная память

Данные в секции связи, которые не находятся в иерархическом отношении друг с другом, нет необходимости группировать в записи. Такие данные классифицируются и определяются как несвязанные элементарные данные. Каждое из этих данных определяется отдельной статьей описания данного, которая начинается специальным номером уровня 77.

В каждой статье описания данного необходимы следующие фразы:

а) номер-уровня-77

б) имя-данного

в) фраза PICTURE (ШАБЛОН) или фраза USAGE IS INDEX (ДЛЯ ИНДЕКСА).

Другие статьи описания данных необязательны и могут использоваться в случае необходимости для завершения описания данного.

4.1.2. Записи секции связи

Элементы данных в секции связи, которые состоят в определенных иерархических отношениях друг с другом, должны быть сгруппированы в записи в соответствии с правилами формирования описаний записей. Данные в секции связи, которые не состоят в иерархических отношениях с другими данными, могут быть описаны как записи, представляющие собой одиночные элементарные данные.

4.1.3. Начальные значения

Фраза VALUE (ЗНАЧЕНИЕ) не должна задаваться в секции связи. Исключение составляют статьи имен-условий (уровень 88).

4.2. **Статья описания файла в модуле межпрограммных связей**

4.2.1. Назначение

Внутри модуля межпрограммных связей статья описания файла в секции файлов определяет внутренние или внешние атрибуты определителя файла, соответствующих записей данных в соответствующих данных. Кроме того, статья описания файла устанавливает, является имя файла локальным или глобальным.

GOST EXPERT ГОСТ 22558-89,

4.2.2. Общий формат

Формат 1

```
FD имя-файла-1

[[IS EXTERNAL]
 [IS GLOBAL]]

[BLOCK CONTAINS [целое-1 TO] целое-2 {RECORDS / CHARACTERS}]

[RECORD {CONTAINS целое-3 CHARACTERS
         IS VARYING IN SIZE [ [FROM целое-4]
                 [TO целое-5] CHARACTERS]
                 [DEPENDING ON имя-данного-1]
         CONTAINS целое-6 TO целое-7
                 CHARACTERS}]

[LABEL {RECORD IS / RECORDS ARE} {STANDARD / OMITTED}]

[VALUE OF {имя-реализации-1 IS {имя-данного-2 / литерал-1}} ...]

[DATA {RECORD IS / RECORDS ARE} {имя-данного-3} ...]

[LINAGE IS {имя-данного-4 / целое-8} LINES [WITH FOOTING AT
   {имя-данного-5 / целое-9}]
      [LINES AT TOP {имя-данного-6 / целое-10}]
      [LINES AT BOTTOM {имя-данного-7 / целое-11}]]
[CODE-SET IS имя-алфавита-1].
```

```
ОФ имя-файла-1

[[ВНЕШНЕЕ]
 [ГЛОБАЛЬНОЕ]]
```


GOST EXPERT ГОСТ 22558-89,

[В БЛОКЕ [ОТ целое-1 ДО] целое-2 { ЗАПИСЕЙ / ЛИТЕР }]

[В ЗАПИСИ { целое-3 ЛИТЕР
ПЕРЕМЕННОЕ ЧИСЛО [[ОТ целое-4] [ДО целое-5] ЛИТЕР] [В ЗАВИСИМОСТИ ОТ имя-данного-1]
ОТ целое-6 ДО целое-7 ЛИТЕР }]

[МЕТКИ { СТАНДАРТНЫ / ОПУЩЕНЫ }]

[{ ЗНАЧЕНИЕ / ЗНАЧ } {имя-реализации-1 { имя-данного-2 / литерал-1 }} ...]

[ЗАПИСИ ДАННЫХ {имя-данного-3} ...]

[ВЕРСТКА { имя-данного-4 / целое-8 } СТРОК [КОНЦОВКА ОТ { имя-данного-5 / целое-9 }]]

[ВЕРХНЕЕ ПОЛЕ { имя-данного-6 / целое-10 }]

[НИЖНЕЕ ПОЛЕ { имя-данного-7 / целое-11 }]]

[АЛФАВИТ имя-алфавита-1].

Формат 2

FD имя-файла-1

[[IS EXTERNAL] / [IS GLOBAL]]

[BLOCK CONTAINS [целое-1 TO] целое-2 { RECORDS / CHARACTERS }]

[RECORD { CONTAINS целое-3 CHARACTERS
IS VARYING IN SIZE [[FROM целое-4] [TO целое-5] CHARACTERS] [DEPENDING ON имя-данного-1]
CONTAINS целое-6 TO целое-7 CHARACTERS }]

GOST EXPERT ГОСТ 22558-89,

[LABEL {RECORD IS / RECORDS ARE} {STANDARD / OMITTED}]

[VALUE OF {имя-реализации-1 IS {имя-данного-2 / литерал-1} ...]

[DATA {RECORD IS / RECORDS ARE} {имя-данного-3} ...].

ОФ имя-файла-1

[ВНЕШНЕЕ] / [ГЛОБАЛЬНОЕ]

[В БЛОКЕ [ОТ целое-1 ДО] целое-2 {ЗАПИСЕЙ / ЛИТЕР}]

[В ЗАПИСИ {целое-3 ЛИТЕР / ПЕРЕМЕННОЕ ЧИСЛО [{ОТ целое-4} [ДО целое-5] ЛИТЕР] [В ЗАВИСИМОСТИ ОТ имя-данного-1] / ОТ целое-6 ДО целое-7 ЛИТЕР}]

[МЕТКИ {СТАНДАРТНЫ / ОПУЩЕНЫ}]

[{ЗНАЧЕНИЕ / ЗНАЧ} {имя-реализации-1 {имя-данного-2 / литерал-1}} ...]

[ЗАПИСИ ДАННЫХ {имя-данного-3} ...].

Формат 3

FD имя-файла-1

[IS EXTERNAL] / [IS GLOBAL]

[BLOCK CONTAINS [целое-1 TO] целое-2 {RECORDS / CHARACTERS}]

[RECORD {CONTAINS целое-3 CHARACTERS / CONTAINS целое-6 TO целое-7 CHARACTERS}]

GOST EXPERT ГОСТ 22558-89,

```
[ LABEL { RECORD IS / RECORDS ARE } { STANDARD / OMITTED } ]
[ VALUE OF {имя-реализации-1 IS { имя-данного-2 / литерал-1 } } ... ]
[CODE-SET IS имя-алфавита]
{ REPORT IS / REPORTS ARE } {имя-отчета-1} ... .
```

ОФ имя-файла-1

```
[ВНЕШНЕЕ]
[ГЛОБАЛЬНОЕ]
[ В БЛОКЕ [ОТ целое-1 ДО] целое-2 { ЗАПИСЕЙ / ЛИТЕР } ]
[ В ЗАПИСИ { целое-3 ЛИТЕР / ОТ целое-6 ДО целое-7 ЛИТЕР } ]
[ МЕТКИ { СТАНДАРТНЫ / ОПУЩЕНЫ } ]
[ { ЗНАЧЕНИЕ / ЗНАЧ } { имя-реализации-1 { имя-данного-2 / литерал-1 } } ... ]
[АЛФАВИТ имя-алфавита-1]
{ ОТЧЕТ / ОТЧЕТЫ } {имя-отчета-1} ... .
```

4.2.3. Синтаксические правила

(1) Формат 1 — это статья описания последовательного файла. Присутствие конкретных фраз в этой статье описания файла зависит от уровня модуля последовательного ввода-вывода, обеспечиваемого реализацией (см. ч. 7, п. 3.2).

(2) Формат 2 — это статья описания относительного файла или индексного файла. Присутствие конкретных фраз в этой статье описания файла зависит от уровня модуля относительного ввода-вывода или модуля индексного ввода-вывода, обеспечиваемого реализацией (см. ч. 8, п. 3.2, ч. 9, п. 3.2).

(3) Формат 3 — это статья описания файла отчетов. Наличие статьи описания для файла отчетов зависит от того, обеспечивается ли модуль генератора отчетов в данной реализации (ч. 13, п. 3.2).

GOST EXPERT ГОСТ 22558-89,

4.2.4. Общие правила

(1) Если статья описания последовательного файла содержит фразу LINAGE (ВЕРСТКА) и фразу EXTERNAL (ВНЕШНЕЕ), данное LINAGE-COUNTER (СЧЕТЧИК-ВЕРСТКИ) является внешним данным. Если статья описания последовательного файла содержит фразу LINAGE (ВЕРСТКА) и фразу GLOBAL (ГЛОБАЛЬНОЕ), специальный регистр LINAGE-COUNTER (СЧЕТЧИК-ВЕРСТКИ) является глобальным именем.

(2) Фраза EXTERNAL (ВНЕШНЕЕ) описана в п. 4.5. Фраза GLOBAL (ГЛОБАЛЬНОЕ) описана в п. 4.6. Все остальные фразы в статье описания файла описаны в соответствующем модуле в этих спецификациях.

4.3. **Статья описания данного в модуле межпрограммных связей**

4.3.1. Назначение

В модуле межпрограммных связей статья описания данного уровня 01 в секции рабочей памяти или в секции файлов устанавливает, какие имена — локальные или глобальные, имеет запись данных и подчиненные ей данные.

В модуле межпрограммных связей статья описания данного уровня 01 в секции рабочей памяти определяет атрибут «внутреннее» или «внешнее» для записи данного и подчиненных ей данных.

4.3.2. Общий формат

```
01 [ имя-данного-1 ]
   [ FILLER       ]

   [REDEFINES имя-данного-2]

   [IS EXTERNAL]
   [IS GLOBAL]

   [ { PICTURE } IS строка-литер ]
   [ { PIC     }                 ]

   [ USAGE IS { BINARY                } ]
              { COMPUTATIONAL         }
              { COMP                  }
              { DISPLAY               }
              { INDEX                 }
              { PACKED-DECIMAL        }

   [ [SIGN IS] { LEADING  } [SEPARATE CHARACTER] ]
               { TRAILING }
```


GOST EXPERT ГОСТ 22558-89,

```
[ OCCURS целое-2 TIMES
      [ { ASCENDING | DESCENDING } KEY IS {имя-данного-3} ... ] ...
  [INDEXED BY {имя-индекса-1} ... ]
  OCCURS целое-1 TO целое-2 TIMES DEPENDING ON
                                  имя-данного-4
      [ { ASCENDING | DESCENDING } KEY IS {имя-данного-3} ... ] ...
    [INDEXED BY {имя-индекса-1} ... ] ]
[ { SYNCHRONIZED | SYNC } [ LEFT | RIGHT ] ]
[ { JUSTIFIED | JUST } RIGHT ]
[BLANK WHEN ZERO]
[VALUE IS литерал-1].
01 [ имя-данного-1 | ЗАП | ЗАПОЛНИТЕЛЬ ]
[ПЕРЕОПРЕДЕЛЯЕТ имя-данного-2]
[ [ВНЕШНЕЕ] | [ГЛОБАЛЬНОЕ] ]
[ { ШАБЛОН | Ш } строка-литер ]
[ ДЛЯ { ВЫЧИСЛЕНИЙ | ВЫЧ | ВЫДАЧИ | ИНДЕКСА }
  ДВОИЧНОЕ | ДЕСЯТИЧНОЕ ]
[ [ЗНАК] { ПЕРВЫЙ | ПОСЛЕДНИЙ } [ОТДЕЛЬНО] ]
```


GOST EXPERT ГОСТ 22558-89,

```
[ ПОВТОРЯЕТСЯ целое-2 РАЗ
      [ ПО { ВОЗРАСТАНИЮ / УБЫВАНИЮ } КЛЮЧА
                                  {имя-данного-3} ... ] ...
      [ИНДЕКСИРУЕТСЯ {имя-индекса-1} ... ]
  ПОВТОРЯЕТСЯ ОТ целое-1 ДО целое-2 РАЗ
      В ЗАВИСИМОСТИ ОТ имя-данного-4
      [ ПО { ВОЗРАСТАНИЮ / УБЫВАНИЮ } КЛЮЧА
                                  {имя-данного-3} ... ] ...
      [ИНДЕКСИРУЕТСЯ {имя-индекса-1} ... ] ]
[ ВЫДЕЛЕНО [ ВЛЕВО / ВПРАВО ] ]
[СДВИНУТО ВПРАВО]
[ПРОБЕЛ КОГДА НУЛЬ]
[ { ЗНАЧЕНИЕ / ЗНАЧ } литерал-1 ].
```

4.3.3. Синтаксические правила

(1) Наличие конкретных фраз в статье описания данного зависит от уровни ядра, обеспечиваемого в данной реализации (см. ч. 6, п. 5.3).

(2) Фраза EXTERNAL (ВНЕШНЕЕ) может быть задана только в статьях описания данных уровня 01 в секции рабочей памяти.

(3) Фраза EXTERNAL (ВНЕШНЕЕ) и фраза REDEFINES (ПЕРЕОПРЕДЕЛЯЕТ) не должны задаваться в одной и той же статье описания данного.

(4) Фраза GLOBAL (ГЛОБАЛЬНОЕ) может быть задана только в статьях описания данных уровня 01.

(5) Имя-данного-1 должно быть задано для любой статьи, содержащей фразу GLOBAL (ГЛОБАЛЬНОЕ) или EXTERNAL (ВНЕШНЕЕ), или для описаний записей, связанных со статьей описания файла, которая содержит фразу EXTERNAL (ВНЕШНЕЕ) или GLOBAL (ГЛОБАЛЬНОЕ).

GOST EXPERT ГОСТ 22558-89,

4.3.4. Общие правила

(1) Фраза EXTERNAL (ВНЕШНЕЕ) описана в п. 4.5. Фраза GLOBAL (ГЛОБАЛЬНОЕ) описана в п. 4.6. Все другие фразы статьи описания данного описываются в модуле Ядро (см. ч. 6, п. 5.3).

4.4. **Статья описания отчета в модуле межпрограммных связей**

4.4.1. Назначение

В модуле межпрограммных связей статья описания отчета в секции отчетов устанавливает, является имя отчета локальным именем или глобальным.

4.4.2. Общий формат

RD имя-отчета-1

[[IS GLOBAL]]

[CODE литерал-1]

[{CONTROL IS / CONTROLS ARE} {имя-данного-1} ... / {FINAL [имя-данного-1] ...}]

[PAGE {LIMIT IS / LIMITS ARE} целое-1 [LINE / LINES] [HEADING целое-2]

[FIRST DETAIL целое-3] [LAST DETAIL целое-4]

FOOTING целое-5].

ОО имя-отчета-1

[[ГЛОБАЛЬНОЕ]]

[С КОДОМ литерал-1]

[УПРАВЛЕНИЕ ПО {имя-данного-1} ... / КОНЦУ [имя-данного-1] ...]

[РАЗМЕР СТРАНИЦЫ целое-1 СТРОК

[ЗАГОЛОВОК целое-2]

[ПЕРВЫЙ ФРАГМЕНТ целое-3]

[ПОСЛЕДНИЙ ФРАГМЕНТ целое-4]

[КОНЦОВКА целое-5]].

4.4.3. Синтаксические правила

(1) Наличие статьи описания отчета зависит от того, обеспечивается ли модуль генератора отчетов в данной реализации (ч. 13, п. 3.5).

GOST EXPERT ГОСТ 22558-89,

4.4.4. Общие правила

(1) Если статья описания отчета содержит фразу GLOBAL (ГЛОБАЛЬНОЕ), специальные регистры LINE-COUNTER (СЧЕТЧИК-СТРОК) и PAGE-COUNTER (СЧЕТЧИК-СТРАНИЦ) являются глобальными именами.

(2) Фраза GLOBAL (ГЛОБАЛЬНОЕ) описана в п. 4.6. Все остальные фразы в статье описания отчета описаны в модуле генератора отчетов в этих спецификациях.

4.5. **Фраза EXTERNAL (ВНЕШНЕЕ)**

4.5.1. Назначение

Фраза EXTERNAL (ВНЕШНЕЕ) указывает, что данное или определитель файла является внешним. Составляющие данные и групповые данные внешней записи данного доступны каждой программе в единице исполнения, которая описывает эту запись.

4.5.2. Общий формат

IS EXTERNAL
ВНЕШНЕЕ

4.5.3. Синтаксические правила

(1) Фраза EXTERNAL (ВНЕШНЕЕ) может быть задана только в статьях описания файлов (см. пп. 4.2, 4.3 настоящей части) или в статьях описания записей в секции рабочей памяти (см. п. 4.3 настоящей части).

(2) В одной и той же программе имя-данного, указанное в качестве субъекта статьи с номером уровня 01, включающей фразу EXTERNAL (ВНЕШНЕЕ), не должно совпадать с именем-данного, указанного для любой другой статьи описания данного, которая включает фразу EXTERNAL (ВНЕШНЕЕ).

(3) Фразу VALUE (ЗНАЧЕНИЕ) нельзя использовать ни в одной статье описания данного, которая включает статью, содержащую фразу EXTERNAL (ВНЕШНЕЕ), или подчиняется такой статье. Фраза VALUE (ЗНАЧЕНИЕ) может быть задана для статей имен-условий, связанных с такими статьями описания данных.

4.5.4. Общие правила

(1) Данные, находящиеся в записи, которую именует фраза имени-данного, являются внешними и могут быть доступны и обрабатываться любой программой в единице исполнения, которая описывает и, возможно, переопределяет их в соответствии со следующими общими правилами.

(2) Если в единице исполнения две или несколько программ описывают одну и ту же внешнюю запись данных, каждое имя-записи соответствующих статей описания записей должно быть одним и тем же, а записи должны определять одно и то же ко-

GOST EXPERT ГОСТ 22558-89,

личество литер в стандартном формате данных. Однако программа, которая описывает внешнюю запись, может содержать статью описания данного с фразой REDEFINES (ПЕРЕОПРЕДЕЛЯЕТ), переопределяющей полную внешнюю запись, и это полное переопределение не должно появляться в идентичном виде в других программах в единице исполнения (см. ч. 6, п. 5.10).

(3) Использование фразы EXTERNAL (ВНЕШНЕЕ) не подразумевает, что соответствующее имя-файла или имя-данного является глобальным именем (п. 4.6 настоящей части).

(4) Определитель файла, связанный с этой статьей описания, является внешним определителем файла.

4.6. **Фраза GLOBAL (ГЛОБАЛЬНОЕ)**

4.6.1. Назначение

Фраза GLOBAL (ГЛОБАЛЬНОЕ) указывает, что имя-данного, имя-файла или имя-отчета является глобальным именем. Глобальное имя доступно каждой программе, содержащейся в программе, которая объявляет это имя.

4.6.2. Общий формат

IS GLOBAL

ГЛОБАЛЬНОЕ

4.6.3. Синтаксические правила

(1) Фраза GLOBAL (ГЛОБАЛЬНОЕ) может быть задана только в статьях описания данных уровня 01 в секции файлов или секции рабочей памяти, статьях описания файлов или статьях описания отчетов.

(2) В одном и том же разделе данных статьи описания данных для любых двух данных, для которых задано одинаковое имя, не должны включать фразу GLOBAL (ГЛОБАЛЬНОЕ).

(3) Если для нескольких файлов задана фраза SAME RECORD AREA (ОБЩАЯ ОБЛАСТЬ ЗАПИСИ), статьи описания записей или статьи описания файлов для этих файлов не должны включать фразу GLOBAL (ГЛОБАЛЬНОЕ).

4.6.4. Общие правила

(1) Имя-данного, имя-файла или имя-отчета, описанное фразой GLOBAL (ГЛОБАЛЬНОЕ), является глобальным именем. Все имена данных, подчиненные глобальному имени, являются глобальными именами. Все имена условий, связанные с глобальным именем, являются глобальными именами.

(2) Оператор в программе, которая прямо или косвенно содержится в программе, описывающей глобальное имя, может обращаться к этому имени без его повторного описания (см. п. 1.3.8 настоящей части).

(3) Если фраза GLOBAL (ГЛОБАЛЬНОЕ) используется в статье описания данного, которая содержит фразу REDEFINES (ПЕРЕОПРЕДЕЛЯЕТ) или RENAMES (ПЕРЕИМЕНОВЫВА-

GOST EXPERT ГОСТ 22558-89,

ЕТ), атрибутом «глобальное» обладает только субъект фразы REDEFINES (ПЕРЕОПРЕДЕЛЯЕТ) или RENAMES (ПЕРЕИМЕНОВЫВАЕТ).

5. РАЗДЕЛ ПРОЦЕДУР В МОДУЛЕ МЕЖПРОГРАММНЫХ СВЯЗЕЙ

5.1. Заголовок раздела процедур

Раздел процедур идентифицируется и должен начинаться следующим заголовком:

PROCEDURE DIVISION [USING {имя-данного-1} . . .].

РАЗДЕЛ ПРОЦЕДУР [ИСПОЛЬЗУЯ {имя-данного-1} . . .].

Фраза USING (ИСПОЛЬЗУЯ) требуется только тогда, когда объектная программа будет вызываться по оператору CALL (ВЫЗВАТЬ), который содержит фразу USING (ИСПОЛЬЗУЯ).

Фраза USING (ИСПОЛЬЗУЯ) в заголовке раздела процедур идентифицирует имена, используемые программой для всех параметров, которые передаются ей из вызывающей программы. Параметры, передаваемые вызываемой программе, идентифицируются во фразе USING (ИСПОЛЬЗУЯ) оператора CALL (ВЫЗВАТЬ) в вызывающей программе. Соответствие между этими двумя списками имен устанавливается по позиционному принципу.

Имя-данного-1 должно быть определено как статья уровня 01 или статья уровня 77 в секции связи. Конкретное слово, определенное пользователем, может появляться в качестве имени-данного-1 только один раз. Статья описания данного для имени-данного-1 не должна содержать фразу REDEFINES (ПЕРЕОПРЕДЕЛЯЕТ). Однако имя-данного-1 может быть объектом фразы REDEFINES (ПЕРЕОПРЕДЕЛЯЕТ) в других местах в секции связи.

Применяются следующие дополнительные правила:

(1) если ссылка на соответствующее данное в операторе CALL (ВЫЗВАТЬ) объявляет передачу параметра по значению, значение данного пересылается во время выполнения оператора CALL (ВЫЗВАТЬ) и помещается в определяемый системой элемент памяти, который имеет атрибуты, объявленные в секции связи для имени-данного-1. Описание данных для каждого параметра во фразе BY CONTENT (ЗНАЧЕНИЕ) оператора CALL (ВЫЗВАТЬ) должно быть таким же, как описание данного для соответствующего параметра во фразе USING (ИСПОЛЬЗУЯ) заголовка раздела процедур, т. е. не должно требоваться какого-либо преобразования, расширения или усечения (см. п. 5.2 настоящей части);

(2) если ссылка на соответствующее данное в операторе CALL (ВЫЗВАТЬ) объявляет передачу параметра ссылкой, объектная

GOST EXPERT ГОСТ 22558-89,

программа выполняется так, как если бы данное в вызываемой программе занимало такую же область в памяти, как и данное в вызывающей программе.

Описание данного в вызываемой программе должно определять такое же количество позиций литер, какое указано в описании соответствующего данного в вызывающей программе;

(3) во всех случаях в вызываемой программе ссылки на имя-данного-1 разрешаются в соответствии с описанием данного в секции связи вызываемой программы;

(4) к данным, определенным в секции связи вызываемой программы, можно обращаться внутри раздела процедур этой программы тогда и только тогда, когда они удовлетворяют одному из следующих условий:

а) они являются операндами фразы USING (ИСПОЛЬЗУЯ) заголовка раздела процедур;

б) они подчиняются операндам фразы USING (ИСПОЛЬЗУЯ) заголовка раздела процедур;

в) они определяются фразой REDEFINES (ПЕРЕОПРЕДЕЛЯЕТ) или RENAMES (ПЕРЕИМЕНОВЫВАЕТ), объект которой удовлетворяет приведенным выше условиям;

г) они являются элементами, подчиненными любому элементу, который удовлетворяет условию в правиле 4в;

д) они являются именами-условий или именами-индексов, связанными с данными, которые удовлетворяют любому из четырех перечисленных выше условий.

в) в п. 1 во фразе USING (ИСПОЛЬЗУЯ) заголовка раздела процедур и во фразе USING (ИСПОЛЬЗУЯ) оператора CALL (ВЫЗВАТЬ) должно допускаться не менее пяти имен данных.

CALL (ВЫЗВАТЬ)

5.2.1. Назначение

Оператор CALL (ВЫЗВАТЬ) вызывает передачу управления от одной объектной программы к другой внутри единицы исполнения.

5.2.2. Общий формат

Формат 1

```
CALL { идентификатор-1 } [ USING
     { литерал-1       }

      [BY REFERENCE] {идентификатор-2} ...  } ...  ]
      BY CONTENT     {идентификатор-2} ...  }

[ON OVERFLOW повелительный-оператор-1] [END-CALL]
```


GOST EXPERT ГОСТ 22558-89,

ВЫЗВАТЬ { идентификатор-1 | литерал-1 }

[ИСПОЛЬЗУЯ { [ССЫЛКУ НА] {идентификатор-2} ... | ЗНАЧЕНИЕ {идентификатор-2} ... } ...]

[ПРИ ПЕРЕПОЛНЕНИИ повелительный-оператор-1]

[КОНЕЦ-ВЫЗВАТЬ]

Формат 2

CALL { идентификатор-1 | литерал-1 }

[USING { [BY REFERENCE] {идентификатор-2} ... | BY CONTENT {идентификатор-2} ... } ...]

[ON EXCEPTION повелительный-оператор-1]
[NOT ON EXCEPTION повелительный-оператор-2]

[END-CALL]

ВЫЗВАТЬ { идентификатор-1 | литерал-1 }

[ИСПОЛЬЗУЯ { [ССЫЛКУ НА] {идентификатор-2} ... | ЗНАЧЕНИЕ {идентификатор-2} ... } ...]

[ПРИ ОШИБКЕ повелительный-оператор-1]
[БЕЗ ОШИБКИ повелительный-оператор-2]

[КОНЕЦ-ВЫЗВАТЬ]

5.2.3. Синтаксические правила

(1) Литерал-1 должен быть нечисловым литералом.

GOST EXPERT ГОСТ 22558-89,

(2) Идентификатор-1 должен быть определен как буквенно-цифровое данное такое, что его значение может быть именем программы.

(3) Каждый операнд во фразе USING (ИСПОЛЬЗУЯ) должен быть предварительно определен как данное в секции файлов, секции рабочей памяти, секции коммуникаций или секции связи и должен быть данным уровня 01, уровня 77 или элементарным данным.

5.2.4. Общие правила

(1) Литерал-1 или содержимое данного, представленного идентификатором-1, является именем вызываемой программы. Программа, в которой появляется оператор CALL (ВЫЗВАТЬ), является вызывающей программой. Если вызываемая программа — это программа Кобола, то литерал-1 или содержимое данного, представленного идентификатором-1, должно быть именем программы, содержащимся в параграфе PROGRAM-ID (ПРОГРАММА) вызываемой программы. Если вызываемая программа — не программа Кобола, правила формирования имени программы определяются реализацией.

(2) Если во время выполнения оператора CALL (ВЫЗВАТЬ) программа, указанная в операторе CALL (ВЫЗВАТЬ), становится доступной для выполнения, управление передается вызываемой программе. После возврата управления от вызываемой программы фраза ON OVERFLOW (ПРИ ПЕРЕПОЛНЕНИИ) или ON EXCEPTION (ПРИ ОШИБКЕ), если она задана, игнорируется, и управление передается на конец оператора CALL (ВЫЗВАТЬ), или, если задана фраза NOT ON EXCEPTION (БЕЗ ОШИБКИ), повелительному оператору-2. При передаче управления повелительному-оператору-2 выполнение продолжается в соответствии с правилами для каждого оператора, указанного в повелительном-операторе-2. Если выполняется оператор ветвления процедур или условный оператор, который вызывает явную передачу управления, управление передается в соответствии с правилами для этого оператора; в противном случае управление передается на конец оператора CALL (ВЫЗВАТЬ) после того, как будет выполнен повелительный-оператор-2.

(3) Если во время выполнения оператора CALL (ВЫЗВАТЬ) устанавливается, что программа, заданная в операторе CALL (ВЫЗВАТЬ), не может стать доступной для выполнения в это время, будет выполнено одно из указанных действий. Ресурсы времени выполнения, которые должны проверяться с целью установления

GOST EXPERT ГОСТ 22558-89,

доступности вызываемой программы для выполнения, определяются реализацией.

а) Если в операторе CALL (ВЫЗВАТЬ) задана фраза ON OVERFLOW (ПРИ ПЕРЕПОЛНЕНИИ) или ON EXCEPTION (ПРИ ОШИБКЕ), управление передается повелительному-оператору-1. Затем выполнение продолжается в соответствии с правилами для каждого оператора, указанного в повелительном-операторе-1. Если выполняется оператор ветвления процедур или условный оператор, который вызывает явную передачу управления, управление передается в соответствии с правилами для этого оператора; в противном случае управление передается на конец оператора CALL (ВЫЗВАТЬ) после выполнения повелительного-оператора-1, а фраза NOT ON EXCEPTION (БЕЗ ОШИБКИ), если она задана, игнорируется.

б) Если фраза ON OVERFLOW (ПРИ ПЕРЕПОЛНЕНИИ) или ON EXCEPTION (ПРИ ОШИБКЕ) не задана в операторе CALL (ВЫЗВАТЬ), фраза NOT ON EXCEPTION (БЕЗ ОШИБКИ), если она задана, игнорируется. Все остальные действия оператора CALL (ВЫЗВАТЬ) определяются реализацией.

(4) Две или несколько программ в единице исполнения могут иметь одно и то же имя-программы, и ссылка в операторе CALL (ВЫЗВАТЬ) на такое имя-программы разрешается по правилам для области действия для имен программ (см. п. 1.3.8.1 настоящей части).

Например, когда только две программы в единице исполнения имеют одинаковое имя, которое указано в операторе CALL (ВЫЗВАТЬ):

а) одно из этих двух программ должна прямо или косвенно содержаться в отдельно компилируемой программе, в которой находится этот оператор CALL (ВЫЗВАТЬ), или в отдельно компилируемой программе, которая сама прямо или косвенно содержит программу, в которой находится этот оператор CALL (ВЫЗВАТЬ), и

б)-вторая из этих двух программ должна быть другой отдельно компилируемой программой.

Механизм, используемый в этом примере, состоит в следующем:

а) если одна из двух программ, имеющих одинаковое имя, которое указано в операторе CALL (ВЫЗВАТЬ), прямо содержится в программе, в которой находится оператор CALL (ВЫЗВАТЬ), эта программа вызывается;

б) если одна из двух программ, имеющих одинаковое имя, которое указано в операторе CALL (ВЫЗВАТЬ), имеет атрибут «общая» и прямо содержится в другой программе, прямо или

GOST EXPERT ГОСТ 22558-89,

косвенно содержащей программу, в которой находится оператор CALL (ВЫЗВАТЬ), вызывается эта общая программа, если только вызывающая программа не содержится в этой общей программе;

в) в остальных случаях вызывается отдельно компилируемая программа.

(5) Если вызываемая программа не имеет атрибута «начальная», тогда она и каждая программа, прямо или косвенно содержащаяся в ней, находятся в начальном состоянии при первом вызове этой программы в единице исполнения и при ее первом вызове после оператора CANCEL (ОСВОБОДИТЬ) для вызываемой программы.

При всех остальных входах в вызываемую программу состояние этой программы и каждой программы, прямо или косвенно содержащейся в ней, остается таким же, каким оно было во время последнего выхода из программы.

(6) Если вызываемая программа имеет атрибут «начальная», она и каждая программа, прямо или косвенно содержащаяся в ней, устанавливается в свое начальное состояние при каждом вызове вызываемой программы в единице исполнения.

(7) Файлы, связанные с внутренними определителями файлов вызываемой программы, не находятся в открытом состоянии, когда программа находится в начальном состоянии (см. п. 2.4 настоящей части).

При всех остальных входах в вызываемую программу состояние и позиционирование всех таких файлов такое, каким оно было при последнем выходе из вызываемой программы.

(8) Процесс вызова программы или выхода из вызываемой программы не изменяет состояние или позиционирование файла, связанного с внешним определителем файла.

(9) Если вызываемая программа — это программа Кобола, фраза USING (ИСПОЛЬЗУЯ) включается в оператор CALL (ВЫЗВАТЬ) только тогда, когда в заголовке раздела процедур вызываемой программы имеется фраза USING (ИСПОЛЬЗУЯ); в этом случае количество операндов в каждой фразе USING (ИСПОЛЬЗУЯ) должно быть одинаковым. Если вызываемая программа — это программа не на Коболе, использование фразы USING (ИСПОЛЬЗУЯ) определяется реализацией.

(10) Последовательность появления имен-данных во фразе USING (ИСПОЛЬЗУЯ) оператора CALL (ВЫЗВАТЬ) и в соот-

GOST EXPERT ГОСТ 22558-89,

ветствующей фразе USING (ИСПОЛЬЗУЯ) в заголовке раздела процедур вызываемой программы устанавливает соответствие между именами-данных, используемыми в вызывающей и вызываемой программах. Это соответствие является позиционным, а не определяется эквивалентностью имен; первое имя-данного в одной фразе USING (ИСПОЛЬЗУЯ) соответствует первому имени-данного во второй фразе, второе имя-данного в одной фразе соответствует второму имени данного во второй фразе USING (ИСПОЛЬЗУЯ) и т. д.

(11) Значения параметров, названных во фразе USING (ИСПОЛЬЗУЯ) оператора CALL (ВЫЗВАТЬ), становятся доступными для вызываемой программы во время выполнения оператора CALL (ВЫЗВАТЬ).

(12) Обе фразы BY CONTENT (ЗНАЧЕНИЕ) и BY REFERENCE (ССЫЛКУ НА) распространяются на все параметры, следующие за ними, до тех пор, пока не встретится другая фраза BY CONTENT (ЗНАЧЕНИЕ) или BY REFERENCE (ССЫЛКУ НА). Если перед первым параметром не задана ни фраза BY CONTENT (ЗНАЧЕНИЕ), ни BY REFERENCE (ССЫЛКУ НА), подразумевается фраза BY REFERENCE (ССЫЛКУ НА).

(13) Если для параметра задана или подразумевается фраза BY REFERENCE (ССЫЛКУ НА), объектная программа выполняется так, как если бы соответствующее данное в вызываемой программе занимало такую же область памяти, как и данное в вызывающей программе. Описание данного в вызываемой программе должно определять такое же количество позиций литер, как и описание соответствующего данного в вызывающей программе.

(14) Если для параметра задана или подразумевается фраза BY CONTENT (ЗНАЧЕНИЕ), вызываемая программа не может изменить значение этого параметра, указанного во фразе USING (ИСПОЛЬЗУЯ) оператора CALL (ВЫЗВАТЬ), хотя вызываемая программа может изменить значение данного, представленного соответствующим именем данного в заголовке раздела процедур вызываемой программы. Описание данного для каждого параметра во фразе BY CONTENT (ЗНАЧЕНИЕ) оператора CALL (ВЫЗВАТЬ) должно быть таким же, как описание данного для соответствующего параметра во фразе USING (ИСПОЛЬЗУЯ) заголовка раздела процедур, т. е. не должно требоваться какого-либо преобразования, расширения или усечения (см. п. 5.1 настоящей части).

(15) Вызываемые программы могут содержать операторы CALL (ВЫЗВАТЬ). Однако вызываемая программа не должна выполнять оператор CALL (ВЫЗВАТЬ), который прямо или косвенно вызывает вызывающую программу. Если оператор CALL

GOST EXPERT ГОСТ 22558-89,

(ВЫЗВАТЬ) выполняется в области действия декларативы, оператор CALL (ВЫЗВАТЬ) не может прямо или косвенно обратиться к какой-либо вызываемой программе, которой было передано управление и выполнение которой еще не завершено.

(16) Фраза END-CALL (КОНЕЦ-ВЫЗВАТЬ) ограничивает область действия оператора CALL (ВЫЗВАТЬ) (см. ч. 4, п. 7).

5.3. **Оператор CANCEL (ОСВОБОДИТЬ)**

5.3.1. Назначение

Оператор CANCEL (ОСВОБОДИТЬ) обеспечивает, что при следующем вызове названной программы она будет находиться в начальном состоянии.

5.3.2. Общий формат

CANCEL { идентификатор-1 / литерал-1 } ...

ОСВОБОДИТЬ { идентификатор-1 / литерал-1 } ...

5.3.3. Синтаксические правила

(1) Литерал-1 должен быть нечисловым.

(2) Идентификатор-1 должен относиться к буквенно-цифровому данному.

5.3.4. Общие правила

(1) Литерал-1 или содержимое данного, представленного идентификатором-1, идентифицирует программу, которая должна быть освобождена.

(2) В результате выполнения явного или неявного оператора CANCEL (ОСВОБОДИТЬ) прекращается всякая логическая связь программы, которая указывается в нем, с единицей исполнения, в которой появляется оператор CANCEL (ОСВОБОДИТЬ). Если программа, названная в успешно выполненном в единице исполнения явном или неявном операторе CANCEL (ОСВОБОДИТЬ), позднее вызывается в этой же единице исполнения, эта программа находится в начальном состоянии (см. пп. 1.3.8, 2.4 и 5.2 настоящей части).

(3) Программа, названная в операторе CANCEL (ОСВОБОДИТЬ) в другой программе, должна быть вызвана этой другой программой (см. пп. 1.3.8 и 5.2 настоящей части).

(4) При выполнении явного или неявного оператора CANCEL (ОСВОБОДИТЬ) все программы, содержащиеся в программе, которая названа в операторе CANCEL (ОСВОБОДИТЬ), тоже освобождаются. Результат такой же, как если бы правильный оператор CANCEL (ОСВОБОДИТЬ) выполнялся для каждой содержащейся программы в последовательности, обратной той, в которой программы появляются в отдельно компилируемой программе.

GOST EXPERT ГОСТ 22558-89,

(5) Программа, названная в операторе CANCEL (ОСВОБОДИТЬ), не должна прямо или косвенно обращаться к программе, которая была вызвана, но еще не выполнила оператор EXIT PROGRAM (ВЫЙТИ ИЗ ПРОГРАММЫ).

(6) Логическая связь с освобожденной программой устанавливается только в результате выполнения следующего оператора CALL (ВЫЗВАТЬ), в котором названо имя этой программы.

(7) Вызываемая программа освобождается посредством указания ее в качестве операнда оператора CANCEL (ОСВОБОДИТЬ), в результате завершения единицы исполнения, в состав которой входит данная программа, или в результате выполнения оператора EXIT PROGRAM (ВЫЙТИ ИЗ ПРОГРАММЫ) в вызываемой программе, имеющей атрибут «начальная».

(8) Никаких действий не выполняется при выполнении явного или неявного оператора CANCEL (ОСВОБОДИТЬ), называющего программу, которая не была вызвана в этой единице исполнения или была вызвана, но в настоящее время освобождена. Управление передается следующему выполнимому оператору после явного оператора CANCEL (ОСВОБОДИТЬ).

(9) Содержимое данных во внешних записях данных, описанных в программе, не изменяется при освобождении этой программы.

(10) Во время выполнения явного или неявного оператора CANCEL (ОСВОБОДИТЬ) выполняется неявный оператор CLOSE (ЗАКРЫТЬ) без всяких необязательных фраз для каждого файла в открытом состоянии, который связан с внутренним определителем файла в программе, названной в явном операторе CANCEL (ОСВОБОДИТЬ). Все процедуры USE (ИСПОЛЬЗОВАТЬ), относящиеся к любому из этих файлов, не выполняются.

5.4. Оператор EXIT PROGRAM (ВЫЙТИ ИЗ ПРОГРАММЫ)

5.4.1. Назначение

Оператор EXIT PROGRAM (ВЫЙТИ ИЗ ПРОГРАММЫ) отмечает логический конец вызываемой программы.

5.4.2. Общий формат

EXIT PROGRAM
ВЫЙТИ ИЗ ПРОГРАММЫ

5.4.3. Синтаксические правила

(1) Если оператор EXIT PROGRAM (ВЫЙТИ ИЗ ПРОГРАММЫ) появляется в последовательности повелительных операторов внутри предложения, он должен быть последним оператором в этой последовательности.

GOST EXPERT ГОСТ 22558-89,

(2) Оператор EXIT PROGRAM (ВЫЙТИ ИЗ ПРОГРАММЫ) не должен появляться в декларативной процедуре, в которой указана фраза GLOBAL (ГЛОБАЛЬНО).

5.4.4. Общие правила

(1) Если оператор EXIT PROGRAM (ВЫЙТИ ИЗ ПРОГРАММЫ) выполняется в программе, которая не находится под управлением вызывающей программы, оператор EXIT PROGRAM (ВЫЙТИ ИЗ ПРОГРАММЫ) вызывает продолжение выполнения программы со следующего выполнимого оператора.

(2) Выполнение оператора EXIT PROGRAM (ВЫЙТИ ИЗ ПРОГРАММЫ) в вызываемой программе, которая не имеет атрибута «начальная», вызывает продолжение выполнения со следующего выполнимого оператора после оператора CALL (ВЫЗВАТЬ) в вызывающей программе. Состояние вызывающей программы не изменяется и идентично ее состоянию во время выполнения оператора CALL (ВЫЗВАТЬ). Исключение составляет только содержимое данных и файлов данных, совместно используемых вызываемой и вызывающей программами, которое могло быть изменено. Состояние вызываемой программы не изменяется, только считается, что достигнут конец области действия для всех операторов PERFORM (ВЫПОЛНИТЬ), выполняемых этой вызванной программой.

(3) За исключением действий, перечисленных в общем правиле (2), выполнение оператора EXIT PROGRAM (ВЫЙТИ ИЗ ПРОГРАММЫ) в вызываемой программе с атрибутом «начальная» эквивалентно также выполнению оператора CANCEL (ОСВОБОДИТЬ), обращающегося к этой программе (см. п. 5.3 настоящей части).

5.5. **Оператор USE (ИСПОЛЬЗОВАТЬ)**

5.5.1. Назначение

В модуле межпрограммных связей оператор USE (ИСПОЛЬЗОВАТЬ) определяет, вызываются ли соответствующие декларативные процедуры во время выполнения любой программы, содержащейся в программе, в которой находится оператор USE (ИСПОЛЬЗОВАТЬ).

5.5.2. Общий формат

```
USE [GLOBAL] AFTER STANDARD { EXCEPTION | ERROR } PROCEDURE
    ON { {имя-файла-1} ... | INPUT | OUTPUT | I-O | EXTEND }
```


GOST EXPERT ГОСТ 22558-89,

ИСПОЛЬЗОВАТЬ [ГЛОБАЛЬНО] ПОСЛЕ СТАНДАРТНОЙ ПРОЦЕДУРЫ ОШИБКИ

ДЛЯ { {имя-файла-1} ... | ВХОДНЫХ | ВЫХОДНЫХ | ВХОДНЫХ-ВЫХОДНЫХ | ДОПОЛНЯЕМЫХ }

5.5.3. Синтаксические правила

(1) Наличие нескольких имен-файлов и фразы EXTEND (ДОПОЛНЯЕМЫХ) зависит от уровня модуля последовательного ввода-вывода, относительного ввода-вывода или индексного ввода-вывода, обеспечиваемого реализацией (см. ч. 7, п. 5.6.4, ч. 8, п. 4.8, ч. 9, п. 4.8).

5.5.4. Общие правила

(1) При вложении программы в другие программы соблюдаются специальные правила предшествования. При применении этих правил только первая уточняющая декларатива будет выбрана для выполнения. Декларатива, выбираемая для выполнения, должна удовлетворять правилам выполнения этой декларативы. Порядок предшествования для выбора декларативы таков:

а) декларатива внутри программы, которая содержит оператор, вызвавший уточняющее условие;

б) декларатива, в которой задана фраза GLOBAL (ГЛОБАЛЬНО) и которая находится в программе, прямо содержащей программу, которая была проверена последней на уточняющую декларативу;

в) любая декларатива, выбранная в результате применения правила 1б для каждой более объемлющей содержащей программы до тех пор, пока правило 1б не будет применено к наиболее объемлющей программе. Если уточняющая декларатива не найдена, ничего не выполняется.

5.6. Оператор USE BEFORE REPORTING (ИСПОЛЬЗОВАТЬ ДО ВЫДАЧИ)

5.6.1. Назначение

В модуле межпрограммных связей оператор USE BEFORE REPORTING (ИСПОЛЬЗОВАТЬ ДО ВЫДАЧИ) устанавливает, вызываются ли соответствующие декларативные процедуры во время выполнения любой программы, содержащейся в программе, в которой находится оператор USE BEFORE REPORTING (ИСПОЛЬЗОВАТЬ ДО ВЫДАЧИ).

GOST EXPERT ГОСТ 22558-89,

5.6.2. Общий формат

USE [GLOBAL] BEFORE REPORTING идентификатор-1

ИСПОЛЬЗОВАТЬ [ГЛОБАЛЬНО] ДО ВЫДАЧИ идентификатор-1

5.6.3. Синтаксические правила

(1) Наличие оператора USE BEFORE REPORTING (ИСПОЛЬЗОВАТЬ ДО ВЫДАЧИ) зависит от того, обеспечивает ли данная реализация модуль генератора отчетов (ч. 13, п. 4.8.2).

5.6.4. Общие правила

(1) При вложении одних программ в другие программы соблюдаются специальные правила предшествования. При применении этих правил только первая декларатива будет выбрана для выполнения. Декларатива, выбираемая для выполнения, должна удовлетворять правилам выполнения этой декларативы. Порядок предшествования для выбора декларативы таков:

а) декларатива внутри программы, в которой находится оператор, вызвавший уточняющее условие;

б) декларатива, в которой задана фраза GLOBAL (ГЛОБАЛЬНО) и которая находится внутри программы, прямо содержащей программу, которая была проверена последней на уточняющую декларативу;

в) любая декларатива, выбранная путем применения правила 1б для каждой более объемлющей содержащей программы до тех пор, пока правило 1б не будет применено к самой объемлющей программе. Если уточняющая декларатива не найдена, ничего не выполняется.

Часть 11. МОДУЛЬ СОРТИРОВКИ-СЛИЯНИЯ

1. ВВЕДЕНИЕ В МОДУЛЬ СОРТИРОВКИ-СЛИЯНИЯ

1.1. Назначение

Модуль сортировки-слияния обеспечивает возможности упорядочения записей одного или более файлов или комбинирования записей двух или более одинаково упорядоченных файлов в соответствии с набором определенных пользователем ключей, содержащихся в каждой записи. При желании пользователь может применить некоторую специальную обработку для каждой отдельной записи, используя процедуры ввода или вывода. Такая специальная обработка может быть применена до и (или) после того, как записи упорядочены оператором SORT (СОРТИРОВАТЬ), или после того, как записи были объединены оператором MERGE (СЛИТЬ).

1.2. Понятия языка

1.2.1. Сортируемый файл

Сортируемый файл — это совокупность записей, которые долж-

GOST EXPERT ГОСТ 22558-89,

ны быть упорядочены оператором SORT (СОРТИРОВАТЬ). Сортируемый файл не имеет меток, которыми может управлять программист, и правила блокирования и распределения внутренней памяти являются внутренней функцией оператора SORT (СОРТИРОВАТЬ).

Операторы RELEASE (ПЕРЕДАТЬ) и RETURN (ВЕРНУТЬ) не определяют буферных областей, блокирования, размещения на катушках. Таким образом, сортируемый файл представляется внутренним файлом, который создается (с помощью оператора RELEASE (ПЕРЕДАТЬ)) из входного файла, обрабатывается (с помощью оператора SORT (СОРТИРОВАТЬ)) и выводится (с помощью оператора RETURN (ВЕРНУТЬ)) в выходной файл.

Сортируемый файл называется в статье управления файлом и описывается в статье описания сортируемого-сливаемого файла. На сортируемый файл ссылаются в операторх RELEASE (ПЕРЕДАТЬ), RETURN (ВЕРНУТЬ) и SORT (СОРТИРОВАТЬ).

1.2.2. Сливаемый файл

Сливаемый файл — это совокупность записей, предназначенных для слияния с помощью оператора MERGE (СЛИТЬ). Программист не может управлять метками сливаемого файла; правила блокирования и распределения внутренней памяти являются внутренней функцией оператора MERGE (СЛИТЬ). Оператор RETURN (ВЕРНУТЬ) не определяет буферных областей, блокирования, размещения на катушках. Таким образом, сливаемый файл представляется внутренним файлом, который создается из входных файлов посредством их слияния (с помощью оператора MERGE (СЛИТЬ)) и выводится (оператором RETURN (ВЕРНУТЬ)) в выходной файл.

Сливаемый файл называется в статье управления файлом и описывается в статье описания сортируемого-сливаемого файла. На сливаемый файл ссылаются в операторах RETURN (ВЕРНУТЬ) и MERGE (СЛИТЬ).

2. РАЗДЕЛ ОБОРУДОВАНИЯ В МОДУЛЕ СОРТИРОВКИ-СЛИЯНИЯ

2.1. **Секция ввода-вывода**

Информация, относящаяся к секции ввода-вывода, содержится в ч. 7, п. 2.1.

2.2. **Параграф FILE-CONTROL (УПРАВЛЕНИЕ-ФАЙЛАМИ)**

Информация, относящаяся к параграфу FILE-CONTROL (УПРАВЛЕНИЕ-ФАЙЛАМИ), содержится в ч. 7, п. 2.2.

2.3. **Статья управления файлом**

2.3.1. Назначение

Статья управления файлом объявляет существенные свойства файла сортировки или файла слияния.

GOST EXPERT ГОСТ 22558-89,

2.3.2. Общий формат

SELECT имя-файла-1 ASSIGN TO {имя-реализации-1 / литерал-1} ... .

ДЛЯ имя-файла-1 НАЗНАЧИТЬ {имя-реализации-1 / литерал-1} ... .

2.3.3. Синтаксические правила

(1) Каждый сортируемый или сливаемый файл, описанный в разделе данных, должен быть назван в одной статье управления файлом. Каждый сортируемый или сливаемый файл, описанный во фразе SELECT (ДЛЯ), должен иметь статью описания сортируемого или сливаемого файла в разделе данных.

(2) Если имя-файла-1 представляет сортируемый или сливаемый файл, то разрешается только фраза ASSIGN (НАЗНАЧИТЬ), которая должна следовать за именем-файла-1 в параграфе FILE-CONTROL (УПРАВЛЕНИЕ-ФАЙЛАМИ).

2.3.4. Общие правила

(1) Фраза ASSIGN (НАЗНАЧИТЬ) указывает связь файла, представленного именем-файла-1, с носителем данных, представленным именем-реализации-1 или литералом-1.

2.4. Параграф I-O-CONTROL (УПРАВЛЕНИЕ-ВВОДОМ-ВЫВОДОМ)

2.4.1. Назначение

Параграф I-O-CONTROL (УПРАВЛЕНИЕ-ВВОДОМ-ВЫВОДОМ) указывает общие области памяти, которые используются различными файлами, включая сортируемые и сливаемые файлы.

2.4.2. Общий формат

I-O-CONTROL.

[[SAME {RECORD / SORT / SORT-MERGE} AREA FOR имя-файла-1

{имя-файла-2} ...] ... .]

УПРАВЛЕНИЕ-ВВОДОМ-ВЫВОДОМ.

[[ОБЩАЯ ОБЛАСТЬ {ЗАПИСИ / СОРТИРОВКИ / СОРТИРОВКИ-СЛИЯНИЯ}

ДЛЯ имя-файла-1 {имя-файла-2} ...] ... .]

2.4.3. Синтаксическое правило

(1) Допустимость варианта RECORD (ЗАПИСИ) во фразе SAME (ОБЩАЯ ОБЛАСТЬ) зависит от уровня реализации модуля последовательного ввода-вывода.

GOST EXPERT ГОСТ 22558-89,

2.4.4. Общее правило

Фраза SAME RECORD/SORT/SORT-MERGE AREA (ОБЩАЯ ОБЛАСТЬ ЗАПИСИ/СОРТИРОВКИ/СОРТИРОВКИ-СЛИЯНИЯ) для модуля сортировки-слияния описана в п. 2.5.

2.5. Фразы SAME RECORD AREA (ОБЩАЯ ОБЛАСТЬ ЗАПИСИ) и SAME SORT/SORT-MERGE AREA (ОБЩАЯ ОБЛАСТЬ СОРТИРОВКИ/СОРТИРОВКИ-СЛИЯНИЯ)

2.5.1. Назначение

Фразы SAME RECORD (ОБЩАЯ ОБЛАСТЬ ЗАПИСИ) и SAME SORT/SORT-MERGE (ОБЩАЯ ОБЛАСТЬ СОРТИРОВКИ/СОРТИРОВКИ-СЛИЯНИЯ) определяют область памяти, которую одновременно используют разные файлы, среди которых имеется хотя бы один сортируемый или сливаемый файл.

2.5.2. Общий формат

SAME { RECORD / SORT / SORT-MERGE } AREA FOR имя-файла-1 {имя-файла-2}...

ОБЩАЯ ОБЛАСТЬ { ЗАПИСИ / СОРТИРОВКИ / СОРТИРОВКИ-СЛИЯНИЯ } ДЛЯ имя-файла-1 {имя-файла-2}...

2.5.3. Синтаксические правила

(1) Каждое имя-файла, указанное во фразе SAME RECORD (ОБЩАЯ ОБЛАСТЬ ЗАПИСИ) или SAME SORT/SORT-MERGE (ОБЩАЯ ОБЛАСТЬ СОРТИРОВКИ/СОРТИРОВКИ-СЛИЯНИЯ), должно быть указано в параграфе FILE-CONTROL (УПРАВЛЕНИЕ-ФАЙЛАМИ) той же самой программы.

(2) Имя-файла-1 и имя-файла-2 не могут ссылаться на определитель внешнего файла.

(3) Варианты SORT (СОРТИРОВКИ) и SORT-MERGE (СОРТИРОВКИ-СЛИЯНИЯ) эквивалентны.

(4) Имя сортируемого или сливаемого файла не должно быть указано во фразе SAME (ОБЩАЯ) без вариантов RECORD (ЗАПИСИ), SORT (СОРТИРОВКИ) или SORT-MERGE (СОРТИРОВКИ-СЛИЯНИЯ).

(5) Фраза SAME (ОБЩАЯ) может быть включена в программу (в любом из трех возможных вариантов) более одного раза, однако:

а) имя-файла не должно появляться более чем в одной фразе SAME RECORD AREA (ОБЩАЯ ОБЛАСТЬ ЗАПИСИ);

б) имя-файла, указывающее сортируемый или сливаемый файл, не может появляться более чем в одной фразе SAME SORT AREA.

GOST EXPERT ГОСТ 22558-89,

(ОБЩАЯ ОБЛАСТЬ СОРТИРОВКИ) или SAME SORT-MERGE AREA (ОБЩАЯ ОБЛАСТЬ СОРТИРОВКИ-СЛИЯНИЯ);

в) если имя-файла, не относящееся к сортируемому или сливаемому файлу, появляется во фразе SAME (ОБЩАЯ ОБЛАСТЬ) и в одной или более фразах SAME SORT AREA (ОБЩАЯ ОБЛАСТЬ СОРТИРОВКИ) или SAME SORT-MERGE AREA (ОБЩАЯ ОБЛАСТЬ СОРТИРОВКИ-СЛИЯНИЯ) (см. ч. 7, п. 2.13), то все файлы, указанные в этой фразе SAME (ОБЩАЯ ОБЛАСТЬ) должны быть указаны в той же фразе SAME SORT AREA (ОБЩАЯ ОБЛАСТЬ СОРТИРОВКИ) или SAME SORT-MERGE AREA (ОБЩАЯ ОБЛАСТЬ СОРТИРОВКИ-СЛИЯНИЯ).

(6) Файлы, указанные во фразах SAME SORT AREA (ОБЩАЯ ОБЛАСТЬ СОРТИРОВКИ), SAME SORT-MERGE AREA (ОБЩАЯ ОБЛАСТЬ СОРТИРОВКИ-СЛИЯНИЯ) или SAME RECORD AREA (ОБЩАЯ ОБЛАСТЬ ЗАПИСИ), могут иметь различную организацию или доступ.

2.5.4. Общие правила

(1) Фраза SAME RECORD AREA (ОБЩАЯ ОБЛАСТЬ ЗАПИСИ) указывает, что два или более файлов, представленных именем-файла-1 и именем-файла-2, должны использовать общую область памяти для обработки текущей логической записи. Все файлы могут быть открыты одновременно. Логическая запись в общей области записи рассматривается как логическая запись каждого открытого выходного файла, имя которого встречается во фразе SAME RECORD AREA (ОБЩАЯ ОБЛАСТЬ ЗАПИСИ), и как логическая запись читавшегося последним входного файла, имя которого встречается в этой же фразе SAME RECORD AREA (ОБЩАЯ ОБЛАСТЬ ЗАПИСИ). Это равносильно неявному переопределению области, то есть записи располагаются с самой левой позиции литеры.

(2) Если используется фраза SAME SORT AREA (ОБЩАЯ ОБЛАСТЬ СОРТИРОВКИ) или SAME SORT-MERGE AREA (ОБЩАЯ ОБЛАСТЬ СОРТИРОВКИ-СЛИЯНИЯ), то хотя бы одно из имен-файлов должно указывать сортируемый или сливаемый файл. Эта фраза указывает, что память используется следующим образом:

а) любая из этих фраз указывает область памяти, которая будет доступна для использования при сортировке или слиянии каждого указанного сортируемого или сливаемого файла. Такая область памяти, резервируемая для слияния или сортировки одного файла, доступна для повторного использования в сортировке или слиянии и для других сортируемых или сливаемых файлов;

б) области памяти, назначенные файлам, не являющимся сортируемыми или сливаемыми файлами, могут быть назначены, при необходимости, для сортируемых или сливаемых файлов, указан-

GOST EXPERT ГОСТ 22558-89

ных в какой-либо из этих фраз. Особенности такого назначения должны быть указаны реализацией;

в) файлы, не являющиеся сортируемыми или сливаемыми, не используют одну и ту же область памяти. Если пользователь хочет, чтобы эти файлы использовали общую область памяти, он должен включить в программу фразу SAME (ОБЩАЯ ОБЛАСТЬ) или SAME RECORD (ОБЩАЯ ОБЛАСТЬ ЗАПИСИ), называющую эти файлы;

г) во время выполнения операторов SORT (СОРТИРОВАТЬ) или MERGE (СЛИТЬ), ссылающихся на файл, указанный во фразах SAME SORT AREA (ОБЩАЯ ОБЛАСТЬ СОРТИРОВКИ) или SAME SORT-MERGE AREA (ОБЩАЯ ОБЛАСТЬ СОРТИРОВКИ-СЛИЯНИЯ), не должен быть открыт никакой файл, не являющийся сортируемым или сливаемым файлом, указанный в этой фразе.

3. РАЗДЕЛ ДАННЫХ В МОДУЛЕ СОРТИРОВКИ-СЛИЯНИЯ

3.1. Секция файлов

Секция файлов расположена в разделе данных исходной программы. Секция файлов определяет структуру сортируемых и сливаемых файлов. Каждый сортируемый или сливаемый файл описывается статьей описания сортируемого-сливаемого файла и одной или более статьями описания записи. Статьи описания записи размещаются непосредственно после статьи описания сортируемого-сливаемого файла.

Общий формат секции файлов в модуле сортировки-слияния приведен ниже.

FILE SECTION.

[статья-описания-сортируемого-сливаемого-файла
{статья-описания-записи}...]...

СЕКЦИЯ ФАЙЛОВ.

[статья-описания-сортируемого-сливаемого-файла
{статья-описания-записи}...]...

3.1.1. Статья описания сортируемого-сливаемого файла

В Кобол-программе статья описания сортируемого-сливаемого файла (статья SD (ОС)) является высшим уровнем организации в секции файлов. После заголовка секции файлов следует статья описания сортируемого-сливаемого файла, состоящая из индикатора уровня SD (ОС), имени-файла и последовательности независимых фраз. Фразы статьи описания сортируемого-сливаемого файла (статьи SD (ОС)) определяют размер и имена записей данных, относящихся к сортируемому или сливаемому файлу.

Для таких файлов не предусмотрены управляемые пользователем процедуры меток, а правила объединения записей в блоки и

GOST EXPERT ГОСТ 22558-89,

выделения внутренней памяти являются внутренней функцией операторов SORT (СОРТИРОВАТЬ) и MERGE (СЛИТЬ).

Статья описания сортируемого-сливаемого файла оканчивается точкой.

3.1.2. Структура описания записи

Описание записи состоит из ряда статей описания данных, описывающих характеристики отдельной записи. Каждая статья описания данного состоит из номера-уровня, за которым следует имя-данного или фраза FILLER (ЗАПОЛНИТЕЛЬ), если указаны, далее может быть указана последовательность независимых фраз. Описание записи может иметь иерархическую структуру, поэтому используемые в статье фразы могут существенно отличаться друг от друга в зависимости от того, следуют ли за ней подчиненные статьи.

Структура описания записи и допустимых в статье описания записи элементов приводится в ч. 4, п. 4.3.2 и ч. 6, п. 5.3. Допустимые в статье описания данных фразы соответствуют уровню модуля ядра, поддерживаемого реализацией.

3.1.3. Начальные значения

Начальные значения данных в секции файлов не определены.

3.2. **Статья описания сортируемого-сливаемого файла**

3.2.1. Назначение

Описание сортируемого или сливаемого файла дает информацию, касающуюся физической структуры и идентификации записей файла, подлежащего сортировке или слиянию.

3.2.2. Общий формат

SD имя-файла-1

```
[ RECORD { CONTAINS целое-1 CHARACTERS
         { IS VARYING IN SIZE [ [FROM целое-2] [TO
             целое-3] CHARACTERS]
               [DEPENDING ON имя-данного-1]
         { CONTAINS целое-4 TO целое-5
                          CHARACTERS                 } ]

[ DATA { RECORD IS  } {имя-данного-2} ... ].
       { RECORDS ARE }
```

OC имя-файла-1

```
[В ЗАПИСИ { целое-1 ЛИТЕР
          { ПЕРЕМЕННОЕ ЧИСЛО [[ОТ целое-2]
                [ДО целое-3] ЛИТЕР]
          { [В ЗАВИСИМОСТИ ОТ имя-данного-1]
          { ОТ целое-4 ДО целое-5 ЛИТЕР          } ]
```


GOST EXPERT ГОСТ 22558-89,

[{ЗАПИСЬ / ЗАПИСИ} ДАННЫХ {имя-данного-2}...].

3.2.3. Синтаксические правила

(1) Индикатор уровня SD (ОС) указывает начало статьи описания сортируемого или сливаемого файла и должен предшествовать имени-файла.

(2) Фразы, следующие за именем-файла-1, необязательны, и порядок их следования не существенен.

(3) Одна или более статей описания записи должны следовать за статьей описания сортируемого или сливаемого файла, однако, никакой оператор ввода-вывода не может быть выполнен для этого файла.

(4) Возможность использования формата 2 фразы RECORD (В ЗАПИСИ) зависит от уровня модуля последовательного ввода-вывода, поддерживаемого реализацией.

3.2.4. Общие правила

(1) Фраза DATA RECORDS (ЗАПИСИ ДАННЫХ) модуля сортировки-слияния аналогична фразе DATA RECORDS (ЗАПИСИ ДАННЫХ) модуля последовательного ввода-вывода. Поэтому правила для фразы DATA RECORDS (ЗАПИСИ ДАННЫХ) см. в ч. 7, п. 3.5. Фраза DATA RECORDS (ЗАПИСИ ДАННЫХ) рассматривается в настоящем стандарте как устаревший элемент и будет удалена в следующей редакции стандарта.

(2) Фраза RECORD (В ЗАПИСИ) модуля сортировки-слияния аналогична фразе RECORD (В ЗАПИСИ) модуля последовательного ввода-вывода. Поэтому правила для фразы RECORD (В ЗАПИСИ) см. ч. 7, п.3.8.

4. РАЗДЕЛ ПРОЦЕДУР В МОДУЛЕ СОРТИРОВКИ-СЛИЯНИЯ

4.1. Оператор MERGE (СЛИТЬ)

4.1.1. Назначение

Оператор MERGE (СЛИТЬ) комбинирует два или более файлов, одинаково упорядоченных по указанному набору ключей, и во время этого процесса делает записи доступными в порядке слияния процедуре вывода или выходному файлу.

4.1.2. Общий формат

MERGE имя-файла-1 {ON {ASCENDING / DESCENDING} KEY {имя-данного-1}...}...
[COLLATING SEQUENCE IS имя-алфавита-1]
USING имя-файла-2 {имя-файла-3}...

GOST EXPERT ГОСТ 22558-89,

{ OUTPUT PROCEDURE IS имя-процедуры-1
  [{ THROUGH / THRU } имя-процедуры-2]
  GIVING {имя-файла-4} ... }

СЛИТЬ имя-файла-1

{ ПО { ВОЗРАСТАНИЮ / УБЫВАНИЮ } КЛЮЧА {имя-данного-1} ... } ...

[АЛФАВИТ имя-алфавита-1]

ИСПОЛЬЗУЯ имя-файла-2 {имя-файла-3} ...

{ ПРОЦЕДУРА ВЫВОДА имя-процедуры-1
  [ПО имя-процедуры-2]
  ПОЛУЧАЯ {имя-файла-4} ... }

4.1.3. Синтаксические правила

(1) Оператор MERGE (СЛИТЬ) может указываться в любом месте раздела процедур, кроме декларатив.

(2) Имя-файла-1 должно быть описано в статье описания сортируемого-сливаемого файла в разделе данных.

(3) Если файл, представленный именем-файла-1, содержит записи переменной длины, размер записей, содержащихся в файлах, представленных именем-файла-2 и именем-файла-3, должен быть не меньше самой короткой и не больше самой длинной из записей, описанных для имени-файла-1. Если файл, представленный именем-файла-1, содержит записи фиксированной длины, размер записей, содержащихся в файлах, представленных именем-файла-2 и именем-файла-3, должен быть не больше, чем самая длинная запись, описанная для имени-файла-1.

(4) Имя-данного-1 является именем ключа и подчиняется следующим правилам:

а) имена ключей должны быть описаны в записях, связанных с именем-файла-1;

б) имена ключей могут уточняться;

в) имена ключей не должны быть групповыми данными, содержащими переменно повторяющиеся данные;

г) если имя-файла-1 имеет более одного описания записи, то имена-данных, указанных как имена ключей, должны быть описаны только в одном из описаний записей. Одни и те же позиции литер, определяемые именами ключей в одной статье описания записи, считаются ключами всех записей этого файла;

д) описания ключей не должны содержать фразу OCCURS (ПОВТОРЯЕТСЯ) или быть подчиненными статьям, имеющим фразу OCCURS (ПОВТОРЯЕТСЯ);

GOST EXPERT ГОСТ 22558-89,

е) если файл, представленный именем-файла-1, содержит записи переменной длины, все имена ключей должны содержаться в первых х позициях литер записи, где х равно минимальному размеру записи, определенному для файла, представленного именем-файла-1.

(5) В разделе данных статьи описания имени-файла-2, имени-файла-3, имени-файла-4 не должны иметь индикатор уровня SD (ОС).

(6) В операторе MERGE (СЛИТЬ) не может быть указано несколько файлов, размещенных на одной катушке.

(7) В пределах одного оператора MERGE (СЛИТЬ) никакое из имен-файлов не может указываться несколько раз.

(8) Никакие два имени-файла в операторе MERGE (СЛИТЬ) не могут быть указаны в одной и той же фразе SAME AREA (ОБЩАЯ ОБЛАСТЬ), SAME SORT AREA (ОБЩАЯ ОБЛАСТЬ СОРТИРОВКИ), SAME SORT-MERGE AREA (ОБЩАЯ ОБЛАСТЬ СОРТИРОВКИ-СЛИЯНИЯ). Исключение составляют только имена-файлов, относящиеся к фразе GIVING (ПОЛУЧАЯ) (см. ч. 7, пп. 2.13 и 2.5 настоящей части).

(9) Слова THRU и THROUGH являются синонимами.

(10) Если имя-файла-4 определяет индексный файл, первое определение имени-данного-1 должно быть связано с фразой ASCENDING (ПО ВОЗРАСТАНИЮ), а данное, представленное именем-данного-1, должно занимать те же позиции литер в записи, что и данное, соответствующее основному ключу записи файла.

(11) Если указана фраза GIVING (ПОЛУЧАЯ), а файл, представленный именем-файла-4, содержит записи переменной длины, размер записей, содержащихся в файле, представленном именем-файла-1, должен быть не меньше, чем самая короткая, и не больше, чем самая длинная запись, описанная для имени-файла-4. Если файл, представленный именем-файла-4, содержит записи фиксированной длины, размер записей, содержащихся в файле, представленном именем-файла-1, должен быть не больше, чем самая длинная запись, описанная для имени-файла-4.

4.1.4. Общие правила

(1) Оператор MERGE (СЛИТЬ) объединяет в один файл все записи, содержащиеся в файлах, указанных именем-файла-2 и именем-файла-3.

(2) Если файл, представленный именем-файла-1, содержит только записи фиксированной длины, всякая запись файла, представленного именем-файла-2 или именем-файла-3, содержащая меньше литерных позиций, чем значение фиксированной длины, дополняется пробелами справа, начиная с первой позиции после последней литеры в записи, когда эта запись помещается в файл, представленный именем-файла-1.

GOST EXPERT ГОСТ 22558-89,

(3) Указанные во фразе KEY (ПО ВОЗРАСТАНИЮ/УБЫВАНИЮ КЛЮЧА) имена-данных перечисляются в операторе MERGE (СЛИТЬ) в порядке уменьшения значимости, независимо от того, как они распределены между фразами KEY (ПО ВОЗРАСТАНИЮ/УБЫВАНИЮ КЛЮЧА). Согласно формату, имя-данного-1 — самый главный ключ, имя-данного-2 — следующий по значимости ключ и т. д.

а) Если указана фраза ASCENDING (ПО ВОЗРАСТАНИЮ КЛЮЧА), то сливаемая последовательность будет создаваться от наименьшего значения данных, указанных именами ключей, к наибольшему значению в соответствии с правилами сравнения в условии отношения.

б) Если указана фраза DESCENDING (ПО УБЫВАНИЮ КЛЮЧА), то сливаемая последовательность будет создаваться от наибольшего значения данных, указанных именами ключей, к наименьшему значению в соответствии с правилами сравнения операндов в условии отношения.

(4) Если, в соответствии с правилами сравнения операндов в условиях отношения, значения всех ключей одной записи данных равны соответствующим значениям ключей одной или нескольких других записей данных, порядок поступления этих записей соответствует порядку указания входных файлов в операторе MERGE (СЛИТЬ); при этом все записи, связанные с одним входным файлом, поступают до поступления записей других входных файлов.

(5) Основная последовательность при сравнении нечисловых данных определяется в начале выполнения оператора MERGE (СЛИТЬ) в следующем порядке старшинства:

а) во-первых, основная последовательность, установленная фразой COLLATING SEQUENCE (АЛФАВИТ), если она указана в операторе MERGE (СЛИТЬ);

б) во-вторых, основная последовательность, установленная как программный алфавит.

(6) Если записи файлов, представленных именем-файла-2 и именем-файла-3, не упорядочены в соответствии с фразами ASCENDING (ПО ВОЗРАСТАНИЮ КЛЮЧА) и DESCENDING (ПО УБЫВАНИЮ КЛЮЧА) оператора MERGE (СЛИТЬ), результат оператора слияния не определен.

(7) Все записи файлов, представленных именем-файла-2 и именем-файла-3, переносятся в файл, представленный именем-файла-1. В начале выполнения оператора MERGE (СЛИТЬ) файлы, представленные именем-файла-2 и именем-файла-3, не должны быть открыты. Для каждого из файлов, представленных именем-файла-2 и именем-файла-3, выполнение оператора MERGE (СЛИТЬ) приводит к следующим действиям:

а) начинается обработка файла так, как будто был выполнен оператор OPEN (ОТКРЫТЬ) с фразой INPUT (ВХОДНОЙ). Если

GOST EXPERT ГОСТ 22558-89,

указана процедура вывода, то обработка начинается до передачи управления процедуре вывода;

б) логические записи получаются и передаются операции слияния. Каждая запись получается таким образом, как будто был выполнен оператор READ (ЧИТАТЬ) с фразами NEXT (СЛЕДУЮЩУЮ) и AT END (В КОНЦЕ);

в) обработка файла завершается так, как будто был выполнен оператор CLOSE (ЗАКРЫТЬ) без каких-либо дополнительных фраз. Если указана процедура вывода, это завершение не выполняется до тех пор, пока управление не будет возвращено после выполнения последнего оператора процедуры вывода.

Эти неявные функции выполняются таким образом, что выполняются все соответствующие процедуры USE AFTER EXCEPTION/ERROR (ИСПОЛЬЗОВАТЬ ПОСЛЕ ПРОЦЕДУРЫ ОШИБКИ).

(8) Процедура вывода может состоять из любой процедуры, необходимой для выбора, изменения или копирования записей, поочередно доступных посредством оператора RETURN (ВЕРНУТЬ) в порядке слияния из файла, представленного именем-файла-1.

Процедура включает все операторы, выполняемые в результате передачи управления по операторам CALL (ВЫЗВАТЬ), EXIT (ВЫЙТИ), GO TO (ПЕРЕЙТИ) и PERFORM (ВЫПОЛНИТЬ) в рамках процедуры вывода, а также все операторы декларативных процедур, выполняемых в результате выполнения операторов, находящихся в области действия процедуры вывода. В области действия процедуры вывода не должен выполняться ни один из операторов MERGE (СЛИТЬ), RELEASE (ПЕРЕДАТЬ), SORT (СОРТИРОВАТЬ) (см. ч. 4, п. 4.4).

(9) Если процедура вывода определена, то управление передается ей в процессе выполнения оператора MERGE (СЛИТЬ). Компилятор вставляет механизм возврата в конец последней секции процедуры вывода. Когда управление достигает последнего оператора в процедуре вывода, механизм возврата обеспечивает окончание слияния, а затем передает управление следующему после оператора MERGE (СЛИТЬ) выполнимому оператору. Перед входом в процедуру вывода процедура слияния доходит до точки, в которой она, если потребуется, может выбрать очередную запись в порядке слияния. Операторы RETURN (ВЕРНУТЬ) в процедуре вывода являются запросами на получение следующей записи.

(10) Во время выполнения процедуры вывода нельзя выполнять операторы, ссылающиеся на файлы, представленные именем-файла-2 или именем-файла-3, или область записи, связанную с именем-файла-2 или именем-файла-3. Во время выполнения любой процедуры, определенной оператором USE AFTER EXCEPTION (ИСПОЛЬЗОВАТЬ ПОСЛЕ ПРОЦЕДУРЫ ОШИБКИ),

GOST EXPERT ГОСТ 22558-89,

неявно вызванной при выполнении оператора MERGE (СЛИТЬ), нельзя выполнять операторы, ссылающиеся на файлы, представленные именем-файла-2, именем-файла-3 или именем-файла-4, или область записи, связанную с именем-файла-2, именем-файла-3 или именем-файла-4.

(11) Если указана фраза GIVING (ПОЛУЧАЯ), сливаемые записи записываются в файл, представленный именем-файла-4, неявной процедурой вывода для оператора MERGE (СЛИТЬ). В начале выполнения оператора MERGE (СЛИТЬ) файл, представленный именем-файла-4, не должен быть открыт. Для каждого файла, представленного именем-файла-4, выполнение оператором MERGE (СЛИТЬ) приводит к следующим действиям:

а) начинается обработка файла так, как будто выполнился оператор OPEN (ОТКРЫТЬ) с фразой OUTPUT (ВЫХОДНОЙ);

б) сливаемые логические записи записываются в файл так, как будто выполнился оператор READ (ПИСАТЬ) без каких-либо необязательных фраз.

Для файла с относительной организацией относительный ключ первой пересылаемой записи содержит значение 1; второй пересылаемой записи — значение 2 и т. д. После выполнения оператора MERGE (СЛИТЬ) содержимое данного, указанного именем относительного ключа, указывает на последнюю запись, возвращенную в файл;

в) обработка файла завершается так, как будто выполнился оператор CLOSE (ЗАКРЫТЬ) без каких-либо необязательных фраз.

Эти неявные функции выполняются так, что будут выполняться связанные с ними процедуры USE AFTER EXCEPTION/ERROR (ИСПОЛЬЗОВАТЬ ПОСЛЕ ПРОЦЕДУРЫ ОШИБКИ), однако выполнение таких процедур USE (ИСПОЛЬЗОВАТЬ) не должно приводить к выполнению каких-либо операторов, ссылающихся на файл, представленный именем-файла-4, или область записи, соответствующую имени-файла-4. При первой попытке записи с нарушением внешне определенных границ файла выполняется указанная для файла процедура USE AFTER STANDARD EXCEPTION/ERROR (ИСПОЛЬЗОВАТЬ ПОСЛЕ СТАНДАРТНОЙ ПРОЦЕДУРЫ ОШИБКИ); если управление возвращается из процедуры USE (ИСПОЛЬЗОВАТЬ) или если такая процедура не указана, обработка файла завершается, как указано выше в п. 11в.

(12) Если файл, представленный именем-файла-4, содержит только записи фиксированной длины, любая запись файла, представленного именем-файла-1, имеющая меньше позиций литер, чем значение фиксированной длины, дополняется пробелами слева направо, начиная с первой позиции литеры после последней литеры записи, когда эта запись возвращается в файл, представленный именем-файла-4.

GOST EXPERT ГОСТ 22558-89,

(13) В программах, содержащих оператор MERGE (СЛИТЬ), может применяться сегментация, однако, имеют место следующие ограничения:

а) если оператор MERGE (СЛИТЬ) указан в секции, которая не является независимым сегментом, то любая процедура вывода, указанная оператором MERGE (СЛИТЬ), должна быть указана либо полностью вне независимых сегментов, либо целиком содержаться в одном независимом сегменте;

б) если оператор MERGE (СЛИТЬ) указан в независимом сегменте, то любая процедура вывода, на которую ссылается этот оператор, должна содержаться либо полностью вне независимых сегментов, либо целиком в пределах того же независимого сегмента, что и оператор MERGE (СЛИТЬ).

4.2. **Оператор RELEASE (ПЕРЕДАТЬ)**

4.2.1. Назначение

Оператор RELEASE (ПЕРЕДАТЬ) передает записи в начальную фазу операции сортировки.

4.2.2. Общий формат

RELEASE имя-записи-1 [FROM идентификатор-1]

ПЕРЕДАТЬ имя-записи-1 [ИЗ ПОЛЯ идентификатор-1]

4.2.3. Синтаксические правила

(1) Имя-записи-1 должно быть именем логической записи в соответствующей статье описания сортируемого или сливаемого файла и может уточняться.

(2) Оператор RELEASE (ПЕРЕДАТЬ) может использоваться только в пределах процедуры ввода, связанной с оператором SORT (СОРТИРОВАТЬ), для сортируемого или сливаемого файла, статья описания файла которого содержит это имя-записи-1.

(3) Имя-записи-1 и идентификатор-1 не должны ссылаться на одну и ту же область памяти.

4.2.4. Общие правила

(1) При выполнении оператора RELEASE (ПЕРЕДАТЬ) запись, указанная именем-записи-1, передается в начальную фазу операции сортировки.

(2) Логическая запись, переданная в результате выполнения оператора RELEASE (ПЕРЕДАТЬ), становится недоступной в области записи, если имя сортируемого или сливаемого файла, соответствующее имени-записи-1, не указано во фразе SAME RECORD AREA (ОБЩАЯ ОБЛАСТЬ ЗАПИСИ). Логическая запись доступна программе как запись других файлов, представленных фразой SAME RECORD AREA (ОБЩАЯ ОБЛАСТЬ ЗАПИСИ), соответствующей выходному файлу так, как и запись файла, соответствующего имени-записи-1.

GOST EXPERT ГОСТ 22558-89,

(3) Результат выполнения оператора RELEASE (ПЕРЕДАТЬ) с фразой FROM (ИЗ ПОЛЯ) эквивалентен выполнению следующих операторов в указанном порядке:

а) оператор

MOVE идентификатор-1 TO имя-записи-1

ПОМЕСТИТЬ идентификатор-1 В имя-записи-1 согласно правилам, указанным для оператора MOVE (ПОМЕСТИТЬ);

б) тот же оператор RELEASE (ПЕРЕДАТЬ) без фразы FROM (ИЗ ПОЛЯ).

(4) После завершения выполнения оператора RELEASE (ПЕРЕДАТЬ) информацию в области, представленной идентификатором-1, остается доступной, даже если недоступна информация в области, представленной именем-записи-1, за исключением случаев, определенных фразой SAME RECORD AREA (ОБЩАЯ ОБЛАСТЬ ЗАПИСИ).

4.3. **Оператор RETURN (ВЕРНУТЬ)**

4.3.1. Назначение

Оператор RETURN (ВЕРНУТЬ) получает либо отсортированные записи в конечной фазе операции сортировки, либо объединенные в один файл записи, полученные при выполнении операции слияния.

4.3.2. Общий формат

RETURN имя-файла-1 RECORD [INTO идентификатор-1]

AT END повелительный-оператор-1

[NOT AT END повелительный-оператор-2]

[END-RETURN]

ВЕРНУТЬ ЗАПИСЬ имя-файла-1 [В идентификатор-1]

В КОНЦЕ повелительный-оператор-1

[НЕ В КОНЦЕ повелительный-оператор-2]

[КОНЕЦ-ВЕРНУТЬ]

4.3.3. Синтаксические правила

(1) Область памяти, связанная с идентификатором-1, и область записи, связанная с именем-файла-1, не должны представлять одну и ту же область памяти.

(2) Имя-файла-1 должно быть описано в статье описания сортируемого-сливаемого файла в разделе данных.

(3) Оператор RETURN (ВЕРНУТЬ) может использоваться только в процедуре вывода, связанной с оператором SORT (СОРТИРОВАТЬ) или MERGE (СЛИТЬ) для имени-файла-1.

4.3.4. Общие правила

(1) Если файл состоит из логических записей нескольких типов, то эти записи автоматически разделяют общую область запи-

GOST EXPERT ГОСТ 22558-89,

си в памяти; это равносильно неявному переопределению области. Значения любых данных, которые лежат вне текущей записи, по окончании выполнения оператора RETURN (ВЕРНУТЬ) не определены.

(2) В результате выполнения оператора RETURN (ВЕРНУТЬ) следующая существующая запись файла, представленного именем-файла-1, становится доступной для обработки в области, связанной с именем-файла-1, в порядке, определенном ключами, перечисленными в операторе MERGE (СЛИТЬ) или SORT (СОРТИРОВАТЬ). Если следующей логической записи в файле, представленном именем-файла-1, не существует, возникает условие конца, и управление передается повелительному-оператору-1 фразы AT END (В КОНЦЕ). Выполнение продолжается согласно правилам для операторов, указанных в повелительном-операторе-1. Если выполняется оператор ветвления процедур или условный оператор, вызывающий явную передачу управления, оно передается согласно правилам для этого оператора; в противном случае после завершения выполнения повелительного-оператора-1 управление передается в точку выхода из оператора RETURN (ВЕРНУТЬ), а фраза NOT AT END (НЕ В КОНЦЕ), если указана, она игнорируется. При наступлении условия конца выполнение оператора RETURN (ВЕРНУТЬ) считается неуспешным и содержимое области записи, соответствующей имени-файла-1, не определено. Оператор RETURN (ВЕРНУТЬ) не может быть выполнен как часть текущей процедуры вывода после выполнения повелительного-оператора-1, указанного фразой AT END (В КОНЦЕ).

(3) Если при выполнении оператора RETURN (ВЕРНУТЬ) условие конца не возникает, то после того, как запись стала доступной, и после выполнения всех неявных пересылок, связанных с фразой INTO (В), управление передается повелительному-оператору-2, если он указан; в противном случае управление передается в точку выхода оператора RETURN (ВЕРНУТЬ).

(4) Фраза END-RETURN (КОНЕЦ-ВЕРНУТЬ) ограничивает область действия оператора RETURN (ВЕРНУТЬ) (см. ч. 4, п. 6.4.3).

(5) Фраза INTO (В) может быть указана в операторе RETURN (ВЕРНУТЬ) в следующих случаях:

а) если в статье описания сортируемого-сливаемого файла имеется только одно описание записи;

б) если все имена-записей, соответствующие имени-файла-1, и данное, представленное идентификатором-1, описывают групповое данное или элементарное буквенно-цифровое данное.

(6) результат выполнения оператора RETURN (ВЕРНУТЬ) с фразой INTO (В) эквивалентен выполнению следующих действий в указанном порядке:

GOST EXPERT ГОСТ 22558-89,

а) выполнению того же оператора RETURN (ВЕРНУТЬ) без фразы INTO (В);

б) текущая запись перемещается из области записи в область, определенную идентификатором-1, согласно правилам для оператора MOVE (ПОМЕСТИТЬ) без фразы CORRESPONDING (СООТВЕТСТВЕННО). Размер текущей записи определяется правилами, указанными для фразы RECORD (В ЗАПИСИ). Если статья описания файла содержит фразу RECORD IS VARYING (В ЗАПИСИ ПЕРЕМЕННОЕ ЧИСЛО ЛИТЕР), пересылка является групповой. Неявный оператор MOVE (ПОМЕСТИТЬ) не выполняется, если выполнение оператора RETURN (ВЕРНУТЬ) было неуспешным. Индексы, относящиеся к идентификатору-1, вычисляются после чтения записи и непосредственно перед ее пересылкой в данное. Запись доступна как в области записи, так и в области данного, представленного идентификатором-1.

4.4. Оператор SORT (СОРТИРОВАТЬ)

4.4.1. Назначение

Оператор SORT (СОРТИРОВАТЬ) создает сортируемый файл, выполняя для этого процедуру ввода или перемещение записей из других файлов, сортирует записи в сортируемом файле по указанному набору ключей и в последней фазе операции сортировки делает доступной каждую запись из сортируемого файла в отсортированном порядке для указанной процедуры вывода или для выходного файла.

4.4.2. Общий формат

SORT имя-файла-1 {ON {ASCENDING / DESCENDING} KEY {имя-данного-1}...}...

[WITH DUPLICATES IN ORDER]

[COLLATING SEQUENCE IS имя-алфавита-1]

{INPUT PROCEDURE IS имя-процедуры-1 [{THROUGH / THRU} имя-процедуры-2]
USING {имя-файла-2}...}

{OUTPUT PROCEDURE IS имя-процедуры-3 [{THROUGH / THRU} имя-процедуры-4
GIVING {имя-файла-3}...}

GOST EXPERT ГОСТ 22558-89,

СОРТИРОВАТЬ имя-файла-1

{ПО { ВОЗРАСТАНИЮ / УБЫВАНИЮ } КЛЮЧА {имя-данного-1} . . .) . . .

[С ДУБЛИРОВАНИЕМ]

[АЛФАВИТ имя-алфавита-1]

{ ПРОЦЕДУРА ВВОДА имя-процедуры-1
  [ПО имя-процедуры-2]
  ИСПОЛЬЗУЯ {имя-файла-2} . . . }

{ ПРОЦЕДУРА ВЫВОДА имя-процедуры-3
  [ПО имя-процедуры-4]
  ПОЛУЧАЯ {имя-файла-3} . . . }

4.4.3. Синтаксические правила

(1) Оператор SORT (СОРТИРОВАТЬ) может указываться в любом месте раздела процедур, за исключением деклараций.

(2) Имя-файла-1 должно быть описано в статье описания сортируемого-сливаемого файла в разделе данных.

(3) Если указана фраза USING (ИСПОЛЬЗУЯ) и файл, представленный именем-файла-1, содержит записи переменной длины, размер записей, содержащихся в файле, представленном именем-файла-2, должен быть не меньше размера самой короткой и не больше размера самой длинной записи, описанной для имени-файла-1. Если файл, представленный именем-файла-1, содержит записи фиксированной длины, размер записей, содержащихся в файле, представленном именем-файла-2, должен быть не больше, чем размер самой длинной записи, описанной для файла, представленного именем-файла-1.

(4) Имя-данного-1 является именем ключа и подчиняется следующим правилам:

а) данные, представленные именами ключей, должны быть описаны в записях, соответствующих имени-файла-1;

б) имена ключей могут уточняться;

в) имена ключей не должны быть групповыми данными, содержащими переменно повторяющиеся данные;

г) если имя-файла-1 имеет более одного описания записи, то данные, указанные именами ключей, могут быть описаны только в одном из описаний записей. Одни и те же позиции литер, определяемые именем-ключа в одной статье описания записи, считаются ключом во всех записях файла;

д) описания ключей не должны содержать фразу OCCURS (ПОВТОРЯЕТСЯ) или быть подчиненными статьям, имеющим фразу OCCURS (ПОВТОРЯЕТСЯ);

GOST EXPERT ГОСТ 22558-89,

е) если файл, представленный именем-файла-1, содержит записи переменной длины, все данные, представленные именами-ключей, должны содержаться в первых x позициях записи, где x равняется минимальному размеру записи, указанному для файла, представленного именем-файла-1.

(5) Слова THRU и THROUGH эквивалентны.

(6) Имя-файла-2 и имя-файла-3 должны быть описаны в разделе данных статьей описания файла, а не статьей описания сортируемого-сливаемого файла.

(7) Файлы, представленные именем-файла-2 и именем-файла-3, могут размещаться на одной и той же катушке.

(8) Если имя-файла-3 относится к индексному файлу, первое указание имени-данного-1 должно быть связано с фразой ASCENDING (ПО ВОЗРАСТАНИЮ), а данное, представленное именем-данного-1, должно занимать те же позиции литер в записи, что и данное, соответствующее основному ключу записи для этого файла.

(9) В одной и той же фразе SAME SORT AREA (ОБЩАЯ ОБЛАСТЬ СОРТИРОВКИ) или SAME SORT-MERGE AREA (ОБЩАЯ ОБЛАСТЬ СОРТИРОВКИ-СЛИЯНИЯ) не могут быть указаны никакие два имени-файла, используемые в одном и том же операторе SORT (СОРТИРОВАТЬ). Имена-файлов, соответствующие фразе GIVING (ПОЛУЧАЯ), не могут указываться в одной и той же фразе SAME (ОБЩАЯ) (см. ч. 7, п. 2.13 и п. 2.5 настоящей части).

(10) Если указана фраза GIVING (ПОЛУЧАЯ) и файл, представленный именем-файла-3, содержит записи переменной длины, размер записей, содержащихся в файле, представленном именем-файла-1, должен быть не меньше размера самой короткой и не больше размера самой длинной записи, описанной для имени-файла-3. Если файл, представленный именем-файла-3, содержит записи фиксированной длины, размер записей, содержащихся в файле, представленном именем-файла-1, должен быть не больше самой длинной записи, описанной для файла, представленного именем-файла-3.

4.4.4. *Общие правила*

(1) Если файл, представленный именем-файла-1, содержит только записи фиксированной длины, любая запись файла, представленного именем-файла-2, содержащая меньше позиций литер, чем запись фиксированной длины, при передаче в файл, представленный именем-файла-1, дополняется пробелами справа, начиная с первой позиции литеры после последней литеры записи.

(2) Имена-данных, указанные в качестве ключей, перечисляются в операторе SORT (СОРТИРОВАТЬ) в порядке убывания значимости. Самое левое имя-данного-1 — самый главный ключ, следующее имя-данного-2 — следующий по значимости ключ и так далее.

GOST EXPERT ГОСТ 22558-89,

а) Если определена фраза ASCENDING (ПО ВОЗРАСТАНИЮ), то отсортированная последовательность будет сформирована, начиная с наименьшего значения данных, указанных в качестве ключей, и кончая наибольшим значением в соответствии с правилами сравнения операндов в условиях отношения.

б) Если указана фраза DESCENDING (ПО УБЫВАНИЮ), то отсортированная последовательность будет сформирована, начиная с наибольшего значения данных, указанных в качестве ключей, и кончая наименьшим значением в соответствии с правилами сравнения операндов.

(3) Если указана фраза DUPLICATES (С ДУБЛИРОВАНИЕМ) и содержимое всех ключей, связанных с одной записью данных, равно содержимому соответствующих ключей, связанных с одной или несколькими другими записями данных, то порядок возврата этих записей следующий:

а) совпадает с порядком указания входных файлов в операторе SORT (СОРТИРОВАТЬ). В заданном входном файле порядок записей совпадает с порядком получения записей из этого файла;

б) если указана процедура ввода, порядок записей совпадает с порядком, в котором эти записи поступают из процедуры ввода.

(4) Если фраза DUPLICATES (С ДУБЛИРОВАНИЕМ) не указана, и содержимое всех ключей, связанных с одной записью данных, равно содержимому соответствующих ключей, связанных с одной или несколькими другими записями данных, то порядок возвращения этих записей не определен.

(5) Основная последовательность при сравнении нечисловых данных, являющихся ключами, определяется в начале выполнения оператора SORT (СОРТИРОВАТЬ) в следующем порядке старшинства:

а) во-первых, основная последовательность, установленная фразой COLLATING SEQUENCE (АЛФАВИТ) в операторе SORT (СОРТИРОВАТЬ), если эта фраза определена;

б) во-вторых, основная последовательность, установленная как программный алфавит.

(6) Выполнение оператора SORT (СОРТИРОВАТЬ) состоит из следующих трех этапов:

а) записи становятся доступными файлу, представленному именем-файла-1, либо благодаря выполнению оператора RELEASE (ПЕРЕДАТЬ) в процедуре ввода, либо благодаря неявному выполнению оператора READ (ЧИТАТЬ) для имени-файла-2. Перед началом этой фазы файл, представленный именем-файла-2, не должен быть открыт. После окончания этой фазы файл, представленный именем-файла-2, не является открытым;

б) файл, представленный именем-файла-1, упорядочивается. На протяжении этого этапа файлы, представленные именем-файла-2 и именем-файла-3, не подвергаются никакой обработке;

GOST EXPERT ГОСТ 22558-89,

в) записи файла, представленного именем-файла-1, становятся доступными в отсортированном виде. Отсортированные записи либо записываются в файл, представленный именем-файла-3, либо становятся доступными для обработки в результате выполнения оператора RETURN (ВЕРНУТЬ) в процедуре вывода. В начале этой фазы файл, представленный именем-файла-3, не должен быть открыт. После окончания это файл, представленный именем-файла-3, не является открытым.

(7) Процедура ввода может состоять из любой процедуры выборки, модификации или копирования записей, которые становятся доступными посредством оператора RELEASE (ПЕРЕДАТЬ) для файла, представленного именем-файла-1. Область действия такой процедуры включает все операторы, выполняющиеся в результате передачи управления по операторам CALL (ВЫЗВАТЬ), EXIT (ВЫЙТИ), GO TO (ПЕРЕЙТИ) и PERFORM (ВЫПОЛНИТЬ), находящимся в области действия процедуры ввода, а также все операторы декларативных процедур, выполняющихся в результате выполнения операторов, находящихся в области действия процедуры ввода. В области действия процедуры ввода не должны выполняться операторы MERGE (СЛИТЬ), RETURN (ВЕРНУТЬ) или SORT (СОРТИРОВАТЬ) (см. ч. 4, п. 4.4).

(8) Если процедура ввода определена, то управление передается ей до того как файл, представленный именем-файла-1, будет упорядочен оператором SORT (СОРТИРОВАТЬ). Компилятор встраивает механизм возврата в конец последнего оператора процедуры ввода. Когда управление достигает последнего оператора в процедуре ввода, то записи, которые были переданы для файла, представленного именем-файла-1, сортируются.

(9) Если указана фраза USING (ИСПОЛЬЗУЯ), все записи файла (файлов), представленных именем-файла-2, передаются файлу, представленному именем-файла-1. Для каждого файла, представленного именем-файла-2, выполнение оператора SORT (СОРТИРОВАТЬ) приводит к следующим действиям:

а) инициируется обработка файла, которая происходит так, как будто выполняется оператор OPEN (ОТКРЫТЬ) с фразой INPUT (ВХОДНОЙ);

б) логические записи извлекаются и передаются операции сортировки, причем каждая запись извлекается так, как если бы выполнился оператор READ (ЧИТАТЬ) с фразами NEXT (СЛЕДУЮЩУЮ) и AT END (В КОНЦЕ). Для файла с относительной организацией, не указанного именем-файла-2 во фразе GIVING (ПОЛУЧАЯ), значение данного, являющегося относительным ключом, после выполнения оператора SORT (СОРТИРОВАТЬ) не определено;

в) обработка файла завершается так, как если бы был выполнен оператор CLOSE (ЗАКРЫТЬ) без каких-либо необязательных

GOST EXPERT ГОСТ 22558-89,

фраз. Это завершение выполняется до сортировки файла, представленного именем-файла-1 в операторе SORT (СОРТИРОВАТЬ).

Эти неявные функции выполняются таким образом, что выполняются и соответствующие процедуры USE AFTER EXCEPTION/ERROR (ИСПОЛЬЗОВАТЬ ПОСЛЕ ОШИБКИ); однако исполнение таких процедур не должно приводить к выполнению каких-либо операторов, обрабатывающих файл, представленный именем-файла-2, либо осуществляющих доступ к области записи, соответствующей имени-файла-2.

(10) Процедура вывода может состоять из любых процедур выборки, модификации или копирования записей, которые поочередно становятся доступными в отсортированном порядке из файла, представленного именем-файла-1 посредством оператора RETURN (ВЕРНУТЬ). Область действия таких процедур включает все операторы, выполняющиеся в результате передачи управления по операторам CALL (ВЫЗВАТЬ), EXIT (ВЫЙТИ), GO TO (ПЕРЕЙТИ) и PERFORM (ВЫПОЛНИТЬ) в области действия процедуры вывода, а также все операторы декларативных процедур, выполняющихся в результате выполнения операторов, находящихся в области действия процедуры вывода. В области действия процедуры вывода не должны находиться операторы MERGE (СЛИТЬ), RELEASE (ПЕРЕДАТЬ), SORT (СОРТИРОВАТЬ) (см. ч. 4, п. 4.4).

(11) Если процедура вывода определена, то управление передается ей после упорядочения файла, представленного именем-файла-1, оператором SORT (СОРТИРОВАТЬ). Компилятор встраивает механизм возврата в конец последнего оператора процедуры вывода, и когда управление достигает последнего оператора процедуры вывода, механизм возврата обеспечивает завершение сортировки, и затем управление передается следующему после оператора SORT (СОРТИРОВАТЬ) выполнимому оператору. Процедура вывода получает управление, когда все записи отсортированы. Операторы RETURN (ВЕРНУТЬ) в процедуре вывода являются запросами на получение следующей записи.

(12) Если указана фраза GIVING (ПОЛУЧАЯ), выполняется неявная процедура вывода, в результате которой все отсортированные записи записываются в файл, представленный именем-файла-3. Для всех файлов, представленных именем-файла-3, выполнение оператора SORT (СОРТИРОВАТЬ) приводит к следующим действиям:

а) обработка файла инициируется так, как будто бы выполнен оператор OPEN (ОТКРЫТЬ) с фразой OUTPUT (ВЫХОДНОЙ). Эта инициация выполняется после выполнения всех процедур ввода;

GOST EXPERT ГОСТ 22558-89,

б) отсортированные логические записи возвращаются и записываются в файл так, как будто выполняется оператор WRITE (ПИСАТЬ) без каких-либо необязательных фраз; для файла с относительной организацией данное, являющееся относительным ключом, для первой возвращенной записи принимает значение 1, для второй — значение 2 и т. д.

После выполнения оператора SORT (СОРТИРОВАТЬ) содержимое данного, являющегося относительным ключом, указывает на последнюю возвращенную в файл запись;

в) обработка файла завершается так, как будто был выполнен оператор CLOSE (ЗАКРЫТЬ) без каких-либо необязательных фраз.

Эти неявные функции выполняются так, что выполняются и соответствующие процедуры USE AFTER EXCEPTION/ERROR (ИСПОЛЬЗОВАТЬ ПОСЛЕ ПРОЦЕДУРЫ ОШИБКИ), однако исполнение таких процедур не должно приводить к выполнению каких-либо операторов, обрабатывающих файл, представленный именем-файла-3, либо осуществляющих доступ к области записи, соответствующей имени-файла-3. При первой попытке записи за пределами внешне определенных границ выполняется процедура USE AFTER STANDARD EXCEPTION/ERROR (ИСПОЛЬЗОВАТЬ ПОСЛЕ СТАНДАРТНОЙ ПРОЦЕДУРЫ ОШИБКИ), указанная для файла; если управление возвращается из такой процедуры или такая процедура не указана, обработка файла завершается, как указано выше в п. 12в.

(13) Если файл, представленный именем-файла-3, содержит только записи фиксированной длины, записи файла, представленного именем-файла-1, содержащие меньше позиций литер, чем запись фиксированной длины, при возвращении записи в файл, представленный именем-файла-3, дополняются пробелами справа, начиная с первой позиции литеры после последней литеры в записи.

(14) В программах, содержащих оператор SORT (СОРТИРОВАТЬ), может применяться сегментация (ч. 16). Однако имеют место следующие ограничения:

а) если оператор SORT (СОРТИРОВАТЬ) указан в секции, которая не является независимым сегментом, то любые процедуры ввода или вывода, указанные оператором SORT (СОРТИРОВАТЬ), должны быть указаны либо вне независимых сегментов, либо целиком содержаться в одном независимом сегменте;

б) если оператор SORT (СОРТИРОВАТЬ) указан в независимом сегменте, то любые процедуры ввода или вывода, на которые ссылается оператор SORT (СОРТИРОВАТЬ), должны содержаться либо полностью вне независимых сегментов, либо целиком в том же независимом сегменте, что и оператор SORT (СОРТИРОВАТЬ).

GOST EXPERT ГОСТ 22558-89,

Часть 12. МОДУЛЬ ОБРАБОТКИ ИСХОДНЫХ ТЕКСТОВ

1. ВВЕДЕНИЕ В МОДУЛЬ ОБРАБОТКИ ИСХОДНЫХ ТЕКСТОВ

1.1. Назначение

Модуль обработки исходных текстов содержит оператор COPY (КОПИРОВАТЬ) | и оператор REPLACE (ЗАМЕНИТЬ) | . Каждый из этих операторов может функционировать независимо от другого или в сочетании с другим для обеспечения возможности вставлять | или заменять | текст исходной программы в процессе компиляции исходной программы.

Тексты, которые становятся доступными компилятору во время компиляции, содержатся в библиотеках Кобола. Эффект интерпретации оператора COPY (КОПИРОВАТЬ) состоит в генерации текста из библиотечного текста, который рассматривается компилятором как часть исходной программы.

Кроме того, исходные программы Кобола могут быть написаны в определяемой программистом нотации, которая может быть превращена во время компиляции в синтаксически правильные фразы и операторы. | Эффект интерпретации оператора REPLACE (ЗАМЕНИТЬ) состоит в замене текста, появляющегося в исходной программе, новым текстом, который рассматривается компилятором как часть исходной программы. |

1.2. Характеристика уровней

Уровень 1 обработки исходных текстов обеспечивает возможность копирования в исходную программу текста из единственной библиотеки. Текст копируется из библиотеки без изменений.

Уровень 2 обработки исходных текстов обеспечивает дополнительную возможность замены в процессе копирования всех появлений указанного литерала, идентификатора, слова или группы слов в библиотечном тексте другим текстом. Уровень 2 обеспечивает также возможности использования во время компиляции нескольких библиотек Кобола и замены текста, появляющегося в исходной программе, новым текстом.

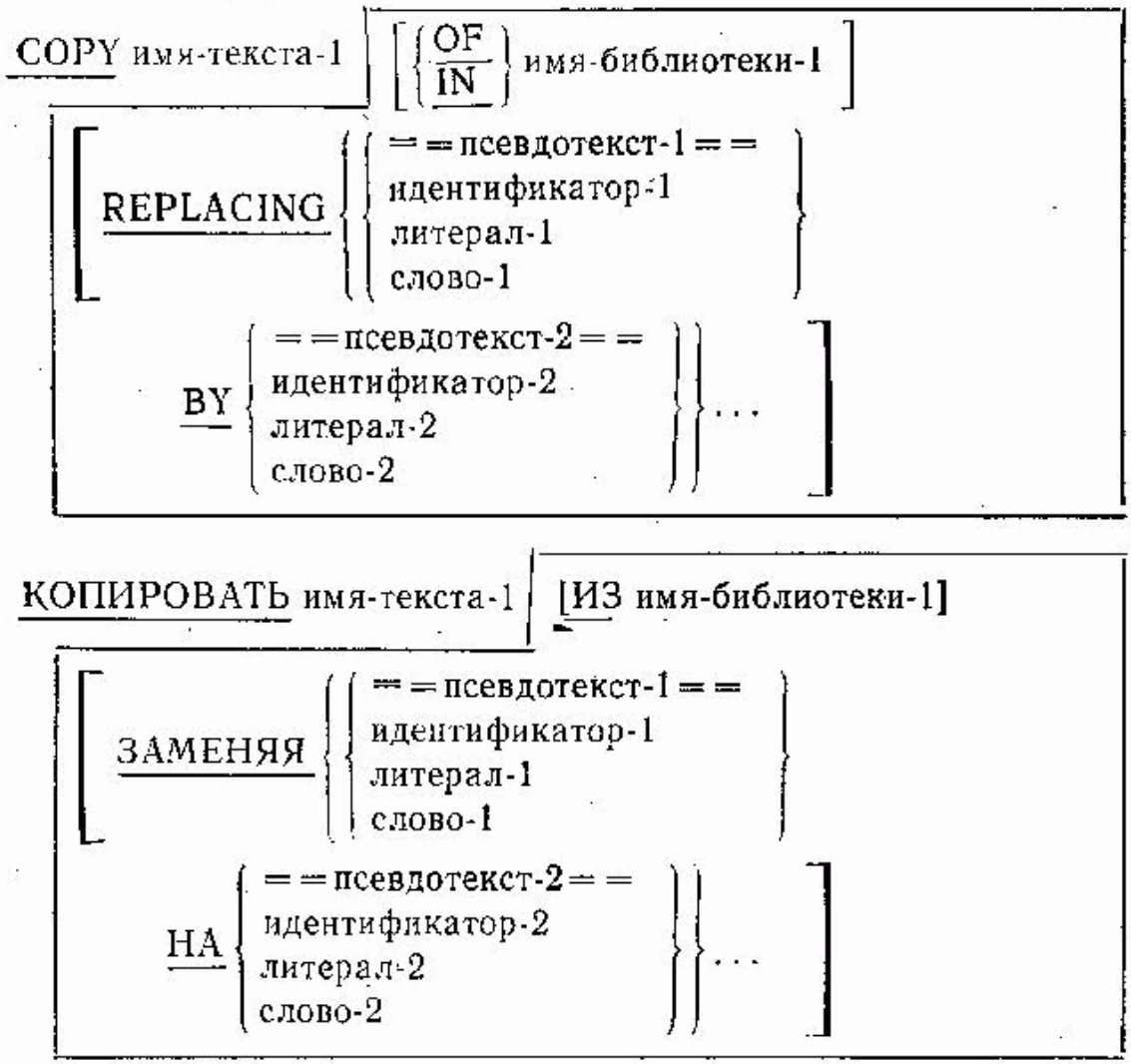
2. ОПЕРАТОР COPY (КОПИРОВАТЬ)

2.1. Назначение

Оператор COPY (КОПИРОВАТЬ) включает текст в исходную программу Кобола.

GOST EXPERT ГОСТ 22558-89,

2.2. Общий формат



2.3. Синтаксические правила

(1) Если во время компиляции доступна более чем одна библиотека Кобола, имя-текста-1 должно уточняться именем-библиотеки-1, идентифицирующим библиотеку Кобола, в которой находится текст, соответствующий имени-текста-1.

В одной библиотеке Кобола каждое имя-текста должно быть уникальным (однозначным).

(2) Оператору COPY (КОПИРОВАТЬ) должен предшествовать пробел. Оператор COPY (КОПИРОВАТЬ) должен заканчиваться разделителем точка.

(3) Псевдотекст-1 должен содержать одно или несколько слов текста.

(4) Псевдотекст-2 может содержать одно или несколько слов текста или не содержать слов текста.

GOST EXPERT ГОСТ 22558-89,

(5) Строка-литер в псевдотексте-1 и псевдотексте-2 может быть продолжена на следующей строке (см. ч. 4, п. 7.2.5).

(6) Слово-1 и слово-2 могут быть любым одиночным словом Кобола, кроме COPY (КОПИРОВАТЬ).

(7) Оператор COPY (КОПИРОВАТЬ) может быть указан в исходной программе всюду, где может появиться строка-литер или разделитель, отличный от закрывающего знака «кавычки», за исключением самого оператора COPY (КОПИРОВАТЬ), внутри которого оператор COPY (КОПИРОВАТЬ) не может появляться.

(8) Реализация должна допускать длину слова текста от 1 до 322 литер в псевдотексте и библиотечном тексте.

(9) Псевдотекст-1 не должен состоять только из разделителя запятая или разделителя точка с запятой.

(10) Если слово COPY (КОПИРОВАТЬ) появляется в статье-комментарии или в том месте, где статья-комментарий может появиться, оно рассматривается как часть статьи-комментария.

2.4. Общие правила

(1) Компиляция исходной программы, содержащей операторы COPY (КОПИРОВАТЬ), логически эквивалентна обработке всех операторов COPY (КОПИРОВАТЬ) до обработки результирующей исходной программы.

(2) Эффект обработки оператора COPY (КОПИРОВАТЬ) состоит в том, что библиотечный текст, связанный с именем-текста-1, копируется в исходную программу, логически заменяя весь оператор COPY (КОПИРОВАТЬ), начиная с зарезервированного слова COPY (КОПИРОВАТЬ) и кончая литерой пунктуации точка включительно.

(3) Если вариант REPLACING (ЗАМЕНЯЯ) не указан, библиотечный текст копируется без изменений.

Если вариант указан, библиотечный текст копируется и каждое сравнившееся вхождение псевдотекста-1, идентификатора-1, слова-1 или литерала-1 в библиотечном тексте заменяется соответствующим псевдотекстом-2, идентификатором-2, словом-2 или литералом-2.

(4) Для целей сравнения идентификатор-1, слово-1 и литерал-1 рассматриваются как псевдотекст, содержащий, соответственно, только идентификатор-1, слово-1 или литерал-1.

(5) Операция сравнения для определения замены текста выполняется следующим образом.

Самое левое слово библиотечного текста, которое не является разделителем запятая или разделителем точка с запятой, является первым словом текста, используемым для сравнения. Лю-

GOST EXPERT ГОСТ 22558-89,

бое слово текста или пробел, предшествующий этому слову текста, копируется в исходную программу. Начиная с первого слова текста, выбранного для сравнения, и первого псевдотекста-1, идентификатора-1, слова-1 или литерала-1, который был указан в варианте REPLACING (ЗАМЕНЯЯ) весь операнд варианта REPLACING (ЗАМЕНЯЯ), который предшествует слову BY (НА), сравнивается с эквивалентным количеством последовательных слов библиотечного текста.

Псевдотекст-1, идентификатор-1, слово-1 или литерал-1 совпадают с библиотечным текстом тогда и только тогда, когда упорядоченная последовательность слов текста, которые образуют псевдотекст-1, идентификатор-1, слово-1 или литерал-1, равна символ за символом упорядоченной последовательности слов библиотечного текста. При сопоставлении каждое вхождение разделителя запятая, точка с запятой или пробел в псевдотексте-1 или в библиотечном тексте рассматривается как один пробел. Любая последовательность из одного или нескольких разделителей пробел рассматривается как один пробел.

Если совпадение не имеет места, сравнение повторяется с каждым следующим последовательным псевдотекстом-1, идентификатором-1, словом-1 или литералом-1, если они указаны, в варианте REPLACING (ЗАМЕНЯЯ), до тех пор пока не будет обнаружено совпадение или не останется ни одного следующего последовательного операнда REPLACING (ЗАМЕНЯЯ).

Если все операнды варианта REPLACING (ЗАМЕНЯЯ) сравнивались и совпадение не произошло, самое левое слово библиотечного текста копируется в исходную программу. Следующее последовательное слово библиотечного текста рассматривается как самое левое слово библиотечного текста и снова начинается цикл сравнения с первым псевдотекстом-1, идентификатором-1, словом-1 или литералом-1, указанным в варианте REPLACING (ЗАМЕНЯЯ).

Если происходит совпадение псевдотекста-1, идентификатора-1, слова-1 или литерала-1 с библиотечным текстом, соответствующий псевдотекст-2, идентификатор-2, слово-2 или литерал-2 помещаются в исходную программу. Слово библиотечного текста, непосредственно следующее за самым правым словом текста, которое участвовало в сравнении, рассматривается теперь как самое левое слово текста. Снова начинается цикл сравнения с первым псевдотекстом-1, идентификатором-1, словом-1 или литералом-1, указанным в варианте REPLACING (ЗАМЕНЯЯ).

Операция сравнения продолжается до тех пор, пока самое правое слово текста в библиотечном тексте или участвовало в успешном сравнении или было рассмотрено в качестве самого

GOST EXPERT ГОСТ 22558-89,

левого слова библиотечного текста и участвовало в полном цикле сравнения.

(6) Строки комментариев и пустые строки, появляющиеся в библиотечном тексте и псевдотексте-1, игнорируются при сопоставлении; последовательность слов текста в библиотечном тексте, если это применимо, и в псевдотексте-1 определяются правилами формата представления (см. ч. 4, п. 7.2).

Строки комментариев и пустые строки, появляющиеся в псевдотексте-2, копируются в результирующую программу без изменений всякий раз, когда псевдотекст-2 помещается в исходную программу в результате замены текста. Строки комментариев и пустые строки, появляющиеся в библиотечном тексте, копируются в результирующую исходную программу без изменений со следующим исключением: строка-комментарий или пустая строка в библиотечном тексте не копируется, если строка-комментарий или пустая строка появляется внутри последовательности слов текста, которые совпадают с псевдотекстом 1.

(7) В библиотечном тексте и псевдотексте допускаются отладочные строки. Слова текста в отладочной строке участвуют в правилах сравнения, как если бы D (T) не появлялась в поле индикатора. Отладочная строка указана в псевдотексте, если отладочная строка начинается в исходной программе после открывающего ограничителя псевдотекста, но до соответствующего закрывающего ограничителя псевдотекста.

(8) Синтаксическая правильность библиотечного текста не может быть установлена независимо. Синтаксическая правильность всей исходной программы Кобола, за исключением операторов COPY (КОПИРОВАТЬ) и REPLACE (ЗАМЕНИТЬ), не может быть определена до тех пор, пока не будут полностью обработаны все операторы COPY (КОПИРОВАТЬ) и REPLACE (ЗАМЕНИТЬ).

(9) Каждое слово текста, копируемое из библиотеки, но не замененное, копируется так, что будет начинаться в результирующей программе в той же области строки, в которой оно начинается в строке в библиотеке. Однако, если слово текста, копируемое из библиотеки, начинается в области A, и за ним следует другое слово текста, которое также начинается в области A этой же строки, и если имеет место замена предшествующего слова текста в строке заменяющим текстом большей длины, следующее слово текста начинается в области B, если оно не может быть начато в области A. Каждое слово текста в псевдотексте-2, кото-

GOST EXPERT ГОСТ 22558-89,

рое должно быть помещено в результирующую программу, начинается в результирующей программе в той же области, в которой оно появляется в псевдотексте-2. Каждый идентификатор-2, литерал-2 или слово-2, которое должно быть помещено в результирующую программу, начинается в результирующей программе в той же области, в которой появилось бы самое левое слово библиотечного текста, которое участвовало в сравнении, если бы оно не заменялось.

Библиотечный текст должен соответствовать правилам формата представления Кобола.

Если в исходную программу как результат оператора COPY (КОПИРОВАТЬ) вводятся дополнительные строки, каждое вводимое слово текста появляется в отладочной строке, если оператор COPY (КОПИРОВАТЬ) начинается в отладочной строке или если вводимое слово текста появляется в отладочной строке в библиотечном тексте.

Когда вводится слово текста, определенное фразой BY (НА), оно появляется в отладочной строке, если первое слово библиотечного текста, которое должно быть заменено, указано в отладочной строке. За исключением предыдущих случаев, только те слова текста, которые указаны в отладочных строках, которые являются отладочными в строках псевдотекста-2, появляются в отладочных строках в результирующей программе. Если какой-нибудь литерал, указанный как литерал-2 или в псевдотексте-2 или библиотечном тексте, является слишком длинным и не может вместиться в одной строке без переноса на другую в результирующей программе и литерал не должен размещаться в отладочной строке, вводятся дополнительные строки продолжения, которые содержат остаток литерала. Если замена требует, чтобы продолжаемый литерал был продолжен в отладочной строке, в программе имеется ошибка.

(10) Слова текста после замены помещаются в исходную программу для компиляции в соответствии с правилами формата представления (см. ч. 4, п. 7.). Если в исходную программу копируются слова текста из псевдотекста-2, дополнительные пробелы могут быть введены только между словами текста туда, где уже имеется пробел (включая подразумеваемый пробел между исходными строками).

(11) Если в исходную программу в результате обработки операторов COPY (КОПИРОВАТЬ) вводятся дополнительные строки, поле индикатора вводимой строки содержит такую же литеру, как строка, на которой начинается текст, подлежащий замене, за исключением случая, когда эта строка содержит дефис; в этом случае вводимая строка содержит пробел.

GOST EXPERT ГОСТ 22558-89,

В случае, когда литерал продолжается на вводимой строке, которая не является отладочной строкой, в поле индикатора помещается дефис.

3. ОПЕРАТОР REPLACE (ЗАМЕНИТЬ)

3.1. **Назначение**

Оператор REPLACE (ЗАМЕНИТЬ) используется для замены текста исходной программы.

3.2. **Общий формат**

Формат 1

REPLACE {==псевдотекст-1==
BY==псевдотекст-2==}...

ЗАМЕНИТЬ {==псевдотекст-1==
НА==псевдотекст-2==}...

Формат 2

REPLACE OFF

ОТКЛЮЧИТЬ ЗАМЕНИТЬ.

3.3. **Синтаксические правила**

(1) Оператор REPLACE (ЗАМЕНИТЬ) может появиться в исходной программе всюду, где может появиться строка-литер. Ему должен предшествовать разделитель точка, кроме случая, когда он является первым оператором отдельно компилируемой программы.

(2) Оператор REPLACE (ЗАМЕНИТЬ) должен заканчиваться разделителем точка.

(3) Псевдотекст-1 должен содержать одно или несколько слов текста.

(4) Псевдотекст-2 может содержать одно или несколько слов текста или не содержать ни одного слова текста.

(5) Строка-литер в псевдотексте-1 и псевдотексте-2 может быть продолжена (см. ч. 4, п. 7.2.5).

(6) Реализация должна допускать слова текста в псевдотексте длиной от 1 до 322 литер.

(7) Псевдотекст-1 не должен состоять только из разделителя запятая или разделителя точка с запятой.

(8) Если слово REPLACE (ЗАМЕНИТЬ) появляется в статье-комментарии или в месте, где может появиться статья-комментарий, оно рассматривается как часть статьи-комментария.

3.4. **Общие правила**

(1) Формат 1 оператора REPLACE (ЗАМЕНИТЬ) определяет текст исходной программы, который должен быть заменен

GOST EXPERT ГОСТ 22558-89,

соответствующим текстом. Каждое совпадающее появление псевдотекста-1 в исходной программе заменяется соответствующим псевдотекстом-2.

(2) Формат 2 оператора REPLACE (ЗАМЕНИТЬ) указывает, что все действующие в настоящий момент замены текста отменяются.

(3) Данное появление оператора REPLACE (ЗАМЕНИТЬ) находится в действии от точки, в которой оно указано, до следующего появления оператора или конца отдельно компилируемой программы соответственно.

(4) Все операторы REPLACE (ЗАМЕНИТЬ), находящиеся в исходной программе, обрабатываются после того, как будут обработаны все операторы COPY (КОПИРОВАТЬ), находящиеся в исходной программе.

(5) Текст, создаваемый в результате обработки оператора REPLACE (ЗАМЕНИТЬ), не должен содержать оператор REPLACE (ЗАМЕНИТЬ).

(6) Операция сравнения для определения замены текста выполняется следующим образом:

а) псевдотекст-1 сравнивается с эквивалентным количеством смежных слов текста исходной программы, начиная с самого левого слова текста исходной программы и первого псевдотекста-1;

б) псевдотекст-1 совпадает с исходным текстом тогда и только тогда, когда упорядоченная последовательность слов текста которая образует псевдотекст-1, равна символ за символом упорядоченной последовательности слов текста исходной программы. При сопоставлении каждое вхождение разделителя запятая, точка с запятой или пробел в псевдотексте-1 или в тексте исходной программы рассматривается как один пробел. Любая последовательность из одного или нескольких разделителей пробел рассматриваются как один пробел;

в) если совпадение не имеет места, сравнение повторяется для каждого следующего последовательного вхождения псевдотекста-1 до тех пор, пока не будет обнаружено совпадение или не останется ни одного следующего последовательного вхождения псевдотекста-1.

г) когда все вхождения псевдотекста-1 сравнивались и совпадение не произошло, следующее последовательное слово текста исходной программы рассматривается как самое левое слово текста исходной программы и снова повторяется цикл сравнения, начиная с первого вхождения псевдотекста-1;

д) если происходит совпадение псевдотекста-1 и текста исходной программы, соответствующий псевдотекст-2 заменяет совпавший текст в исходной программе. Слово текста исходной программы, непосредственно следующее за самым правым сло-

GOST EXPERT ГОСТ 22558-89,

вом текста, которое участвовало в сравнении, рассматривается теперь как самое левое слово текста исходной программы. Цикл сравнения слова начинается с первого вхождения псевдотекста-1;

е) операция сравнения продолжается до тех пор, пока самое правое слово текста в тексте исходной программы, которое принадлежит области действия оператора REPLACE (ЗАМЕНИТЬ), или участвовало в успешном сравнении, или было рассмотрено в качестве самого левого слова текста исходной программы и участвовало в полном цикле сравнения.

(7) Строки комментариев и пустые строки, появляющиеся в тексте исходной программы и в псевдотексте-1, при сопоставлении игнорируются; следование слов текста в тексте исходной программы и псевдотекста-1 определяется правилами формата представления (см. ч. 4, п. 7.2).

Строки комментариев и пустые строки в псевдотексте-2 помещаются в результирующую программу без изменения всякий раз, когда псевдотекст-2 помещается в исходную программу в результате замены текста. Строка комментария и пустая строка в тексте исходной программы не заменяется, если эта строка комментария или пустая строка появляется внутри последовательности слов текста, которые совпадают с псевдотекстом-1.

(8) В псевдотексте допускаются отладочные строки. Слова текста в отладочной строке участвуют в правилах сравнения, как если бы D (Т) не появлялось в поле индикатора.

(9) Синтаксическая правильность текста исходной программы, за исключением операторов COPY (КОПИРОВАТЬ) и REPLACE (ЗАМЕНИТЬ), не может быть установлена до тех пор, пока не будут полностью обработаны все операторы COPY (КОПИРОВАТЬ) и REPLACE (ЗАМЕНИТЬ).

(10) Слова текста, вводимые в исходную программу в результате обработки оператора REPLACE (ЗАМЕНИТЬ), помещаются в исходную программу в соответствии с правилами формата представления (см. ч. 4, п. 7). Когда слова текста из псевдотекста-2 вводятся в исходную программу, дополнительные пробелы могут быть введены только между словами текста, где уже имеется пробел (включая подразумеваемый пробел между исходными строками).

(11) Если в исходную программу в результате обработки операторов REPLACE (ЗАМЕНИТЬ) вводятся дополнительные строки, поле индикатора вводимой строки содержит такую же литеру, как строка, на которой начинается текст, подлежащий замене, за исключением случая, когда эта же строка содержит дефис; в этом случае вводимая строка содержит пробел.

Если какой-нибудь литерал в псевдотексте-2 является слишком длинным и не может вместиться в одной строке без перено-

GOST EXPERT ГОСТ 22558-89,

са на другую строку в результирующей программе и литерал не должен размещаться в отладочной строке, вводятся дополнительные строки продолжения, которые содержат остаток литерала. Если замена требует, чтобы продолжаемый литерал был продолжен в отладочной строке, в программе имеется ошибка.

Часть 13. МОДУЛЬ ГЕНЕРАТОРА ОТЧЕТОВ

1. ВВЕДЕНИЕ В МОДУЛЬ ГЕНЕРАТОРА ОТЧЕТОВ

1.1. **Назначение**

Модуль генератора отчетов обеспечивает средства составления отчетов посредством определения физического представления отчета, не требуя, по возможности, задания необходимых для этого процедур.

Для определения логической организации отчета использована иерархия уровней. Каждый отчет подразделяется на группы отчета, которые, в свою очередь, подразделяются на последовательности данных. Такая иерархическая структура делает возможными явные ссылки на отдельные группы отчета наряду с неявными ссылками на другие уровни иерархии. Группа отчета содержит одно или несколько данных, которые располагаются на одной или нескольких строках.

1.2. **Понятия языка**

1.2.1. Файл отчетов

Файл отчетов — это выходной файл с последовательной организацией, имеющий статью описания файла, содержащую фразу REPORT (ОТЧЕТ). Содержимое файла отчетов состоит из записей, которые будут записываться в файл под управлением системы управления генератором отчетов (СУГО).

Файл отчетов именуется в статье управления файлом и описывается статьей описания файла, содержащей фразу REPORT (ОТЧЕТ). Доступ к файлу отчетов обеспечивается операторами OPEN (ОТКРЫТЬ), GENERATE (ГЕНЕРИРОВАТЬ), INITIATE (НАЧАТЬ), SUPPRESS (ПОДАВИТЬ), TERMINATE (ЗАКОНЧИТЬ), USE AFTER STANDARD EXCEPTION PROCEDURE (ИСПОЛЬЗОВАТЬ ПОСЛЕ СТАНДАРТНОЙ ПРОЦЕДУРЫ ОШИБКИ), USE BEFORE REPORTING (ИСПОЛЬЗОВАТЬ ДО ВЫДАЧИ) и CLOSE (ЗАКРЫТЬ).

1.2.2. Специальный регистр PAGE-COUNTER (СЧЕТЧИК-СТРАНИЦ)

Зарезервированное слово PAGE-COUNTER (СЧЕТЧИК-СТРАНИЦ) служит именем счетчика страниц, который генерируется для каждой статьи описания отчета, определенной в секции отчетов раздела данных. Неявное описание счетчика соответствует описа-

GOST EXPERT ГОСТ 22558-89,

нию целого без знака в диапазоне значений от 1 до 999999, а его использование определяется реализацией. Значение PAGE-COUNTER (СЧЕТЧИК-СТРАНИЦ) обеспечивается СУГО и используется программой для нумерации страниц отчета. На PAGE-COUNTER (СЧЕТЧИК-СТРАНИЦ) можно ссылаться только во фразе SOURCE (ИСТОЧНИК) в секции отчетов и в операторах раздела процедур (п. 3.5.5 настоящей части).

1.2.3. Специальный регистр LINE-COUNTER (СЧЕТЧИК-СТРОК)

Зарезервированное слово LINE-COUNTER (СЧЕТЧИК-СТРОК) является именем счетчика строк, порождаемого для каждой статьи описания отчета, определенной в секции отчетов раздела данных. Этот счетчик неявно описан как целое без знака в диапазоне значений от 0 до 999999, а его использование определяется реализацией. Значение LINE-COUNTER (СЧЕТЧИК-СТРОК) обеспечивается системой управления генератором отчетов и используется для определения вертикального расположения отчета.

На LINE-COUNTER (СЧЕТЧИК-СТРОК) можно ссылаться только во фразе SOURCE (ИСТОЧНИК) в секции отчетов и в операторах раздела процедур; однако изменять значение LINE-COUNTER (СЧЕТЧИК-СТРОК) может только система управления генератором отчетов (СУГО) (п. 3.5.6 настоящей части).

1.2.4. Индексирование

Счетчики сумм и специальные регистры LINE-COUNTER (СЧЕТЧИК-СТРОК) и PAGE-COUNTER (СЧЕТЧИК-СТРАНИЦ) не могут быть использованы в качестве индексов в секции отчетов.

2. РАЗДЕЛ ОБОРУДОВАНИЯ В МОДУЛЕ ГЕНЕРАТОРА ОТЧЕТОВ

2.1. Секция ввода-вывода

Информация, относящаяся к секции ввода-вывода, находится в ч. 7, п. 2.1.

2.2. Параграф FILE-CONTROL (УПРАВЛЕНИЕ-ФАЙЛАМИ)

Информация, относящаяся к параграфу FILE-CONTROL (УПРАВЛЕНИЕ-ФАЙЛАМИ) находится в ч. 7, п. 2.2.

2.3. Статья управления файлом

2.3.1. Назначение

Статья управления файлом объявляет соответствующие физические атрибуты файла отчетов.

2.3.2. Общий формат

SELECT [OPTIONAL] имя-файла-1

ASSIGN TO { имя-реализации-1 / литерал-1 } ...

[RESERVE целое-1 [AREA / AREAS]]

GOST EXPERT ГОСТ 22558-89,

[[ORGANIZATION IS] SEQUENTIAL]

[PADDING CHARACTER IS {имя-данного-1 / литерал-2}]

[RECORD DELIMITER IS {STANDARD-1 / имя-реализации-2}]

[ACCESS MODE IS SEQUENTIAL]

[FILE STATUS IS имя-данного-2].

ДЛЯ [НЕОБЯЗАТЕЛЬНОГО] имя-файла-1

НАЗНАЧИТЬ {имя-реализации-1 / литерал-1} ...

[РЕЗЕРВИРОВАТЬ целое-1 ОБЛАСТЕЙ]

[[ОРГАНИЗАЦИЯ] ПОСЛЕДОВАТЕЛЬНАЯ]

[ЛИТЕРА ЗАПОЛНИТЕЛЬ {имя-данного-1 / литерал-2}]

[ОГРАНИЧИТЕЛЬ ЗАПИСИ {СТАНДАРТ-А / имя-реализации-2}]

[ДОСТУП ПОСЛЕДОВАТЕЛЬНЫЙ]

[СОСТОЯНИЕ ФАЙЛА имя-данного-2].

2.3.3. Синтаксические правила

(1) Фраза SELECT (ДЛЯ) должна быть первой в статье управления файлом. Следующие за ней фразы могут появляться в любом порядке.

(2) Каждый файл отчетов, описанный в разделе данных, должен определяться только один раз в параграфе FILE-CONTROL (УПРАВЛЕНИЕ-ФАЙЛАМИ). Каждый файл отчетов, определенный фразой SELECT (ДЛЯ), должен иметь в разделе данных этой же программы статью описания файла, содержащую фразу REPORT (ОТЧЕТ).

(3) Литерал-1 должен быть печисловым литералом и не должен быть стандартной константой. Смысл и правила для допустимого содержимого имени-реализации-1 и значения литерала-1 определяются реализацией.

(4) Допустимость отдельных фраз в статье управления файлом для файла отчетов зависит от уровня модуля последовательного ввода-вывода, поддерживаемого реализацией (см. ч. 7, п. 2.2).

2.3.4. Общие правила

(1) Если определитель файла, на который ссылается имя-файла-1, является внешним определителем файла (см. ч. 10, п. 4.5), все

GOST EXPERT ГОСТ 22558-89,

статьи управления файлом, ссылающиеся на этот определитель файла, в единице исполнения должны иметь:

а) одну и ту же спецификацию фразы OPTIONAL (НЕОБЯЗАТЕЛЬНОГО);

б) согласующуюся спецификацию для имени-реализации-1 или литерала-1 во фразе ASSIGN (НАЗНАЧИТЬ). Реализация определяет правила согласования для имени-реализации-1 или литерала-1;

в) согласующуюся спецификацию для имени-реализации-2 во фразе RECORD DELIMITER (ОГРАНИЧИТЕЛЬ ЗАПИСИ). Реализация определяет правила согласования для имени-реализации-2;

г) одно и то же значение целого-1 во фразе RESERVE (РЕЗЕРВИРОВАТЬ);

д) одну и ту же организацию;

е) один и тот же метод доступа;

ж) одну и ту же спецификацию для фразы PADDING CHARACTER (ЛИТЕРА ЗАПОЛНИТЕЛЬ).

(2) Фраза OPTIONAL (НЕОБЯЗАТЕЛЬНОГО) применяется только к файлу отчетов, открытому в режиме дополнения. Ее спецификация требуется только для файла отчетов, наличие которого необязательно во время каждого выполнения объектной программы.

(3) Фраза ASSIGN (НАЗНАЧИТЬ) определяет связь между файлом отчетов, на который ссылается имя-файла-1, и запоминающей средой, на которую ссылается имя-реализации-1 или литерал-1.

(4) Файл отчетов имеет последовательную организацию. Таким образом все фразы статьи управления файлом для файла отчетов в общем формате п. 2.3.2 настоящей части относятся к модулю последовательного ввода-вывода (см. ч. 7, п. 2.3.2).

2.4. Параграф I-O-CONTROL (УПРАВЛЕНИЕ-ВВОДОМ-ВЫВОДОМ)

2.4.1. Назначение

Параграф I-O-CONTROL (УПРАВЛЕНИЕ-ВВОДОМ-ВЫВОДОМ) определяет область памяти, которая будет использоваться различными файлами, и размещение нескольких файлов на одной катушке.

2.4.2. Общий формат

I-O-CONTROL.

[[SAME AREA FOR имя-файла-1 {имя-файла-2}...] ...

[MULTIPLE FILE TAPE CONTAINS {имя-файла-3

[POSITION целое-1]}...]...]

УПРАВЛЕНИЕ-ВВОДОМ-ВЫВОДОМ.

GOST EXPERT ГОСТ 22558-89,

[[ОБЩАЯ ОБЛАСТЬ ДЛЯ имя-файла-1
{имя-файла-2} ...] ...
[НА ОДНОЙ КАТУШКЕ {имя-файла-3 [ПОЗИЦИЯ
целое-1]} ...] ...]

2.4.3 Синтаксические правила

(1) Порядок появления фраз несущественен.

(2) Имя-файла, представляющее файл отчетов, может появляться во фразах MULTIPLE FILE TAPE (НА ОДНОЙ КАТУШКЕ) или SAME (ОБЩАЯ), для которой не указан вариант RECORD (ЗАПИСИ).

(3) Допустимость отдельных фраз в параграфе I-O-CONTROL (УПРАВЛЕНИЕ-ВВОДОМ-ВЫВОДОМ) для файла отчетов зависит от уровня модуля последовательного ввода-вывода, поддерживаемого реализацией (см. ч. 7, п. 2.10).

2.4.4. Общие правила

(1) Фраза MULTIPLE FILE TAPE (НА ОДНОЙ КАТУШКЕ) приведена в ч. 7, п. 2.11.

(2) Фраза SAME (ОБЩАЯ) приведена в ч. 7, п. 2.13.

3. РАЗДЕЛ ДАННЫХ В МОДУЛЕ ГЕНЕРАТОРА ОТЧЕТОВ

3.1. Секция файлов

Секция файлов находится в разделе данных исходной программы. Секция файлов определяет структуру файлов отчетов. Каждый файл отчетов определяется статьей описания файла, содержащей фразу REPORT (ОТЧЕТ). За статьей описания файла для файла отчетов не следуют статьи описания записи.

Общий формат секции файлов в модуле генератора отчетов приведен ниже.

FILE SECTION.

[статья-описания-файла-отчетов] ...
СЕКЦИЯ ФАЙЛОВ.

[статья-описания-файла-отчетов] ...

В Кобол-программе статья описания файла (статья FD (ОФ)) представляет наивысший уровень организации в секции файлов. За заголовком секции файлов следует статья описания файла, состоящая из индикатора уровня FD (ОФ), имени-файла и ряда независимых фраз. Для файла отчетов статья описания файла должна содержать фразу REPORT (ОТЧЕТ), определяющую имена отчетов, заносимых в файл отчетов. За статьей описания файла для файла отчетов не могут следовать никакие статьи описания записей.

GOST EXPERT ГОСТ 22558-89,

3.2. Статья описания файла

3.2.1. Назначение

Статья описания файла предоставляет информацию о физической структуре, идентификации и именах-отчетов, относящихся к файлу отчетов.

3.2.2. Общий формат

FD имя-файла-1

[LABEL { RECORD IS / RECORDS ARE } { STANDARD / OMITTED }]

[BLOCK CONTAINS [целое-1 TO] целое-2 { RECORDS / CHARACTERS }]

[RECORD { CONTAINS целое-3 CHARACTERS / CONTAINS целое-4 TO целое-5 CHARACTERS }]

[VALUE OF { имя-реализации-1 IS { имя-данного-1 / литерал-1 } } ...]

[CODE-SET IS имя-алфавита-1]

{ REPORT IS / REPORTS ARE } {имя-отчета-1} . . . .

ОФ имя-файла-1

[В БЛОКЕ [ОТ целое-1 ДО] целое-2 { ЗАПИСЕЙ / ЛИТЕР }]

[В ЗАПИСИ { целое-3 ЛИТЕР / ОТ целое-4 ДО целое-5 ЛИТЕР }]

[МЕТКИ { СТАНДАРТНЫ / ОПУЩЕНЫ }]

[{ ЗНАЧЕНИЕ / ЗНАЧ } { имя-реализации-1 { имя-данного-1 / литерал-1 } } ...]

[АЛФАВИТ имя-алфавита-1]

{ ОТЧЕТ / ОТЧЕТЫ } {имя-отчета-1} . . . .

GOST EXPERT ГОСТ 22558-89,

3.2.3. Синтаксические правила

(1) Индикатор уровня FD (ОФ) определяет начало статьи описания файла и должен предшествовать имени файла отчетов.

(2) Фразы, которые следуют за именем-файла-1, могут появляться в любом порядке.

(3) Имя-файла-1 может относиться только к последовательному файлу.

(4) За статьей описания файла для файла отчетов не могут следовать статьи описания записей.

(5) На субъект статьи описания файла, определяющей фразу REPORT (ОТЧЕТ), можно ссылаться в разделе процедур только в операторах USE (ИСПОЛЬЗОВАТЬ), CLOSE (ЗАКРЫТЬ) или OPEN (ОТКРЫТЬ) с фразой OUTPUT (ВЫХОДНОЙ) или EXTEND (ДОПОЛНЯЕМЫЙ).

(6) Допустимость отдельных фраз в такой статье описания файла зависит от уровня модуля последовательного ввода-вывода, поддерживаемого реализацией (см. ч. 7, п. 3.2).

3.2.4. Общие правила

(1) Статья описания файла связывает имя-файла-1 с определителем файла.

(2) Структура логической записи файла, связанного с именем-файла-1, определяется реализацией.

(3) Все фразы статьи описания файла в п. 3.2 для файла отчетов, за исключением фразы REPORT (ОТЧЕТ), описаны в модуле последовательного ввода-вывода (см. ч. 7, п. 3.2).

(4) Фраза REPORT (ОТЧЕТ) представлена в п. 3.3 настоящей части.

3.3. Фраза REPORT (ОТЧЕТ)

3.3.1. Назначение

Фраза REPORT (ОТЧЕТ) указывает имена отчетов, образующих файл отчетов.

3.3.2. Общий формат

$$\left\{\begin{array}{l}\underline{\text{REPORT}}\ \text{IS}\\ \underline{\text{REPORTS}}\ \text{ARE}\end{array}\right\}\{\text{имя-отчета-1}\}\ldots$$

$$\left\{\begin{array}{l}\underline{\text{ОТЧЕТ}}\\ \underline{\text{ОТЧЕТЫ}}\end{array}\right\}\{\text{имя-отчета-1}\}\ldots,$$

3.3.3. Синтаксические правила

(1) Каждое имя-отчета, указанное во фразе REPORT (ОТЧЕТ), должно быть субъектом статьи описания отчета в секции отчетов. Порядок появления имен-отчетов во фразе не существенен.

(2) Каждое имя-отчета может появляться только в одной фразе REPORT (ОТЧЕТ).

GOST EXPERT ГОСТ 22558-89,

(3) На субъект статьи описания файла, содержащей фразу REPORT (ОТЧЕТ), можно ссылаться в разделе процедур только в операторах USE (ИСПОЛЬЗОВАТЬ), CLOSE (ЗАКРЫТЬ) или OPEN (ОТКРЫТЬ) с фразой OUTPUT (ВЫХОДНОЙ) или EXTEND (ДОПОЛНЯЕМЫЙ).

3.3.4. Общие правила

(1) Присутствие нескольких имен-отчетов во фразе REPORT (ОТЧЕТ) указывает, что файл содержит несколько отчетов.

(2) После выполнения оператора INITIATE (НАЧАТЬ) и до выполнения оператора TERMINATE (ЗАКОНЧИТЬ) для одного и того же файла отчетов файл отчетов находится под управлением системы управления генератором отчетов. Когда файл отчетов находится под управлением СУГО, никакой оператор ввода-вывода, ссылающийся на этот файл отчетов, не может выполняться.

(3) Если соответствующий определитель файла является внешним определителем файла, каждая статья описания файла в единице исполнения, связанная с этим определителем файла, должна описывать файл как файл отчетов.

3.4. **Секция отчетов**

Секция отчетов располагается в разделе данных исходной программы. Секция отчетов описывает отчеты, которые будут записаны в файл отчетов. Описание каждого отчета должно начинаться статьей описания отчета (статьей RD (OO)), за которой следуют одна или несколько статей описания групп отчета.

Ниже приводится общий формат секции отчетов.

REPORT SECTION.

[статья-описания-отчета
{статья-описания-группы-отчета}...] ...

СЕКЦИЯ ОТЧЕТОВ.

[статья-описания-отчета
{статья-описания-группы-отчета}...] ...

3.4.1. Статья описания отчета

Статья описания отчета (статья RD (OO)) определяет имя отчета, формат каждой страницы отчета, указывая вертикальные границы поля страницы, в котором может быть отпечатан каждый тип группы отчета. Статья описания отчета также указывает управляющие данные. При составлении отчета изменения значений управляющих данных приводят к порождению групп отчета, называемых управляемыми группами.

Каждый отчет, названный во фразе REPORT (ОТЧЕТ) статьи описания файла в секции файлов, должен быть субъектом статьи описания отчета в секции отчетов. Более того, каждый отчет в секции отчетов должен быть назван только в одной статье описания файла.

3.4.2. Статья описания группы отчета

Группы отчета, образующие отчет, описываются вслед за ста-

GOST EXPERT ГОСТ 22558-89,

тьей описания отчета. Описание каждой группы отчета начинается статьей описания группы отчета, имеющей номер уровня 01 и фразу TYPE (ТИП). Статье описания группы отчета могут подчиняться статьи описания групповых и элементарных данных, которые более подробно описывают характеристики группы отчета.

3.5. **Статья описания отчета**

3.5.1. Назначение

Статья описания отчета именует отчет, указывает идентифицирующие литеры, которые вставляются в начале каждой печатаемой строки отчета, и описывает его физическую структуру и организацию.

3.5.2. Общий формат

RD имя-отчета-1

[CODE литерал-1]

[{CONTROL IS / CONTROLS ARE} {имя-данного-1} ... / FINAL [имя-данного-1] ...]

[PAGE {LIMIT IS / LIMITS ARE} целое-1 {LINE / LINES}

[HEADING целое-2]

[FIRST DETAIL целое-3] [LAST DETAIL целое-4]

[FOOTING целое-5]].

ОО имя-отчета-1

[С КОДОМ литерал-1]

[УПРАВЛЕНИЕ ПО {{имя-данного-1} ... / КОНЦУ [имя-данного-1] ...}]

[РАЗМЕР СТРАНИЦЫ целое-1 СТРОК [ЗАГОЛОВОК

целое-2]

[ПЕРВЫЙ ФРАГМЕНТ целое-3]

[ПОСЛЕДНИЙ ФРАГМЕНТ целое-4]

[КОНЦОВКА целое-5]].

3.5.3. Синтаксические правила

(1) Имя-отчета-1 должно указываться в одной и только одной фразе REPORTS (ОТЧЕТЫ).

(2) Порядок появления фраз, следующих за именем-отчета-1, не существенен.

GOST EXPERT ГОСТ 22558-89,

(3) Имя-отчета-1 является наивысшим допустимым уточнителем, который может быть указан для LINE-COUNTER (СЧЕТЧИК-СТРОК), PAGE-COUNTER (СЧЕТЧИК-СТРАНИЦ) и для всех имен-данных, определяемых в секции отчетов.

3.5.4. Общие правила

(1) Фразы CODE (С КОДОМ), CONTROL (УПРАВЛЕНИЕ) и PAGE (РАЗМЕР СТРАНИЦЫ) представлены начиная с п. 3.6 настоящей части.

3.5.5. Правила для счетчика страниц

(1) PAGE-COUNTER (СЧЕТЧИК-СТРАНИЦ) — это зарезервированное слово, используемое для ссылки на специальный регистр, автоматически создаваемый для каждого отчета, указанного в секции отчетов (см. ч. 4, п. 4.2.2.1.3.3.1 и п. 1.2.2 настоящей части).

(2) В секции отчетов ссылка на PAGE-COUNTER (СЧЕТЧИК-СТРАНИЦ) разрешается только во фразе SOURCE (ИСТОЧНИК). В разделе процедур PAGE-COUNTER (СЧЕТЧИК-СТРАНИЦ) может быть использован в любом контексте, в котором может быть указано данное или значение целого.

(3) Если в программе имеется более одного PAGE-COUNTER (СЧЕТЧИК-СТРАНИЦ), то обращение к каждому из них в разделе процедур должно быть уточнено именем-отчета.

В секции отчетов неуточненная ссылка на PAGE-COUNTER (СЧЕТЧИК-СТРАНИЦ) уточняется неявно именем отчета, в статье описания которого сделана ссылка. Всякий раз, когда имеется ссылка на PAGE-COUNTER (СЧЕТЧИК-СТРАНИЦ) другого отчета, PAGE-COUNTER (СЧЕТЧИК-СТРАНИЦ) должен быть явно уточнен соответствующим именем-отчета.

(4) Выполнение оператора INITIATE (НАЧАТЬ) приводит к установке системой управления генератором отчетов счетчика страниц соответствующего отчета в единицу.

(5) Счетчик страниц автоматически увеличивается на единицу каждый раз, когда система управления генератором отчетов осуществляет переход на следующую страницу.

(6) Счетчик страниц может быть изменен операторами раздела процедур.

3.5.6. Правила для счетчика строк

(1) LINE-COUNTER (СЧЕТЧИК-СТРОК) — это зарезервированное слово, используемое для ссылки на специальный регистр, автоматически создаваемый для каждого отчета, указанного в секции отчетов (см. ч. 4, п. 4.2.2.1.3.3.1 и п. 1.2.3 настоящей части).

(2) В секции отчетов ссылка на LINE-COUNTER (СЧЕТЧИК-СТРОК) разрешена только во фразе SOURCE (ИСТОЧНИК). В разделе процедур LINE-COUNTER (СЧЕТЧИК-СТРОК) может быть использован в любом контексте, в котором может быть ука-

GOST EXPERT ГОСТ 22558-89,

зано данное или значение целого. Однако менять содержимое счетчика строк может только система управления генератором отчетов.

(3) Если в программе имеется более одного счетчика строк, ссылка на каждый из них в разделе процедур должна уточняться именем-отчета.

В секции отчетов неуточненная ссылка на LINE-COUNTER (СЧЕТЧИК-СТРОК) неявно уточняется именем отчета, в статье описания которого сделана ссылка. Каждый раз, когда имеется ссылка на LINE-COUNTER (СЧЕТЧИК-СТРОК) другого отчета, LINE-COUNTER (СЧЕТЧИК-СТРОК) должен быть явно уточнен соответствующим именем-отчета.

(4) Выполнение оператора INITIATE (НАЧАТЬ) приводит к установке системой управления генератором отчетов счетчика строк соответствующего отчета в нуль. Система управления генератором отчетов также автоматически переустанавливает счетчик строк в нуль при переходе на следующую страницу.

(5) На значение счетчика строк не влияет обработка непечатаемых групп отчета или печатаемых групп отчета, печать которых отменяется посредством оператора SUPPRESS (ПОДАВИТЬ).

(6) В момент представления каждой печатаемой строки значение счетчика строк указывает номер строки, на которой представляется печатаемая строка. Значение счетчика строк после представления группы отчета управляется правилами представления для группы отчета (п. 3.10 настоящей части).

3.6. **Фраза CODE (С КОДОМ)**

3.6.1. Назначение

Фраза CODE (С КОДОМ) указывает литерал из двух литер, идентифицирующий каждую печатаемую строку отчета как принадлежащую к определенному отчету.

3.6.2. Общий формат

CODE литерал-1

С КОДОМ литерал-1

3.6.3. Синтаксические правила

(1) Литерал-1 должен быть нечисловым литералом, состоящим из двух литер.

(2) Если фраза CODE (С КОДОМ) указана для некоторого отчета в файле, то она должна быть указана для всех отчетов этого файла.

3.6.4. Общие правила

(1) Если указана фраза CODE (С КОДОМ), литерал-1 автоматически помещается в первые две позиции литер каждой логической записи отчета.

(2) Позиции, занимаемые литералом-1, не включаются в описание печатаемой строки, но включаются в размер логической записи.

GOST EXPERT ГОСТ 22558-89,

3.7. Фраза CONTROL (УПРАВЛЕНИЕ)

3.7.1. Назначение

Фраза CONTROL (УПРАВЛЕНИЕ) устанавливает для отчета уровни иерархии управления.

3.7.2. Общий формат

$$\left\{\begin{matrix}\underline{\text{CONTROL}}\ \text{IS}\\ \underline{\text{CONTROLS}}\ \text{ARE}\end{matrix}\right\}\left\{\begin{matrix}\{\text{имя-данного-1}\}\ldots\\ \underline{\text{FINAL}}\ [\text{имя-данного-1}]\ \ldots\end{matrix}\right\}$$

$$\underline{\text{УПРАВЛЕНИЕ ПО}}\left\{\begin{matrix}\{\text{имя-данного-1}\}\ldots\\ \underline{\text{КОНЦУ}}\ [\text{имя-данного-1}]\ \ldots\end{matrix}\right\}$$

3.7.3. Синтаксические правила

(1) Имя-данного-1 не должно определяться в секции отчетов. Имя-данного-1 может быть уточненным.

(2) Каждое повторение имени-данного-1 должно идентифицировать различные данные.

(3) Имя-данного-1 не должно иметь подчиненных данных с переменным числом вхождений.

3.7.4. Общие правила

(1) Имя-данного-1 и слово FINAL (ПО КОНЦУ) указывают уровни иерархии управления. Если указано FINAL (ПО КОНЦУ), то это самый высокий уровень управления, имя-данного-1 определяет старший уровень управления, следующее повторение имени-данного-1 — промежуточный уровень управления и т. д. Последнее повторение имени-данного-1 определяет младший уровень управления.

(2) При выполнении первого по времени оператора GENERATE (ГЕНЕРИРОВАТЬ) для данного отчета система управления генератором отчетов запоминает значения всех управляющих данных, связанных с этим отчетом. При следующих выполнениях всех операторов GENERATE (ГЕНЕРИРОВАТЬ) для данного отчета СУГО проверяет, не изменились ли значения управляющих данных. Изменение значения любого управляющего данного приводит к прерыванию управления. Это прерывание управления связано с наивысшим уровнем иерархии управления, для которого отмечено изменение значения (п. 4.3 настоящей части).

(3) Система управления генератором отчетов (СУГО) определяет наличие прерывания управления путем сравнения содержимого каждого управляющего данного с предыдущим содержимым каждого управляющего данного, сохраненным во время предыдущего выполнения оператора GENERATE (ГЕНЕРИРОВАТЬ) для этого же отчета. Сравнение осуществляется следующим образом:

а) если управляющее данное является числовым данным, проверка отношения является сравнением двух числовых операндов;

б) если управляющее данное является индексным данным, проверка отношения является сравнением двух индексных данных;

GOST EXPERT ГОСТ 22558-89,

в) в остальных случаях проверка отношения является сравнением двух нечисловых операндов.

Проверка отношения неравенства объясняется в соответствующих параграфах (см. ч. 6, п. 6.3.1.1).

(4) CONTROL IS FINAL (УПРАВЛЕНИЕ ПО КОНЦУ) используется, если наиболее объемлющая управляемая группа в отчете не связана с управляющим именем-данного.

3.8. **Фраза PAGE (РАЗМЕР СТРАНИЦЫ)**

3.8.1. Назначение

Фраза PAGE (РАЗМЕР СТРАНИЦЫ) определяет длину страницы и вертикальные подразделения, на которых представляются группы отчета.

3.8.2. Общий формат

PAGE [LIMIT IS / LIMITS ARE] целое-1 [LINE / LINES] [HEADING целое-2]

[FIRST DETAIL целое-3] [LAST DETAIL целое-4]

[FOOTING целое-5]

РАЗМЕР СТРАНИЦЫ целое-1 СТРОК [ЗАГОЛОВОК целое-2]

[ПЕРВЫЙ ФРАГМЕНТ целое-3]

[ПОСЛЕДНИЙ ФРАГМЕНТ целое-4]

[КОНЦОВКА целое-5]

3.8.3. Синтаксические правила

(1) Все указанные в формате варианты фразы могут быть записаны в произвольном порядке.

(2) Целое-1 должно быть не более чем трехзначное число.

(3) Целое-2 должно быть больше или равно единице.

(4) Целое-3 должно быть больше или равно целому-2.

(5) Целое-4 должно быть больше или равно целому-3.

(6) Целое-5 должно быть больше или равно целому-4.

(7) Целое-1 должно быть больше или равно целому-5.

(8) Следующие правила указывают вертикальные подразделения страницы, на которых могут представляться группы отчета различного типа, если указана фраза PAGE (РАЗМЕР СТРАНИЦЫ) (п. 3.8.5 настоящей части).

а) Если группа отчета, являющаяся заголовком отчета, должна быть представлена на отдельной странице, то описание группы должно определять ее представление в вертикальном подразделении страницы, начиная от номера строки, указанного целым-2, до номера строки, указаного целым-1, включительно.

Если группа отчета, являющаяся заголовком отчета, не должна быть представлена на отдельной странице, то описание группы дол-

GOST EXPERT ГОСТ 22558-89,

жно определять ее представление, начиная от номера строки, указанного целым-2, до номера строки, на единицу меньшего целого-3, включительно.

б) Если указана группа отчета типа заголовок страницы, то ее описание должно определять ее представление в вертикальном подразделении страницы, начиная от номера строки, указанного целым-2, до номера строки, на единицу меньшего целого-3, включительно.

в) Если указана группа отчета управляемый заголовок или фрагмент, описание каждой из них должно определять представление соответствующей группы в вертикальном подразделении страницы, начиная от номера строки, указанного целым-3, до номера строки, указанного целым-4, включительно.

г) Если указана группа отчета управляемая концовка, ее описание должно определять ее представление в вертикальном подразделении страницы, начиная от номера строки, указанного целым-3, до номера строки, указанного целым-5, включительно.

д) Если указана группа отчета концовка страницы, ее описание должно определять ее представление в вертикальном подразделении страницы, начиная от номера строки, на единицу большего целого-5, до номера строки, указанного целым-1, включительно.

е) Описание группы отчета концовка отчета, которая должна быть представлена на отдельной странице, должно определять ее представление в вертикальном подразделении страницы, начиная от номера строки, указанного целым-2, до номера строки, указанного целым-1 включительно.

Описание группы отчета концовка отчета, которая не должна представляться на отдельной странице, должно определять ее представление, начиная от номера строки, указанного целым-5 плюс 1, до номера строки, указанного целым-1, включительно.

(9) Все группы отчета должны быть описаны так, чтобы каждая из них могла быть представлена на одной странице. СУГО не производит разбиение групп отчета, выходящих за границы страницы.

3.8.4. Общие правила

(1) Значения целых, указанные во фразе PAGE (РАЗМЕР СТРАНИЦЫ), определяют вертикальный формат страницы отчета.

а) Целое-1 определяет размер страницы отчета, указывая число строк, доступных на каждой странице.

б) Вариант HEADING целое-2 (ЗАГОЛОВОК целое-2) определяет номер первой из строк, на которой может быть представлена группа отчета заголовок страницы или заголовок отчета.

в) Вариант FIRST DETAIL (ПЕРВЫЙ ФРАГМЕНТ) определяет номер первой из строк, на которой может быть представлена группа тела отчета. Заголовок отчета (без варианта NEXT GROUP NEXT PAGE (СЛЕДУЮЩАЯ ГРУППА НА СЛЕДУЮЩЕЙ

GOST EXPERT ГОСТ 22558-89,

СТРАНИЦЕ)) и заголовок страницы не могут быть представлены на строке с номером равным или больше целому-3.

г) Вариант LAST DETAIL (ПОСЛЕДНИЙ ФРАГМЕНТ) указывает номер последней строки, на которой могут быть представлены управляемый заголовок или фрагмент.

д) Вариант FOOTING целое-5 (КОНЦОВКА целое-5) определяет номер последней строки, на которой может быть представлена группа отчета управляемая концовка. Группа отчета концовка отчета (без варианта LINE целое-1 NEXT PAGE (НОМЕР СТРОКИ целое-1 НА СЛЕДУЮЩЕЙ СТРАНИЦЕ)) и концовка страницы должны располагаться на строке, следующей за строкой с номером, указанным целым-5.

(2) Если указана фраза PAGE (РАЗМЕР СТРАНИЦЫ) для ее опущенных вариантов имеют место следующие соглашения по умолчанию:

а) если вариант HEADING (ЗАГОЛОВОК) опущен, целое-2 предполагается равным единице;

б) если вариант FIRST DETAIL (ПЕРВЫЙ ФРАГМЕНТ) опущен, целое-3 предполагается равным целому-2;

в) если оба варианта LAST DETAIL (ПОСЛЕДНИЙ ФРАГМЕНТ) и FOOTING (КОНЦОВКА) опущены, целое-4 и целое-5 предполагаются равными целому-1;

г) если вариант FOOTING (КОНЦОВКА) указан, а вариант LAST DETAIL (ПОСЛЕДНИЙ ФРАГМЕНТ) опущен, целое-4 предполагается равным целому-5;

д) если вариант LAST DETAIL (ПОСЛЕДНИЙ ФРАГМЕНТ) указан, а вариант FOOTING (КОНЦОВКА) опущен, целому-5 присваивается значение целого-4.

(3) Если фраза PAGE (РАЗМЕР СТРАНИЦЫ) не указана, отчет состоит из одной страницы неопределенной длины.

(4) Правила представления для каждого типа группы отчета определены в соответствующем параграфе (п. 3.10 настоящей части).

3.8.5. Области страницы

Ниже описаны области страницы, устанавливаемые фразой PAGE (РАЗМЕР СТРАНИЦЫ).

Группы отчета, которые могут быть представлены в области	Номер первой строки области	Номер последней строки области
Заголовок отчета, описанный с вариантом NEXT GROUP NEXT PAGE (СЛЕДУЮЩАЯ ГРУППА НА СЛЕДУЮЩЕЙ СТРАНИЦЕ)	Целое-2	Целое-1
Концовка отчета, описанная вариантом LINE целое-1 NEXT PAGE (НОМЕР СТРОКИ целое-1 НА СЛЕДУЮЩЕЙ СТРАНИЦЕ)	Целое-2	Целое-1

GOST EXPERT ГОСТ 22558-89,

Группы отчета, которые могут быть представлены в области	Номер первой строки области	Номер последней строки области
Заголовок отчета, описанный без варианта NEXT GROUP NEXT PAGE (СЛЕДУЮЩАЯ ГРУППА НА СЛЕДУЮЩЕЙ СТРАНИЦЕ) Заголовок страницы	Целое-2	Целое-3 минус 1
Управляемый заголовок Фрагмент	Целое-3	Целое-4
Управляемая концовка	Целое-2	Целое-5
Концовка страницы Концовка отчета, описанная без варианта LINE целое-1 NEXT PAGE (НОМЕР СТРОКИ целое-1 НА СЛЕДУЮЩЕЙ СТРАНИЦЕ)	Целое-5 плюс 1	Целое-1

3.9. Статья описания группы отчета

3.9.1. Назначение

Статья описания группы отчета определяет характеристики группы отчета и отдельных данных в группе отчета.

3.9.2. Общий формат

Формат 1

```
01 [имя-данного-1]
   [ LINE NUMBER IS { целое-1 [ON NEXT PAGE] } ]
                    { PLUS целое-2            }
   [ NEXT GROUP IS { целое-3    } ]
                   { PLUS целое-4 }
                   { NEXT PAGE    }
```


GOST EXPERT ГОСТ 22558-89,

```
                    ┌ REPORT HEADING ┐
                    └ RH             ┘
                    ┌ PAGE HEADING   ┐
                    └ PH             ┘
                    ┌ CONTROL HEADING ┐ ┌ имя-данного-2 ┐
                    └ CH              ┘ └ FINAL         ┘
TYPE IS             ┌ DETAIL          ┐
                    └ DE              ┘
                    ┌ CONTROL FOOTING ┐ ┌ имя-данного-3 ┐
                    └ CF              ┘ └ FINAL         ┘
                    ┌ PAGE FOOTING    ┐
                    └ PF              ┘
                    ┌ REPORT FOOTING  ┐
                    └ RF              ┘
```

[[USAGE IS] DISPLAY].

01 [имя-данного-1]

```
[ НОМЕР СТРОКИ ┌ целое-1 [НА СЛЕДУЮЩЕЙ СТРАНИЦЕ] ┐ ]
               └ ПЛЮС целое-2                     ┘

[ СЛЕДУЮЩАЯ ГРУППА ┌ целое-3                  ┐ ]
                   │ ПЛЮС целое-4             │
                   └ НА СЛЕДУЮЩЕЙ СТРАНИЦЕ    ┘ ]
```


GOST EXPERT ГОСТ 22558-89,

ТИП { ЗАГОЛОВОК ОТЧЕТА / ЗО }
{ ЗАГОЛОВОК СТРАНИЦЫ / ЗС }
{ УПРАВЛЯЕМЫЙ ЗАГОЛОВОК / УЗ } ПО { имя-данного-2 / КОНЦУ }
{ ФРАГМЕНТ / ФР }
{ УПРАВЛЯЕМАЯ КОНЦОВКА / УК } ПО { имя-данного-3 / КОНЦУ }
{ КОНЦОВКА СТРАНИЦЫ / КС }
{ КОНЦОВКА ОТЧЕТА / КО }
[ДЛЯ ВЫДАЧИ].

Формат 2

номер-уровня [имя-данного-1]

[LINE NUMBER IS { целое-1 [ON NEXT PAGE] / PLUS целое-2 }]

[[USAGE IS] DISPLAY].

номер-уровня [имя-данного-1]

[НОМЕР СТРОКИ { целое-1 [НА СЛЕДУЮЩЕЙ СТРАНИЦЕ] / ПЛЮС целое-2 }]

[ДЛЯ ВЫДАЧИ].

GOST EXPERT ГОСТ 22558-89,

Формат 3

номер-уровня [имя-данного-1]

{PICTURE / PIC} IS строка-литер

[[USAGE IS] DISPLAY]

[[SIGN IS] {LEADING / TRAILING} [SEPARATE CHARACTER]]

[{JUSTIFIED / JUST} RIGHT]

[BLANK WHEN ZERO]

[LINE NUMBER IS {целое-1 [ON NEXT PAGE] / PLUS целое-2}]

[COLUMN NUMBER IS целое-3]

{SOURCE IS идентификатор-1
VALUE IS литерал-1
[SUM {идентификатор-2}... [UPON {имя-данного-2}...]]...
[RESET ON {имя-данного-3 / FINAL}]}

[GROUP INDICATE].

номер-уровня [имя-данного-1]

{ШАБЛОН / Ш} строка-литер

[ДЛЯ ВЫДАЧИ]

[[ЗНАК] {ПЕРВЫЙ / ПОСЛЕДНИЙ} [ОТДЕЛЬНО]]

[СДВИНУТО ВПРАВО]

[ПРОБЕЛ КОГДА НУЛЬ]

GOST EXPERT ГОСТ 22558-89,

```
[ НОМЕР СТРОКИ { целое-1 [НА СЛЕДУЮЩЕЙ СТРАНИЦЕ] } ]
               { ПЛЮС целое-2                     }

[НОМЕР СТОЛБЦА целое-3]

{ ИСТОЧНИК идентификатор-1

  { ЗНАЧЕНИЕ } литерал-1
  { ЗНАЧ     }

  {СУММА {идентификатор-2} ... [ДЛЯ {имя-данного-2} ... ]} ...

    [ СБРОСИТЬ ПО { имя-данного-3 } ]
                  { КОНЦУ         }
}

[ОПРЕДЕЛЯЕТ ГРУППУ].
```

3.9.3. Синтаксические правила

(1) Статья описания группы отчета может указываться только в секции отчетов.

(2) Фразы могут быть записаны в любом порядке, за исключением имени-данного, которое (если указывается) должно следовать непосредственно за номером-уровня.

(3) В формате 2 номер-уровня может быть любым целым от 02 до 48 включительно, в формате 3 — от 02 до 49 включительно.

(4) Описание группы отчета может состоять из одного, двух или трех уровней иерархии.

а) Первая из статей, описывающих группу отчета, должна быть представлена форматом 1.

б) Статьи формата 2 или 3 могут непосредственно подчиняться статье формата 1.

в) По крайней мере одна статья формата 3 должна подчиняться непосредственно статье формата 2.

г) Статьи формата 3 должны определять элементарные данные.

(5) В статье формата 1 имя-данного-1 обязательно только в следующих случаях:

а) если на группу отчета типа фрагмент имеется ссылка в операторе GENERATE (ГЕНЕРИРОВАТЬ);

б) если на группу отчета типа фрагмент имеется ссылка в вариате UPON (ДЛЯ) фразы SUM (СУММА);

в) если на группу отчета имеется ссылка в операторе USE BEFORE REPORTING (ИСПОЛЬЗОВАТЬ ДО ВЫДАЧИ);

GOST EXPERT ГОСТ 22558-89,

г) если имя группы отчета типа управляемый заголовок используется для уточнения ссылки на счетчик суммы.

Если указано имя-данного-1, на него можно ссылаться только в операторе GENERATE (ГЕНЕРИРОВАТЬ), варианте UPON (ДЛЯ) фразы SUM (СУММА), операторе USE BEFORE REPORTING (ИСПОЛЬЗОВАТЬ ДО ВЫДАЧИ) или использовать как уточнитель счетчика суммы.

(6) Статья формата 2 должна содержать, по крайней мере, одну необязательную фразу.

(7) В статье формата 2 имя-данного-1 не обязательно. При наличии имени-данного оно может быть использовано только для уточнения ссылок на счетчик суммы.

(8) В секции отчетов фраза об использовании используется только для печатаемых данных.

а) Если фраза об использовании указана в статье формата 3, эта статья должна определять печатаемое данное.

б) Если фраза об использовании появляется в статье формата 1 или 2, по крайней мере одна подчиненная ей статья должна определять печатаемое данное.

(9) Статья, содержащая фразу LINE NUMBER (НОМЕР СТРОКИ), не должна иметь подчиненных статей, содержащих эту фразу.

(10) Статьи формата 3 подчиняются следующим ограничениям:

а) фраза GROUP INDICATE (ОПРЕДЕЛЯЕТ ГРУППУ) может указываться только в группе отчета типа фрагмент;

б) фраза SUM (СУММА) может указываться только в группе отчета типа управляемая концовка;

в) статья, содержащая фразу COLUMN NUMBER (НОМЕР СТОЛБЦА), но не содержащая фразу LINE NUMBER (НОМЕР СТРОКИ), должна быть подчинена статье, содержащей фразу LINE NUMBER (НОМЕР СТРОКИ);

г) имя-данного-1 не обязательно, но может быть указано в любой статье. Однако на имя-данного-1 можно ссылаться только тогда, когда статья определяет счетчик суммы;

д) статья, содержащая фразу VALUE (ЗНАЧЕНИЕ), должна также содержать фразу COLUMN NUMBER (НОМЕР СТОЛБЦА).

(11) Ниже представлены все допустимые комбинации фраз для статей формата 3. Таблицу следует читать слева направо вдоль выбранной строки.

«О» указывает, что наличие фразы обязательно.

«Р» указывает, что наличие фразы разрешается, но фраза не является обязательной.

«—» указывает, что фраза не разрешается.

GOST EXPERT ГОСТ 22558-89,

PIC (Ш)	COLUMN (НОМЕР СТОЛБЦА)	SOURCE (ИСТОЧНИК)	SUM (СУММА)	VALUE (ЗНАЧЕНИЕ)	JUST (СДВИНУТО)	BLANK WHEN ZERO (ПРОБЕЛ КОГДА НУЛЬ)	GROUP INDICATE (ОПРЕДЕЛЯЕТ ГРУППУ)	USAGE (ДЛЯ ВЫДАЧИ)	SIGN (ЗНАК)	LINE NUMBER (НОМЕР СТРОКИ)
О	—	—	О	—	—	—	—	—	Р	Р
О	О	—	О	—	—	Р	—	Р	Р	Р
О	Р	О	—	—	Р	—	Р	Р	Р	Р
О	Р	О	—	—	—	Р	Р	Р	Р	Р
О	О	—	—	О	Р	—	Р	Р	Р	Р

3.9.4. Общие правила

(1) Формат 1 относится к статье группы отчета. Группа отчета определяется этой статьей и всеми подчиненными ей статьями.

(2) Фразы BLANK WHEN ZERO (ПРОБЕЛ КОГДА НУЛЬ), JUSTIFIED (СДВИНУТО) и PICTURE (ШАБЛОН) в модуле генератора отчетов такие же, как соответствующие фразы в модуле ядра. Поэтому описание этих фраз см. в ч. 7, пп. 5.4, 5.6 и 5.9, соответственно. Остальные фразы статьи описания группы отчета представлены, начиная с п. 3.11 настоящей части.

3.10. Таблицы правил представления

3.10.1. Назначение

В таблицах и правилах, приведенных ниже, определяются:

(1) допустимые комбинации фраз LINE NUMBER (НОМЕР СТРОКИ) и NEXT GROUP (СЛЕДУЮЩАЯ ГРУППА) для каждого из типов группы отчета;

(2) требования к использованию этих фраз;

(3) интерпретация этих фраз системой управления генератором отчетов.

3.10.2. Пояснение к таблицам

Для каждого из следующих типов групп отчета — заголовок отчета, заголовок страницы, концовка страницы, концовка отчета, — имеется отдельная таблица правил представления. Кроме того, в таблице правил представления тела группы объединены группы отчета типов фрагмент, управляемый заголовком и управляемая концовка (п. 3.10.8 настоящей части).

В столбцах 1 и 2 таблицы правил представления выписаны все допустимые комбинации фраз LINE NUMBER (НОМЕР СТРОКИ) и NEXT GROUP (СЛЕДУЮЩАЯ ГРУППА) для соответствующе-

GOST EXPERT ГОСТ 22558-89,

го типа группы отчета. Для определения набора правил представления, применяемого для определенной комбинации фраз LINE NUMBER (НОМЕР СТРОКИ) и NEXT GROUP (СЛЕДУЮЩАЯ ГРУППА), следует читать таблицу правил представления слева направо вдоль выбранной строки.

Столбцы применяемых правил в таблице правил представления разделены на две части. В первой части указаны правила, применяемые, если описание отчета содержит фразу PAGE (РАЗМЕР СТРАНИЦЫ), и во второй части указаны правила, применяемые, если фраза PAGE (РАЗМЕР СТРАНИЦЫ) отсутствует. Назначение правил, указанных в столбцах применяемых правил, приведено ниже.

(1) Правила верхней и нижней границы

Эти правила определяют вертикальные подразделения страницы, в рамках которых может быть представлена указанная группа отчета.

При отсутствии фразы PAGE (РАЗМЕР СТРАНИЦЫ) печатаемый отчет рассматривается как не разделенный по вертикали. Поэтому в таблицах правила верхней границы и правила нижней границы не указаны для описания отчета, в котором фраза PAGE (РАЗМЕР СТРАНИЦЫ) опущена.

(2) Правила проверки вместимости

Правила проверки вместимости применимы только к группам тела отчета, и поэтому правила проверки вместимости указаны только в таблице правил представления группы тела отчета. Во время выполнения система управления генератором отчетов применяет правила проверки вместимости для определения, может ли быть представлена соответствующая группа тела отчета на странице, на которой в текущее время располагается отчет.

Тем не менее, даже для групп тела отчета нет правил проверки вместимости, если фраза PAGE (РАЗМЕР СТРАНИЦЫ) опущена в статье описания отчета.

(3) Правила позиции первой печатаемой строки

Правила позиции первой печатаемой строки определяют местоположение, где система управления генератором отчетов может представить первую печатаемую строку данной группы отчета.

Таблицы этих правил не определяют местоположение, где система управления генератором отчетов может представить вторую и последующие (если таковые имеются) строки группы отчета. Отдельные общие правила определяют, где могут быть представлены вторая и последующие строки группы отчета. Эта информация содержится в общих правилах фразы LINE NUMBER (НОМЕР СТРОКИ) (п. 3.15 настоящей части).

(4) Правила следующей группы

Правила следующей группы относятся к особенностям использования фразы NEXT GROUP (СЛЕДУЮЩАЯ ГРУППА).

GOST EXPERT ГОСТ 22558-89,

(5) Правила результирующей установки счетчика строк

Конечные значения, помещаемые системой управления генератором отчетов в специальный регистр LINE-COUNTER (СЧЕТЧИК-СТРОК) после представления группы отчета определяются правилами результирующей установки счетчика строк.

3.10.3. Обозначение в таблице для фразы LINE NUMBER (НОМЕР СТРОКИ)

В столбце 1 таблицы правил представления используются следующие сокращенные обозначения для описания последовательности фраз LINE NUMBER (НОМЕР СТРОКИ), которые могут появиться в описании группы отчета:

(1) А — обозначает одну или несколько фраз LINE NUMBER (НОМЕР СТРОКИ) без варианта NEXT PAGE (НА СЛЕДУЮЩЕЙ СТРАНИЦЕ), задающих абсолютное значение номера строки, появляющихся одна за другой в последовательности фраз LINE NUMBER (НОМЕР СТРОКИ) в статье описания группы отчета. В дальнейшем такие фразы будем называть абсолютными;

(2) О — обозначает одну или несколько последовательных фраз LINE NUMBER (НОМЕР СТРОКИ) в статье описания группы отчета, задающих относительное смещение номера строки; такие фразы будем называть относительными;

(3) СС — обозначает одну или несколько абсолютных последовательных фраз LINE NUMBER (НОМЕР СТРОКИ) в статье описания группы отчета, первая из которых и только она содержит вариант NEXT PAGE (НА СЛЕДУЮЩЕЙ СТРАНИЦЕ).

Появление в одной строке столбца из двух обозначений указывает, что в последовательности фраз COLUMN NUMBER (НОМЕР СТОЛБЦА) имеются обе указанные последовательности. Например «А О» указывает, что в статье описания группы отчета за последовательностью фраз типа «А» (определенных в правиле 1 выше) непосредственно следует последовательность фраз типа «О» (определенных в правиле 2).

3.10.4. Область применимости правил для фразы LINE NUMBER (НОМЕР СТРОКИ)

Все правила представления применимые к последовательности «А О», применимы также к последовательности «А».

Все правила представления, применимые к последовательности «СС О», применимы также к последовательности «СС».

3.10.5. Понятие сохраняемой позиции следующей группы

Сохраняемая позиция следующей группы представляет данное, доступное только системе управления генератором отчетов. Если абсолютная фраза NEXT GROUP (СЛЕДУЮЩАЯ ГРУППА) указывает значение вертикального расположения, которое не может быть достигнуто на текущей странице, СУГО запоминает это значение в сохраняемой позиции следующей группы. После осущест-

GOST EXPERT ГОСТ 22558-89,

вления перехода к следующей странице СУГО располагает следующую группу тела отчета, используя сохраняемое значение следующей группы.

3.10.6. Правила представления группы типа заголовок отчета

В табл. 1 указаны соответствующие правила представления для всех допустимых комбинаций фраз LINE NUMBER (НОМЕР СТРОКИ) и NEXT GROUP (СЛЕДУЮЩАЯ ГРУППА) в группе отчета типа заголовок отчета.

(1) Правило верхней границы

Номер первой строки, на которой может быть представлена группа типа заголовок отчета является номером строки, указанным в варианте HEADING (ЗАГОЛОВОК) фразы PAGE (РАЗМЕР СТРАНИЦЫ).

(2) Правила нижней границы:

а) номер последней строки, на которой может быть представлена группа типа заголовок отчета, является номером строки, полученным в результате вычитания единицы из значения целого-3 в варианте FIRST DETAIL (ПЕРВЫЙ ФРАГМЕНТ) фразы PAGE (РАЗМЕР СТРАНИЦЫ);

б) номер последней строки, на которой может быть представлена группа типа заголовок отчета, равняется номеру строки, указанному целым-1 фразы PAGE (РАЗМЕР СТРАНИЦЫ).

(3) Правила позиции первой печатаемой строки:

а) первая печатаемая строка группы типа заголовок отчета представляется на строке с номером, указанным целым в соответствующей фразе LINE NUMBER (НОМЕР СТРОКИ);

б) первая печатаемая строка группы типа заголовок отчета представляется на строке с номером, полученным в результате сложения целого, указанного в первой фразе LINE NUMBER (НОМЕР СТРОКИ), и значения, полученного в результате вычитания единицы из целого-2 в варианте HEADING (ЗАГОЛОВОК) фразы PAGE (РАЗМЕР СТРАНИЦЫ);

в) группа типа заголовок отчета не представляется;

г) первая печатаемая строка группы типа заголовок отчета представляется на строке с номером, равным значению результата сложения содержимого соответствующего LINE-COUNTER (СЧЕТЧИК-СТРОК) (в данном случае равного нулю) с целым, указанным первой фразой LINE NUMBER (НОМЕР СТРОКИ).

(4) Правила следующей группы:

GOST EXPERT ГОСТ 22558-89,

Таблица 1

**		Применяемые правила***						
		Фраза PAGE (РАЗМЕР СТРАНИЦЫ) указана					Фраза PAGE (РАЗМЕР СТРАНИЦЫ) опущена	
Последовательность фраз LINE NUMBER (НОМЕР СТРОКИ)*	Фраза NEXT GROUP (СЛЕДУЮЩАЯ ГРУППА)	Верхняя граница	Нижняя граница	Позиция первой печатаемой строки	Позиция следующей группы	Результирующая установка LINE-COUNTER (СЧЕТЧИК-СТРОК)	Позиция первой печатаемой строки	Результирующая установка LINE-NUMBER (СЧЕТЧИК-СТРОК)
1	2	3	4	5	6	7	8	9
A O	Абсолютная	1	2а	3а	4а	5а	Запрещенная комбинация+	
A O	Относительная	1	2а	3а	4б	5б	Запрещенная комбинация+	
A O	NEXT PAGE (НА СЛЕДУЮЩЕЙ СТРАНИЦЕ)	1	2б	3а	4в	5в	Запрещенная комбинация+	
A O	—	1	2а	3а	—	5г	Запрещенная комбинация+	
O	Абсолютная	1	2а	3б	4а	5а	Запрещенная комбинация++	
O	Относительная	1	2а	3б	4б	5б	3г	5б
O	NEXT PAGE (НА СЛЕДУЮЩЕЙ СТРАНИЦЕ)	1	2б	3б	4в	5в	Запрещенная комбинация++	
O	—	1	2а	3б	—	5г	3г	5г
—	—	—	—	3в	—	5д	3в	5д

* Описание сокращений, используемых в столбце 1, см. п. 3.10.3.

** Знак «—» в столбцах 1 и 2 указывает на отсутствие названных фраз в статье описания группы отчета.

*** Знак « » в столбцах применяемых правил означает отсутствие правил представления для данной комбинации фраз LINE NUMBER (НОМЕР СТРОКИ) и NEXT GROUP (СЛЕДУЮЩАЯ ГРУППА).

\+ См. п. 3.15, фраза LINE NUMBER (НОМЕР СТРОКИ).

++ См. п. 3.16, фраза NEXT GROUP (СЛЕДУЮЩАЯ ГРУППА).

а) целое, указанное фразой NEXT GROUP (СЛЕДУЮЩАЯ ГРУППА), должно быть больше номера строки, на которой представляется последняя печатаемая строка группы типа заголовок отчета. Кроме того, целое, указанное фразой NEXT GROUP (СЛЕДУЮЩАЯ ГРУППА), должно быть меньше номера строки, указанного значением целого-3 варианта FIRST DETAIL (ПЕРВЫЙ ФРАГМЕНТ) фразы PAGE (РАЗМЕР СТРАНИЦЫ);

б) сумма целого, указанного фразой NEXT GROUP (СЛЕДУЮЩАЯ ГРУППА), и номер строки, на которой представляется последняя печатаемая строка группы типа заголовок отчета, должна быть меньше значения целого-3 в варианте FIRST DETAIL (ПЕРВЫЙ ФРАГМЕНТ) фразы PAGE (РАЗМЕР СТРАНИЦЫ);

в) фраза NEXT GROUP NEXT PAGE (СЛЕДУЮЩАЯ ГРУППА НА СЛЕДУЮЩЕЙ СТРАНИЦЕ) означает, что только группа типа заголовок отчета представляется на первой странице отчета. Система управления генератором отчетов не вырабатывает никакой другой группы отчета во время позиционирования на первой странице отчета.

(5) Правила результирующей установки счетчика строк:

а) после представления группы типа заголовок отчета система управления генератором отчетов помещает целое, указанное фразой NEXT GROUP (СЛЕДУЮЩАЯ ГРУППА), в LINE-COUNTER (СЧЕТЧИК-СТРОК) в качестве результирующей установки счетчика строк;

б) после представления группы типа заголовок отчета система управления генератором отчетов в качестве результирующей установки счетчика строк помещает сумму целого, заданного фразой NEXT GROUP (СЛЕДУЮЩАЯ ГРУППА) и номера строки, на которой представлена последняя печатаемая строка группы типа заголовок отчета;

в) после представления группы типа заголовок отчета система управления генератором отчетов в качестве результирующей установки счетчика строк помещает нуль в счетчик строк;

г) после представления группы типа заголовок отчета результирующей установкой счетчика строк является номер строки, на которой предстаавлена последняя печатаемая строка группы заголовок отчета;

д) при обработке непечатаемой группы отчета значение LINE-COUNTER (СЧЕТЧИК-СТРОК) не изменяется.

3.10.7. Правила представления группы типа заголовок страницы

В табл. 2 приведены правила представления группы типа заголовок страницы при всех допустимых комбинациях фраз LINE NUMBER (НОМЕР СТРОКИ) и NEXT GROUP (СЛЕДУЮЩАЯ ГРУППА).

GOST EXPERT ГОСТ 22558-89,

Правила представления группы отчета типа заголовок страницы следующие.

(1) Правило верхней границы

Если группа типа заголовок отчета должна быть представлена на той же странице, на которой представляется группа отчета типа заголовок страницы, то номер первой строки, на которой может быть представлена группа типа заголовок страницы, на единицу больше результирующего регистра LINE-COUNTER (СЧЕТЧИК-СТРОК), установленного после представления заголовка отчета.

В противном случае номер первой строки, на которой может быть представлена группа отчета типа заголовок страницы, определяется вариантом HEADING (ЗАГОЛОВОК) фразы PAGE (РАЗМЕР СТРАНИЦЫ).

(2) Правило нижней границы

Номер последней строки, на которой может быть представлена группа отчета типа заголовок страницы, это номер, полученный в результате вычитания единицы от значения целого-3, указанного в варианте FIRST DETAIL (ПЕРВЫЙ ФРАГМЕНТ) фразы PAGE (РАЗМЕР СТРАНИЦЫ).

Таблица 2

**		Применяемые правила***				
		Фраза PAGE (РАЗМЕР СТРАНИЦЫ) указана****				
Последовательность фраз LINE NUMBER (НОМЕР СТРОКИ)	Фраза NEXT GROUP (СЛЕДУЮЩАЯ ГРУППА)	Верхняя граница	Нижняя граница	Позиция первой печатаемой строки	Следующая группа	Результирующая установка LINE-COUNTER (СЧЕТЧИК-СТРОК)
1	2	3	4	5	6	7
A O	—	1	2	3а	—	4а
O	—	1	2	3б	—	4а
—	—	—	—	3в	—	4б

* Описание сокращений, используемых в столбце 1, см. п. 3.10.3.

** Знак «—» в столбце 1 или 2 указывает на отсутствие названной фразы в статье описания группы отчета.

*** Знак «—» в столбце применяемых правил указывает на отсутствие названного правила для данной комбинации фраз LINE NUMBER (НОМЕР СТРОКИ) и NEXT GROUP (СЛЕДУЮЩАЯ ГРУППА).

**** Если в статье описания отчета опущена фраза PAGE (РАЗМЕР СТРАНИЦЫ), группа отчета заголовок страницы не может быть определена (п. 3.20 настоящей части).

(3) Правила первой печатаемой строки:

а) первая печатаемая строка группы отчета типа заголовок страницы представляется на строке, номер которой указан целым в ее фразе LINE NUMBER (НОМЕР СТРОКИ);

GOST EXPERT ГОСТ 22558-89,

б) если группа отчета типа заголовок отчета будет представлена на той же странице, на которой представляется группа отчета типа заголовок страницы, но номер строки, на которой представляется первая печатаемая строка группы типа заголовок страницы, определяется как сумма результирующего значения LINE-COUNTER (СЧЕТЧИК-СТРОК), установка которого выполнена при представлении группы типа заголовок отчета, и целого в первой фразе LINE NUMBER (НОМЕР СТРОКИ) группы отчета типа заголовок страницы.

Иначе номер строки, на которой представляется первая печатаемая строка группы отчета типа заголовок страницы, определяется как сумма целого первой фразы LINE NUMBER (НОМЕР СТРОКИ) группы типа заголовок страницы и уменьшенного на единицу значения целого-2 в варианте HEADING (ЗАГОЛОВОК) фразы PAGE (РАЗМЕР СТРАНИЦЫ).

(4) Правила результирующей установки LINE-COUNTER (СЧЕТЧИК-СТРОК):

а) результирующая установка LINE-COUNTER (СЧЕТЧИК-СТРОК) — номер строки, на которой представлена последняя печатаемая строка группы отчета типа заголовок страницы;

в) LINE-COUNTER (СЧЕТЧИК-СТРОК) не изменяется при обработке непечатаемой группы отчета.

3.10.8. Правила представления группы тела отчета

В табл. 3 приведены соответствующие правила представления для всех допустимых комбинаций фраз LINE NUMBER (НОМЕР СТРОКИ) и NEXT GROUP (СЛЕДУЮЩАЯ ГРУППА) в группах типа управляемый заголовок, фрагмент и управляемая концовка. Правила представления групп тела отчета следующие.

(1) Правило верхней границы

Номер первой строки, на которой может быть представлена группа тела отчета, указывается вариантом FIRST DETAIL (ПЕРВЫЙ ФРАГМЕНТ) фразы PAGE (РАЗМЕР СТРАНИЦЫ).

(2) Правила нижней границы

Номер последней строки, на которой может быть представлена группа отчета типа управляемый заголовок или фрагмент, указывается вариантом LAST DETAIL (ПОСЛЕДНИЙ ФРАГМЕНТ) фразы PAGE (РАЗМЕР СТРАНИЦЫ).

Номер последней строки, на которой может быть представлена группа типа управляемых концовка, указывается вариантом FOOTING (КОНЦОВКА) фразы PAGE (РАЗМЕР СТРАНИЦЫ).

(3) Правила проверки вместимости:

а) если значение LINE-COUNTER (СЧЕТЧИК-СТРОК) меньше целого, указанного первой из фраз LINE NUMBER (НОМЕР СТРОКИ), задающих абсолютное значение, группа тела отчета будет представлена на странице, на которой в данный момент представляется отчет.

GOST EXPERT ГОСТ 22558-89,

Правила представления группы тела отчета Таблица 3

**		Применяемые правила***							
		Фраза PAGE (РАЗМЕР СТРАНИЦЫ) указана						Фраза PAGE (РАЗМЕР СТРАНИЦЫ) опущена	
Последовательность фраз LINE NUMBER (НОМЕР СТРОКИ)*	Фраза NEXT GROUP (СЛЕДУЮЩАЯ ГРУППА)	Верхняя граница	Нижняя граница	Проверка вместимости	Позиция первой печатаемой строки	Следующая группа	Результирующая установка LINE-COUNTER (СЧЕТЧИК-СТРОК)	Позиция первой печатаемой строки	Результирующая установка LINE-COUNTER (СЧЕТЧИК-СТРОК)
1	2	3	4	5	6	7	8	9	10
A O	Абсолютная	1	2	3а	4а	5	6а	Запрещенная комбинация+	
A O	Относительная	1	2	3а	4а	—	6б	Запрещенная комбинация+	
A O	NEXT PAGE (НА СЛЕДУЮЩЕЙ СТРАНИЦЕ)	1	2	3а	4а	—	6в	Запрещенная комбинация+	
A O	—	1	2	3а	4а	—	6г	Запрещенная комбинация+	
O	Абсолютная	1	2	3б	4б	5	6а	Запрещенная комбинация++	
O	Относительная	1	2	3б	4б	—	6б	4г	6е
O	NEXT PAGE (НА СЛЕДУЮЩЕЙ СТРАНИЦЕ)	1	2	3б	4б	—	6в	Запрещенная комбинация+++	
O	—	1	2	3б	4б	—	6г	4г	6г
CC O	Абсолютная	1	2	3в	4а	5	6а	Запрещенная комбинация+	
CC O	Относительная	1	2	3в	4а	—	6б	Запрещенная комбинация+	
CC O	NEXT PAGE (НА СЛЕДУЮЩЕЙ СТРАНИЦЕ)	1	2	3в	4а	—	6в	Запрещенная комбинация+	
CC O	—	1	2	3в	4а	—	6г	Запрещенная комбинация+	
—	—	—	—	—	4в	—	6д	4в	6д

* Описание сокращений, используемых в столбце 1, см. п. 3.10.3, обозначение фразы LINE NUMBER (НОМЕР СТРОКИ).

** Знак «—» в столбце 1 или 2 указывает, что названная фраза отсутствует в статье описания группы отчета.

*** Знак «—» в столбцах применяемых правил обозначает отсутствие названного правила для данной комбинации фраз LINE NUMBER (НОМЕР СТРОКИ) и NEXT GROUP (СЛЕДУЮЩАЯ ГРУППА).

* См. п. 3.15, фраза LINE NUMBER (НОМЕР СТРОКИ).

** См. п. 3.16, фраза NEXT GROUP (СЛЕДУЮЩАЯ ГРУППА).

В противном случае СУГО обеспечивает переход на следующую страницу. После обработки группы типа заголовок страницы (если такая группа определена), СУГО определяет, была ли установлена сохраняемая позиция следующей группы при представлении последней группы тела отчета на предыдущей странице (см. далее правило 6а). Если сохраняемая позиция следующей группы не была установлена, группа тела отчета будет представлена на странице, на которой в данный момент представляется отчет. Если же сохраняемая позиция следующей группы была установлена, СУГО присваивает регистру LINE-COUNTER (СЧЕТЧИК-СТРОК) значение сохраняемой позиции следующей группы, устанавливает в нуль значение сохраняемой позиции следующей группы и повторно применяет правило 3а проверки вместимости;

б) если некоторая группа тела представляется на текущей странице, СУГО вычисляет пробную сумму в рабочей области. Пробная сумма вычисляется путем сложения значения LINE-COUNTER (СЧЕТЧИК-СТРОК) и всех целых, указанных фразами LINE NUMBER (НОМЕР-СТРОКИ) группы отчета. Если полученная сумма не больше определенной для данной группы нижней границы ее представления, группа отчета представляется на текущей странице. Если же полученная сумма превышает определенную для данной группы нижнюю границу, СУГО обеспечивает переход к следующей странице. После обработки группы типа заголовок страницы (если таковой определен), СУГО вновь применяет правило 3б проверки вместимости.

Если на текущей странице отчета не была еще представлена ни одна группа тела отчета, СУГО определяет, было ли установлено значение сохраняемой позиции следующей группы при представлении последней группы тела отчета на предыдущей странице (см. ниже правило 6а результирующей установки LINE-COUNTER (СЧЕТЧИК-СТРОК).

Если значение сохраняемой позиции следующей группы не было установлено, группа тела отчета будет представляться на текущей странице отчета.

Если значение сохраняемой позиции следующей группы было установлено, СУГО помещает значение сохраняемой позиции следующей группы LINE-COUNTER (СЧЕТЧИК-СТРОК), устанавли-

GOST EXPERT ГОСТ 22558-89,

вает в нуль значение сохраняемой позиции следующей группы и повторно вычисляет пробную сумму в рабочей области.

Пробная сумма вычисляется сложением сумм значения LINE-COUNTER (СЧЕТЧИК-СТРОК) с единицей и целыми, указанными во фразе LINE NUMBER (НОМЕР СТРОКИ), за исключением первой фразы группы тела отчета. Если полученная сумма не превосходит определенную для данной группы нижнюю границу, тело группы отчета представляется на текущей странице. Если же полученная сумма больше определенной для данной группы нижней границы, СУГО обеспечивает переход на следующую страницу. После обработки группы типа заголовок страницы (если таковая определена) СУГО представляет группу тела отчета на этой странице;

в) если на текущей странице была уже представлена группа тела отчета, СУГО осуществляет переход к следующей странице. После обработки группы заголовок страницы (если таковая определена) СУГО повторно применяет правило 3в вместимости. Если на текущей странице отчета не представлена ни одна группа тела отчета, СУГО определяет, было ли установлено при представлении последней группы тела отчета на предыдущей странице значение сохраняемой позиции следующей группы (см. правило 6а заключительной установки LINE-COUNTER (СЧЕТЧИК-СТРОК). Если значение сохраняемой позиции следующей группы не установлено, группа тела отчета представляется на текущей странице отчета. Если значение сохраняемой позиции следующей группы установлено, СУГО помещает установленное значение в LINE-COUNTER (СЧЕТЧИК-СТРОК) и переустанавливает значение сохраняемой позиции следующей группы в нуль. Если значение LINE-COUNTER (СЧЕТЧИК-СТРОК) меньше целого, указанного в варианте FIRST (ПЕРВЫЙ) абсолютной фразы LINE NUMBER (НОМЕР СТРОКИ), группа тела отчета представляется на текущей странице отчета. В противном случае СУГО осуществляет переход к следующей странице. После обработки группы типа заголовок страницы (если таковая определена) СУГО представляет группу тела отчета на этой странице.

(4) Правила позиции первой печатаемой строки:

а) первая печатаемая строка группы тела отчета представляется на строке с номером, указанным целым соответствующей фразы LINE NUMBER (НОМЕР СТРОКИ);

б) если значение LINE-COUNTER (СЧЕТЧИК-СТРОК) равно или больше номера строки, указанного вариантом FIRST DETAIL (ПЕРВЫЙ ФРАГМЕНТ) фразы PAGE (РАЗМЕР СТРАНИЦЫ), и на текущей странице отчета еще не была представлена ни одна группа тела отчета, первая печатаемая строка данной группы тела отчета представляется на строке, непосредственно следующей за

GOST EXPERT ГОСТ 22558-89,

строкой, номер которой указан значением LINE-COUNTER (СЧЕТЧИК-СТРОК).

Если значение LINE-COUNTER (СЧЕТЧИК-СТРОК) равно или больше номера строки, указанного вариантом FIRST DETAIL (ПЕРВЫЙ ФРАГМЕНТ) фразы PAGE (РАЗМЕР СТРАНИЦЫ), и если на текущей странице отчета уже была представлена некоторая группа тела отчета, то первая печатаемая строка данной группы тела отчета представляется на строке, номер которой равен сумме значения LINE-COUNTER (СЧЕТЧИК-СТРОК) и целого первой фразы LINE NUMBER (НОМЕР СТРОКИ) текущей группы тела отчета.

Если значение LINE-COUNTER (СЧЕТЧИК-СТРОК) меньше номера строки, указанного FIRST DETAIL (ПЕРВЫЙ ФРАГМЕНТ) фразы PAGE (РАЗМЕР СТРАНИЦЫ), первая печатаемая строка группы тела отчета представляется на строке, номер которой указан вариантом FIRST DETAIL (ПЕРВЫЙ ФРАГМЕНТ);

в) группа тела отчета не представляется;

г) номер строки, на которой представляется первая печатаемая строка, определяется суммой содержимого LINE-COUNTER (СЧЕТЧИК-СТРОК) и целого, указанного первой фразой LINE NUMBER (НОМЕР СТРОКИ).

(5) Правило следующей группы

Абсолютное значение номера строки, указанное целым фразы NEXT GROUP (СЛЕДУЮЩАЯ ГРУППА), должно определять номер строки, не меньший значения, указанного во фразе FIRST DETAIL (ПЕРВЫЙ ФРАГМЕНТ) фразы PAGE (РАЗМЕР СТРАНИЦЫ), и не больший значения, указанного во фразе FOOTING (КОНЦОВКА) той же фразы PAGE (РАЗМЕР СТРАНИЦЫ).

(6) Правила результирующей установки LINE-COUNTER (СЧЕТЧИК-СТРОК):

а) если представленная последней группа тела отчета является управляемой концовкой, не связанной с наивысшим уровнем, на котором СУГО было обнаружено прерывание управления, то результирующее значение LINE-COUNTER (СЧЕТЧИК-СТРОК) устанавливается равным номеру строки, на которой была представлена последняя печатаемая строка группы управляемая концовка.

Во всех остальных случаях СУГО сравнивает номер строки, на которой была представлена последняя печатаемая строка группы тела, с целым, указанным фразой NEXT GROUP (СЛЕДУЮЩАЯ ГРУППА). Если сравниваемый номер строки меньше этого целого, СУГО устанавливает LINE-COUNTER (СЧЕТЧИК-СТРОК) равным значению, представленному целым во фразе NEXT GROUP (СЛЕДУЮЩАЯ ГРУППА). Если же сравниваемый номер строки больше или равен целому, указанному фразой NEXT GROUP (СЛЕДУЮЩАЯ ГРУППА), СУГО помещает в LINE-COUNTER

GOST EXPERT ГОСТ 22558-89,

(СЧЕТЧИК СТРОК) значение, равное номеру строки, указанному в варианте FOOTING (КОНЦОВКА) фразы PAGE (РАЗМЕР СТРАНИЦЫ); кроме того значение сохраняемой позиции следующей группы устанавливается равным целому, указанному фразой NEXT GROUP (СЛЕДУЮЩАЯ ГРУППА);

б) если представленная последней группа тела отчета является управляемой концовкой и не связана с наивысшим уровнем, на котором СУГО было обнаружено прерывание управления, то результирующее значение LINE-COUNTER (СЧЕТЧИК-СТРОК) устанавливается равным номеру строки, на которой была представлена последняя печатаемая строка группы управляемая концовка.

Во всех остальных случаях СУГО вычисляет в рабочем поле пробную сумму. Пробная сумма является суммой целого, указанного фразой NEXT GROUP (СЛЕДУЮЩАЯ ГРУППА), и номера строки, на которой представлена последняя печатаемая строка группы тела отчета. Если пробная сумма меньше номера строки, указанного в варианте FOOTING (КОНЦОВКА) фразы PAGE (РАЗМЕР СТРАНИЦЫ), СУГО помещает эту сумму в LINE-COUNTER (СЧЕТЧИК-СТРОК) в качестве его результирующего значения. Если пробная сумма равна или больше номера строки, указанного в варианте FOOTING (КОНЦОВКА) фразы PAGE (РАЗМЕР СТРАНИЦЫ), то СУГО помещает в LINE-COUNTER (СЧЕТЧИК-СТРОК) в качестве его результирующего значения значение номера строки, указанного в варианте FOOTING (КОНЦОВКА) фразы PAGE (РАЗМЕР СТРАНИЦЫ);

в) если представленная группа отчета является группой отчета управляемая концовка и она не связана с наивысшим уровнем, на котором СУГО было обнаружено прерывание управления, результирующим значением LINE-COUNTER (СЧЕТЧИК-СТРОК) является номер строки, на которой была представлена последняя печатаемая строка группы управляемая концовка.

Во всех остальных случаях СУГО в качестве результирующего значения помещает в LINE-COUNTER (СЧЕТЧИК-СТРОК) значение номера строки, указанного в варианте FOOTING (КОНЦОВКА) фразы PAGE (РАЗМЕР СТРАНИЦЫ);

г) результирующим значением LINE-COUNTER (СЧЕТЧИК-СТРОК) является номер строки, на которой была представлена последняя печатаемая строка группы тела отчета;

д) при обработке непечатаемой группы тела отчета LINE-COUNTER (СЧЕТЧИК-СТРОК) не изменяется;

е) если представленная последней группа тела отчета является управляемой концовкой, но она не связана с наивысшим уровнем, на котором СУГО было обнаружено прерывание управления, то результирующим значением LINE-COUNTER (СЧЕТЧИК-СТРОК) является значение, равное номеру строки, на которой бы-

GOST EXPERT ГОСТ 22558-89,

ла представлена последняя печатаемая строка управляемой концовки.

Во всех остальных случаях в качестве результирующего значения СУГО помещает в LINE-COUNTER (СЧЕТЧИК-СТРОК) сумму номера строки, на которой была представлена последняя печатаемая строка, и целого, указанного фразой NEXT GROUP (СЛЕДУЮЩАЯ ГРУППА).

3.10.9. Правила представления концовки страницы

В табл. 4 приведены правила представления группы типа концовка страницы для всех допустимых комбинаций фраз LINE NUMBER (НОМЕР СТРОКИ) и NEXT GROUP (СЛЕДУЮЩАЯ ГРУППА).

Таблица 4

Правила представления концовки страницы

**		Применяемые правила**				
		Фраза PAGE (РАЗМЕР СТРАНИЦЫ) указана****				
Последовательность фраз LINE NUMBER (НОМЕР СТРОКИ)	Фраза NEXT GROUP (СЛЕДУЮЩАЯ ГРУППА)	Верхняя граница	Нижняя граница	Позиция первой печатаемой строки	Следующая группа	Результирующая установка LINE-COUNTER (СЧЕТЧИК-СТРОК)
1	2	3	4	5	6	7
A O	Абсолютная	1	2	3а	4а	5а
A O	Относительная	1	2	3а	4б	5б
A O	—	1	2	3а	—	5в
				3б	—	5г

* Описание сокращений, используемых в столбце 1 см. п. 3.10.3 настоящей части.

** Знак «—» в столбце 1 или 2 указывает на отсутствие фразы в статье описания группы отчета.

*** Знак «—» в столбце применяемых правил обозначает отсутствие названного правила для данной комбинации фраз LINE NUMBER (НОМЕР СТРОКИ) и NEXT GROUP (СЛЕДУЮЩАЯ ГРУППА).

**** Если в статье описания отчета фраза PAGE (РАЗМЕР СТРАНИЦЫ) опущена, группа концовка страницы не может быть определена (п. 3.20 настоящей части).

Правила представления концовки страницы следующие.

(1) Правило верхней границы

Номер первой строки, на которой может быть представлена группа типа концовка страницы, равен увеличенному на единицу значению целого-5, указанного вариантом FOOTING (КОНЦОВКА) фразы PAGE (РАЗМЕР СТРАНИЦЫ).

(2) Правило нижней границы

GOST EXPERT ГОСТ 22558-89,

Номер последней строки, на которой может быть представлена группа типа концовка страницы, равен целому-1, указанному фразой PAGE (РАЗМЕР СТРАНИЦЫ).

(3) Правила позиции первой печатаемой строки:

а) первая печатаемая строка группы типа концовка страницы представляется на строке, номер которой определяется ее фразой LINE NUMBER (НОМЕР СТРОКИ);

б) группа типа концовка страницы не представляется.

(4) Правила следующей группы:

а) целое, указанное фразой NEXT GROUP (СЛЕДУЮЩАЯ ГРУППА), должно быть больше номера строки, на которой представляется последняя печатаемая строка группы типа концовка страницы. Кроме того, целое, указанное фразой NEXT GROUP (СЛЕДУЮЩАЯ ГРУППА) не должно превышать номера строки, указанного целым-1 фразы PAGE (РАЗМЕР СТРАНИЦЫ);

б) сумма целого, указанного фразой NEXT GROUP (СЛЕДУЮЩАЯ ГРУППА), и номера строки, на которой представляется последняя печатаемая строка группы типа концовка страницы, не должна превышать целого-1, указанного фразой PAGE (РАЗМЕР СТРАНИЦЫ).

(5) Правила результирующей установки LINE-COUNTER (СЧЕТЧИК СТРОК):

а) после представления группы типа концовка страницы СУГО помещает в LINE-COUNTER (СЧЕТЧИК-СТРОК) в качестве результирующего значения целое, указанное фразой NEXT GROUP (СЛЕДУЮЩАЯ ГРУППА);

б) после представления группы концовка страницы СУГО помещает в качестве результирующего значения в LINE-COUNTER (СЧЕТЧИК-СТРОК) значение, равное сумме целого, указанного фразой NEXT GROUP (СЛЕДУЮЩАЯ ГРУППА), и номера строки, на которой была представлена последняя печатаемая строка группы типа концовка страницы;

в) после представления группы типа концовка страницы результирующее значение LINE-COUNTER (СЧЕТЧИК-СТРОК) равно номеру строки, на которой была представлена последняя печатаемая строка группы типа концовка страницы;

г) при обработке непечатаемой группы отчета LINE-COUNTER (СЧЕТЧИК-СТРОК) не изменяется.

3.10.10. Правила представления концовки отчета

В табл. 5 приведены соответствующие правила представления группы типа концовка отчета при всех допустимых комбинациях LINE NUMBER (НОМЕР СТРОКИ) и NEXT GROUP (СЛЕДУЮЩАЯ ГРУППА).

GOST EXPERT ГОСТ 22558-89,

Таблица 5

**		Применяемые правила***						
		Фраза PAGE (РАЗМЕР СТРАНИЦЫ) указана					Фраза PAGE (РАЗМЕР СТРАНИЦЫ) опущена	
Последовательность фраз LINE NUMBER (НОМЕР СТРОКИ)*	Фраза NEXT GROUP (СЛЕДУЮЩАЯ ГРУППА)	Верхняя граница	Нижняя граница	Позиция первой печатаемой строки	Следующая группа	Результирующая установка LINE-COUNTER (СЧЕТЧИК-СТРОК)	Позиция первой печатаемой строки	Результирующая установка LINE-COUNTER (СЧЕТЧИК-СТРОК)
А О	—	1а	2	3а	—	4а	Запрещенная комбинация+	
О	—	1а	2	3б	—	4а	3г	4а
СС О	—	1б	2	3в	—	4а	Запрещенная комбинация+	
—	—	—	—	3д	—	4б	3д	4б

* Описание сокращений, используемых в столбце 1, см. п. 3.10.3.

** Знак «—» в столбце 1 и 2 указывает, что названная фраза отсутствует в статье описания группы отчета.

*** Знак «—» в столбце применяемых правил обозначает отсутствие названного правила для данной комбинации фраз LINE NUMBER (НОМЕР СТРОКИ) и NEXT GROUP (СЛЕДУЮЩАЯ ГРУППА).

\+ См. п. 3.15 настоящей части.

ГОСТ 22558-89

Правила представления концовки отчета следующие.

(1) Правила верхней границы:

а) если группа отчета типа концовка страницы должна быть представлена на текущей странице отчета, номер первой строки, на которой она может быть представлена, должен быть на единицу больше результирующего значения LINE-COUNTER (СЧЕТЧИК-СТРОК) после представления группы типа концовка страницы.

В остальных случаях номер первой строки, на которой может быть представлена группа типа концовка отчета, устанавливается равным увеличенному на единицу значению целого-5 фразы PAGE (РАЗМЕР СТРАНИЦЫ);

б) номер первой строки, на которой может быть представлена группа типа концовка отчета, равен номеру строки, указанному в варианте HEADING (ЗАГОЛОВОК) фразы PAGE (РАЗМЕР СТРАНИЦЫ).

(2) Правило нижней границы

Номер последней строки, на которой может быть представлена группа типа концовка отчета определяется целым-1 фразы PAGE (РАЗМЕР СТРАНИЦЫ).

(3) Правила позиции первой печатаемой строки:

а) первая печатаемая строка группы типа концовка отчета представляется на строке, указанной целым фразы LINE NUMBER (НОМЕР СТРОКИ) для концовки отчета;

б) если группа типа концовка отчета должна быть представлена на текущей странице, то номер строки, на которой представляется ее первая печатаемая строка группы концовка отчета, определяется суммой результирующего значения LINE-COUNTER (СЧЕТЧИК-СТРОК) после представления группы концовка отчета и целого, указанного в первой из фраз LINE NUMBER (НОМЕР СТРОКИ) в описании группы отчета типа концовка отчета. В остальных случаях номер строки, на которой представляется первая печатаемая строка группы концовка отчета, определяется суммой целого в первой фразе LINE NUMBER (НОМЕР СТРОКИ) группы отчета типа REPORT FOOTING (КОНЦОВКА ОТЧЕТА) и номера строки, указанного значением целого-5 варианта FOOTING (КОНЦОВКА) фразы PAGE (РАЗМЕР СТРАНИЦЫ);

в) фраза NEXT PAGE (НА СЛЕДУЮЩЕЙ СТРАНИЦЕ) в первой из абсолютных фраз LINE NUMBER (НОМЕР СТРОКИ) указывает, что группа типа концовка отчета представляется на отдельной странице, на которой никакая другая группа не представлена. Первая печатаемая строка группы типа концовка отчета представляется на строке, номер которой указан целым фразы LINE NUMBER (НОМЕР СТРОКИ) группы отчета типа концовка отчета;

GOST EXPERT ГОСТ 22558-89,

г) номер строки, на которой представляется первая печатаемая строка группы типа концовка отчета, определяется суммой целого первой фразы LINE NUMBER (НОМЕР СТРОКИ) и значения LINE-COUNTER (СЧЕТЧИК-СТРОК);

д) группа типа концовка отчета не представляется.

(4) Правила результирующей установки LINE-COUNTER (СЧЕТЧИК-СТРОК):

а) результирующее значение LINE-COUNTER (СЧЕТЧИК-СТРОК) равно номеру строки, на которой представляется последняя печатаемая строка группы концовка отчета;

б) при обработке непечатаемой группы отчета значение LINE-COUNTER (СЧЕТЧИК-СТРОК) не изменяется.

3.11. **Фраза COLUMN NUMBER (НОМЕР СТОЛБЦА)**

3.11.1. Назначение

Фраза COLUMN NUMBER (НОМЕР СТОЛБЦА) идентифицирует данное как печатаемое и указывает позицию данного на печатаемой строке.

3.11.2. Общий формат

COLUMN NUMBER целое-1

НОМЕР СТОЛБЦА целое-1

3.11.3. Синтаксические правила

(1) Фраза COLUMN NUMBER (НОМЕР СТОЛБЦА) может быть указана в группе отчета только на элементарном уровне. Если фраза COLUMN NUMBER (НОМЕР СТОЛБЦА) указана, она должна находиться в статье, содержащей фразу LINE NUMBER (НОМЕР СТРОКИ), или в статье, подчиненной статье, содержащей фразу LINE NUMBER (НОМЕР СТРОКИ).

(2) Печатаемые данные должны определяться в порядке возрастания номеров столбцов в пределах отдельной печатаемой строки таким образом, что каждое печатаемое данное занимает единственную последовательность позиций смежных литер.

3.11.4. Общие правила

(1) Фраза COLUMN NUMBER (НОМЕР СТОЛБЦА) указывает, что на печатаемой строке представляется объект фразы SOURCE (ИСТОЧНИК), объект фразы VALUE (ЗНАЧЕНИЕ) или счетчик суммы, определяемый фразой SUM (СУММА). Отсутствие фразы COLUMN NUMBER (НОМЕР СТОЛБЦА) указывает, что статья не будет представлена на печатаемой строке.

(2) Целое-1 указывает номер позиции на строке самой левой литеры печатаемого данного.

(3) СУГО проставляет пробелы на всех позициях печатаемой строки, не занятых печатаемыми данными.

(4) Самая левая позиция печатаемой строки соответствует номеру столбца 1.

3.12. **Фраза «имя-данного»**

3.12.1. Назначение

GOST EXPERT ГОСТ 22558-89,

Имя-данного определяет имя описываемого данного.

3.12.2. Общий формат

Имя-данного-1

3.12.3. Синтаксические правила

(1) В секции отчетов имя-данного-1 не обязательно определять в статье описания данного.

3.12.4. Общие правила

(1) В секции отчетов имя-данного-1 должно быть задано в следующих случаях:

а) если имя-данного-1 представляет группу отчета, на которую имеется ссылка в операторах GENERATE (ГЕНЕРИРОВАТЬ) или USE (ИСПОЛЬЗОВАТЬ) в разделе процедур;

б) если в разделе процедур или секции отчетов необходима ссылка на счетчик суммы;

в) если в варианте UPON (ДЛЯ) фразы SUM (СУММА) имеется ссылка на группу отчета типа фрагмент;

г) если имя-данного-1 необходимо для уточнения счетчика суммы.

3.13. Фраза GROUP INDICATE (ОПРЕДЕЛЯЕТ ГРУППУ)

3.13.1. Назначение

Фраза GROUP INDICATE (ОПРЕДЕЛЯЕТ ГРУППУ) указывает, что соответствующее печатаемое данное представляется только при первом появлении содержащей его группы после прерывания управления или продвижения страницы.

3.13.2. Общий формат

GROUP INDICATE

ОПРЕДЕЛЯЕТ ГРУППУ

3.13.3. Синтаксические правила

(1) Фраза GROUP INDICATE (ОПРЕДЕЛЯЕТ ГРУППУ) должна быть указана только в статье описания группы отчета типа фрагмент, определяющей печатаемое данное.

3.13.4. Общие правила

(1) Если фраза GROUP INDICATE (ОПРЕДЕЛЯЕТ ГРУППУ) указана, фразы SOURCE (ИСТОЧНИК) или VALUE (ЗНАЧЕНИЕ) и варианты этих фраз являются пробелами во всех случаях, кроме следующих:

а) при первом представлении в начале каждой группы типа фрагмент, или

б) при первом после каждого продвижения страницы представлении соответствующей группы типа фрагмент, или

в) при первом после каждого прерывания управления представлении соответствующей группы типа фрагмент.

(2) Если в статье описания отчета не указана ни фраза RAGE (РАЗМЕР СТРАНИЦЫ), ни фраза CONTROL (УПРАВЛЕНИЕ), печатаемое данное, определяемое фразой GROUP INDICATE (ОПРЕДЕЛЯЕТ ГРУППУ), представляется при первом после выпол-

GOST EXPERT ГОСТ 22558-89,

нения оператора INITIATE (НАЧАТЬ) представлении соответствующей группы отчета типа фрагмент. При последующих представлениях этой группы печатаемые данные, описанные с фразой SOURCE (ИСТОЧНИК) или VALUE (ЗНАЧЕНИЕ), представляются пробелами.

3.14. **Номер-уровня**

3.14.1. Назначение

Номер-уровня указывает позицию данного в иерархической структуре группы отчета.

3.14.2. Общий формат

номер-уровня

3.14.3. Синтаксические правила

(1) Номер-уровня необходим как первый элемент каждой статьи описания данного.

(2) Статьи описания данных, подчиненных статье RD (ОО), могут иметь номера уровней только от 01 до 49.

3.14.4. Общие правила

(1) Номер уровня 01 идентифицирует первую статью в группе отчета.

(2) Несколько статей уровня 01, подчиненных статье описания отчета с индикатором уровня RD (ОО), не представляют неявное переопределение одной и той же области.

3.15. **Фраза LINE NUMBER (НОМЕР СТРОКИ)**

3.15.1. Назначение

Фраза LINE NUMBER (НОМЕР СТРОКИ) определяет вертикальное расположение информации соответствующей группы отчета.

3.15.2. Общий формат

LINE NUMBER IS { целое-1 [ON NEXT PAGE] / PLUS целое-2 }

НОМЕР СТРОКИ { целое-1 [НА СЛЕДУЮЩЕЙ СТРАНИЦЕ] / ПЛЮС целое-2 }

3.15.3. Синтаксические правила

(1) Целое-1 и целое-2 должны быть не более чем трехзначными.

Целое-1 и целое-2 должны быть такими, чтобы любая представляемая строка группы отчета находилась в пределах вертикального подразделения страницы, определенного для данного типа группы отчета фразой PAGE (РАЗМЕР СТРАНИЦЫ) (см. п. 3.8 настоящей части).

Целое-2 может равняться нулю.

(2) В статье описания группы отчета статья, содержащая фразу LINE NUMBER (НОМЕР СТРОКИ), не должна содержать подчиненную статью с такой же фразой.

GOST EXPERT ГОСТ 22558-89,

(3) В статье описания группы отчета все фразы LINE NUMBER (НОМЕР СТРОКИ), задающие абсолютную позицию (абсолютные фразы), должны предшествовать всем фразам LINE NUMBER (НОМЕР СТРОКИ), задающим относительную позицию (относительным фразам).

(4) В статье описания группы отчетов следующие одна за другой абсолютные фразы LINE NUMBER (НОМЕР СТРОКИ) должны указывать целые в возрастающем порядке. Целые не обязательно должны быть последовательными числами.

(5) Если в статье описания отчета опущена фраза PAGE (РАЗМЕР СТРАНИЦЫ), то в статьях описания групп этого отчета могут указываться только относительные фразы LINE NUMBER (НОМЕР СТРОКИ).

(6) В статье описания группы отчета вариант NEXT PAGE (НА СЛЕДУЮЩЕЙ СТРАНИЦЕ) может указываться только один раз и только в первой фразе LINE NUMBER (НОМЕР СТРОКИ) статьи описания этой группы отчета.

(7) Фраза LINE NUMBER (НОМЕР СТРОКИ) с вариантом NEXT PAGE (НА СЛЕДУЮЩЕЙ СТРАНИЦЕ) может указываться только в описании групп тела отчета и в группе типа концовка отчета.

(8) Каждая статья, определяющая печатаемое данное (см. п. 3.11 настоящей части), должна или содержать фразу LINE NUMBER (НОМЕР СТРОКИ), или подчиняться статье, содержащей фразу LINE NUMBER (НОМЕР СТРОКИ).

(9) Первая фраза LINE NUMBER (НОМЕР СТРОКИ), указанная в группе отчета типа концовка страницы должна быть абсолютной фразой.

3.15.4. **Общие правила**

(1) Для установления каждой печатаемой строки группы отчета должна быть указана фраза LINE NUMBER (НОМЕР СТРОКИ).

(2) СУГО осуществляет вертикальное позиционирование, определяемое фразой LINE NUMBER (НОМЕР СТРОКИ), до представления установленной этой фразой печатаемой строки.

(3) Целое-1 указывает абсолютный номер строки. Абсолютный номер строки определяет номер строки, на которой представляется печатаемая строка.

(4) Целое-2 указывает относительный номер строки. Если относительная фраза LINE NUMBER (НОМЕР СТРОКИ) является не первой фразой LINE NUMBER (НОМЕР СТРОКИ) статьи описания группы отчета, то номер строки, на которой представляется печатаемая строка, определяется как сумма номера строки, на которой была представлена предыдущая печатаемая строка, и целого-2 из относительной фразы LINE NUMBER (НОМЕР СТРО-

GOST EXPERT ГОСТ 22558-89,

КИ). Если целое-2 равно нулю, строка будет печататься на той же строке, что и печатаемая строка.

Если относительная фраза LINE NUMBER (НОМЕР СТРОКИ) является первой фразой LINE NUMBER (НОМЕР СТРОКИ) в статье описания группы отчета, то номер строки, на которой представляется печатаемая строка, определяется по специальным правилам (см. п. 3.10 настоящей части).

(5) Фраза NEXT PAGE (НА СЛЕДУЮЩЕЙ СТРАНИЦЕ) указывает, что группа отчета будет представлена начиная с указанного номера строки на новой странице (см. п. 3.10 настоящей части).

3.16. Фраза NEXT GROUP (СЛЕДУЮЩАЯ ГРУППА)

3.16.1. Назначение

Фраза NEXT GROUP (СЛЕДУЮЩАЯ ГРУППА) задает информацию о вертикальном позиционировании на странице после представления последней строки группы отчета.

3.16.2. Общий формат

NEXT GROUP IS { целое-1 | PLUS целое-2 | NEXT PAGE }

СЛЕДУЮЩАЯ ГРУППА { целое-1 | ПЛЮС целое-2 | НА СЛЕДУЮЩЕЙ СТРАНИЦЕ }

3.16.3. Синтаксические правила

(1) Статья описания группы отчета не должна содержать фразу NEXT GROUP (СЛЕДУЮЩАЯ ГРУППА), если описание этой группы отчета не содержит хотя бы одну фразу LINE NUMBER (НОМЕР СТРОКИ).

(2) Целое-1 и целое-2 не должны быть больше чем трехзначными числами.

(3) Если в статье описания отчета опущена фраза PAGE (РАЗМЕР СТРАНИЦЫ), то в любой статье описания группы этого отчета может быть указана только относительная фраза NEXT GROUP (СЛЕДУЮЩАЯ ГРУППА).

(4) Вариант NEXT PAGE (НА СЛЕДУЮЩЕЙ СТРАНИЦЕ) фразы NEXT GROUP (СЛЕДУЮЩАЯ ГРУППА) не может быть указан в группе отчета типа концовка страницы.

(5) Фраза NEXT GROUP (СЛЕДУЮЩАЯ ГРУППА) не может быть указана для групп типа концовка отчета или заголовок страницы.

3.16.4. Общие правила

(1) Позиционирование страницы, указанное фразой NEXT GROUP (СЛЕДУЮЩАЯ ГРУППА), выполняется после пред-

GOST EXPERT ГОСТ 22558-89,

ставления группы отчета, в описании которой эта фраза указана (см. п. 3.10 настоящей части).

(2) Для определения нового значения LINE-COUNTER (СЧЕТЧИК-СТРОК) СУГО использует информацию о вертикальном позиционировании, задаваемую фразой NEXT GROUP (СЛЕДУЮЩАЯ ГРУППА) и фразами TYPE (ТИП) и PAGE (РАЗМЕР СТРАНИЦЫ), и текущие значение LINE-COUNTER (СЧЕТЧИК-СТРОК) (см. п. 3.10 настоящей части).

(3) Фраза NEXT GROUP (СЛЕДУЮЩАЯ ГРУППА) игнорируется СУГО, если она указана для группы отчета типа управляемая концовка, уровень которой отличается от наивысшего уровня, на котором обнаружено нарушение управления.

(4) Фраза NEXT GROUP (СЛЕДУЮЩАЯ ГРУППА) группы отчета относится к группе тела отчета, которая должна быть представлена следующей, и поэтому может влиять на расположение следующей группы тела при ее представлении. Фраза NEXT GROUP (СЛЕДУЮЩАЯ ГРУППА) группы типа заголовок отчета может влиять на расположение группы типа заголовок страницы при ее представлении. Фраза NEXT GROUP (СЛЕДУЮЩАЯ ГРУППА) группы типа концовка страницы может влиять на расположение группы типа концовка отчета при ее представлении (см. п. 3.10 настоящей части).

3.17. **Фраза SIGN (ЗНАК)**

3.17.1. Назначение

Фраза SIGN (ЗНАК) определяет позицию и представление знака числа, если имеется необходимость описать это явно.

3.17.2. Общий формат

[SIGN IS] $\left\{\begin{matrix}\text{LEADING}\\\text{TRAILING}\end{matrix}\right\}$ SEPARATE CHARACTER

[ЗНАК] $\left\{\begin{matrix}\text{ПЕРВЫЙ}\\\text{ПОСЛЕДНИЙ}\end{matrix}\right\}$ ОТДЕЛЬНО

3.17.3. Синтаксические правила

(1) Фраза SIGN (ЗНАК) может быть указана только в статье описания числового данного, шаблон которого содержит литеру S (3).

(2) Статьи описания числовых данных, к которым относится фраза SIGN (ЗНАК), должны быть описаны, явно или неявно, с USAGE IS DISPLAY (ДЛЯ ВЫДАЧИ).

(3) Если фраза SIGN (ЗНАК) включена в статью описания группы отчета, должен быть указан вариант SEPARATE CHARACTER (ОТДЕЛЬНО)

3.17.4. Общие правила

(1) Необязательная фраза SIGN (ЗНАК), если имеется, указывает в статье описания числового данного позицию и представ-

GOST EXPERT ГОСТ 22558-89,

ление знака числа этого данного. Фраза SIGN (ЗНАК) применяется только к статьям описания числовых данных, шаблон которых содержит литеру S (З); S (З) указывает на наличие (но не на представление или позицию) знака числа.

(2) Для числового данного, в статье описания которого не содержится фраза SIGN (ЗНАК), а шаблон содержит литеру S (З) предполагается знак числа, но ни позиция, ни представление знака числа литерой S (З) не определяются. В этом случае реализация определяет позицию и представление знака числа. Общее правило (3) не применяется к данным, описанным таким образом.

(3) Поскольку фраза SIGN (ЗНАК) в статье описания группы отчета должна содержать вариант SEPARATE CHARACTER (ОТДЕЛЬНО), то:

а) предполагается, что знак числа будет занимать позицию первой (или, соответственно, последней) литеры элементарного числового данного; эта позиция литеры не является цифровой позицией;

б) литера S (З) в строке литер шаблона учитывается при определении размера данного (в терминах литер стандартного формата данных);

в) знаками числа для положительных и отрицательных чисел являются литеры стандартного формата данных «+» и «—» соответственно.

(4) Каждая статья описания числового данного, содержащая шаблон с литерой S (З), является статьей описания числового данного со знаком. Если фраза SIGN (ЗНАК) относится к такой статье, и для вычислений или сравнения необходимо преобразование, преобразование производится автоматически.

3.18. **Фраза SOURCE (ИСТОЧНИК)**

3.18.1. Назначение

Фраза SOURCE (ИСТОЧНИК) идентифицирует посылаемое данное, которое помещается в соотнесенное ему печатаемое данное, определенное в статье описания группы отчета.

3.18.2. Общий формат

SOURCE IS идентификатор-1

ИСТОЧНИК идентификатор-1

3.18.3. Синтаксические правила

(1) Идентификатор-1 может быть определен в любой секции раздела данных. Если идентификатор-1 является данным, определенным в секции отчетов, то это должен быть:

а) счетчик строк или

б) счетчик страниц, или

в) счетчик суммы в том же отчете, в котором указана фраза SOURCE (ИСТОЧНИК).

(2) Идентификатор-1 указывает посылаемое данное неявного оператора MOVE (ПОМЕСТИТЬ), выполняемого СУГО для пересылки данного, указанного идентификатором-1, в печатаемое дан-

GOST EXPERT ГОСТ 22558-89,

нос. Идентификатор-1 должен быть определен в соответствии с правилами для посылаемых данных оператора MOVE (ПОМЕСТИТЬ) (см. ч. 6, п. 6.19).

3.18.4. Общие правила

(1) СУГО формирует печатаемые строки группы отчета непосредственно перед представлением группы отчета (п. 3.20 настоящей части). При этом выполняются неявные операторы MOVE (ПОМЕСТИТЬ), определенные фразами SOURCE (ИСТОЧНИК).

3.19. **Фраза SUM (СУММА)**

3.19.1. Назначение

Фраза SUM (СУММА) устанавливает счетчик суммы и указывает имена данных, подлежащих суммированию.

3.19.2. Общий формат

{SUM {идентификатор-1} ... [UPON {имя-данного-1} ...]} ...

[RESET ON {имя-данного-2 / FINAL}]

{СУММА {идентификатор-1} ... [ДЛЯ {имя-данного-1} ...]} ...

[СБРОСИТЬ ПО {имя-данного-2 / КОНЦУ}]

3.19.3. Синтаксические правила

(1) Данное, являющееся субъектом статьи описания группы отчета с фразой SUM (СУММА), не может быть определено как буквенно-цифровое. Идентификатор-1 должен ссылаться на числовое данное. Если идентификатор-1 определен в секции отчетов, идентификатор-1 должен представлять счетчик суммы.

Если фраза UPON (ДЛЯ) опущена, то указанные во фразе SUM (СУММА) идентификаторы, представляющие в свою очередь счетчики сумм, должны быть определены либо в той же группе отчета, которая содержит эту фразу SUM (СУММА), либо в группе отчета, находящейся на более низком уровне управляющей иерархии этого отчета.

Если указана фраза UPON (ДЛЯ), то идентификаторы, указанные во фразе SUM (СУММА), не могут представлять счетчики сумм.

(2) Имя-данного-1 должно быть именем группы отчета типа фрагмент, описанной в том же отчете, что и группа типа управляемая концовка, описание которой содержит фразу SUM (СУММА). Имя-данного-1 может уточняться именем-отчета.

(3) Фраза SUM (СУММА) может появляться только в описании группы отчета типа управляемая концовка.

(4) Имя-данного-2 должно быть одним из имен данных, указанных во фразе CONTROL (УПРАВЛЕНИЕ) для данного отчета. Уровень управления для имени-данного-2 не может быть ниже

GOST EXPERT ГОСТ 22558-89,

уровня управления, соответствующего группе отчета, в описании которой появляется вариант RESET (СБРОСИТЬ).

Если указана фраза RESET ON FINAL (СБРОСИТЬ ПО КОНЦУ), то для этого отчета должна быть также указана фраза CONTROL (УПРАВЛЕНИЕ).

(5) Наивысшим допустимым уточнителем для счетчика суммы является имя-отчета.

3.19.4. Общие правила

(1) Фраза SUM (СУММА) уточняет счетчик суммы. Счетчик суммы является порожденным компилятором числовым данным со знаком. Размер и положение десятичной точки счетчика суммы зависит от категории данного, определенного статьей описания группы отчета с фразой SUM (СУММА). Размер и положение десятичной точки определяются следующим образом:

а) если соответствующее данное числовое, размер и положение десятичной точки счетчика суммы такие же как и этого данного;

б) если соответствующее данное числовое редактируемое, размер счетчика суммы равняется числу цифровых позиций этого данного и положение десятичной точки такое же, как и в этом данном;

в) если соответствующее данное буквенно-цифровое или буквенно-цифровое редактируемое, размер счетчика суммы равняется размеру этого данного за исключением литер редактирования, или 18 литерам (меньшему из них), и счетчик суммы является целым.

(2) Во время выполнения СУГО прибавляет к счетчику суммы значение каждого данного, на которое ссылается идентификатор-1. Это сложение согласуется с правилами для арифметических операторов (см. ч. 6, пп. 6.4.4, 6.4.5).

(3) Статье описания элементарного данного отчета соответствует только один счетчик суммы, независимо от количества фраз SUM (СУММА), указанных в этой статье.

(4) Если статья описания элементарного печатаемого данного отчета содержит фразу SUM (СУММА), счетчик суммы служит в качестве данного-источника. СУГО помещает данное, содержащееся в счетчике суммы, в печатаемое данное для представления согласно правилам оператора MOVE (ПОМЕСТИТЬ).

(5) Если имя-данного появляется как субъект статьи описания элементарного данного отчета, содержащей фразу SUM (СУММА), имя-данного является именем счетчика суммы, имя-данного не является именем печатаемого данного, которое также может определяться этой статьей.

(6) Значения счетчиков сумм допустимо изменять с помощью операторов раздела процедур.

(7) Суммирование в счетчиках сумм значений данных, представленных идентификаторами, осуществляется системой управления генератором отчетов во время выполнения операторов GENE-

GOST EXPERT ГОСТ 22558-89,

RATE (ГЕНЕРИРОВАТЬ) и TERMINATE (ЗАКОНЧИТЬ). Имеются три категории увеличения счетчика суммы, называемые подытоживанием, концовочным суммированием и нарастающим итогом. Подытоживание исполняется только во время выполнения операторов GENERATE (ГЕНЕРИРОВАТЬ) и после обработки любого прерывания управления, но до обработки группы отчета типа фрагмент (см. п. 4.3 настоящей части). Концовочное суммирование и нарастающий итог исполняются во время обработки групп отчета типа управляемая концовка (п. 3.20 настоящей части).

(8) Вариант UPON (ДЛЯ) обеспечивает возможность выполнять выборочное подытоживание только для тех групп отчета типа фрагмент, на которые ссылаются в этом варианте.

(9) Момент времени, в который СУГО прибавляет каждое слагаемое к счетчику суммы, зависит от характеристик слагаемого:

а) если слагаемое является счетчиком суммы, определенным в той же группе отчета типа управляемая концовка, то накопление этого слагаемого в счетчике суммы определяется как концовочное суммирование.

Концовочное суммирование происходит, когда имеют место прерывание управления и обработка группы отчета типа управляемая концовка.

Концовочное суммирование выполняется соответственно последовательности, в которой определены счетчики сумм в группе отчета управляемая концовка. Это значит, что завершается все концовочное суммирование в первом счетчике суммы, определенном в группе отчета управляемая концовка, а затем завершается все концовочное суммирование во втором счетчике суммы, определенном в группе отчета типа управляемая концовка. Эта процедура повторяется пока не завершаются все операции концовочного суммирования.

Если одно из слагаемых является счетчиком суммы, определенным статьей описания данного, содержащей фразу SUM (СУММА), во время суммирования используется начальное значение этого счетчика;

б) если слагаемое является счетчиком суммы, определенным в группе отчета типа управляемая концовка с более низким уровнем в иерархии управления, накопление этого слагаемого в счетчике суммы определяется как нарастающее суммирование. Счетчик суммы в группе отчета типа управляемая концовка на более низком уровне иерархии управления наращивается, когда имеют место прерывание управления и обработка группы отчета типа управляемая концовка для этого уровня в иерархии управления;

в) если слагаемое не является счетчиком суммы, накопление в счетчике суммы такого рода слагаемого определяется как подытоживание. Если фраза SUM (СУММА) содержит вариант UPON (ДЛЯ) слагаемые накапливаются при выполнении оператора

GOST EXPERT ГОСТ 22558-89,

GENERATE (ГЕНЕРИРОВАТЬ) для указанных групп отчета типа фрагмент. Если фраза SUM (СУММА) указана без варианта UPON (ДЛЯ), слагаемые, не являющиеся счетчиками сумм, накапливаются при выполнении любого оператора GENERATE имя-данного (ГЕНЕРИРОВАТЬ имя-данного) для отчета, в описании которого указана фраза SUM (СУММА).

(10) Если два или более идентификаторов ссылаются на одно и то же слагаемое, это слагаемое прибавляется к счетчику суммы столько раз, сколько раз на него имеется ссылка во фразе SUM (СУММА). Допустимо в варианте UPON (ДЛЯ) указывать два или больше имен-данных, ссылающихся на одну и ту же группу типа фрагмент. Если для такой группы типа фрагмент задан оператор GENERATE имя-данного (ГЕНЕРИРОВАТЬ имя-данного), суммирование выполняется повторно столько раз, сколько раз это имя-данного появляется в варианте UPON (ДЛЯ)

(11) Подытоживание при выполнении оператора GENERATE имя-отчета (ГЕНЕРИРОВАТЬ имя-отчета) описано ниже (п. 4.3 настоящей части).

(12) При отсутствии явного варианта RESET (СБРОСИТЬ) СУГО устанавливает счетчик суммы в нуль во время обработки группы отчета типа управляемая концовка, в которой определяется счетчик суммы. Если указан явный вариант RESET (СБРОСИТЬ), СУГО устанавливает счетчик суммы в нуль во время обработки соответствующего уровня управляющей иерархии (п. 3.20 настоящей части).

Счетчики сумм первоначально устанавливаются СУГО в нуль во время выполнения оператора INITIATE (НАЧАТЬ) для отчета, в котором они определены.

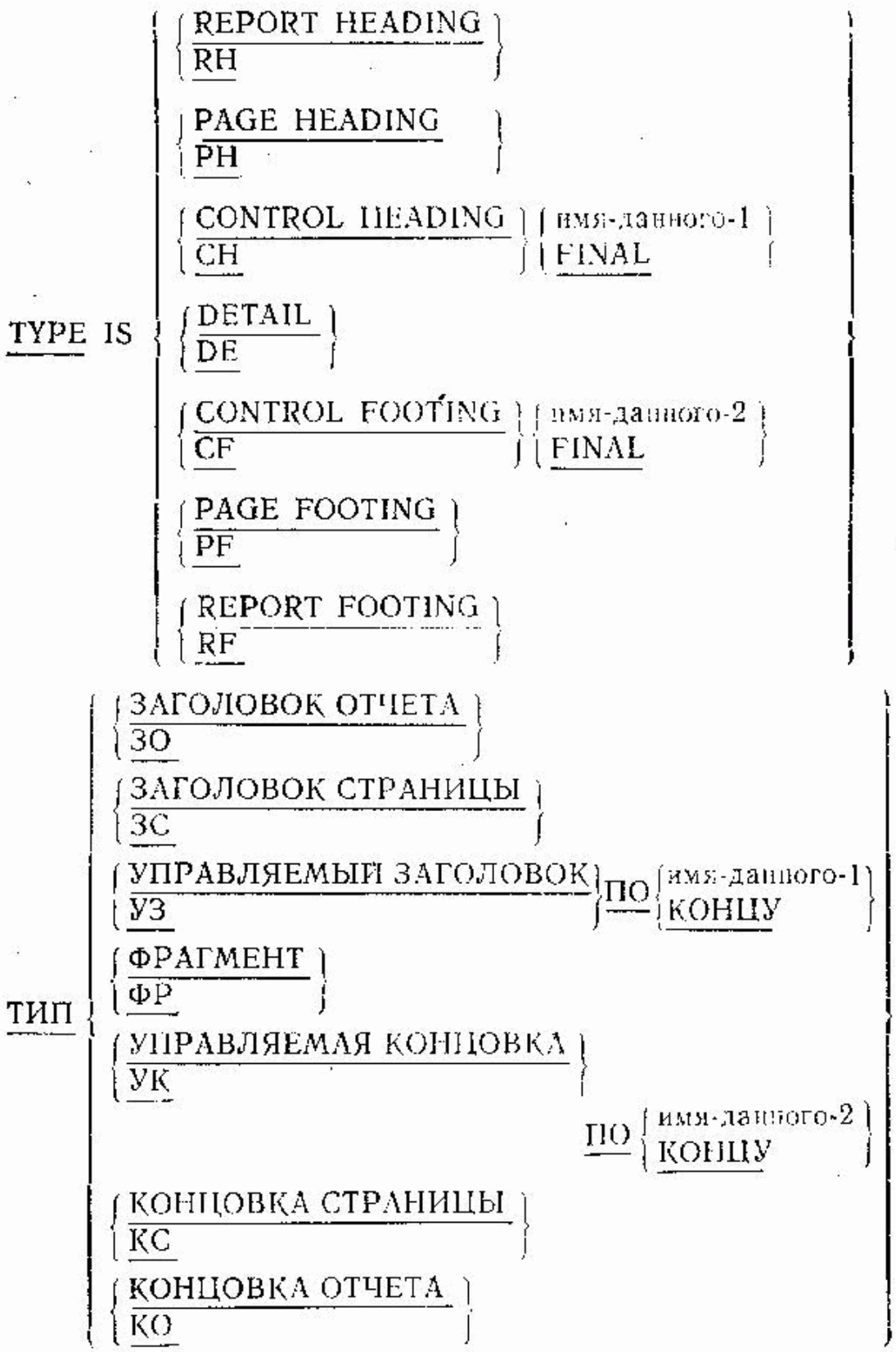
3.20. Фраза TYPE (ТИП)

3.20.1. Назначение

Фраза TYPE (ТИП) определяет, к какому типу принадлежит группа отчета, описание которой содержит эту фразу, и указывает время, в которое группа отчета должна обрабатываться СУГО.

GOST EXPERT ГОСТ 22558-89,

3.20.2. Общий формат



GOST EXPERT ГОСТ 22558-89,

3.20.3. Синтаксические правила

(1) RH (ЗО) является сокращением REPORT HEADING (ЗАГОЛОВОК ОТЧЕТА).

PH (ЗС) является сокращением PAGE HEADING (ЗАГОЛОВОК СТРАНИЦЫ).

CH (УЗ) является сокращением CONTROL HEADING (УПРАВЛЯЕМЫЙ ЗАГОЛОВОК).

DE (ФР) является сокращением DETAIL (ФРАГМЕНТ).

CF (УК) является сокращением CONTROL FOOTING (УПРАВЛЯЕМАЯ КОНЦОВКА).

PF (КС) является сокращением PAGE FOOTING (КОНЦОВКА СТРАНИЦЫ).

RF (КО) является сокращением REPORT FOOTING (КОНЦОВКА ОТЧЕТА).

(2) Каждая из групп отчета, определяемая фразами REPORT HEADING (ЗАГОЛОВОК ОТЧЕТА), PAGE HEADING (ЗАГОЛОВОК СТРАНИЦЫ), CONTROL HEADING FINAL (УПРАВЛЯЕМЫЙ ЗАГОЛОВОК ПО КОНЦУ), CONTROL FOOTING FINAL (УПРАВЛЯЕМАЯ КОНЦОВКА ПО КОНЦУ), PAGE FOOTING (КОНЦОВКА СТРАНИЦЫ) и REPORT FOOTING (КОНЦОВКА ОТЧЕТА) может указываться в описании отчета не более одного раза.

(3) Группы отчета типа заголовок страницы и концовка страницы могут быть указаны только тогда, когда в статье описания соответствующего отчета указана фраза PAGE (РАЗМЕР СТРАНИЦЫ).

(4) Имя-данного-1, имя-данного-2 и FINAL (ПО КОНЦУ), если указаны в фразе TYPE (ТИП), должны быть указаны и во фразе CONTROL (УПРАВЛЕНИЕ) соответствующей статьи описания отчета. Для каждого из указанных во фразе CONTROL (УПРАВЛЕНИЕ) статьи описания отчета имен-данных или FINAL (ПО КОНЦУ) может быть определено не более одной группы отчета типа управляемый заголовок и не более одной группы типа управляемая концовка. Тем не менее ни группа отчета типа управляемый заголовок, ни группа отчета типа управляемая концовка не являются обязательными для имени-данного или FINAL (ПО КОНЦУ), указанных во фразе CONTROL (УПРАВЛЕНИЕ) статьи описания отчета.

(5) В группах отчета типа управляемая концовка, заголовок страницы, концовка страницы и концовка отчета фразы SOURCE (ИСТОЧНИК) и соответствующие операторы USE (ИСПОЛЬЗОВАТЬ) не могут ссылаться на следующие данные:

а) групповые данные, содержащие управляющее данное;

б) данные, подчиненные управляющему данному;

в) данные, переопределяющие или переименовывающие любые позиции управляющего данного.

GOST EXPERT ГОСТ 22558-89,

В группах отчета типа заголовок страницы и концовка страницы фразы SOURCE (ИСТОЧНИК) и операторы USE (ИСПОЛЬЗОВАТЬ) не могут ссылаться на имена управляющих данных.

(6) Если в разделе процедур указан оператор GENERATE имя-отчета (ГЕНЕРИРОВАТЬ имя-отчета), соответствующая статья описания отчета должна содержать не более одной группы отчета типа фрагмент. Если для такого отчета не указан ни один оператор GENERATE имя-данного (ГЕНЕРИРОВАТЬ имя-данного), группа отчета типа фрагмент может не определяться.

(7) Описание отчета должно включать по крайней мере одну группу тела отчета.

3.20.4. Общие правила

(1) Группы отчета типа фрагмент порождаются СУГО как прямой результат операторов GENERATE (ГЕНЕРИРОВАТЬ). Обработка групп отчета других типов является автоматической функцией СУГО.

(2) Вариант REPORT HEADING (ЗАГОЛОВОК ОТЧЕТА) указывает группу отчета, порождаемую СУГО только один раз в отчете как первую группу этого отчета. Группа типа заголовок отчета порождается в момент выполнения первого по времени оператора GENERATE (ГЕНЕРИРОВАТЬ) для этого отчета.

(3) Вариант PAGE HEADING (ЗАГОЛОВОК СТРАНИЦЫ) указывает группу отчета, которая порождается СУГО как первая группа для каждой страницы этого отчета, кроме следующих случаев:

а) группа отчета типа заголовок страницы не порождается для страницы, которая должна содержать только группу типа заголовок отчета или концовка отчета;

б) группа типа заголовок страницы порождается как вторая группа отчета для страницы, когда ей предшествует группа типа заголовок отчета, которая не представляется на отдельной странице (см. п. 3.10 настоящей части).

(4) Вариант CONTROL HEADING (УПРАВЛЯЕМЫЙ ЗАГОЛОВОК) указывает группу отчета, которая обрабатывается СУГО в начале управляемой группы и соотнесена управляющему имени-данного или управлению FINAL (ПО КОНЦУ) и порождается в момент выполнения первого по времени оператора GENERATE (ГЕНЕРИРОВАТЬ) для этого отчета. При выполнении любого оператора GENERATE (ГЕНЕРИРОВАТЬ), если СУГО обнаруживает прерывание управления, порождаются группы отчета типа управляемый заголовок, соотнесенные наивысшему обнаруженному уровню прерывания управления и более низким уровням.

(5) Вариант DETAIL (ФРАГМЕНТ) указывает группу отчета, порождаемую СУГО при выполнении соответствующего оператора GENERATE (ГЕНЕРИРОВАТЬ).

GOST EXPERT ГОСТ 22558-89,

(6) Вариант CONTROL FOOTING (УПРАВЛЯЕМАЯ КОНЦОВКА) указывает группу отчета, порождаемую СУГО в конце управляемой группы для соответствующего управляющего именованного.

В случае управления FINAL (ПО КОНЦУ) группа типа управляемая концовка порождается только один раз в отчете как последняя группа тела этого отчета. При выполнении любого оператора GENERATE (ГЕНЕРИРОВАТЬ), если СУГО обнаруживает прерывание управления, порождается группа отчета типа управляемая концовка, соответствующая наивысшему уровню прерывания управления и более низким уровням. Если для некоторого отчета был выполнен хотя бы один оператор GENERATE (ГЕНЕРИРОВАТЬ), то при выполнении оператора TERMINATE (ЗАКОНЧИТЬ) порождаются все группы типа управляемая концовка для этого отчета (см. п. 4.7 настоящей части).

(7) Вариант PAGE FOOTING (КОНЦОВКА СТРАНИЦЫ) указывает группу отчета, которая порождается СУГО как последняя группа отчета на каждой странице за исключением следующих случаев:

а) группа отчета типа концовка страницы не порождается для страницы, которая должна содержать только группу типа заголовок отчета или только группу типа концовка отчета;

б) группа типа концовка страницы порождается как предпоследняя группа отчета на странице, если за ней следует группа типа концовка отчета, не представляемая на отдельной странице (см. п. 3.10 настоящей части).

(8) Вариант REPORT FOOTING (КОНЦОВКА ОТЧЕТА) указывает группу отчета, которая порождается СУГО только один раз в отчете как последняя группа этого отчета. Группа типа концовка отчета порождается во время выполнения соответствующего оператора TERMINATE (ЗАКОНЧИТЬ), если до этого был выполнен хотя бы один оператор GENERATE (ГЕНЕРИРОВАТЬ) для этого отчета (см. п. 4.7 настоящей части).

(9) При порождении групп отчета типа заголовок отчета, заголовок страницы, управляемый заголовок, концовка страницы или концовка отчета СУГО выполняет последовательность действий, описанных ниже:

а) если указана процедура USE BEFORE REPORTING (ИСПОЛЬЗОВАТЬ ДО ВЫДАЧИ) со ссылкой на именованную группы отчета, выполняется процедура, определяемая оператором USE (ИСПОЛЬЗОВАТЬ);

б) если был выполнен оператор SUPPRESS (ПОДАВИТЬ) или группа отчета не печатаемая, дальнейшая обработка группы отчета не производится;

в) если оператор SUPPRESS (ПОДАВИТЬ) не был выполнен и группа отчета печатаемая, СУГО формирует печатаемые строки

GOST EXPERT ГОСТ 22558-89,

и представляет группу отчета соответственно правилам представления для группы отчета этого типа (см. п. 3.10 настоящей части).

(10) Ниже приводится последовательность действий, выполняемых СУГО при порождении группы типа управляемая концовка.

Правила для оператора GENERATE (ГЕНЕРИРОВАТЬ) указывают, что при распознавании прерывания управления СУГО порождает группы отчета типа управляемая концовка начиная от самого низкого уровня в порядке возрастания до наивысшего уровня прерывания управления. При этом нужно отметить, что если фраза RESET (СБРОСИТЬ) в описании отчета указывает управляющее имя-данного, выполняются действия, описанные в п. 10е, даже если для этого управляющего имени-данного не определена группа отчета типа управляемая концовка.

а) Производится концовочное суммирование, т. е. все счетчики сумм, определенные в этой группе отчета, являющиеся операндами фраз SUM (СУММА) в этой же группе отчета, прибавляются к порождаемым этими фразами счетчикам сумм (см. п. 3.19 настоящей части).

б) Производится нарастающее суммирование, т. е. все счетчики сумм, определенные в этой группе отчета, являющиеся операндами фраз SUM (СУММА) в группах отчета типа управляемая концовка, относящихся к более высоким уровням иерархии управления, прибавляются к порождаемым этими фразами счетчикам сумм (см. п. 3.19 настоящей части).

в) Если имеется процедура USE BEFORE REPORTING (ИСПОЛЬЗОВАТЬ ДО ВЫДАЧИ) со ссылкой на имя-данного группы отчета, выполняется процедура, определенная оператором USE (ИСПОЛЬЗОВАТЬ).

г) Если был выполнен оператор SUPPRESS (ПОДАВИТЬ) или группа отчета непечатаемая, СУГО следующими выполняет действия, описанные в п. 10е ниже.

д) Если оператор SUPPRESS (ПОДАВИТЬ) не был выполнен и группа отчета печатаемая, СУГО формирует печатаемые строки и представляет группу отчета соответственно правилам представления для групп отчета типа управляемая концовка.

е) Затем СУГО сбрасывает те счетчики сумм, обновление которых должно выполняться при обработке этого уровня иерархии управления (см. п. 3.19 настоящей части).

(11) Обработка группы отчета типа фрагмент, выполняемая по оператору GENERATE имя-отчета (ГЕНЕРИРОВАТЬ имя-отчета), описана в п. 11а— 11д ниже.

Если в описании отчета указана только одна группа отчета типа фрагмент, обработка фрагмента выполняется генератором отчетов по оператору GENERATE имя-отчета (ГЕНЕРИРОВАТЬ имя-отчета) в соответствии с п. 11а—11д, описанными ниже. Эти дейст-

GOST EXPERT ГОСТ 22558-89,

вия выполняются так, как если бы выполнялся оператор GENERATE имя-данного (ГЕНЕРИРОВАТЬ имя-данного) для этого фрагмента.

Если в описании отчета нет ни одной группы отчета типа фрагмент, обработка, выполняемая по оператору GENERATE имя-отчета (ГЕНЕРИРОВАТЬ имя-отчета), описана в 11а.

Это действие выполняется так, как если бы описание отчета содержало только одну группу отчета типа фрагмент и выполнялся бы оператор GENERATE имя-данного (ГЕНЕРИРОВАТЬ имя-данного).

а) Выполняется любое подытоживание, предназначенное для определенной группы отчета типа фрагмент (см. п. 3.19 настоящей части).

б) Если имеется процедура USE BEFORE REPORTING (ИСПОЛЬЗОВАТЬ ДО ВЫДАЧИ) со ссылкой на имя-данного группы отчета, выполняется процедура USE (ИСПОЛЬЗОВАТЬ).

в) Если был выполнен оператор SUPPRESS (ПОДАВИТЬ) или группа отчета непечатаемая, то ее дальнейшая обработка не производится.

г) Если группа отчета типа фрагмент порождается по оператору GENERATE имя-отчета (ГЕНЕРИРОВАТЬ имя-отчета), дальнейшая обработка группы отчета не производится.

д) Если ни 11в, ни 11г не применяются, формируются печатаемые строки и группа отчета представляется соответственно правилам представления для групп отчета типа фрагмент (см. п. 3.10 настоящей части).

(12) При обработке групп отчета типа управляемый заголовок, управляемая концовка или фрагмент (см. общие правила 9, 10, 11), СУГО может прерывать обработку этой группы тела после установления, что эта группа должна быть представлена, и выполнить переход на следующую страницу (и, соответственно, порождение групп отчета типа концовка страницы или заголовок страницы) до представления группы тела.

(13) При обработке прерывания управления значения управляющих данных, которые использовались СУГО для обнаружения прерывания управления, будут называться предыдущими значениями.

а) При обработке по прерыванию управления группы отчета типа управляемая концовка любые ссылки на управляющие данные в процедуре, определенной оператором USE (ИСПОЛЬЗОВАТЬ) или фразе SOURCE (ИСТОЧНИК), связанные с этой группой типа управляемая концовка, относятся к предыдущим значениям.

б) При выполнении оператора TERMINATE (ЗАКОНЧИТЬ) и при порождении групп отчета типа управляемая концовка и концовка отчета СУГО обеспечивает доступность набора предыдущих

GOST EXPERT ГОСТ 22558-89,

значений управляющих данных фразе SOURCE (ИСТОЧНИК) или ссылкам процедуры, определенной оператором USE (ИСПОЛЬЗОВАТЬ), как если бы прерывание управления было распознано для управляющего имени-данного самого старшего уровня.

в) Ссылки на все остальные данные в группах отчета и относящихся к ним процедурах операторов USE (ИСПОЛЬЗОВАТЬ) относятся к текущим значениям данных во время обработки этой группы отчета.

3.21. **Фраза USAGE (ОБ ИСПОЛЬЗОВАНИИ)**

3.21.1. Назначение

Фраза USAGE (об использовании) определяет формат данного в памяти машины.

3.21.2. Общий формат

[USAGE IS] DISPLAY

ДЛЯ ВЫДАЧИ

3.21.3. Синтаксические правила

(1) Фраза USAGE (об использовании) может быть записана в любой статье описания данного.

(2) Если фраза USAGE (об использовании) записана в статье описания группового данного, она может быть также записана в статье описания подчиненного элементарного данного или группового данного.

(3) Фраза USAGE (об использовании) для группы отчета может определять только USAGE IS DISPLAY (ДЛЯ ВЫДАЧИ).

3.21.4. Общие правила

(1) Если фраза USAGE (об использовании) записана на уровне группового данного, она относится к каждому элементарному данному в группе.

(2) Фраза USAGE (об использовании) указывает, в каком виде представлено данное в памяти машины. Это не влияет на использование данного, хотя спецификации некоторых операторов в разделе процедур могут накладывать ограничения на фразу об использовании для операндов, на которые имеются ссылки в операторах. Фраза USAGE (об использовании) может влиять на основание системы счисления и тип представления литер данного.

(3) Фраза USAGE IS DISPLAY (ДЛЯ ВЫДАЧИ) указывает, что формат данного является стандартным форматом данного.

(4) Если фраза USAGE (об использовании) не указана для элементарного данного или для любой группы, к которой принадлежит данное, предполагается неявная фраза DISPLAY (ДЛЯ ВЫДАЧИ).

3.22. **Фраза VALUE (ЗНАЧЕНИЕ)**

3.22.1. Назначение

Фраза VALUE (ЗНАЧЕНИЕ) определяет значение печатаемого данного секции отчетов.

GOST EXPERT ГОСТ 22558-89,

3.22.2. Общий формат

VALUE IS литерал-1

{ЗНАЧЕНИЕ / ЗНАЧ} литерал-1

3.22.3. Синтаксические правила

(1) Числовой литерал со знаком должен соответствовать строке литер шаблона числового данного со знаком.

(2) Числовой литерал во фразе VALUE (ЗНАЧЕНИЕ) должен иметь значение из диапазона значений, указанного фразой PICTURE (ШАБЛОН) и не может иметь значение, которое могло бы потребовать усечения ненулевых цифр. Нечисловой литерал во фразе VALUE (ЗНАЧЕНИЕ) данного должен не превышать размер, указанный во фразе PICTURE (ШАБЛОН).

3.22.4. Общие правила

(1) Фраза VALUE (ЗНАЧЕНИЕ) не должна противоречить другим фразам в описании данного или в описании иерархии данного. Применяются следующие правила:

а) если категория данного числовая, литерал-1 во фразе VALUE (ЗНАЧЕНИЕ) должен быть числовым;

б) если категория данного буквенная, буквенно-цифровая, буквенно-цифровая редактируемая или числовая редактируемая, литерал-1 во фразе VALUE (ЗНАЧЕНИЕ) должен быть нечисловым литералом. Литерал выравнивается в данном как если бы данное было описано как буквенно-цифровое (см. ч. 4, п. 4.3.6). Литеры редактирования во фразе PICTURE (ШАБЛОН) учитываются при определении размера данного, но не влияют на инициацию данного (см. ч. 6, п. 5.9). Поэтому значение для редактируемого данного должно быть указано в отредактированной форме;

в) на инициацию не влияют фразы BLANK WHEN ZERO (ПРОБЕЛ КОГДА НУЛЬ) или JUSTIFIED (СДВИНУТО), которые могут быть указаны.

(2) Если в секции отчетов элементарная статья отчета, содержащая фразу VALUE (ЗНАЧЕНИЕ), не содержит фразу GROUP INDICATE (ОПРЕДЕЛЯЕТ ГРУППУ), то печатаемому данному присваивается указанное значение каждый раз при печати соответствующей группы отчета. Однако если кроме фразы VALUE (ЗНАЧЕНИЕ) имеется также фраза GROUP INDICATE (ОПРЕДЕЛЯЕТ ГРУППУ), указанное значение представляется только при существовании определенных условий во время выполнения (см. п. 3.13 настоящей части).

4. РАЗДЕЛ ПРОЦЕДУР В МОДУЛЕ ГЕНЕРАТОРА ОТЧЕТОВ

4.1. Общее описание

Если в исходной Кобол-программе имеется оператор USE BEFORE REPORTING (ИСПОЛЬЗОВАТЬ ДО ВЫДАЧИ) модуля гене-

GOST EXPERT ГОСТ 22558-89,

ратора отчетов, раздел процедур содержит декларативные процедуры. Ниже приведен общий формат раздела процедур, когда имеется оператор USE BEFORE REPORTING (ИСПОЛЬЗОВАТЬ ДО ВЫДАЧИ) и (или) USE AFTER STANDARD EXCEPTION PROCEDURE (ИСПОЛЬЗОВАТЬ ПОСЛЕ СТАНДАРТНОЙ ПРОЦЕДУРЫ ОШИБКИ).

PROCEDURE DIVISION.

DECLARATIVES.

{имя-секции SECTION.

оператор USE { AFTER STANDARD EXCEPTION PROCEDURE | BEFORE REPORTING }

[имя-параграфа.
[предложение] ...] ...}...

END DECLARATIVES.

{имя-секции SECTION.

[имя-параграфа.
[предложение] ...] ...}...

РАЗДЕЛ ПРОЦЕДУР.

ДЕКЛАРАТИВЫ.

{СЕКЦИЯ имя-секции.

оператор ИСПОЛЬЗОВАТЬ { ПОСЛЕ СТАНДАРТНОЙ ПРОЦЕДУРЫ ОШИБКИ | ДО ВЫДАЧИ }.

[имя-параграфа.
[предложение] ...] ...}...

КОНЕЦ ДЕКЛАРАТИВ

{СЕКЦИЯ имя-секции.

[имя-параграфа.
[предложение] ...] ...}...

4.2. Оператор CLOSE (ЗАКРЫТЬ)

4.2.1. Назначение

Оператор CLOSE (ЗАКРЫТЬ) завершает обработку катушек (томов) и файлов с необязательной перемоткой и (или) замком или удалением, где это применимо.

GOST EXPERT ГОСТ 22558-89,

4.2.2. Общий формат



4.2.3. Синтаксические правила

(1) Файлы, на которые ссылается оператор CLOSE (ЗАКРЫТЬ), могут иметь различную организацию или доступ.

(2) Допустимость вариантов в операторе CLOSE (ЗАКРЫТЬ) зависит от уровня модуля последовательного ввода-вывода, поддерживаемого реализацией (см. ч. 7, п. 4.2).

4.2.4. Общие правила

За исключением особо оговоренных случаев в ниже приведенных общих правилах, термины «катушка» и «том» являются синонимами и полностью взаимозаменяемы в операторе CLOSE (ЗАКРЫТЬ). Интерпретация последовательных файлов массовой памяти логически эквивалентна интерпретации файлов на ленте или аналогичных последовательных носителях. Файл, содержащийся в многофайловой ленточной среде, логически рассматривается как последовательный однокатушечный (однотомный) файл.

(1) Оператор CLOSE (ЗАКРЫТЬ) может быть выполнен только для файла, который был открыт.

(2) Для того, чтобы показать действие различных типов оператора CLOSE (ЗАКРЫТЬ) для различных носителей данных, все файлы отчетов разделяются на следующие категории:

а) без катушек (томов). Файл, носитель которого такой, что для него понятие перемотки катушек (томов) не имеет смысла;

б) последовательный однокатушечный (однотомный). Последовательный файл, который полностью располагается на одной катушке (томе);

в) последовательный многокатушечный (многотомный). Последовательный файл, который располагается на нескольких катушках (томах).

(3) Результаты выполнения каждого типа оператора CLOSE (ЗАКРЫТЬ) для каждой категории файла приведены в таблице.

Значения символов А—З в таблице приведены ниже.

А. Влияние на обработанные ранее катушки (тома) выходного файла отчетов.

GOST EXPERT ГОСТ 22558-89,

Формат оператора CLOSE (ЗАКРЫТЬ)	Категория файла		
	Без катушек (томов)	Последовательный однокатушечный (однотомный)	Последовательный многокатушечный (многотомный)
CLOSE (ЗАКРЫТЬ)	В	В, Ж	А, В, Ж
CLOSE WITH LOCK (ЗАКРЫТЬ С ЗАМКОМ)	В, Д	В, Д, Ж	А, В, Д, Ж
CLOSE WITH NO REWIND (ЗАКРЫТЬ БЕЗ ПЕРЕМОТКИ)	В, З	Б, В	А, Б, В
CLOSE REEL/UNIT (ЗАКРЫТЬ КАТУШКУ/ТОМ)	Е	Е, Ж	Е, Ж
CLOSE REEL/UNIT FOR REMOVAL (ЗАКРЫТЬ КАТУШКУ/ТОМ С УДАЛЕНИЕМ)	Е	Г, Е, Ж	Г, Е, Ж

Все катушки (тома) в файле отчетов, предшествующие текущей катушке (тому), закрываются, если для них не выполнялся оператор CLOSE REEL (ЗАКРЫТЬ КАТУШКУ) или CLOSE UNIT (ЗАКРЫТЬ ТОМ).

Б. Текущая катушка не перематывается.

Текущая катушка (том) остается в текущей позиции.

В. Закрыть выходной файл отчетов.

Если для файла указаны записи меток, метки обрабатываются в соответствии со стандартной процедурой обработки меток, определенной реализацией. Действия оператора CLOSE (ЗАКРЫТЬ) не определены, если записи меток указаны, но отсутствуют, или когда записи меток не указаны, но присутствуют.

Если записи меток не указаны для файла отчетов, метки не обрабатываются, но выполняются операции закрытия, определенные реализацией.

Г. Удаление катушки (тома).

Если это применимо, производится перемотка текущей катушки или тома и логическое удаление их из единицы исполнения, однако, катушка или том могут стать снова доступными в порядке расположения катушек или томов в файле отчета, если за выполнением оператора CLOSE (ЗАКРЫТЬ) без варианта REEL (КАТУШКУ) или UNIT (ТОМ) для этого файла отчетов будет выполнен оператор OPEN (ОТКРЫТЬ) для этого же файла отчетов.

Д. Закрыть с замком.

Файл отчетов закрывается (запирается) и не может быть открыт во время выполнения данной единицы исполнения.

GOST EXPERT ГОСТ 22558-89,

Е. Закрыть катушку или том.

Выходной файл отчетов (*носитель в виде катушки или тома*).

Выполняются следующие операции:

1) стандартная процедура обработки конечных меток катушки или тома;

2) смена катушки (тома). Указатель текущего тома обновляется и указывает на новую катушку (том);

3) Выполняется стандартная процедура обработки начальных меток катушки (тома);

4) Следующая операция занесения записи, относящаяся к этому *файлу, заносит следующую логическую запись на следующую* катушку (том) файла.

Выходной файл отчета (носитель не в виде катушки или тома).

Выполнение этого оператора считается успешным. Файл остается открытым, и никакие действия, кроме указанных в общем правиле 4, не выполняются.

Ж. Перемотка.

Текущая катушка или аналогичное устройство устанавливается на физическое начало.

Оператор CLOSE (ЗАКРЫТЬ) выполняется так, как будто нет необязательных фраз.

(4) Выполнение оператора CLOSE (ЗАКРЫТЬ) приводит к обновлению значения состояния ввода-вывода, связанного с именем файла-1 (см. ч. 7, п. 1.3.5).

(5) Все начатые отчеты, связанные с файлом отчетов, должны быть закончены посредством выполнения оператора TERMINATE (ЗАКОНЧИТЬ) до выполнения оператора CLOSE (ЗАКРЫТЬ) для этого файла отчетов.

(6) После успешного завершения оператора CLOSE (ЗАКРЫТЬ) без фразы REEL (КАТУШКУ) или UNIT (ТОМ) файл отчетов *больше не входит в число открытых файлов и файл отчетов* больше не связан с определителем файла.

(7) Если в операторе CLOSE (ЗАКРЫТЬ) указано более одного имени-файла-1, результат выполнения этого оператора CLOSE (ЗАКРЫТЬ) такой же, как если бы отдельный оператор CLOSE (ЗАКРЫТЬ) *был написан для каждого имени-файла-1* в том же порядке, в каком эти имена файлов указаны в операторе CLOSE (ЗАКРЫТЬ).

4.3. Оператор GENERATE (ГЕНЕРИРОВАТЬ)

4.3.1. Назначение

Оператор GENERATE (ГЕНЕРИРОВАТЬ) побуждает СУГО составить отчет в соответствии с описанием этого отчета в секции отчетов раздела данных.

4.3.2. Общий формат

GENERATE { имя-данного-1 / имя-отчета-1 }

GOST EXPERT ГОСТ 22558-89,

ГЕНЕРИРОВАТЬ { имя-данного-1 / имя-отчета-1 }

4.3.3. Синтаксические правила

(1) Имя-данного-1 должно относиться к группе отчета типа фрагмент и может уточняться именем отчета.

(2) Имя-отчета-1 может использоваться только тогда, когда в описании этого отчета содержится:

а) фраза CONTROL (УПРАВЛЕНИЕ) и

б) не более одной группы отчета типа фрагмент, и

в) по крайней мере одна группа тела отчета.

4.3.4. Общие правила

(1) По оператору GENERATE имя-отчета-1 (ГЕНЕРИРОВАТЬ имя-отчета-1) СУГО выполняет итоговую обработку. Если все выполняемые для отчета операторы GENERATE (ГЕНЕРИРОВАТЬ) имеют вид GENERATE имя-отчета-1 (ГЕНЕРИРОВАТЬ имя-отчета-1), то составляемый отчет называется итоговым отчетом. Итоговый отчет — отчет, в котором не представлена ни одна группа отчета типа фрагмент.

(2) По оператору GENERATE имя-данного-1 (ГЕНЕРИРОВАТЬ имя-данного-1) выполняется обработка фрагмента, которая включает определенную обработку, специфическую для группы отчета типа фрагмент, указанную оператором GENERATE (ГЕНЕРИРОВАТЬ). Обычно выполнение оператора GENERATE имя-данного-1 (ГЕНЕРИРОВАТЬ имя-данного-1) приводит к представлению указанной группы отчета типа фрагмент.

(3) При выполнении первого по времени оператора GENERATE (ГЕНЕРИРОВАТЬ) для данного отчета СУГО запоминает значения управляющих данных. Во время выполнения второго и последующих операторов для того же отчета и до распознавания прерывания управления СУГО использует этот набор управляющих данных для проверки наличия прерывания управления. Если встретилось прерывание управления, СУГО запоминает новую последовательность значений управляющих данных, которая с этого времени используется для определения прерывания управления, пока не встретится очередное прерывание управления.

(4) В процессе представления отчета, если СУГО должна осуществить переход на новую страницу для представления группы тела отчета, выполняются автоматические функции обработки групп отчета типа заголовок страницы и концовка страницы, если они определены (см. п. 3.10 настоящей части).

(5) При выполнении первого по времени оператора GENERATE (ГЕНЕРИРОВАТЬ) для данного отчета СУГО обрабатывает названные ниже группы отчета в порядке их перечисления (если они определены в описании отчета), а также группы отчета типа заголовок страницы и концовка страницы, как это описано в об-

GOST EXPERT ГОСТ 22558-89,

щем правиле 4. Действия, выполняемые при обработке каждого типа группы отчета, объяснены в соответствующем параграфе (см. п. 3.20 настоящей части).

а) Обрабатывается группа отчета типа заголовок отчета.

б) Обрабатывается группа отчета типа заголовок страницы.

в) Обрабатываются все группы отчета типа управляемый заголовок, начиная от старшего уровня иерархии управления к младшим уровням.

г) Если выполняется оператор GENERATE имя-данного-1 (ГЕНЕРИРОВАТЬ имя-данного-1), выполняется обработка указанной группы отчета типа фрагмент. Если выполняется оператор GENERATE имя-отчета-1 (ГЕНЕРИРОВАТЬ имя-отчета-1), то выполняются некоторые действия, сопровождающие обработку группы отчета типа фрагмент (см. п. 3.20 настоящей части).

(6) В процессе выполнения последующих по времени операторов GENERATE (ГЕНЕРИРОВАТЬ) для данного отчета выполняется ряд описанных ниже действий в порядке следования их описания, а также в соответствии с общим правилом 4 обрабатываются группы типа заголовок страницы и концовка страницы. Действия, выполняемые при обработке каждого типа группы отчета объяснены в соответствующем параграфе (см. п. 3.20 настоящей части).

а) Проверка на наличие прерывания управления. Правила определения равенства управляющих данных такие же, как и для условий отношения.

Если встретилось прерывание управления, то:

1) для процедур, определенных оператором USE (ИСПОЛЬЗОВАТЬ) и фраз SOURCE (ИСТОЧНИК), соответствующих группам отчета типа управляемая концовка, обеспечивается возможность доступа к значениям управляющих данных, вызвавшим данное прерывание управления (см. п. 3.20 настоящей части);

2) обработка групп отчета типа управляемая концовка в порядке от младшего к старшим уровням иерархии управления. Обрабатываются только те группы отчета типа управляемая концовка, уровень иерархии которых не старше самого высокого уровня, на котором встретилось прерывание управления;

3) обработка групп отчета типа управляемый заголовок в порядке от старшего к младшим уровням иерархии. Обрабатываются только те из них, уровень иерархии управления которых не старше самого высокого уровня, на котором встретилось прерывание управления.

б) Если выполняется оператор GENERATE имя-данного-1 (ГЕНЕРИРОВАТЬ имя-данного-1), производится обработка указанной группы отчета типа фрагмент. Если выполняется оператор GENERATE имя-отчета-1 (ГЕНЕРИРОВАТЬ имя-отчета-1), то выполняются некоторые действия, входящие в обработку группы отчета типа фрагмент (см. п. 3.20 настоящей части).

GOST EXPERT ГОСТ 22558-89,

(7) Операторы GENERATE (ГЕНЕРИРОВАТЬ) для отчета могут быть выполнены только после выполнения оператора INITIATE (НАЧАТЬ) и до выполнения оператора TERMINATE (ЗАКОНЧИТЬ) для этого отчета.

4.4. **Оператор INITIATE (НАЧАТЬ)**

4.4.1. Назначение

Оператор INITIATE (НАЧАТЬ) побуждает систему управления генератором отчетов начать составление отчета.

4.4.2. Общий формат

INITIATE {имя-отчета-1}...

НАЧАТЬ {имя-отчета-1}...

4.4.3. Синтаксические правила

(1) Каждое имя-отчета-1 должно быть определено статьей описания отчета в секции отчетов раздела данных.

4.4.4. Общие правила

(1) Оператор INITIATE (НАЧАТЬ) выполняет следующие действия для каждого указанного отчета:

а) все счетчики сумм устанавливаются в нуль;

б) LINE-COUNTER (СЧЕТЧИК-СТРОК) устанавливается в нуль;

в) PAGE-COUNTER (СЧЕТЧИК-СТРАНИЦ) устанавливается в единицу.

(2) Оператор INITIATE (НАЧАТЬ) не открывает файл, с которым связан отчет. Более того, для этого файла до выполнения оператора INITIATE (НАЧАТЬ) должен быть выполнен оператор OPEN (ОТКРЫТЬ) с вариантом OUTPUT (ВЫХОДНОЙ) или EXTEND (ДОПОЛНЯЕМЫЙ).

(3) Следующий оператор INITIATE (НАЧАТЬ) для имени-отчета-1 не может быть выполнен до тех пор, пока не будет выполнен оператор TERMINATE (ЗАКОНЧИТЬ) для этого отчета, следующий за предыдущим оператором INITIATE (НАЧАТЬ).

(4) Если в операторе INITIATE (НАЧАТЬ) указано более одного имени-отчета, результат выполнения этого оператора такой, как если бы был записан отдельный оператор INITIATE (НАЧАТЬ) для каждого имени-отчета в том порядке, в каком имена-отчетов упоминались в операторе INITIATE (НАЧАТЬ).

4.5. **Оператор OPEN (ОТКРЫТЬ)**

4.5.1. Назначение

Оператор OPEN (ОТКРЫТЬ) инициирует обработку файлов отчетов.

4.5.2. Общий формат

OPEN { OUTPUT {имя-файла-1 [WITH NO REWIND]}... } ...
{ EXTEND {имя-файла-2}... }

GOST EXPERT ГОСТ 22558-89,

ОТКРЫТЬ { ВЫХОДНОЙ (имя-файла-1 [БЕЗ ПЕРЕМОТКИ]) ... / ДОПОЛНЯЕМЫЙ (имя-файла-2) ... } ...

4.5.3. Синтаксические правила

(1) Оператор OPEN (ОТКРЫТЬ) для файла отчетов может содержать только вариант OUTPUT (ВЫХОДНОЙ) или EXTEND (ДОПОЛНЯЕМЫЙ).

(2) Допустимость вариантов в операторе OPEN (ОТКРЫТЬ) зависит от уровня модуля последовательного ввода-вывода, поддерживаемого реализацией (см. ч. 7, п. 4.3).

4.5.4. *Общие правила*

(1) Успешное выполнение оператора OPEN (ОТКРЫТЬ) делает файл доступным для обработки и переводит его в режим открытого файла. Успешное выполнение оператора OPEN (ОТКРЫТЬ) связывает файл с именем-файла посредством определителя файла.

Файл доступен, если он физически имеется в наличии и распознан системой управления вводом-выводом. Приведенная ниже таблица демонстрирует результаты открытия доступных и недоступных файлов.

Фраза	Файл доступен	Файл недоступен
OUTPUT (ВЫХОДНОЙ)	Нормальное открытие; файл не содержит записей	Открытие приводит к созданию файла
EXTEND (ДОПОЛНЯЕМЫЙ)	Нормальное открытие	Открытие неуспешное
EXTEND (ДОПОЛНЯЕМЫЙ) (необязательный файл)	Нормальное открытие	Открытие приводит к созданию файла

(2) Если файл не открыт, не может быть выполнен ни один оператор с явной или неявной ссылкой на файл, за исключением оператора OPEN (ОТКРЫТЬ).

(3) Оператор OPEN (ОТКРЫТЬ) для файла отчетов должен быть выполнен до выполнения оператора INITIATE (НАЧАТЬ) *для любого отчета, содержащегося в этом файле.*

(4) Файл отчетов может быть открыт с вариантом OUTPUT (ВЫХОДНОЙ) или EXTEND (ДОПОЛНЯЕМЫЙ) в одной и той же единице исполнения.

После первого выполнения оператора OPEN (ОТКРЫТЬ) для файла отчетов перед каждым последующим выполнением операто-

GOST EXPERT ГОСТ 22558-89,

ра OPEN (ОТКРЫТЬ) для этого файла должен быть выполнен оператор CLOSE (ЗАКРЫТЬ) без вариантов REEL (КАТУШКУ), UNIT (ТОМ) или LOCK (С ЗАМКОМ) для этого файла.

(5) Если для файла определены записи меток, начальные метки обрабатываются следующим образом:

а) когда указана фраза OUTPUT (ВЫХОДНОЙ), выполнение оператора OPEN (ОТКРЫТЬ) приводит к записи меток в соответствии с процедурами, определенными реализацией для записи выходных меток.

Действия оператора OPEN (ОТКРЫТЬ) не определены, когда записи меток указаны, но в файле отсутствуют, или не указаны, но присутствуют.

(6) Если во время выполнения оператора OPEN (ОТКРЫТЬ) возникает условие противоречия свойств файла, выполнение оператора OPEN (ОТКРЫТЬ) неуспешно (см. ч. 7, п. 1.3.7).

(7) Вариант NO REWIND (БЕЗ ПЕРЕМОТКИ) должен использоваться только в следующих случаях:

а) для последовательных однокатушечных (однотомных) файлов (см. п. 4.2 настоящей части);

б) для последовательных файлов, целиком содержащихся на одной катушке ленты в среде многофайловых лент (см. ч. 7, п. 2.11).

(8) Вариант NO REWIND (БЕЗ ПЕРЕМОТКИ) игнорируется, если он не применим к внешнему носителю, на котором располагается файл.

(9) Если внешний носитель для файла допускает перемотку, применяются следующие правила:

а) если ни один из вариантов EXTEND (ДОПОЛНЯЕМЫЙ) или NO REWIND (БЕЗ ПЕРЕМОТКИ) не указан, выполнение оператора OPEN (ОТКРЫТЬ) приводит к переустановке файла в начало;

б) если задана фраза NO REWIND (БЕЗ ПЕРЕМОТКИ), при выполнении оператора OPEN (ОТКРЫТЬ) переустановка файла не производится; файл уже должен быть установлен в начало до выполнения оператора OPEN (ОТКРЫТЬ).

(10) Если задан вариант EXTEND (ДОПОЛНЯЕМЫЙ), при выполнении оператора OPEN (ОТКРЫТЬ) файл устанавливается непосредственно за последней логической записью этого файла. Последней логической записью последовательного файла является последняя занесенная в файл запись.

(11) Если задан вариант EXTEND (ДОПОЛНЯЕМЫЙ) и фраза LABEL RECORDS (МЕТКИ) указывает, что записи меток присутствуют, выполнение оператора OPEN (ОТКРЫТЬ) включает следующие действия:

а) начальные метки файла обрабатываются только для однокатушечных (однотомных) файлов;

GOST EXPERT ГОСТ 22558-89,

б) начальные метки катушки (тома) обрабатываются на последней имеющейся катушке (томе), как если бы файл открывался с вариантом INPUT (ВХОДНОЙ);

в) имеющиеся конечные метки файла обрабатываются, как если бы файл был открыт с вариантом INPUT (ВХОДНОЙ). Эти метки затем удаляются;

г) затем обработка продолжается, как если бы файл был открыт как OUTPUT (ВЫХОДНОЙ).

(12) Интерпретация файла, содержащегося в среде многофайловых лент, логически эквивалентна интерпретации последовательного файла, содержащегося в среде однофайловых лент.

(13) Если совокупность файлов размещена на одной катушке ленты и на один из этих файлов имеется ссылка в операторе OPEN (ОТКРЫТЬ), то применяются следующие правила:

а) одновременно в открытом состоянии может находиться не более одного файла;

б) если один из файлов, соответствующих имени-файла, является субъектом оператора OPEN (ОТКРЫТЬ) с вариантом OUTPUT (ВЫХОДНОЙ), во время выполнения оператора OPEN (ОТКРЫТЬ) на соответствующей катушке должны уже существовать все файлы, номер позиций которых меньше, чем номер позиции данного файла. Кроме того, в это время на катушке не могут существовать файлы, номер позиции которых больше номера позиции данного файла;

в) каждый из файлов должен быть последовательным файлом.

(14) Для необязательного файла, являющегося недоступным, успешное выполнение оператора OPEN (ОТКРЫТЬ) с вариантом EXTEND (ДОПОЛНЯЕМЫЙ) создает файл. Это создание происходит так, как если бы в указанном порядке выполнялись следующие операторы:

OPEN OUTPUT *имя-файла*.
CLOSE *имя-файла*.
ОТКРЫТЬ ВЫХОДНОЙ *имя-файла*.
ЗАКРЫТЬ *имя-файла*.

За этими операторами следует выполнение оператора OPEN (ОТКРЫТЬ), указанного в исходной программе.

Успешное выполнение оператора OPEN (ОТКРЫТЬ) с вариантом OUTPUT (ВЫХОДНОЙ) создает файл. После успешного создания такой файл не содержит записей данных.

(15) После успешного выполнения оператора OPEN (ОТКРЫТЬ) указатель текущего тома устанавливается на:

а) катушку (том), содержащую последнюю логическую запись дополняемого файла;

б) новую катушку (том) для недоступного выходного или дополняемого файла.

GOST EXPERT ГОСТ 22558-89,

(16) Выполнение оператора OPEN (ОТКРЫТЬ) приводит к обновлению значения состояния ввода-вывода, соответствующего имени-файла (см. ч. 7, п. 1.3.5).

(17) Если в операторе OPEN (ОТКРЫТЬ) указано более одного имени-файла, результат выполнения этого оператора OPEN (ОТКРЫТЬ) такой, как если бы отдельный оператор OPEN (ОТКРЫТЬ) был написан для каждого имени-файла в том порядке, как они указаны в операторе OPEN (ОТКРЫТЬ).

(18) Минимальный и максимальный размеры записей файла устанавливаются во время создания файла и не могут изменяться в дальнейшем.

4.6. Оператор SUPPRESS (ПОДАВИТЬ)

4.6.1. Назначение

Оператор SUPPRESS побуждает систему управления генератором отчетов исключить представление группы отчета.

4.6.2. Общий формат

SUPPRESS PRINTING

ПОДАВИТЬ ПЕЧАТЬ

4.6.3. Синтаксические правила

(1) Оператор SUPPRESS (ПОДАВИТЬ) может указываться только в процедуре USE BEFORE REPORTING (ИСПОЛЬЗОВАТЬ ДО ВЫДАЧИ).

4.6.4. Общие правила

(1) Оператор SUPPRESS (ПОДАВИТЬ) исключает представление только группы отчета, названной в процедуре USE (ИСПОЛЬЗОВАТЬ), в которой появляется оператор SUPPRESS (ПОДАВИТЬ).

(2) Оператор SUPPRESS (ПОДАВИТЬ) должен быть выполнен каждый раз, когда необходимо исключить представление определенной группы отчета.

(3) При выполнении оператора SUPPRESS (ПОДАВИТЬ) система управления генератором отчетов информируется о запрещении следующих функций группы отчета:

а) представление печатных строк группы отчета;

б) обработка всех фраз LINE (НОМЕР СТРОКИ) для группы отчета;

в) обработка фразы NEXT GROUP (СЛЕДУЮЩАЯ ГРУППА) для группы отчета;

г) корректировка регистра LINE-COUNTER (СЧЕТЧИК-СТРОК).

4.7. Оператор TERMINATE (ЗАКОНЧИТЬ)

4.7.1. Назначение

Оператор TERMINATE (ЗАКОНЧИТЬ) побуждает систему управления генератором отчетов завершить обработку указанных отчетов.

GOST EXPERT ГОСТ 22558-89,

4.7.2. Общий формат

TERMINATE {имя-отчета-1}...

ЗАКОНЧИТЬ {имя-отчета-1}...

4.7.3. Синтаксические правила

(1) Имя-отчета-1 должно быть определено статьей описания отчета в секции отчетов раздела данных.

4.7.4. Общие правила

(1) Оператор TERMINATE (ЗАКОНЧИТЬ) приводит к выработке всех групп отчета типа управляемая концовка, начиная от управляемой концовки, соответствующей младшему уровню иерархии управления. Затем вырабатывается группа типа концовка отчета. СУГО обеспечивает доступность предыдущего набора значений управляющих данных фразам SOURCE (ИСТОЧНИК) для управляемой концовки и концовки отчета и процедурам USE (ИСПОЛЬЗОВАТЬ), как если бы прерывание управления было распознано для управляющего имени-данного самого старшего уровня иерархии управления.

(2) Если для данного отчета между выполнением операторов INITIATE (НАЧАТЬ) и TERMINATE (ЗАКОНЧИТЬ) не был выполнен ни один оператор GENERATE (ГЕНЕРИРОВАТЬ), оператор TERMINATE (ЗАКОНЧИТЬ) не приводит к обработке никаких групп отчета и выполнению соответствующих действий СУГО.

(3) В процессе представления отчета, когда СУГО должна осуществить переход на новую страницу для представления группы тела отчета, она выполняет автоматические функции обработки групп отчета типа заголовок страницы и концовка страницы, если они определены (см. п. 3.10 настоящей части).

(4) Оператор TERMINATE (ЗАКОНЧИТЬ) может быть выполнен для отчета, если только оператору TERMINATE (ЗАКОНЧИТЬ) предшествовал по времени оператор INITIATE (НАЧАТЬ) для этого отчета и никакой оператор TERMINATE (ЗАКОНЧИТЬ) еще не был выполнен для него.

(5) Если в операторе TERMINATE (ЗАКОНЧИТЬ) указано более одного имени-отчета, результат выполнения такого оператора TERMINATE (ЗАКОНЧИТЬ) такой же, как если бы отдельный оператор TERMINATE (ЗАКОНЧИТЬ) был записан для каждого имени-отчета в том же порядке, в каком имена-отчетов указаны в операторе TERMINATE (ЗАКОНЧИТЬ).

(6) Оператор TERMINATE (ЗАКОНЧИТЬ) не закрывает файл, с которым связан отчет; для этого файла должен быть выполнен оператор CLOSE (ЗАКРЫТЬ). Каждый отчет, для которого начата обработка, должен быть завершен до выполнения оператора CLOSE (ЗАКРЫТЬ) для соответствующего файла.

GOST EXPERT ГОСТ 22558-89,

4.8. Оператор USE AFTER STANDARD EXCEPTION PROCEDURE (ИСПОЛЬЗОВАТЬ ПОСЛЕ СТАНДАРТНОЙ ПРОЦЕДУРЫ ОШИБКИ)

4.8.1. Назначение

Оператор USE AFTER STANDARD EXCEPTION PROCEDURE (ИСПОЛЬЗОВАТЬ ПОСЛЕ СТАНДАРТНОЙ ПРОЦЕДУРЫ ОШИБКИ) указывает процедуры для обработки ошибок ввода-вывода, являющиеся дополнением к стандартным процедурам, обеспечиваемым системой управления вводом-выводом.

4.8.2. Общий формат

USE AFTER STANDARD { EXCEPTION / ERROR } PROCEDURE ON

{ {имя-файла-1}... / OUTPUT / EXTEND }

ИСПОЛЬЗОВАТЬ ПОСЛЕ СТАНДАРТНОЙ ПРОЦЕДУРЫ ОШИБКИ

ДЛЯ { {имя-файла-1}... / ВЫХОДНЫХ / ДОПОЛНЯЕМЫХ }

4.8.3. Синтаксические правила

(1) Оператор USE (ИСПОЛЬЗОВАТЬ), если он имеется, должен непосредственно следовать за заголовком секции в декларативной части раздела процедур и должен быть один в предложении. Остальная часть декларативной секции должна состоять из нуля, одного или нескольких процедурных параграфов, определяющих процедуры, которые должны использоваться.

(2) Сам оператор USE (ИСПОЛЬЗОВАТЬ) никогда не выполняется; он только определяет условия, вызывающие выполнение указанных после него процедур.

(3) Появление имени-файла-1 в операторе USE (ИСПОЛЬЗОВАТЬ) не может требовать одновременного выполнения более чем одной процедуры USE (ИСПОЛЬЗОВАТЬ).

(4) Слова ERROR и EXCEPTION являются синонимами.

(5) Файлы, на которые имеются явные или неявные ссылки в операторе USE (ИСПОЛЬЗОВАТЬ), могут иметь различную организацию или доступ.

(6) Каждый из вариантов OUTPUT (ВЫХОДНЫХ) или EXTEND (ДОПОЛНЯЕМЫХ) может указываться только один раз в декларативной части раздела процедур.

GOST EXPERT ГОСТ 22558-89,

4.8.4. Общие правила

(1) Декларативные процедуры могут быть включены в любую исходную Кобол-программу, независимо от того, содержит ли эта программа другую программу или сама содержится в другой программе. Декларатива вызывается тогда, когда во время выполнения программы выполняются условия, описанные в операторе USE (ИСПОЛЬЗОВАТЬ), предшествующем декларативе. Только одна декларатива внутри отдельно скомпилированной программы, содержащей оператор, который вызвал уточняющее условие, вызывается тогда, когда во время выполнения программы возникает какое-либо из условий, описанных в операторе USE (ИСПОЛЬЗОВАТЬ), предшествующем декларативе. Если не существует уточняющей декларативы в отдельно скомпилированной программе, то декларатива не выполняется.

(2) Внутри декларативной процедуры не должно быть обращений к каким-либо процедурам в недекларативной части раздела процедур.

(3) К именам процедур, связанным с оператором USE (ИСПОЛЬЗОВАТЬ), могут быть обращения в другой декларативной секции или в недекларативной процедуре только посредством оператора PERFORM (ВЫПОЛНИТЬ).

(4) Когда имя-файла-1 указано явно, то к имени-файла-1 не применяется никакой другой оператор USE (ИСПОЛЬЗОВАТЬ).

(5) Процедуры, связанные с оператором USE (ИСПОЛЬЗОВАТЬ), выполняются системой управления вводом-выводом после завершения стандартной программы обработки ошибки ввода-вывода при неуспешном выполнении операции ввода-вывода, за исключением того, что вариант AT END (В КОНЦЕ) имеет старшинство. При выполнении процедур соблюдаются следующие правила:

а) если указано имя-файла-1, то выполняется соответствующая процедура при возникновении условия, описанного в операторе USE (ИСПОЛЬЗОВАТЬ);

б) если указано OUTPUT (ВЫХОДНЫХ) то соответствующая процедура выполняется при возникновении условия, описанного в операторе USE (ИСПОЛЬЗОВАТЬ) для любого файла, открытого как OUTPUT (ВЫХОДНОЙ) или же находящегося в процессе открытия в режиме вывода, за исключением файлов, ссылка на которые посредством имени-файла-1 имеется в другом операторе USE (ИСПОЛЬЗОВАТЬ), описывающем такое же условие;

в) если указано EXTEND (ДОПОЛНЯЕМЫХ), то соответствующая процедура выполняется при возникновении условия, описанного в операторе USE (ИСПОЛЬЗОВАТЬ), для любого файла, открытого как EXTEND (ДОПОЛНЯЕМЫЙ) или находящегося в процессе открытия в режиме дополнения, за исключением файлов, ссылки на которые посредством имени-файла-1 имеются в другом

GOST EXPERT ГОСТ 22558-89,

операторе USE (ИСПОЛЬЗОВАТЬ), описывающем такое же условие.

(6) После выполнения процедуры, связанной с оператором USE (ИСПОЛЬЗОВАТЬ), управление передается вызывающей программе в системе управления вводом-выводом.

Если значение состояния ввода-вывода не указывает на критическую ошибку ввода-вывода, то система управления вводом-выводом возвращает управление оператору, следующему за оператором ввода-вывода. Если значение состояния ввода-вывода указывает на критическую ошибку, то действие определяется реализацией.

(7) Внутри процедуры, связанной с оператором USE (ИСПОЛЬЗОВАТЬ), не должны выполняться никакие операторы, которые могут потребовать выполнение процедуры, связанной с другим оператором USE (ИСПОЛЬЗОВАТЬ), вызванной ранее и еще не вернувшей управление вызвавшей ее программе.

4.9. **Оператор USE BEFORE REPORTING (ИСПОЛЬЗОВАТЬ ДО ВЫДАЧИ)**

4.9.1. Назначение

Оператор USE BEFORE REPORTING (ИСПОЛЬЗОВАТЬ ДО ВЫДАЧИ) указывает операторы раздела процедур, которые выполняются непосредственно перед представлением группы отчета, описанной в секции отчетов раздела процедур.

4.9.2. Общий формат

USE BEFORE REPORTING идентификатор-1

ИСПОЛЬЗОВАТЬ ДО ВЫДАЧИ идентификатор-1

4.9.3. Синтаксические правила

(1) Оператор USE BEFORE REPORTING (ИСПОЛЬЗОВАТЬ ДО ВЫДАЧИ), если он указан, должен непосредственно следовать за заголовком секции в декларативной части раздела процедур и должен быть единственным в предложении. Остальная часть секции должна состоять из нуля, одного или нескольких параграфов, определяющих подлежащие использованию процедуры.

(2) Идентификатор-1 должен относиться к группе отчета. Идентификатор-1 не должен появляться более чем в одном операторе USE BEFORE REPORTING (ИСПОЛЬЗОВАТЬ ДО ВЫДАЧИ).

(3) Операторы GENERATE (ГЕНЕРИРОВАТЬ), INITIATE (НАЧАТЬ) или TERMINATE (ЗАКОНЧИТЬ) не должны появляться в процедуре USE BEFORE REPORTING (ИСПОЛЬЗОВАТЬ ДО ВЫДАЧИ).

Оператор PERFORM (ВЫПОЛНИТЬ) в процедуре, относящейся к оператору USE BEFORE REPORTING (ИСПОЛЬЗОВАТЬ ДО ВЫДАЧИ), не должен иметь в своей области действия операторы GENERATE (ГЕНЕРИРОВАТЬ), INITIATE (НАЧАТЬ) или TERMINATE (ЗАКОНЧИТЬ).

GOST EXPERT ГОСТ 22558-89,

(4) Процедура, связанная с оператором USE BEFORE REPORTING (ИСПОЛЬЗОВАТЬ ДО ВЫДАЧИ), не должна изменять значений ни одного из управляющих данных.

(5) Оператор USE BEFORE REPORTING (ИСПОЛЬЗОВАТЬ ДО ВЫДАЧИ) сам не выполняется, он только определяет условия, которые вызывают выполнение связанной с ним процедуры.

4.9.4. Общие правила

(1) Декларативные процедуры могут быть включены в любую исходную Кобол-программу, независимо от того, содержит ли она другую программу или сама содержится в другой программе. Декларатива вызывается непосредственно перед представлением названной группы отчета во время выполнения программы. Группа отчета указывается идентификатором-1 в операторе USE BEFORE REPORTING (ИСПОЛЬЗОВАТЬ ДО ВЫДАЧИ), предшествующем процедуре.

(2) Внутри декларативной процедуры не должно быть обращений к каким-либо недекларативным процедурам.

(3) К именам-процедур, связанным с оператором USE BEFORE REPORTING (ИСПОЛЬЗОВАТЬ ДО ВЫДАЧИ), могут быть обращения в другой декларативной секции или в недекларативной процедуре только посредством оператора PERFORM (ВЫПОЛНИТЬ).

(4) В операторе USE BEFORE REPORTING (ИСПОЛЬЗОВАТЬ ДО ВЫДАЧИ) указанные процедуры выполняются системой управления генератором отчетов непосредственно перед представлением названной группы отчета (см. п. 3.20 настоящей части).

(5) Внутри процедуры, связанной с оператором USE (ИСПОЛЬЗОВАТЬ), не должны выполняться никакие операторы, которые могут потребовать выполнение процедуры, связанной с другим оператором USE (ИСПОЛЬЗОВАТЬ), вызванной ранее и еще не вернувшей управление вызвавшей ее программе.

Часть 14. МОДУЛЬ КОММУНИКАЦИЙ

1. ВВЕДЕНИЕ В МОДУЛЬ КОММУНИКАЦИЙ

1.1. **Назначение**

Модуль коммуникаций обеспечивает возможность получения, обработки и создания сообщений или их частей. Он позволяет с помощью системы управления сообщениями связываться с коммуникационными устройствами.

1.2. Характеристика уровней

Уровень 1 коммуникаций обеспечивает ограниченные возможности для статьи описания коммуникации. В разделе процедур уровень 1 обеспечивает ограниченные возможности операторов RECEIVE (ПОЛУЧИТЬ) и SEND (ПОСЛАТЬ) и полные возможности

GOST EXPERT ГОСТ 22558-89,

оператора ACCEPT MESSAGE COUNT (ПРИНЯТЬ ЧИСЛО СООБЩЕНИЙ).

Уровень 2 коммуникаций обеспечивает полные возможности для статьи описания коммуникации. В разделе процедур уровень 2 обеспечивает полные возможности операторов ACCEPT MESSAGE COUNT (ПРИНЯТЬ ЧИСЛО СООБЩЕНИЙ), DISABLE (ЗАПРЕТИТЬ), ENABLE (РАЗРЕШИТЬ), PURGE (ОЧИСТИТЬ), RECEIVE (ПОЛУЧИТЬ), SEND (ПОСЛАТЬ).

2. РАЗДЕЛ ДАННЫХ В МОДУЛЕ КОММУНИКАЦИЙ

2.1. Секция коммуникаций

Секция коммуникаций находится в разделе данных исходной программы. Секция коммуникаций описывает элементы данных исходной программы, которые будут использованы для интерфейса между системой управления сообщениями и программой. Эта область интерфейса определяется статьей описания коммуникации. За статьей описания коммуникации может следовать одна или несколько статей описания записей или не следовать ни одной.

Общий формат секции коммуникаций показан ниже:

COMMUNICATION SECTION.
[статья-описания-коммуникации
[статья-описания-записи] ...] ...
СЕКЦИЯ КОММУНИКАЦИЙ.
[статья-описания-коммуникации
[статья-описания-записи] ...] ...

2.1.1. Статья описания коммуникации

В программе на Коболе статья описания коммуникации представляет собой высший уровень организации в секции коммуникаций. За заголовком секции коммуникаций следует статья описания коммуникации, содержащая индикатор уровня CD (OK), имя-данного и последовательность независимых фраз. Статья заканчивается точкой.

Для статьи описания коммуникации для ввода фразы указывают очереди [и подочереди], дату и время сообщения, символический источник, длину текста, ключи состояния и конца, число сообщений. Для статьи описания коммуникации для вывода фразы указывают число адресатов, длину текста, ключи состояния и ошибки, символические адресаты.

Для статьи описания коммуникации для ввода-вывода фразы указывают дату и время сообщения, символический терминал, длину текста, ключи состояния и конца.

2.1.2. Структура описания записи

Область записи, связанная со статьей описания коммуникации, может быть неявно переопределена пользователем статьей описа-

GOST EXPERT ГОСТ 22558-89,

ния записи, написанной непосредственно за статьей описания коммуникации.

Описание записи состоит из набора статей описания данных, которые описывают характеристики отдельной записи. Каждая статья описания данного состоит из номера уровня, имени данного или фразы FILLER (ЗАПОЛНИТЕЛЬ), и следующей за ними последовательности независимых фраз. Описание записи может иметь иерархическую структуру, поэтому используемые в статье фразы зависят от наличия подчиненных статей. Структура описания записи и допустимые элементы в статье описания записи приводятся в ч. 4, пп. 4.3.1.3 и 5.3. Употребление отдельных фраз в статье описания данного зависит от уровня модуля ядра, поддерживаемого реализацией.

2.2. Статья описания коммуникации

2.2.1. Назначение

Описание коммуникации определяет область связи между системой управления сообщениями и программой Кобола.

2.2.2. Общий формат

Формат 1

```
CD имя-коммуникации-1 FOR [INITIAL] INPUT
    [[SYMBOLIC QUEUE IS имя-данного-1]
     [SYMBOLIC SUB-QUEUE-1 IS имя-данного-2]
     [SYMBOLIC SUB-QUEUE-2 IS имя-данного-3]
     [SYMBOLIC SUB-QUEUE-3 IS имя-данного-4]
     [MESSAGE DATE IS имя-данного-5]
     [MESSAGE TIME IS имя-данного-6]
     [SYMBOLIC SOURCE IS имя-данного-7]
     [TEXT LENGTH IS имя-данного-8]
     [END KEY IS имя-данного-9]
     [STATUS KEY IS имя-данного-10]
     [MESSAGE COUNT IS имя-данного-11]]
    [имя-данного-1, имя-данного-2, имя-данного-3,
     имя-данного-4 имя-данного-5, имя-данного-6,
     имя-данного-7, имя-данного-8, имя-данного-9,
     имя-данного-10, имя-данного-11].
```


GOST EXPERT ГОСТ 22558-89,

ОК имя-коммуникации-1 ДЛЯ [НАЧАЛЬНОГО] ВВОДА

[[СИМВОЛИЧЕСКАЯ ОЧЕРЕДЬ имя-данного-1]

[СИМВОЛИЧЕСКАЯ ПОДОЧЕРЕДЬ-1 имя-данного-2]
[СИМВОЛИЧЕСКАЯ ПОДОЧЕРЕДЬ-2 имя-данного-3]
[СИМВОЛИЧЕСКАЯ ПОДОЧЕРЕДЬ-3 имя-данного-4]

[ДАТА СООБЩЕНИЯ имя-данного 5]
[ВРЕМЯ СООБЩЕНИЯ имя-данного-6]
[СИМВОЛИЧЕСКИЙ ИСТОЧНИК имя-данного-7]
[ДЛИНА ТЕКСТА имя-данного-8]
[КЛЮЧ КОНЦА имя-данного-9]
[КЛЮЧ СОСТОЯНИЯ имя-данного-10]
[ЧИСЛО СООБЩЕНИЙ имя-данного-11]]

[имя-данного-1, имя-данного-2, имя-данного-3,
имя-данного-4, имя-данного-5, имя-данного-6,
имя-данного-7, имя-данного-8, имя-данного-9,
имя-данного-10, имя-данного-11]

Формат 2

CD имя-коммуникации-1 FOR OUTPUT

[DESTINATION COUNT IS имя-данного-1]
[TEXT LENGTH IS имя-данного-2]
[STATUS KEY IS имя-данного-3]

[DESTINATION TABLE OCCURS целое-1 TIMES
[INDEXED BY {имя-индекса-1}...]]

[ERROR KEY IS имя-данного-4]
[SYMBOLIC DESTINATION IS имя-данного-5].

GOST EXPERT ГОСТ 22558-89,

ОК имя-коммуникации-1 ДЛЯ ВЫВОДА
[ЧИСЛО АДРЕСАТОВ имя-данного-1]
[ДЛИНА ТЕКСТА имя-данного-2]
[КЛЮЧ СОСТОЯНИЯ имя-данного-3]
[ТАБЛИЦА АДРЕСАТОВ ПОВТОРЯЕТСЯ целое-1 РАЗ
[ИНДЕКСИРУЕТСЯ {имя-индекса-1} . . .]]
[КЛЮЧ ОШИБКИ имя-данного-4]
[СИМВОЛИЧЕСКИЙ АДРЕСАТ имя-данного-5].

Формат 3

CD имя-коммуникации-1 FOR [INITIAL] I-O
[[MESSAGE DATE IS имя-данного-1]
[MESSAGE TIME IS имя-данного-2]
[SYMBOLIC TERMINAL IS имя-данного-3]
[TEXT LENGTH IS имя-данного-4]
[END KEY IS имя-данного-5]
[STATUS KEY IS имя-данного-6]]
[имя-данного-1, имя-данного-2, имя-данного-3, имя-данного-4, имя-данного-5, имя-данного-6]

ОК имя-коммуникации-1 ДЛЯ НАЧАЛЬНОГО ВВОДА-ВЫВОДА
[[ДАТА СООБЩЕНИЯ имя-данного-1]
[ВРЕМЯ СООБЩЕНИЯ имя-данного-2]
[СИМВОЛИЧЕСКИЙ ТЕРМИНАЛ имя-данного-3]
[ДЛИНА ТЕКСТА имя-данного-4]
[КЛЮЧ КОНЦА имя-данного-5]
[КЛЮЧ СОСТОЯНИЯ имя-данного-6]]
[имя-данного-1 имя-данного-2, имя-данного-3, имя-данного-4, имя-данного-5, имя-данного-6]

GOST EXPERT ГОСТ 22558-89,

2.2.3. Синтаксические правила

Все форматы

(1) Статья CD (ОК) может появиться только в секции коммуникаций.

Форматы 1 и 3

(2) В пределах одной программы фраза INITIAL (НАЧАЛЬНОГО) может быть указана только в одной статье описания коммуникации. Эта фраза не может использоваться в программе, для которой в заголовке раздела процедур указан вариант USING (ИСПОЛЬЗУЯ).

(3) Кроме фразы INITIAL (НАЧАЛЬНОГО), все фразы могут быть написаны в любом порядке.

(4) Если ни одна из фраз в формате не указана, за статьей описания коммуникации CD (ОК) должна следовать статья уровня 01 описания данного. В любом случае за статьей CD (ОК) может следовать статья уровня 01 описания данного.

Формат 1

(5) Статьи описания записей, следующие за CD (ОК) для ввода, неявно переопределяют эту запись и должны описывать запись из 87 литер. Допускается неоднократное переопределение этой записи. Однако фразу VALUE (ЗНАЧЕНИЕ) может содержать только первое переопределение. Система управления сообщениями будет всегда обращаться к записи в соответствии с описаниями данных, определенными в общем правиле 2 (см. ч. 6, п. 5.15).

(6) Имя-данного-1 по имя-данного-11 не должны дублироваться в данной статье CD (ОК). Любое из этих имен данных может быть заменено зарезервированным словом FILLER (ЗАПОЛНИТЕЛЬ).

Формат 2

(7) Необязательные фразы могут следовать в любом порядке.

(8) Если в статье описания коммуникации не задана ни одна из фраз, за статьей CD (ОК) должна следовать статья описания данного уровня 01.

(9) Статьи описания записей, следующие за CD (ОК) для вывода, неявно переопределяют эту запись. Допускается многократное переопределение этой записи. Однако фразу VALUE (ЗНАЧЕНИЕ) может содержать только первое переопределение. Система управления сообщениями обращается к записи в соответствии с описаниями данных, определенными в общих правилах (см. ч. 6, п. 5.15).

(10) Имя-данного-1, имя-данного-2, имя-данного-3, имя-данного-4, имя-данного-5 не должны дублироваться внутри CD (ОК).

GOST EXPERT ГОСТ 22558-89,

(11) Если не задана фраза DESTINATION TABLE OCCURS (ТАБЛИЦА АДРЕСАТОВ ПОВТОРЯЕТСЯ), то предполагается одна область ключа ошибки и одна область символического адресата. В этом случае при обращении к имени-данного-4 и (или) имени-данного-5 не допускается индексирование.

(12) Если задана фраза DESTINATION TABLE OCCURS (ТАБЛИЦА АДРЕСАТОВ ПОВТОРЯЕТСЯ), то к этим данным можно обращаться только используя индексирование.

(13) Уровень 1 модуля коммуникаций требует, чтобы значение имени-данного-1 и целого-2 было равно 1. Уровень 2 никаких ограничений на значение имени-данного-1 и целого-2 не накладывает.

Формат 3

(14) Статьи описания записей, следующие за CD (ОК) для ввода-вывода, неявно переопределяют эту запись и должны описывать запись из 33 литер стандартного формата данных. Допускается многократное переопределение этой записи, однако фразу VALUE (ЗНАЧЕНИЕ) может содержать только первое переопределение. Система управления сообщениями обращается к записи в соответствии с правилами, определенными в общих правилах (см. ч. 6, п. 5.15).

(15) Имя-данного-1, имя-данного-2, имя-данного-3, имя-данного-4, имя-данного-5 и имя-данного-6 не должны дублироваться в данной статье CD (ОК). Любое из этих имен данных может быть заменено зарезервированным словом FILLER (ЗАПОЛНИТЕЛЬ).

2.2.4. Общие правила

Формат 1

(1) Информация CD (ОК) для ввода служит для связи между системой управления сообщениями и программой, так как задает информацию о сообщении, которое будет обрабатываться. Эта информация не поступает с терминала как часть сообщения.

(2) Для каждой статьи CD (ОК) для ввода выделяется область из 87 смежных литер в стандартном формате данных. Эта область записи определяется для системы управления сообщениями следующим образом:

а) фраза SYMBOLIC QUEUE (СИМВОЛИЧЕСКАЯ ОЧЕРЕДЬ) определяет имя-данного-1 как имя элементарного буквенно-цифрового данного из 12 литер, занимающего в записи позиций 1—12;

б) фраза SYMBOLIC SUB-QUEUE-1 (СИМВОЛИЧЕСКАЯ ПОДОЧЕРЕДЬ-1) определяет имя-данного-2 как имя элементарного буквенно-цифрового данного из 12 литер, занимающего в записи позиции 13—24;

GOST EXPERT ГОСТ 22558-89,

в) фраза SYMBOLIC SUB-QUEUE-2 (СИМВОЛИЧЕСКАЯ ПОДОЧЕРЕДЬ-2) определяет имя-данного-3 как имя элементарного буквенно-цифрового данного из 12 литер, занимающего в записи позиции 25—36;

г) фраза SYMBOLIC SUB-QUEUE-3 (СИМВОЛИЧЕСКАЯ ПОДОЧЕРЕДЬ-3) определяет имя-данного-4 как имя элементарного буквенно-цифрового данного из 12 литер, занимающего в записи позиции 37—48;

д) фраза MESSAGE DATE (ДАТА СООБЩЕНИЯ) определяет имя-данного-5 как имя данного, неявно описанного как целое из 6 цифр без знака и занимающего в записи позиции 49—54;

ж) фраза MESSAGE TIME (ВРЕМЯ СООБЩЕНИЯ) определяет имя-данного-6 как имя данного, неявно описанного как целое из 8 цифр без знака и занимающего в записи позиции 55—62;

з) фраза SYMBOLIC SOURCE (СИМВОЛИЧЕСКИЙ ИСТОЧНИК) определяет имя-данного-7 как имя элементарного буквенно-цифрового данного из 12 литер, занимающего в записи позиции 63—74;

и) фраза TEXT LENGTH (ДЛИНА ТЕКСТА) определяет имя-данного-8 как имя данного, неявно описанного как целое из 4 цифр без знака и занимающего в записи позиции 75—78;

к) фраза END KEY (КЛЮЧ КОНЦА) определяет имя-данного-9 как имя элементарного буквенно-цифрового данного из одной литеры, занимающей в записи позицию 79;

л) фраза STATUS KEY (КЛЮЧ СОСТОЯНИЯ) определяет имя-данного-10 как имя элементарного буквенно-цифрового данного из двух литер, занимающего в записи позиции 80—81;

м) фраза MESSAGE COUNT (ЧИСЛО СООБЩЕНИЙ) определяет имя-данного-11 как имя элементарного данного, неявно описанного как целое из 6 цифр без знака и занимающего в записи позиции 82—87.

Вместо указанных выше фраз могут быть использованы имена-данных, которые, взятые по порядку, соответствуют именам-данных, определенным этими фразами.

В любом случае предполагается неявное описание записи, как это представлено ниже.

GOST EXPERT ГОСТ 22558-89,

Неявное описание

```
01 имя-данного-0.
   02 имя-данного-1 PICTURE X(12).
   02 имя-данного-2 PICTURE X(12).
   02 имя-данного-3 PICTURE X(12).
   02 имя-данного-4 PICTURE X(12).
   02 имя-данного-5 PICTURE 9(6).
   02 имя-данного-6 PICTURE 9(8).
   02 имя-данного-7 PICTURE X(12).
   02 имя-данного-8 PICTURE 9(4).
   02 имя-данного-9 PICTURE X.
   02 имя-данного-10 PICTURE XX.
   02 имя-данного-11 PICTURE 9(6).
```

Неявное описание

```
01 имя-данного-0.
   02 имя-данного-1
      ШАБЛОН X(12).
   02 имя-данного-2
      ШАБЛОН X(12).
   02 имя-данного-3
      ШАБЛОН X(12).
   02 имя-данного-4
      ШАБЛОН X(12).
   02 имя-данного-5
      ШАБЛОН 9(6).
```

Комментарий

```
SYMBOLIC QUEUE
  SYMBOLIC SUB-QUEUE-1
  SYMBOLIC SUB-QUEUE-2
  SYMBOLIC SUB-QUEUE-3
MESSAGE DATE
MESSAGE TIME
SYMBOLIC SOURCE
TEXT LENGTH
END KEY
STATUS KEY
MESSAGE COUNT
```

Комментарий

```
СИМВОЛИЧЕСКАЯ ОЧЕРЕДЬ
  СИМВОЛИЧЕСКАЯ ПОДОЧЕРЕДЬ-1
  СИМВОЛИЧЕСКАЯ ПОДОЧЕРЕДЬ-2
  СИМВОЛИЧЕСКАЯ ПОДОЧЕРЕДЬ-3
ДАТА СООБЩЕНИЯ
```


GOST ГОСТ 22558-89

Неявное описание	Комментарий
02 имя-данного-6 ШАБЛОН 9(8).	ВРЕМЯ СООБЩЕНИЯ
02 имя-данного-7 ШАБЛОН X(12).	СИМВОЛИЧЕСКИЙ ИСТОЧНИК
02 имя-данного-8 ШАБЛОН 9(4).	ДЛИНА ТЕКСТА
02 имя-данного-9 ШАБЛОН X.	КЛЮЧ КОНЦА
02 имя-данного-10 ШАБЛОН XX.	КЛЮЧ СОСТОЯНИЯ
02 имя-данного-11 ШАБЛОН 9(6).	ЧИСЛО СООБЩЕНИЙ

(3) Когда значения имени-данного-2, имени-данного-3, имени-данного-4 не будут использоваться, они должны представлять собой пробелы.

(4) Имя-данного-1, имя-данного-2, имя-данного-3 и имя-данного-4 содержат символические имена, обозначающие соответственно очереди и подочереди. Все символические имена должны образовываться по правилам для системных имен и должны быть предварительно определены для системы управления сообщениями.

(5) Оператор RECEIVE (ПОЛУЧИТЬ) вызывает последовательное получение следующего сообщения или его части из очереди, как указано фразами CD (ОК).

Во время выполнения оператора RECEIVE (ПОЛУЧИТЬ), область CD (ОК) для ввода должна содержать в качестве значения имени-данного-1 имя символической очереди. Элементы данных имя-данного-2, имя-данного-3 и имя-данного-4 могут содержать имена символических подочередей или пробелы.

Когда задан некоторый уровень структуры очереди, должны быть заданы и все более высокие уровни. Если заданы не все уровни иерархии очереди, то система управления сообщениями определяет следующее сообщение или его часть, которое доступно в очереди и(или) подочереди, указанной CD (ОК) для ввода.

После выполнения оператора RECEIVE (ПОЛУЧИТЬ) значения имени-данного-1 по имя-данного-4 будут представлять символические имена всех уровней структуры очереди.

(6) Если программа обработки сообщения вызывается системой управления сообщениями, символические имена уровней

GOST EXPERT ГОСТ 22558-89,

структуры очереди, связанной с этой обработкой, помещаются в имя-данного-1 по имя-данного-4, определенные в статье CD (ОК) с фразой INITIAL (НАЧАЛЬНОГО). Во всех остальных случаях запуска программ значения имени-данного-1 по имя-данного-4, связанные с такой статьей, представляются пробелами.

Засылка пробелов или символических имен заканчивается до выполнения первого оператора раздела процедур.

Выполнение последующего оператора RECEIVE (ПОЛУЧИТЬ) для тех же значений имени-данного-1 по имя-данного-4 приводит к получению того же сообщения, которое вызвало запуск программы. Только в этот момент будет обновлена оставшаяся часть области связи.

(7) Если система управления сообщениями пытается вызвать программу, не содержащую фразу INITIAL (НАЧАЛЬНОГО) в статье CD (ОК), результат не определен.

(8) При выполнении оператора RECEIVE (ПОЛУЧИТЬ) система управления сообщениями помещает дату, когда было распознано, что сообщение завершено, в форме 'ГГММДД' (год, месяц, день) в имя-данного-5. Значение имени-данного-5 обновляется системой управления сообщениями только во время выполнения оператора RECEIVE (ПОЛУЧИТЬ).

(9) При выполнении оператора RECEIVE (ПОЛУЧИТЬ) система управления сообщениями помещает значение момента времени завершения сообщения в форме 'ЧЧММССХХ' (часы, минуты, секунды, сотые доли секунды) в имя-данного-6. Значение имени-данного-6 обновляется системой управления сообщениями только во время выполнения оператора RECEIVE (ПОЛУЧИТЬ).

(10) При выполнении оператора RECEIVE (ПОЛУЧИТЬ) система управления сообщениями помещает в имя-данного-7 символическое имя терминала, который является источником передаваемого сообщения.

Это символическое имя должно удовлетворять правилам образования системных имен.

Однако, если символическое имя терминала неизвестно системе управления сообщениями, имя-данного-7 будет содержать пробелы.

(11) Значением имени-данного-8 система управления сообщениями указывает число позиций литер, заполненных в результате выполнения оператора RECEIVE (ПОЛУЧИТЬ).

(12) Значение имени-данного-9 устанавливается системой управления сообщениями во время выполнения оператора RECEIVE (ПОЛУЧИТЬ) по следующим правилам:

а) для оператора RECEIVE MESSAGE (ПОЛУЧИТЬ СООБЩЕНИЕ):

GOST EXPERT ГОСТ 22558-89,

1) если обнаружен конец группы, то значение имени-данного-9 устанавливается равным 3;

2) если обнаружен конец сообщения, то значение имени-данного-9 устанавливается равным 2;

3) если передается часть сообщения, то значение имени-данного-9 устанавливается равным нулю;

б) для оператора RECEIVE SEGMENT (ПОЛУЧИТЬ СЕГМЕНТ):

1) Если обнаружен конец группы, то значение имени-данного-9 устанавливается равным 3;

2) если обнаружен конец сообщения, то значение имени-данного-9 устанавливается равным 2;

3) если обнаружен конец сегмента, то значение имени-данного-9 устанавливается равным 1;

4) если передается только часть сообщения, то значение имени-данного-9 устанавливается равным 0;

в) если одновременно выполняются несколько перечисленных условий, то значение имени-данного-9 определяется первым выполненным условием в перечисленном выше порядке.

(13) Значение имени-данного-10 указывает состояние выполненных перед этим операторов RECEIVE (ПОЛУЧИТЬ) ACCEPT MESSAGE COUNT (ПРИНЯТЬ ЧИСЛО СООБЩЕНИЙ), ENABLE INPUT (РАЗРЕШИТЬ ВВОД), DISABLE INPUT (ЗАПРЕТИТЬ ВВОД).

Соответствие значения имени-данного-1 состоянию выполнения отражено в табл. 1.

(14) Значение имени-данного-11 указывает число сообщений, имеющихся в очереди, подочереди-1 и т. д. Система управления сообщениями обновляет это значение при выполнении оператора ACCEPT MESSAGE COUNT (ПРИНЯТЬ ЧИСЛО СООБЩЕНИЙ).

Формат 2

(15) Информация, определенная статьей CD (ОК) для вывода, не посылается на терминал. Она служит для связи между системой управления сообщениями и программой и задает информацию о сообщении, которое обрабатывается.

(16) На уровне 1 для каждого CD (ОК) для вывода выделяется непрерывная область записи из 23 литер. На уровне 2 для каждого CD (ОК) для вывода выделяется непрерывная область записи длиной (10 плюс (13 умножить на целое-1)) литер. Неявное описание этой области записи следующее:

а) фраза DESTINATION COUNT (ЧИСЛО АДРЕСАТОВ) определяет имя-данного-1 как имя данного, неявно описанного как целое без знака и занимающего в записи позиции 1—4;

GOST EXPERT ГОСТ 22558-89,

б) фраза TEXT LENGTH (ДЛИНА ТЕКСТА) определяет имя-данного-2 как имя данного, неявно описанного как целое из 4 цифр без знака и занимающего в записи позиции 5—8;

в) фраза STATUS KEY (КЛЮЧ СОСТОЯНИЯ) определяет имя-данного-3 как элементарное буквенно-цифровое данное из 2 литер, занимающее в записи позиции 9, 10;

г) позиции литер 11—23 и каждый набор по 13 литер за ними образуют таблицу элементов со следующим описанием:

1) фраза ERROR KEY (КЛЮЧ ОШИБКИ) определяет имя-данного-4 как имя элементарного буквенно-цифрового данного из одной литеры;

2) фраза SYMBOLIC DESTINATION (СИМВОЛИЧЕСКИЙ АДРЕСАТ) определяет имя-данного-5 как имя элементарного буквенно-цифрового данного из 12 литер.

Использование всех этих фраз определяет запись, неявное описание которой приведено ниже.

GOST EXPERT ГОСТ 22558-89,

GOST ГОСТ 22558-89

Неявное описание	Комментарий
01 имя-данного-0.	
02 имя-данного-1 PICTURE 9(4).	DESTINATION COUNT
02 имя-данного-2 PICTURE 9(4).	TEXT LENGTH
02 имя-данного-3 PICTURE XX.	STATUS KEY
02 имя-данного [OCCURS целое-1 TIMES].	DESTINATION TABLE
03 имя-данного-4 PICTURE X.	ERROR KEY
03 имя-данного-5 PICTURE X(12).	SYMBOLIC DESTINATION

Неявное описание	Комментарий
01 имя-данного-8.	
02 имя-данного-1 ШАБЛОН 9(4).	ЧИСЛО АДРЕСАТОВ
02 имя-данного-2 ШАБЛОН 9(4).	ДЛИНА ТЕКСТА
02 имя-данного-3 ШАБЛОН XX.	КЛЮЧ СОСТОЯНИЯ
02 имя-данного [ПОВТОРЯЕТСЯ] [целое-1 РАЗ].	ТАБЛИЦА АДРЕСАТОВ
03 имя-данного-4 ШАБЛОН X.	КЛЮЧ ОШИБКИ
03 имя-данного-5 ШАБЛОН X(12).	СИМВОЛИЧЕСКИЙ АДРЕСАТ

(17) При выполнении операторов SEND (ПОСЛАТЬ), PURGE (ОЧИСТИТЬ), ENABLE OUTPUT (РАЗРЕШИТЬ ВЫВОД), DISABLE OUTPUT (ЗАПРЕТИТЬ ВЫВОД) значение имени-данного-1 указывает системе управления сообщениями число символических адресатов, которые надо использовать из области, определяемой именем-данного-5.

Система управления сообщениями находит первый символический адресат в первом элементе таблицы адресатов, определенной именем-данного-5, следующий символический адресат — в следующем элементе этой таблицы и т. д. до элемента номер которого совпадает со значением имени-данного-1.

Если при выполнении оператора SEND (ПОСЛАТЬ), PURGE (ОЧИСТИТЬ), ENABLE OUTPUT (РАЗРЕШИТЬ ВЫВОД), DISABLE OUTPUT (ЗАПРЕТИТЬ ВЫВОД) значение имени-данного-1 находится вне диапазона чисел от 1 до целого-1, устанавливается ключ ошибки и выполнение оператора прекращается.

(18) Пользователь должен обеспечить, чтобы значение имени-данного-1 было допустимым в момент выполнения операторов SEND (ПОСЛАТЬ), PURGE (ОЧИСТИТЬ), DISABLE OUTPUT (ЗАПРЕТИТЬ ВЫВОД), ENABLE OUTPUT (РАЗРЕШИТЬ ВЫВОД).

(19) Во время выполнения оператора SEND (ПОСЛАТЬ) система управления сообщениями рассматривает значение имени-данного-2 как число крайних левых позиций литер в поле, определяемом идентификатором в операторе SEND (ПОСЛАТЬ), из которого надо передавать данные (п. 3.6 настоящей части).

(20) Каждое вхождение имени-данного-5 содержит символический адресат, предварительно сообщенный системе управления сообщениями. Имена символических адресатов должны соответствовать правилам образования системных имен.

(21) Значение имени-данного-3 указывает состояние выполнения оператора SEND (ПОСЛАТЬ), PURGE (ОЧИСТИТЬ), ENABLE OUTPUT (РАЗРЕШИТЬ ВЫВОД), DISABLE OUTPUT (ЗАПРЕТИТЬ ВЫВОД). Соответствие значения имени-данного-3 состоянию выполнения операторов приведено в табл. 1.

(22) Если при выполнении операторов DISABLE OUTPUT (ЗАПРЕТИТЬ ВЫВОД), ENABLE OUTPUT (РАЗРЕШИТЬ ВЫВОД), PURGE (ОЧИСТИТЬ), SEND (ПОСЛАТЬ) система управления сообщениями определяет, что имеет место ошибка, значения имени-данного-3 и всех вхождений имени-данного-4 обновляются вплоть до и включая вхождение, указанное значением имени-данного-1.

GOST EXPERT ГОСТ 22558-89,

Соответствие между значением данного имя-данного-4 и ключом ошибки определено в табл. 2.

Формат 3

(23) Информация, определенная статьей CD (ОК) для ввода-вывода служит для связи между системой управления сообщениями и программой и задает информацию о сообщении, которое обрабатывается. Эта информация не передается с терминала как часть сообщения.

(24) Для каждого CD (ОК) для ввода-вывода выделяется непрерывная область записи из 33 литер. Эта область записи определяется для системы управления сообщениями следующим образом:

а) фраза MESSAGE DATE (ДАТА СООБЩЕНИЯ) определяет имя-данного-1 как имя данного, неявно описанного как целое из 6 цифр без знака, занимающего в записи позиции 1—6;

б) фраза MESSAGE TIME (ВРЕМЯ СООБЩЕНИЯ) определяет имя-данного-2 как имя данного, неявно описанного как целое из 8 цифр без знака, занимающего в записи позиции литер 7—14;

в) фраза SYMBOLIC TERMINAL (СИМВОЛИЧЕСКИЙ ТЕРМИНАЛ) определяет имя-данного-3 как имя элементарного буквенно-цифрового данного, состоящего из 12 литер и занимающего в записи позиции литер 15—26;

г) фраза TEXT LENGTH (ДЛИНА ТЕКСТА) определяет имя-данного-4 как имя элементарного данного, неявно описанного как целое из 4 цифр без знака, занимающего в записи позиции литер 27—30;

д) фраза END KEY (КЛЮЧ КОНЦА) определяет имя-данного-5 как имя элементарного буквенно-цифрового данного из 1 литеры и занимающего в записи позицию 31;

е) фраза STATUS KEY (КЛЮЧ СОСТОЯНИЯ) определяет имя-данного-6 как имя элементарного буквенно-цифрового данного из 2 литер и занимающего в записи позиции литер 32, 33.

Вместо указанных выше фраз могут быть использованы имена-данных, которые, взятые по порядку, соответствуют именам-данных, определенным этими фразами.

В любом случае предполагается неявное описание записи, которое представлено ниже.

GOST EXPERT ГОСТ 22558-89,

690

Неявное описание	Комментарий
01 имя-данного-0.	
02 имя-данного-1 PICTURE 9(6).	MESSAGE DATE
02 имя-данного-2 PICTURE 9(8).	MESSAGE TIME
02 имя-данного-3 PICTURE X(12).	SYMBOLIC TERMINAL
02 имя-данного-4 PICTURE 9(4).	TEXT LENGTH
02 имя-данного-5 PICTURE X.	END KEY
02 имя-данного-6 PICTURE XX.	STATUS KEY
01 имя-данного-0.	
02 имя-данного-1 ШАБЛОН 9(6).	ДАТА СООБЩЕНИЯ
02 имя-данного-2 ШАБЛОН 9(8).	ВРЕМЯ СООБЩЕНИЯ
02 имя-данного-3 ШАБЛОН X(12).	СИМВОЛИЧЕСКИЙ ТЕРМИНАЛ
02 имя-данного-4 ШАБЛОН 9(4).	ДЛИНА ТЕКСТА
02 имя-данного-5 ШАБЛОН X.	КЛЮЧ КОНЦА
02 имя-данного-6 ШАБЛОН XX.	КЛЮЧ СОСТОЯНИЯ

GOST ГОСТ 22558-89

(25) Если программа обработки сообщений вызывается системой управления сообщениями, то выполнение первого оператора RECEIVE (ПОЛУЧИТЬ) для статьи CD (ОК) для ввода-вывода с фразой INITIAL (НАЧАЛЬНОГО) приводит к получению того же сообщения, которое вызвало запуск программы.

(26) Имя-данного-1 имеет формат 'ГГММДД' (год, месяц, день). Его значение представляет дату, когда было распознано завершение сообщения системой управления сообщениями.

Значение имени-данного-1 обновляется системой управления сообщениями только во время выполнения оператора RECEIVE (ПОЛУЧИТЬ).

(27) Имя-данного-2 имеет формат 'ЧЧММССДД' (часы, минуты, секунды, сотые доли секунды) и его значение представляет время, когда системой управления сообщениями было распознано завершение сообщения.

Значение имени-данного-2 обновляется системой управления сообщениями только во время выполнения оператора RECEIVE (ПОЛУЧИТЬ).

(28) Если программа обработки сообщений вызывается системой обработки сообщений, то символическое имя терминала, являющегося источником сообщения, активирующего эту программу, помещается в имя-данного-3 статьи CD (ОК) для ввода-вывода с фразой INITIAL (НАЧАЛЬНОГО), если она применяется. Это символическое имя должно удовлетворять правилам образования системных имен.

Во всех других случаях значением имени-данного-3 статьи CD (ОК) для ввода-вывода с фразой INITIAL (НАЧАЛЬНОГО) будут пробелы.

Засылка символического имени или пробелов заканчивается до выполнения первого оператора раздела процедур.

(29) Если система управления сообщениями пытается вызвать программу, не содержащую фразу INITIAL (НАЧАЛЬНОГО) в статье CD (ОК), результат не определен.

(30) Если фраза INITIAL (НАЧАЛЬНОГО) используется в статье CD (ОК) для ввода-вывода и программа вызывается программой управления сообщениями, то значение имени-данного-3 не должно изменяться программой. Если это значение изменить, то выполнение любого оператора, использующего имя-коммуникация-1, будет неуспешным и значением имени-данного-6 будет код, означающий неизвестный источник или адресат (см. табл. 1).

(31) Для статьи CD (ОК) для ввода-вывода без фразы INITIAL (НАЧАЛЬНОГО) или статьи CD (ОК) для ввода-вывода с фразой INITIAL (НАЧАЛЬНОГО), но когда программа не вызывается программой управления сообщениями, до выпол-

GOST EXPERT ГОСТ 22558-89,

нения первого оператора, использующего имя-коммуникации-1, программа должна обеспечить символическое имя источника или адресата в имени-данного-3.

После выполнения первого оператора, использующего имя-коммуникации-1, значение данного имя-данного-3 не должно изменяться программой. Если это значение изменить, то выполнение любого оператора, использующего имя-коммуникации-1, будет неуспешным, и значением имени-данного-6 будет код, означающий неизвестный источник или адресат (см. табл. 1).

(32) В качестве значения имени-данного-4 система управления сообщениями указывает число позиций литер, заполненных в результате выполнения оператора RECEIVE (ПОЛУЧИТЬ).

Во время выполнения оператора SEND (ПОСЛАТЬ) система управления сообщениями рассматривает значение имени-данного-4 как число крайних левых позиций в поле, используемом в операторе SEND (ПОСЛАТЬ), из которого надо передавать данные (п. 3.6 настоящей части).

(33) Значение имени-данного-5 устанавливается системой управления сообщениями во время выполнения оператора RECEIVE (ПОЛУЧИТЬ) согласно следующим правилам.

а) Для оператора RECEIVE MESSAGE (ПОЛУЧИТЬ СООБЩЕНИЕ):

1) если обнаружен конец группы, то значение имени-данного-5 устанавливается равным 3;

2) если обнаружен конец сообщения, то значение имени-данного-5 устанавливается равным 2;

3) если передается часть сообщения, то значение имени-данного-5 устанавливается равным 0.

б) Для оператора RECEIVE SEGMENT (ПОЛУЧИТЬ СЕГМЕНТ):

1) если обнаружен конец группы, то значение имени-данного-5 устанавливается равным 3;

2) если обнаружен конец сообщения, то значение имени-данного-5 устанавливается равным 2;

3) если обнаружен конец сегмента, то значение имени-данного-5 устанавливается равным 1;

4) если передается только часть сообщения, то значение имени-данного-5 устанавливается равным 0.

в) Если одновременно удовлетворяются несколько из перечисленных выше условий, то значение имени-данного-5 определяется первым выполненным условием в порядке перечисления.

(34) Значение имени-данного-6 определяет ключ состояния выполнения предшествующих операторов DISABLE (ЗАПРЕТИТЬ), ENABLE (РАЗРЕШИТЬ), PURGE (ОЧИСТИТЬ), ACCEPT (ПРИНЯТЬ), SEND (ПОСЛАТЬ).

GOST EXPERT ГОСТ 22558-89,

Таблица 1

Комментарий	13	Значение кдюча состояния	DISABLE OUTPUT² (ЗАПРЕТИТЬ ВЫВОД²)	ENABLE OUTPUT² (РАЗРЕШИТЬ ВЫВОД²)	DISABLE INPUT² (ЗАПРЕТИТЬ ВВОД²)	DISABLE INPUT-O TERMINAL² (ЗАПРЕТИТЬ ВВОД/ВЫВОД С ТЕРМИНАЛА²)	DISABLE OUTPUT² (ЗАПРЕТИТЬ ВЫВОД²)	ENABLE INPUT-O TERMINAL² (РАЗРЕШИТЬ ВВОД/ВЫВОД С ТЕРМИНАЛА²)	ENABLE INPUT² (РАЗРЕШИТЬ ВВОД²)	ACCEPT MESSAGE COUNT (ПРИНЯТЬ ЧИСЛО СООБЩЕНИЙ)	PURGE² (ОЧИСТИТЬ²)	SEND (ПОСЛАТЬ) ДЛЯ СП (ОК) ДЛЯ ВЫВОДА	SEND (ПОСЛАТЬ) ДЛЯ СП (ОК) ДЛЯ ВВОДА-ВЫВОДА	RECEIVE (ПОЛУЧИТЬ)
Ошибки не обнаружено. Выполнение оператора завершено	00	00	×	×	×	×			×	×			×	×
Один или несколько адресатов запрещены. Выполнение оператора завершено (см. п. 2.2.6)	10	10									×	×	×	
Адресат запрещен. Никакие действия не предпринимаются	10	10											×	
Символический источник, одна или несколько очередей и адресатов запрещены/разрешены² (см. п. 2.2.6)	15	15	×	×	×	×			×	×				
Один или несколько адресатов неизвестны. Для известных адресатов действие завершается (см. п. 2.2.6)	20	20	×	×							×	×	×	

ГОСТ 22558-89,

Продолжение табл. 1

Значение кнопки состояния	RECEIVE (ПОЛУЧИТЬ)	SEND (ПОСЛАТЬ) для CD (ОК) для ввода-вывода	SEND (ПОСЛАТЬ) для CD (ОК) для вывода	PURGE (ОЧИСТИТЬ)	ACCEPT MESSAGE COUNT (ПРИНЯТЬ ЧИСЛО СООБЩЕНИЙ)	ENABLE INPUT (РАЗРЕШИТЬ ВВОД)	ENABLE INPUT/I-O TERMINAL (РАЗРЕШИТЬ ВВОД/ВВОД-ВЫВОД С ТЕРМИНАЛА)	ENABLE OUTPUT (РАЗРЕШИТЬ ВЫВОД)	DISABLE INPUT (ЗАПРЕТИТЬ ВВОД)	DISABLE INPUT/I-O TERMINAL (ЗАПРЕТИТЬ ВВОД/ВВОД-ВЫВОД С ТЕРМИНАЛА)	DISABLE OUTPUT (ЗАПРЕТИТЬ ВЫВОД)	Комментарий
1	2	3	4	5	6	7	8	9	10	11	12	13
20	×			×	×			×				Одна или несколько очередей или подочередей неизвестны. Никакие действия не предпринимаются
21	×					×			×			Символический источник неизвестен. Никакие действия не предпринимаются
30			×	×			×			×		Значение данного DESTINATION COUNT (ЧИСЛО АДРЕСАТОВ) недопустимо. Никакие действия не предпринимаются
40					×	×	×	×	×	×		Пароль недействителен. Никакие действия не предпринимаются
50		×	×									Длина текста больше чем длина посылаемого поля, предоставленного идентификатором

GOST EXPERT ГОСТ 22558-89,

Продолжение табл. 1

Комментарий	13	Значение ключа состояния	12	DISABLE OUTPUT² (ЗАПРЕТИТЬ ВЫВОД²)	11	DISABLE INPUT/I-O TERMINAL² (ЗАПРЕТИТЬ ВВОД/ВВОД-ВЫВОД С ТЕРМИНАЛА²)	10	DISABLE INPUT² (ЗАПРЕТИТЬ ВВОД²)	9	ENABLE OUTPUT² (РАЗРЕШИТЬ ВЫВОД²)	7	ENABLE INPUT/I-O TERMINAL² (РАЗРЕШИТЬ ВВОД/ВВОД-ВЫВОД С ТЕРМИНАЛА²)	7	ENABLE INPUT² (РАЗРЕШИТЬ ВВОД²)	6	ACCEPT MESSAGE COUNT (ПРИНЯТЬ ЧИСЛО СООБЩЕНИЙ)	5	RESET (ОЧИСТИТЬ²)	4	SEND (ПОСЛАТЬ) ДЛЯ СД (ОК) ДЛЯ ВЫВОДА	3	SEND (ПОСЛАТЬ) ДЛЯ СД (ОК) ДЛЯ ПРОЦЕДУР-ВЫВОДА	2	RECEIVE (ПОЛУЧИТЬ)	1
Часть сообщения с нулевым счетчиком литер или не определен идентификатор-1. Никакие действия не предпринимаются		60																			×	×	
Превышены возможности выходной очереди (см. п. 2.2.6)		65																		×			
Один или несколько адресатов не имеют порций, связанных с ними. Выполнение оператора завершается для других адресатов²		70																×					
Произошла комбинация по крайней мере хотя бы двух ключей состояния со значениями 10, 15 и 20 (см. п. 2.2.6)		80		×				×		×				×				×					
Состояния, определяемые реализацией.		9x																					

GOST EXPERT ГОСТ 22558-89,

Соответствие значения имени-данного-6 состоянию отражено в табл. 1.

2.2.5. Условия ключа состояния коммуникации

Табл. 1 указывает на возможные значения имени-данного-10 формата 1, имени-данного-3 формата 2 и имени-данного-6 формата 3 при завершении выполнения перечисленных в ней операторов. Символ × означает, что соответствующее значение ключа состояния имеет смысл для данного оператора. Символ[2] означает элемент уровня 2, недоступный на уровне 1.

2.2.6. Значение ключа ошибки

В табл. 2 показаны возможные значения имени-данного-4 формата 2 при выполнении перечисленных операторов. Символ × означает, что соответствующее значение ключа ошибки имеет смысл для данного оператора.

Символ[2] означает элемент уровня 2, недоступный на уровне 1.

Таблица 2

SEND (ПОСЛАТЬ)	PURGE[2] (ОЧИСТИТЬ[2])	ENABLE OUTPUT[2] (РАЗРЕШИТЬ ВЫВОД[2])	DISABLE OUTPUT[2] (ЗАПРЕТИТЬ ВЫВОД[2])	Значение ключа ошибки	Комментарий
×	×	×	×	0	Ошибка не обнаружена
×	×	×	×	1	Символический адресат не известен
×	×			2	Символический адресат запрещен
	×			4	Ни одна из частей сообщения не имеет символического адресата[2]
		×	×	5	Символический адресат уже был разрешен/запрещен[2]
×				6	Возможности выходной очереди превышены
				7—9	Зарезервированы для дальнейшего использования
				A—Z	Условия, определяемые реализацией

GOST EXPERT ГОСТ 22558-89,

3. РАЗДЕЛ ПРОЦЕДУР В МОДУЛЕ КОММУНИКАЦИЙ

3.1. Оператор ACCEPT MESSAGE COUNT (ПРИНЯТЬ ЧИСЛО СООБЩЕНИЙ)

3.3.1. Назначение

Оператор ACCEPT MESSAGE COUNT (ПРИНЯТЬ ЧИСЛО СООБЩЕНИЙ) делает доступным число полных сообщений в очереди.

3.1.2. Общий формат

ACCEPT имя-коммуникации-1 MESSAGE COUNT

ПРИНЯТЬ ЧИСЛО СООБЩЕНИЙ имя-коммуникации-1

3.1.3. Синтаксические правила

(1) Имя-коммуникации-1 должно относиться к описанию коммуникации для ввода.

3.1.4. Общие правила

(1) Оператор ACCEPT MESSAGE COUNT (ПРИНЯТЬ ЧИСЛО СООБЩЕНИЙ) обновляет счетчик числа сообщений, связанный с именем-коммуникации-1, для указания числа полных сообщений, существующих в структуре очереди, определяемой содержимым данных, указанных от имени-данного-1 во фразе SYMBOLIC QUEUE (СИМВОЛИЧЕСКАЯ ОЧЕРЕДЬ) | по имя-данного-4 во фразе SYMBOLIC SUB-QUEUE-3 (СИМВОЛИЧЕСКАЯ ПОДОЧЕРЕДЬ-3) | в области, на которую ссылается имя-коммуникации-1.

(2) Во время выполнения оператора ACCEPT MESSAGE COUNT (ПРИНЯТЬ ЧИСЛО СООБЩЕНИЙ) область, определенная статьей описания коммуникации, должна содержать, по крайней мере, имя символической очереди, которую надо проверить. Проверка условия приводит к обновлению областей, определенных именем-данного-10 во фразе STATUS KEY (КЛЮЧ СОСТОЯНИЯ) и именем-данного-11 во фразе MESSAGE COUNT (ЧИСЛО СООБЩЕНИЙ), связанных с данной статьей описания коммуникаций (см. п. 2.1 настоящей части).

3.2. Оператор DISABLE (ЗАПРЕТИТЬ)

3.2.1. Назначение

Оператор DISABLE (ЗАПРЕТИТЬ) извещает систему управления сообщениями о том, что она должна запретить передачу между указанными выходными очередями и адресатами или указанными источниками и входными очередями или между программой и указанным источником или адресатом для ввода-вывода.

Фраза WITH KEY (КЛЮЧ) рассматривается в настоящем стандарте как устаревшая и будет удалена из следующей редакции стандарта.

GOST EXPERT ГОСТ 22558-89,

3.2.2. Общий формат

DISABLE { INPUT [TERMINAL] / I-O TERMINAL / OUTPUT } имя-коммуникации-1
[WITH KEY { идентификатор-1 / литерал-1 }]

ЗАПРЕТИТЬ { ВВОД [С ТЕРМИНАЛА] / ВВОД-ВЫВОД С ТЕРМИНАЛА / ВЫВОД }
имя-коммуникации-1
[КЛЮЧ { идентификатор-1 / литерал-1 }]

3.2.3. Синтаксические правила

(1) Имя-коммуникации-1 должно относиться к описанию коммуникации для ввода, если в операторе указана фраза INPUT (ВВОД).

(2) Имя-коммуникации-1 должно относиться к описанию коммуникации для ввода-вывода, если указана фраза I-O TERMINAL (ВВОД-ВЫВОД С ТЕРМИНАЛА).

(3) Имя-коммуникации должно относиться к описанию коммуникации для вывода, если в операторе указана фраза OUTPUT (ВЫВОД).

(4) Литерал-1 и значение идентификатора-1 должны быть определены как буквенно-цифровые.

3.2.4. Общие правила

(1) Оператор DISABLE (ЗАПРЕТИТЬ) обеспечивает логическое рассоединение системы управления сообщениями с заданными источниками или адресатами. Если логическое рассоединение уже имеет место или если оно должно обеспечиваться какими-либо средствами, внешними к программе, оператор DISABLE (ЗАПРЕТИТЬ) в этой программе не требуется.

Если выполняется оператор DISABLE (ЗАПРЕТИТЬ), в котором указан уже рассоединенный источник или адресат, за исключением того, что значение ключа состояния указывает на это условие, никаких действий не производится.

Оператор DISABLE (ЗАПРЕТИТЬ) не влияет на логический путь передачи данных между программой на Коболе и системой управления сообщениями.

(2) Система управления сообщениями обеспечит, чтобы выполнение оператора DISABLE (ЗАПРЕТИТЬ) приводило к ло-

GOST EXPERT ГОСТ 22558-89,

гическому рассоединению в кратчайшее время, когда источник или адресат становится неактивным. Выполнение оператора DISABLE (ЗАПРЕТИТЬ) никогда не прерывает передачу сообщения на терминал или с него.

(3) Фраза INPUT (ВВОД) без фразы TERMINAL (С ТЕРМИНАЛА) указывает на прекращение логической связи между очередями и подочередями, указанными содержимым от имени-данного-1 во фразе SYMBOLIC QUEUE (СИМВОЛИЧЕСКАЯ ОЧЕРЕДЬ) по имя-данного-4 во фразе SYMBOLIC SUB-QUEUE-3 (СИМВОЛИЧЕСКАЯ ПОДОЧЕРЕДЬ-3) области, на которую ссылается имя-коммуникации-1, и всеми источниками, связанными с ними.

(4) Фраза INPUT (ВВОД) с необязательной фразой TERMINAL (С ТЕРМИНАЛА) указывает на прекращение логической связи между источником, определенным значением имени-данного-7 во фразе SYMBOLIC SOURCE (СИМВОЛИЧЕСКИЙ ИСТОЧНИК), и всеми очередями и подочередями.

(5) Фраза I-O TERMINAL (ВВОД-ВЫВОД С ТЕРМИНАЛА) указывает на прекращение логической связи между источником (определенным значением имени-данного-3 во фразе SYMBOLIC TERMINAL (СИМВОЛИЧЕСКИЙ ТЕРМИНАЛ), и программой.

(6) Фраза OUTPUT (ВЫВОД) указывает на прекращение логической связи для всех адресатов, определенных значениями каждого из экземпляров имени-данного-5, количество которых определяется значением имени-данного-1 из области, на которую ссылается имя-коммуникации-1.

(7) Литерал-1 или значение идентификатора-1 будет сравниваться с паролем, заданым в системе. Оператор DISABLE (ЗАПРЕТИТЬ) будет выполняться, если только литерал-1 или значение идентификатора-1 совпадает с системным паролем. В противном случае лишь обновляется значение данного STATUS KEY (КЛЮЧ СОСТОЯНИЯ) в области, на которую ссылается имя-коммуникации-1.

Система управления сообщениями должна уметь обрабатывать пароль длиной от 1 до 10 литер включительно.

3.3. **Оператор ENABLE (РАЗРЕШИТЬ)**

3.3.1. Назначение

Оператор ENABLE (РАЗРЕШИТЬ) сообщает системе управления сообщениями о разрешении обмена данными между очередями для вывода и адресатами или заданными источниками и очередями для ввода или между программой и одним заданным источником или адресатом для ввода-вывода. Фраза WITH KEY (КЛЮЧ) рассматривается в настоящем стандарте как устаревший элемент и будет удалена в следующей редакции.

GOST ГОСТ 22558-89,

3.3.2. Общий формат

```
ENABLE { INPUT [TERMINAL] }
       { I-O TERMINAL     } имя-коммуникации-1
       { OUTPUT           }
  [ WITH KEY { идентификатор-1 } ]
             { литерал-1       }

РАЗРЕШИТЬ { ВВОД [С ТЕРМИНАЛА]   }
          { ВВОД-ВЫВОД С ТЕРМИНАЛА } имя-коммуникации-1
          { ВЫВОД                }
  [ КЛЮЧ { идентификатор-1 } ]
         { литерал-1       }
```

3.3.3. Синтаксические правила

(1) Имя-коммуникации-1 должно относиться к описанию коммуникации для ввода, если задана фраза INPUT (ВВОД).

(2) Имя-коммуникации-1 должно относиться к описанию коммуникации для ввода-вывода, если задана фраза I-O TERMINAL (ВВОД-ВЫВОД С ТЕРМИНАЛА).

(3) Имя-коммуникации-1 должно относиться к описанию коммуникации для вывода, если задана фраза OUTPUT (ВЫВОД).

(4) Литерал-1 и значение идентификатора-1 должны быть определены как буквенно-цифровые.

3.3.4. Общие правила

(1) Оператор ENABLE (РАЗРЕШИТЬ) обеспечивает логическое соединение системы управления сообщениями с заданными источниками или адресатами. Если это логическое соединение уже имеет место или обеспечивается какими-либо другими средствами, внешними по отношению к этой программе, то оператор ENABLE (РАЗРЕШИТЬ) в ней не требуется.

Никаких действий не производится, если оператор ENABLE (РАЗРЕШИТЬ) выполняется с указанным источником или адресатом, которые уже соединены, за исключением того, что значение ключа состояния указывает на это условие. Оператор ENABLE (РАЗРЕШИТЬ) не влияет на логический путь передачи данных между программой на Коболе и системой управления сообщениями.

(2) Фраза INPUT (ВВОД) без необязательной фразы TERMINAL (С ТЕРМИНАЛА) указывает на активизацию логического пути между очередью и подочередями, определенными

GOST EXPERT ГОСТ 22558-89,

значением от имени-данного-1 во фразе SYMBOLIC QUEUE (СИМВОЛИЧЕСКАЯ ОЧЕРЕДЬ) по имя-данного-4 во фразе SYMBOLIC SUB-QUEUE-3 (СИМВОЛИЧЕСКАЯ ПОДОЧЕРЕДЬ-3), из области, на которую ссылается имя-коммуникации-1, и всеми источниками, связанными с ними.

(3) Фраза INPUT (ВВОД) с необязательной фразой TERMINAL (С ТЕРМИНАЛА) указывает на активизацию логического пути между источником, определенным значением имени-данного-7, во фразе SYMBOLIC SOURCE (СИМВОЛИЧЕСКИЙ ИСТОЧНИК), и всеми ему соответствующими очередями и подочередями.

(4) Фраза I-O TERMINAL (ВВОД-ВЫВОД С ТЕРМИНАЛА) указывает на активизацию логического пути между источником, определенным значением имени-данного-3 во фразе SYMBOLIC TERMINAL (СИМВОЛИЧЕСКИЙ ТЕРМИНАЛ), и программой.

(5) Фраза OUTPUT (ВЫВОД) указывает на активизацию логических путей для всех адресатов, определенных значениями каждого из экземпляров имени-данного-5, количество которых определяется значением имени-данного-1 из области, на которую ссылается имя-коммуникации-1.

(6) Литерал-1 или значение идентификатора-1 сравнивается с паролем, заданным в системе. Если литерал-1 или значение идентификатора-1 совпадает с этим паролем, оператор ENABLE (РАЗРЕШИТЬ) будет выполнен. В противном случае лишь обновляется значение данного STATUS KEY (КЛЮЧ СОСТОЯНИЯ), связанного с именем-коммуникации-1.

Система управления сообщениями должна уметь обрабатывать пароль длиной от 1 до 10 литер включительно.

3.4. **Оператор PURGE (ОЧИСТИТЬ)**

3.4.1. Назначение

Оператор PURGE (ОЧИСТИТЬ) исключает из системы управления сообщениями незаконченное сообщение, переданное одним или более оператором SEND (ПОСЛАТЬ).

3.4.2. Общий формат

PURGE имя-коммуникации-1

ОЧИСТИТЬ имя-коммуникации-1

3.4.3. Синтаксические правила

(1) Имя-коммуникации-1 должно относиться к статье CD (OK) для вывода или к статье CD (OK) для ввода-вывода.

3.4.4. Общие правила

(1) Выполнение оператора PURGE (ОЧИСТИТЬ) указывает системе управления сообщениями уничтожить все неоконченные сообщения, ожидающие передачи адресатам, определенным в статье CD (OK) для имени-коммуникации-1.

GOST EXPERT ГОСТ 22558-89,

(2) Сообщение, связанное с EMI (ИКЩ) или EGI (ИКГ), не затрагивается при выполнении оператора PURGE (ОЧИСТИТЬ).

(3) Значение ключа состояния и ключа ошибки, если они используются, из области, на которую ссылается имя-коммуникации-1, обновляются системой управления сообщениями (см. п. 2.2 настоящей части).

3.5. Оператор RECEIVE (ПОЛУЧИТЬ)

3.5.1. Назначение

Оператор RECEIVE (ПОЛУЧИТЬ) делает доступным сообщение | или сегмент сообщения | и соответствующую информацию о них.

3.5.2. Общий формат

RECEIVE имя-коммуникации-1 { MESSAGE / SEGMENT } INTO идентификатор-1

[NO DATA повелительный-оператор-1]

[WITH DATA повелительный-оператор-2]

[END-RECEIVE]

ПОЛУЧИТЬ { СЕГМЕНТ / СООБЩЕНИЕ } имя-коммуникации-1 В идентификатор-1

[НЕТ ДАННЫХ повелительный-оператор-1]

[ЕСТЬ ДАННЫЕ повелительный-оператор-2]

[КОНЕЦ-ПОЛУЧИТЬ]

3.5.3. Синтаксические правила

(1) Имя-коммуникации-1 должно относиться к описанию коммуникации для ввода | или для ввода-вывода |.

3.5.4. Общие правила

(1) Если имя-коммуникации-1 относится к описанию коммуникации для ввода, то значения данных, указанных от имени-данного-1 во фразе SYMBOLIC QUEUE (СИМВОЛИЧЕСКАЯ ОЧЕРЕДЬ) | по имя-данного-4 во фразе SYMBOLIC SUB-QUEUE-3 | (СИМВОЛИЧЕСКАЯ ПОДОЧЕРЕДЬ-3), | из области, на которую ссылается имя-коммуникации-1, определяют структуру очереди, содержащей сообщение (см. п. 2.2 настоящей части).

GOST EXPERT ГОСТ 22558-89,

(2) Если имя-коммуникации-1 относится к описанию коммуникации для ввода-вывода, то значение данного, указанного именем-данного-3 во фразе SYMBOLIC TERMINAL (СИМВОЛИЧЕСКИЙ ТЕРМИНАЛ), связанного с именем-коммуникации-1, определяет источник сообщения.

(3) Сообщение | , сегмент сообщения либо часть сообщения | или сегмента | передается в область, указанную идентификатором-1, с выравниванием влево; дополнение сообщения до размера области пробелами не производится.

(4) Если при выполнении оператора RECEIVE (ПОЛУЧИТЬ) система управления сообщениями делает данные доступными в области, определенной идентификатором-1, фраза NO DATA (НЕТ ДАННЫХ) игнорируется и управление передается в конец оператора RECEIVE (ПОЛУЧИТЬ) или повелительному-оператору-2, если задана фраза WITH DATA (ЕСТЬ ДАННЫЕ). Если управление передается повелительному-оператору-2, то выполнение продолжается согласно правилам, определенным для каждого оператора повелительного-оператора-2. Если это оператор ветвления процедур или условный оператор который явно передает управление, то передача управления осуществляется согласно правилам для-используемых операторов; в противном случае после выполнения повелительного-оператора-2 управление передается в конец оператора RECEIVE (ПОЛУЧИТЬ).

(5) Если при выполнении оператора RECEIVE (ПОЛУЧИТЬ) система управления сообщениями не делает данные доступными в области, определенной идентификатором-1, выполняется одно из трех перечисленных ниже действий. Условия, при которых данные недоступны, определяются реализацией.

а) Если в операторе RECEIVE (ПОЛУЧИТЬ) задана фраза NO DATA (НЕТ ДАННЫХ), то выполнение оператора RECEIVE (ПОЛУЧИТЬ) заканчивается указанием на завершение действия и управление передается повелительному-оператору-1. Выполнение продолжается согласно правилам, определенным для каждого оператора, указанного в повелительном-операторе-1.

Если это оператор ветвления процедур или условный оператор, который явно передает управление, то передача управления осуществляется согласно правилам для используемых операторов; в противном случае после выполнения повелительного-оператора-1 управление передается в конец оператора RECEIVE (ПОЛУЧИТЬ), а фраза WITH DATA (ЕСТЬ ДАННЫЕ), если указана, игнорируется.

б) Если фраза NO DATA (НЕТ ДАННЫХ) не задана в операторе RECEIVE (ПОЛУЧИТЬ), то выполнение объектной программы приостанавливается до тех пор, пока данные не будут доступны в области, определенной идентификатором-1.

GOST EXPERT ГОСТ 22558-89,

в) Если одна или несколько очередей [или подочередей] не известны системе управления сообщениями, то устанавливается соответствующий код и управление передается как в случае доступности данных.

(6) При каждом выполнении оператора RECEIVE (ПОЛУЧИТЬ) данные, определенные именем-коммуникации-1, обновляются системой управления сообщениями (см. п. 2.2 настоящей части).

(7) Однократное выполнение оператора RECEIVE (ПОЛУЧИТЬ) никогда не приводит к передаче более одного сообщения, если задана фраза MESSAGE (СООБЩЕНИЕ), или одного сегмента, если задана фраза SEGMENT (СЕГМЕНТ), в область, определенную идентификатором-1. Вместе с тем, система управления сообщениями не передает никакой части сообщения объектной программе до тех пор, пока сообщение не станет полностью доступно во входной очереди, даже если задана фраза SEGMENT (СЕГМЕНТ) оператора RECEIVE (ПОЛУЧИТЬ).

(8) При использовании фразы MESSAGE (СООБЩЕНИЕ) индикаторы конца сегмента игнорируются, а передача данных происходит по следующим правилам:

а) если размер сообщения совпадает с размером области, определенной идентификатором-1, сообщение запоминается в этой области;

б) если размер сообщения меньше размера области, то оно выравнивается к самой левой позиции литеры области идентификатора-1; позиции, не занятые сообщением, не изменяются;

в) если размер сообщения больше размера области, то сообщение заполняет область идентификатора-1 слева направо, начиная с самой левой литеры сообщения. В уровне 1 расположение остатка сообщения не определено. В уровне 2 остаток можно передать в область, определенную идентификатором-1, следующими операторами RECEIVE (ПОЛУЧИТЬ), относящимися к той же очереди, подочереди и так далее. Остаток сообщения рассматривается как новое сообщение и к нему применимы приведенные выше правила 8а, 8б, 8в;

г) если с текстом передаваемого оператором RECEIVE (ПОЛУЧИТЬ) сообщения связан индикатор конца группы, то предполагается существование индикатора конца сообщения.

(9) Если используется фраза SEGMENT (СЕГМЕНТ), применяются следующие правила передачи данных:

а) если размер сегмента совпадает с размером области, определенной идентификатором-1, сегмент запоминается в этой области;

б) если размер сегмента меньше размера области, то сегмент выравнивается к самой левой позиции литеры области идентн-

GOST EXPERT ГОСТ 22558-89,

фикатора-1; позиции, не занятые сообщением, не изменяются;

в) если размер сегмента больше размера области идентификатора-1, сегмент заполняет область слева направо, начиная с самой левой литеры сегмента. Остаток сегмента может быть передан в область, определенную идентификатором-1, следующими операторами RECEIVE (ПОЛУЧИТЬ), относящимися к той же очереди, подочереди и так далее. Остаток сегмента рассматривается как новый сегмент и к нему применимы приведенные выше правила 9а, 9б, 9в;

г) если с текстом, доступным по оператору RECEIVE (ПОЛУЧИТЬ), связан индикатор конца группы или индикатор конца сообщения, предполагается существование индикатора конца сегмента.

(10) Когда выполнение оператора RECEIVE (ПОЛУЧИТЬ) делает доступным часть сообщения, только последующее выполнение операторов RECEIVE (ПОЛУЧИТЬ) в этом же исполняемом модуле может привести к передаче оставшейся части сообщения.

(11) Фраза END-RECEIVE (КОНЕЦ-ПОЛУЧИТЬ) ограничивает область действия оператора RECEIVE (ПОЛУЧИТЬ) (см. ч. 4, п. 6.4.3).

3.6. **Оператор SEND (ПОСЛАТЬ)**

3.6.1. Назначение

Оператор SEND (ПОСЛАТЬ) приводит к передаче сообщения, сегмента либо части сообщения или сегмента в одну /или несколько/ выходных очередей посредством программы управления сообщениями.

3.6.2. Общий формат

Формат 1

SEND имя-коммуникации-1 FROM идентификатор-1

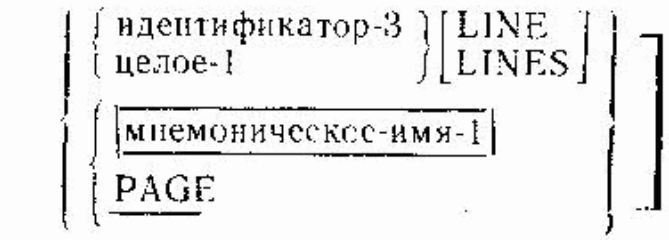
ПОСЛАТЬ имя-коммуникации-1 ИЗ ПОЛЯ идентификатор-1

Формат 2

SEND имя-коммуникации-1 [FROM идентификатор-1]

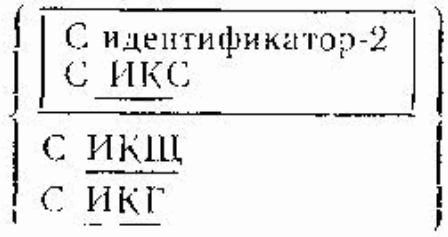
{ WITH идентификатор-2 / WITH ESI / WITH EMI / WITH EGI } [{ BEFORE / AFTER } ADVANCING ...]

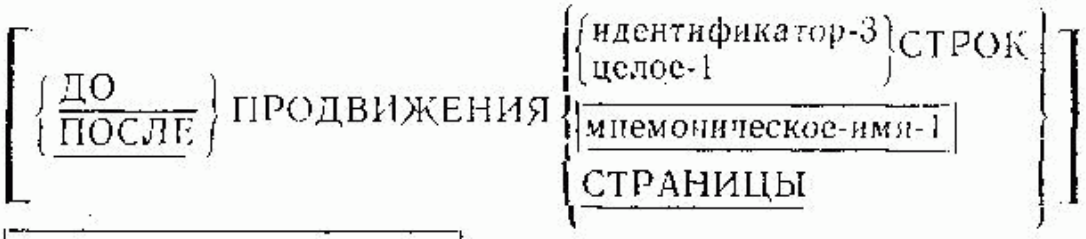
GOST EXPERT ГОСТ 22558-89,

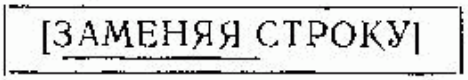




ПОСЛАТЬ имя-коммуникации-1 [ИЗ ПОЛЯ идентификатор-1]







3.6.3. Синтаксические правила

(1) Имя-коммуникации-1 должно относиться к описанию коммуникации для вывода или для ввода-вывода.

(2) Идентификатор-2 должен представлять целое из одной цифры без знака.

(3) Идентификатор-3 должен представлять целое.

(4) Если используется мнемоническое-имя-1, оно идентифицирует конкретные свойства, определяемые реализацией. Мнемоническое-имя-1 определяется в параграфе SPECIAL-NAMES (СПЕЦИАЛЬНЫЕ-ИМЕНА) раздела оборудования.

(5) Целое-1 или значение идентификатора-3 может быть нулем.

3.6.4. Общие правила

Все форматы

(1) Когда приемное коммуникационное устройство ориентировано на фиксированный размер строки (например дисплей, перфо-

GOST EXPERT ГОСТ 22558-89,

карточное устройство вывода, печатающее устройство), выполняется следующее:

а) каждое сообщение | или сегмент сообщения | начинается в крайней левой позиции литеры физической строки;

б) если размер сообщения | или сегмента сообщения | меньше, чем размер физической строки, сообщение дополняется пробелами справа;

в) избыточные литеры сообщения | или сегмента сообщения | не усекаются. После заполнения физической строки она выводится на устройство. Избыточные литеры переносятся на следующую строку.

(2) Когда приемное коммуникационное устройство ориентировано на обработку сообщений переменной длины (как, например, перфоленточное устройство вывода, другая ЭВМ), каждое сообщение | или сегмент сообщения | начинается со следующей доступной позиции литеры коммуникационного устройства.

(3) Во время выполнения оператора SEND (ПОСЛАТЬ) система управления сообщениями интерпретирует значение длины текста в области, к которой относится имя-коммуникации-1, как заданное пользователем число позиций литер, начиная от крайней слева, в области идентификатора-1, из которой будут передаваться данные.

Если значение длины текста равно нулю, то никакие литеры идентификатора-1 не передаются.

Если значение длины текста лежит вне диапазона чисел от нуля до размера идентификатора-1 включительно, то значение ключа состояния указывает на ошибку и данные не передаются.

(4) Во время выполнения оператора SEND (ПОСЛАТЬ) значение ключа состояния, связанного с именем-коммуникации-1, обновляется системой управления сообщениями.

(5) При наличии в значении идентификатора-1 специальных символов управления результат выполнения оператора не определен.

(6) | Однократное выполнение оператора SEND (ПОСЛАТЬ) формата 1 приводит к передаче в систему управления сообщениями только одной части сообщения или сегмента сообщения. |

Однократное выполнение оператора SEND (ПОСЛАТЬ) формата 2 передает системе управления сообщениями не более одного сообщения | или сегмента сообщения | в соответствии со | значением идентификатора-2 или | заданным индикатором | ESI (ИКС), | EMI (ИКЩ), EGI (ИКГ).

GOST EXPERT ГОСТ 22558-89,

Однако система управления сообщениями передаст сообщение коммуникационному устройству только тогда, когда все сообщение будет полностью у нее.

(7) Во время выполнения единицы исполнения размещение части сообщения, не законченного по EMI (ИКЩ), EGI (ИКГ) или не уничтоженного оператором PURGE (ОЧИСТИТЬ), не определено.

Для системы управления сообщениями такое сообщение логически не существует и, следовательно, не может быть послано адресату.

(8) Если выполнение оператора SEND (ПОСЛАТЬ) передает системе управления сообщениями часть сообщения, то только последующее выполнение оператора SEND (ПОСЛАТЬ) в той же единице исполнения может привести к передаче оставшейся части сообщения.

Формат 2

(9) Значение идентификатора-2 указывает, что значение идентификатора-1, если он определен, имеет связанный с ним индикатор конца сегмента, индикатор конца сообщения, индикатор конца группы или не имеет индикатора (что означает передачу части сегмента или сообщения). Если идентификатор-1 не определен, то только индикатор пересылается системе управления сообщениями.

Значение идентификатора-2	Тип индикатора, связанный со значением идентификатора-1	Комментарий
0	Нет индикатора	Часть сообщения или сегмента
1	ESI (ИКС) (индикатор конца сегмента)	Конец текущего сегмента
2	EMI (ИКЩ) (индикатор конца сообщения)	Конец текущего сообщения
3	EGI (ИКГ) (индикатор конца группы)	Конец текущей группы сообщений

Любое другое значение идентификатора-2 будет трактоваться как нуль. В этом случае значение ключа состояния, связанного с именем-коммуникации-1, указывает на ошибку и данные не передаются.

(10) Фраза WITH EGI (С ИКГ) указывает системе управления сообщениями, что закончена группа сообщений.

GOST EXPERT ГОСТ 22558-89,

Фраза WITH EMI (С ИКЩ) указывает системе управления сообщениями, что сообщение закончено.

Фраза WITH ESI (С ИКС) указывает системе управления сообщениями что сегмент сообщения закончен.

Система управления сообщениями распознает эти индикаторы и устанавливает необходимые средства управления группой, сообщением и сегментом.

(11) Иерархия индикаторов конца следующая: EGI (ИКГ), EMI (ИКЩ), ESI (ИКС). Индикатору EGI (ИКГ) могут не предшествовать ESI (ИКС), EMI (ИКЩ). Индикатору EMI (ИКЩ) может не предшествовать ESI (ИКС).

(12) Фраза ADVANCING (ПРОДВИЖЕНИЯ) обеспечивает вертикальное позиционирование каждого сообщения или сегмента сообщения на коммуникационном устройстве, на котором возможно вертикальное позиционирование. Если вертикальное позиционирование невозможно на данном устройстве, система управления сообщениями игнорирует заданное или подразумеваемое вертикальное позиционирование.

(13) Если задан идентификатор-2 и его значение равно нулю, то система управления сообщениями игнорирует фразу ADVANCING (ПРОДВИЖЕНИЯ) и фразу REPLACING (ЗАМЕНЯЯ), если они указаны.

(14) На устройстве, где возможно вертикальное позиционирование, а фраза ADVANCING (ПРОДВИЖЕНИЯ) не задана, реализация должна обеспечивать автоматическое продвижение, как если бы пользователь указал AFTER ADVANCING 1 LINE (ПОСЛЕ ПРОДВИЖЕНИЯ 1 СТРОК).

(15) Если явно или неявно указана фраза ADVANCING (ПРОДВИЖЕНИЯ) и вертикальное позиционирование возможно, применяются следующие правила:

а) если задан идентификатор-3 или целое, то литеры, передаваемые на коммуникационное устройство, будут перемещены вертикально вниз на количество строк, равное значению идентификатора-3 или целого;

б) если значение данного, на которое ссылается идентификатор-3, отрицательно, результат не определен;

в) если задано мнемоническое-имя-1, литеры, передаваемые на устройство, будут размещены по правилам, определенным реализацией для этого устройства;

GOST EXPERT ГОСТ 22558-89,

г) если используется фраза BEFORE (ДО), сообщение | или | | сегмент сообщения | передается на коммуникационное устройство до вертикального позиционирования в соответствии с указанными выше правилами;

д) если используется фраза AFTER (ПОСЛЕ), сообщение | или сегмент | передается на коммуникационное устройство после вертикального позиционирования в соответствии с указанными выше правилами;

ж) если задана фраза PAGE (СТРАНИЦЫ), то литеры, передаваемые на коммуникационное устройство, будут переданы на устройство до или после (в зависимости от используемого варианта) того, как устройство переведено на следующую страницу.

Если фраза PAGE (СТРАНИЦЫ) задана, но понятие страницы не имеет смысла для данного устройства, то продвижение должно быть обеспечено реализацией как если бы пользователь указал фразу BEFORE (ДО) или AFTER ADVANCING 1 LINE (ПОСЛЕ ПРОДВИЖЕНИЯ 1 СТРОК) (в зависимости от используемого варианта).

(16) Если принимающее коммуникационное устройство является устройством отображения символов, на котором возможно на одной и той же позиции представлять одну или более литер и, если устройство позволяет выбрать: либо последовательность литер накладывается на литеру, уже выведенную на дисплей, либо литера замещает другую литеру, предварительно выведенную на строку дисплея, то:

| а) если фраза REPLACING (ЗАМЕНЯЯ) указана, то литеры, передаваемые при помощи оператора SEND (ПОСЛАТЬ), замещают все литеры, ранее переданные на эту же строку, начиная с крайней левой позиции;

б) | если фраза REPLACING (ЗАМЕНЯЯ) не указана, то | литеры, передаваемые при помощи оператора SEND (ПОСЛАТЬ), будут накладываться на литеры, предварительно пересланные на эту же строку, начиная с крайней левой позиции.

(17) Если принимающее коммуникационное устройство не обеспечивает замещения символов, | независимо от того, указана фраза REPLACING (ЗАМЕНЯЯ) или нет, | литеры, переданные при помощи оператора SEND (ПОСЛАТЬ), могут быть наложены на символы, предварительно переданные на ту же строку, начиная с крайней левой позиции строки.

(18) Если принимающее коммуникационное устройство не обеспечивает наложения одного или более символов на одну и ту же позицию, | независимо от того указана фраза REPLACING (ЗАМЕНЯЯ) или нет, | символы, переданные при помощи оператора SEND (ПОСЛАТЬ), замещают все символы, предваритель-

GOST EXPERT ГОСТ 22558-89,

но переданные на ту же строку, начиная с крайней левой позиции строки.

Часть 15. МОДУЛЬ ОТЛАДКИ

1. ВВЕДЕНИЕ В МОДУЛЬ ОТЛАДКИ

1.1. **Назначение**

Модуль отладки предоставляет средства для описания пользователем своего алгоритма отладки, включающего условия, по которым можно следить за данными или процедурами во время выполнения объектной программы.

Выбор объектов, подлежащих слежению, и соответствующей выдаваемой информации является обязанностью пользователя. Средства отладки в Коболе обеспечивают удобный доступ к соответствующей информации.

Модуль отладки является устаревшим элементом в настоящем стандарте и будет удален в следующей редакции стандарта.

1.2. **Характеристика уровней**

Уровень 1 отладки предоставляет основные средства отладки, включающие выборочное слежение за процедурами.

Уровень 2 отладки предоставляет полные средства отладки, имеющиеся в Коболе.

1.3. **Понятия языка**

1.3.1. Возможности отладки

Модуль отладки в языке Кобол поддерживает следующие возможности:

а) переключатель времени компиляции — фраза WITH DEBUGGING MODE (В РЕЖИМЕ ОТЛАДКИ);

б) переключатель, действующий во время исполнения;

в) оператор USE FOR DEBUGGING (ИСПОЛЬЗОВАТЬ ДЛЯ ОТЛАДКИ);

г) специальный регистр DEBUG-ITEM (ДАННЫЕ-ОТЛАДКИ).

1.3.2. Специальный регистр DEBUG-ITEM (ДАННЫЕ-ОТЛАДКИ)

Зарезервированное слово DEBUG-ITEM (ДАННЫЕ-ОТЛАДКИ) является именем специального регистра, содержащего отладочную информацию и автоматически порождаемого реализацией. Для каждой программы порождается единственный регистр DEBUG-ITEM (ДАННЫЕ-ОТЛАДКИ). DEBUG-ITEM (ДАННЫЕ-ОТЛАДКИ) имеет подчиненные данные, имена которых также являются зарезервированными словами.

1.3.3. Переключатель времени компиляции

Переключатель времени компиляции управляет компиляцией отладочных строк; он устанавливается фразой WITH DEBUGGING

GOST EXPERT ГОСТ 22558-89,

MODE (В РЕЖИМЕ ОТЛАДКИ), которая должна быть указана в параграфе SOURCE-COMPUTER (ИСХОДНАЯ-МАШИНА). Если эта фраза в программе указана, все отладочные строки компилируются как указано в этом разделе документа. Если эта фраза в программе не указана, все отладочные строки и секции рассматриваются при компиляции как строки комментариев.

1.3.4. Переключатель, действующий во время исполнения

Переключатель, действующий во время исполнения, динамически активизирует отладочные коды, встроенные компилятором. К этому переключателю нельзя обращаться в Кобол-программе, он управляется вне среды Кобола. Если он «включен», разрешаются все указанные в исходной программе отладочные действия. Если этот переключатель «выключен», действия, описанные в п. 3.2 настоящей части, подавляются; при этом нет необходимости перекомпиляции исходной программы. Если в исходной программе фраза WITH DEBUGGING MODE (В РЕЖИМЕ ОТЛАДКИ) не указана, то переключатель, действующий во время исполнения, не оказывает влияния на выполнение объектной программы.

2. РАЗДЕЛ ОБОРУДОВАНИЯ В МОДУЛЕ ОТЛАДКИ

2.1. Фраза WITH DEBUGGING MODE (В РЕЖИМЕ ОТЛАДКИ)

2.1.1. Назначение

Фраза WITH DEBUGGING MODE (В РЕЖИМЕ ОТЛАДКИ) указывает, что все отладочные секции должны компилироваться Если эта фраза не указана, все отладочные секции компилируются так, как если бы они были строками комментариев.

2.1.2. Общий формат

SOURCE-COMPUTER. [имя-машины [WITH DEBUGGING MODE] .]

ИСХОДНАЯ-МАШИНА. [имя-машины [В РЕЖИМЕ ОТЛАДКИ] .]

2.1.3. Общие правила

(1) Если в параграфе SOURCE-COMPUTER (ИСХОДНАЯ-МАШИНА) секции конфигурации программы указана фраза WITH DEBUGGING MODE (В РЕЖИМЕ ОТЛАДКИ), компилируются все операторы USE FOR DEBUGGING (ИСПОЛЬЗОВАТЬ ДЛЯ ОТЛАДКИ).

(2) Если в секции конфигурации программы в параграфе SOURCE-COMPUTER (ИСХОДНАЯ-МАШИНА) фраза WITH DEBUGGING MODE (В РЕЖИМЕ ОТЛАДКИ) не указана, все операторы USE FOR DEBUGGING (ИСПОЛЬЗОВАТЬ ДЛЯ ОТЛАДКИ) и все соответствующие отладочные секции рассматриваются при компиляции как строки комментариев.

GOST EXPERT ГОСТ 22558-89,

3. РАЗДЕЛ ПРОЦЕДУР В МОДУЛЕ ОТЛАДКИ

3.1. **Общее описание**

Если оператор USE FOR DEBUGGING (ИСПОЛЬЗОВАТЬ ДЛЯ ОТЛАДКИ) из модуля отладки задается в исходной Кобол-программе, раздел процедур содержит декларативные процедуры. Ниже показан общий формат раздела процедур при задании оператора USE FOR DEBUGGING (ИСПОЛЬЗОВАТЬ ДЛЯ ОТЛАДКИ)

```
PROCEDURE DIVISION.
DECLARATIVES.
  {имя-секции SECTION.
   оператор USE FOR DEBUGGING.
   [имя-параграфа.
     [предложение] ...] ...}...
END DECLARATIVES.
  {имя-секции SECTION.
   [имя-параграфа.
   [предложение] ...] ...}...
РАЗДЕЛ ПРОЦЕДУР.
ДЕКЛАРАТИВЫ.
  {СЕКЦИЯ имя-секции.
    оператор ИСПОЛЬЗОВАТЬ ДЛЯ ОТЛАДКИ.
   [имя-параграфа.
     [предложение] ...] ...}...
КОНЕЦ ДЕКЛАРАТИВ.
  {СЕКЦИЯ имя-секции.
   [имя-параграфа.
     [предложение] ...] ...}...
```

3.2. **Оператор USE FOR DEBUGGING (ИСПОЛЬЗОВАТЬ ДЛЯ ОТЛАДКИ)**

3.2.1. Назначение

В операторе USE FOR DEBUGGING (ИСПОЛЬЗОВАТЬ ДЛЯ ОТЛАДКИ) пользователь указывает данные, за которыми необходимо следить посредством соответствующей отладочной секции.

3.2.2. Общий формат

```
USE FOR DEBUGGING ON
   { [имя-коммуникации-1
      [ALL REFERENCES OF] идентификатор-1
      имя-файла-1]
     имя-процедуры-1
     ALL PROCEDURES }...
```


GOST EXPERT ГОСТ 22558-89,

ИСПОЛЬЗОВАТЬ ДЛЯ ОТЛАДКИ ПРИ

{ имя-коммуникации-1 / [ВСЕХ ССЫЛКАХ НА] идентификатор-1 / имя-файла-1 / имя-процедуры-1 / ВСЕХ ПРОЦЕДУРАХ } ...

3.2.3. Синтаксические правила

(1) Если отладочные секции указаны, то они должны следовать друг за другом непосредственно за заголовком DECLARATIVES (ДЕКЛАРАТИВЫ).

(2) В отладочной секции в операторах, отличных от оператора USE FOR DEBUGGING (ИСПОЛЬЗОВАТЬ ДЛЯ ОТЛАДКИ), нельзя обращаться к процедурам, содержащимся в недекларативной части раздела процедур.

(3) Операторы, появляющиеся вне отладочных секций, не должны ссылаться на имена-процедур, определенные в отладочных секциях.

(4) Операторы, появляющиеся в одной из отладочных секций, могут ссылаться на имена-процедур, определенные в другой отладочной секции, только посредством оператора PERFORM (ВЫПОЛНИТЬ).

Исключение составляет только оператор USE FOR DEBUGGING (ИСПОЛЬЗОВАТЬ ДЛЯ ОТЛАДКИ).

(5) Имена-процедур, определенные в отладочных секциях, не должны появляться в операторах USE FOR DEBUGGING (ИСПОЛЬЗОВАТЬ ДЛЯ ОТЛАДКИ).

(6) Каждый из идентификаторов, имен-файлов, имен-коммуникаций или имен-процедур может появляться только в одном операторе USE FOR DEBUGGING (ИСПОЛЬЗОВАТЬ ДЛЯ ОТЛАДКИ) и только один раз.

(7) Фраза ALL PROCEDURES (ПРИ ВСЕХ ПРОЦЕДУРАХ) может появляться только один раз в программе.

(8) Если указана фраза ALL PROCEDURES (ПРИ ВСЕХ ПРОЦЕДУРАХ), то ни в каком другом операторе USE FOR DEBUGGING (ИСПОЛЬЗОВАТЬ ДЛЯ ОТЛАДКИ) не должны указываться имена-процедур.

(9) Идентификатор-1 не должен представлять данные, определенные в секции отчетов, за исключением счетчиков сумм.

(10) Если статья описания данного, представленного идентификатором-1, содержит фразу OCCURS (ПОВТОРЯЕТСЯ) или подчинена статье с фразой OCCURS (ПОВТОРЯЕТСЯ), то идентификатор-1 должен указываться без обычного необходимого индексирования.

GOST EXPERT ГОСТ 22558-89,

(11) Ссылки на специальный регистр DEBUG-ITEM (ДАННЫЕ-ОТЛАДКИ) допускаются только в пределах отладочных секций.

(12) Идентификатор-1 не должен быть модификацией ссылки.

3.2.4. Общие правила

(1) Операторы, встречающиеся в секции отладки, не вызывают автоматического выполнения секции отладки.

(2) Если в операторе USE FOR DEBUGGING (ИСПОЛЬЗОВАТЬ ДЛЯ ОТЛАДКИ) задано имя-файла-1, соответствующая секция отладки выполняется:

а) после выполнения операторов OPEN (ОТКРЫТЬ) или CLOSE (ЗАКРЫТЬ), ссылающихся на имя-файла-1;

б) после выполнения оператора READ (ЧИТАТЬ) (после других указанных процедур USE (ИСПОЛЬЗОВАТЬ)), не вызвавшего выполнения соответствующего повелительного оператора, указанного фразами AT END (В КОНЦЕ) или INVALID KEY (ПРИ ОШИБКЕ КЛЮЧА);

в) после выполнения операторов DELETE (УДАЛИТЬ) или START (ПОДВЕСТИ), ссылающихся на имя-файла-1.

(3) Если в операторе USE FOR DEBUGGING (ИСПОЛЬЗОВАТЬ ДЛЯ ОТЛАДКИ) указано имя-процедуры-1, соответствующая отладочная секция выполняется:

а) непосредственно перед каждым выполнением названной процедуры;

б) непосредственно после выполнения оператора ALTER (ИЗМЕНИТЬ), ссылающегося на имя-процедуры-1.

(4) Фраза ALL PROCEDURES (ПРИ ВСЕХ ПРОЦЕДУРАХ) вызывает выполнение действий, указанных в общем правиле 3 для всех имен-процедур программы, кроме процедур, определенных в отладочных секциях.

(5) Если указана фраза ALL REFERENCES OF идентификатор-1 (ПРИ ВСЕХ ССЫЛКАХ НА идентификатор-1), соответствующая отладочная секция выполняется для каждого оператора, явно ссылающегося на идентификатор-1 в каждом из следующих случаев:

а) в случае операторов WRITE (ПИСАТЬ) или REWRITE (ОБНОВИТЬ) выполнение отладочной секции происходит до выполнения операторов WRITE (ПИСАТЬ) или REWRITE (ОБНОВИТЬ), но после выполнения неявных перемещений, вызванных наличием в указанных операторах фразы FROM (ИЗ ПОЛЯ);

б) в случае оператора GO TO (ПЕРЕЙТИ) с фразой DEPENDING ON (В ЗАВИСИМОСТИ ОТ) соответствующая

GOST EXPERT ГОСТ 22558-89,

отладочная секция выполняется непосредственно перед передачей управления и до выполнения отладочной секции, связанной с именем-процедуры, которой передается управление;

в) в случае оператора PERFORM (ВЫПОЛНИТЬ), ссылающегося на идентификатор-1 посредством фраз VARYING (МЕНЯЯ), AFTER (ЗАТЕМ) или UNTIL (ДО) — непосредственно после присвоения начального значения, изменения или вычисления значения данного, представленного идентификатором-1;

г) для всех других операторов Кобола — непосредственно после выполнения оператора.

Если ссылка на идентификатор-1 производится во фразе, которая не выполняется, соответствующая отладочная секция также не выполняется.

(6) Если указан идентификатор-1 без фразы ALL REFERENCES OF (ПРИ ВСЕХ ССЫЛКАХ НА), соответствующая секция отладки выполняется в следующих случаях:

а) в случае операторов WRITE (ПИСАТЬ) или REWRITE (ОБНОВИТЬ) соответствующая отладочная секция выполняется непосредственно перед выполнением этих операторов и после неявных перемещений, указанных фразой FROM (ИЗ ПОЛЯ);

б) в случае оператора PERFORM (ВЫПОЛНИТЬ), ссылающегося на идентификатор-1 посредством фраз VARYING (МЕНЯЯ), AFTER (ЗАТЕМ), UNTIL (ДО), непосредственно после присвоения начального значения, модификации и вычисления значения данного, представленного идентификатором-1;

в) для всех других операторов Кобола, явно ссылающихся на идентификатор-1, непосредственно после выполнения оператора, приводящего к изменению значения данного, представленного идентификатором-1.

Если ссылка на идентификатор-1 производится во фразе, которая не выполняется, соответствующая отладочная секция не выполняется.

(7) Независимо от количества ссылок на некоторый идентификатор в пределах одного оператора, соответствующая отладочная секция выполняется для одного выполнения оператора не более одного раза. Исключение составляет оператор PERFORM (ВЫПОЛНИТЬ), вызывающий итеративное выполнение процедуры, для которого соответствующая отладочная секция может выполняться один раз для каждой итерации. Каждое отдельное вхождение повелительного глагола в повелительном операторе рассматривается с точки зрения отладочных действий как отдельный оператор.

(8) Если в операторе USE FOR DEBUGGING (ИСПОЛЬЗОВАТЬ ДЛЯ ОТЛАДКИ) указано имя-коммуникации-1, соответствующая отладочная секция выполняется:

GOST EXPERT ГОСТ 22558-89,

а) после выполнения операторов ENABLE (РАЗРЕШИТЬ), DISABLE (ЗАПРЕТИТЬ) и SEND (ПОСЛАТЬ), ссылающихся на имя-коммуникации-1;

б) после выполнения ссылающегося на имя-коммуникации-1 оператора RECEIVE (ПОЛУЧИТЬ), не вызывающего выполнение повелительного оператора, указанного во фразе NO DATA (НЕТ ДАННЫХ);

в) после выполнения ссылающегося на имя-коммуникации-1 оператора ACCEPT MESSAGE COUNT (ПРИНЯТЬ ЧИСЛО СООБЩЕНИЙ).

(9) Ссылка на идентификатор-1, имя-коммуникации-1, имя-файла-1 или имя-процедуры-1 как на уточнитель не является ссылкой на этот элемент для отладки, описанной в вышеперечисленных общих правилах.

(10) С каждым выполнением отладочных секций связывается специальный регистр DEBUG-ITEM (ДАННЫЕ-ОТЛАДКИ), в котором представляется информация об условиях, вызвавших данное выполнение отладочной секции.

DEBUG-ITEM (ДАННЫЕ-ОТЛАДКИ) имеет следующее неявное описание:

```
01 DEBUG-ITEM.
   02 DEBUG-LINE PICTURE IS X(6).
   02 FILLER PICTURE IS X VALUE IS SPACE.
   02 DEBUG-NAME PICTURE IS X(30).
   02 FILLER PICTURE IS X VALUE IS SPACE.
   02 DEBUG-SUB-1 PICTURE IS S9999 SIGN IS LEADING
      SEPARATE CHARACTER.
   02 FILLER PICTURE IS X VALUE IS SPACE.
   02 DEBUG-SUB-2 PICTURE IS S9999 SIGH IS LEADING
      SEPARATE CHARACTER.
   02 FILLER PICTURE IS X VALUE IS SPACE.
   2 DEBUG-SUB-3 PICTURE IS S9999 SIGN IS LEADING
      SEPARATE CHARACTER.
   02 FILLER PICTURE IS X VALUE IS SPACE.
   02  DEBUG-CONTENTS PICTURE IS X(n).
01 ДАННЫЕ-ОТЛАДКИ.
   02 СТРОКА-ОТЛАДКИ ШАБЛОН X(6).
   02 ЗАПОЛНИТЕЛЬ ШАБЛОН X ЗНАЧЕНИЕ ПРОБЕЛ.
   02 ИМЯ-ОТЛАДКИ ШАБЛОН X(30).
   02 ЗАПОЛНИТЕЛЬ ШАБЛОН X ЗНАЧЕНИЕ ПРОБЕЛ.
   02 ИНДЕКС-ОТЛАДКИ-1 ШАБЛОН S9999 ЗНАК ПЕРВЫЙ
   ОТДЕЛЬНО.
   02 ЗАПОЛНИТЕЛЬ ШАБЛОН X ЗНАЧЕНИЕ ПРОБЕЛ.
```


GOST EXPERT ГОСТ 22558-89,

02 ИНДЕКС-ОТЛАДКИ-2 ШАБЛОН 39999 ЗНАК ПЕРВЫЙ ОТДЕЛЬНО.
02 ЗАПОЛНИТЕЛЬ ШАБЛОН X ЗНАЧЕНИЕ ПРОБЕЛ.
02 ИНДЕКС-ОТЛАДКИ-3 ШАБЛОН 39999 ЗНАК ПЕРВЫЙ ОТДЕЛЬНО.
02 ЗАПОЛНИТЕЛЬ ШАБЛОН X ЗНАЧЕНИЕ ПРОБЕЛ.
02 ЗНАЧЕНИЕ-ОТЛАДКИ ШАБЛОН X(n).

(11) Перед каждым выполнением отладочной секции значения данных, соотнесенных DEBUG-ITEM (ДАННЫЕ-ОТЛАДКИ), заполняются пробелами. Затем значения подчиненных ему данных обновляются в соответствии с нижеприведенными общими правилами, непосредственно перед передачей управления этой отладочной секции. Значения данных, не указанных в нижеследующих общих правилах, представляются пробелами.

Обновление выполняется в соответствии с правилами для оператора MOVE (ПОМЕСТИТЬ), за единственным исключением, состоящим в том, что перемещение в DEBUG-CONTENTS (ЗНАЧЕНИЕ-ОТЛАДКИ) неявно рассматривается как элементарное перемещение буквенно-цифрового в буквенно-цифровое без преобразования данных из одной формы внутреннего представления в другую.

(12) DEBUG-LINE (СТРОКА-ОТЛАДКИ) является определенным реализацией средством идентификации исходного оператора.

(13) DEBUG-NAME (ИМЯ-ОТЛАДКИ) содержит первые 30 литер имени, вызвавшего выполнение отладочной секции.

Все уточнители имени отделяются в значении данного DEBUG-NAME (ИМЯ-ОТЛАДКИ) словом IN или OF (ИЗ).
Индексы при их наличии не включаются в это значение.

(14) Если ссылка на данное, вызвавшее выполнение отладочной секции, индексирована, то в DEBUG-SUB-1 (ИНДЕКС-ОТЛАДКИ-1), DEBUG-SUB-2 (ИНДЕКС-ОТЛАДКИ-2), DEBUG-SUB-3 (ИНДЕКС-ОТЛАДКИ-3) соответственно помещаются номера вхождений каждого из необходимых уровней индексирования.

(15) Размер данного DEBUG-CONTENTS (ЗНАЧЕНИЕ-ОТЛАДКИ) должен допускать представление необходимых значений, определяемых последующими общими правилами.

(16) Если отладочная секция вызывается вследствие первого выполнения первой процедуры программы, не принадлежащей к декларативной части, то выполняются следующие условия:

а) DEBUG-LINE (СТРОКА-ОТЛАДКИ) идентифицирует первый оператор этой процедуры;

б) DEBUG-NAME (ИМЯ-ОТЛАДКИ) содержит имя этой процедуры;

GOST EXPERT ГОСТ 22558-89,

в) DEBUG-CONTENTS (ЗНАЧЕНИЕ ОТЛАДКИ) содержит значение 'START PROGRAM' («НАЧАЛО ПРОГРАММЫ»).

(17) Если выполнение отладочной секции вызвано ссылкой на имя-процедуры-1 в операторе ALTER (ИЗМЕНИТЬ), то имеет место следующее:

а) DEBUG-LINE (СТРОКА-ОТЛАДКИ) идентифицирует этот оператор ALTER (ИЗМЕНИТЬ);

б) DEBUG-NAME (ИМЯ-ОТЛАДКИ) содержит имя-процедуры-1;

в) DEBUG-CONTENTS (ЗНАЧЕНИЕ-ОТЛАДКИ) содержит имя процедуры, указанное фразой PROCEED TO (ДЛЯ ПЕРЕХОДА К) оператора ALTER (ИЗМЕНИТЬ).

(18) Если выполнение отладочной секции вызвано передачей управления при выполнении оператора GO TO (ПЕРЕЙТИ), то имеет место следующее:

а) DEBUG-LINE (СТРОКА-ОТЛАДКИ) идентифицирует указанный оператор GO TO (ПЕРЕЙТИ), который передает управление имени-процедуры-2;

б) DEBUG-NAME (ИМЯ-ОТЛАДКИ) содержит имя-процедуры-1.

(19) Если выполнение отладочной секции вызвано ссылкой на имя-процедуры-1, указанной во фразах INPUT (ПРОЦЕДУРА ВВОДА) или OUTPUT (ПРОЦЕДУРА ВЫВОДА) оператора SORT (СОРТИРОВАТЬ) или MERGE (СЛИТЬ), то имеет место следующее:

а) DEBUG-LINE (СТРОКА-ОТЛАДКИ) идентифицирует оператор SORT (СОРТИРОВАТЬ) или MERGE (СЛИТЬ), ссылающийся на имя-процедуры-1;

б) DEBUG-NAME (ИМЯ-ОТЛАДКИ) содержит имя-процедуры-1;

в) DEBUG-CONTENTS (ЗНАЧЕНИЕ-ОТЛАДКИ) содержит:

1) в случае фразы INPUT (ПРОЦЕДУРА ВВОДА) оператора SORT (СОРТИРОВАТЬ) — 'SORT INPUT' («ВВОД СОРТИРОВКИ»);

2) в случае фразы OUTPUT (ПРОЦЕДУРА ВЫВОДА) оператора SORT (СОРТИРОВАТЬ) — 'SORT OUTPUT' («ВЫВОД СОРТИРОВКИ»);

3) в случае фразы OUTPUT (ПРОЦЕДУРА ВЫВОДА) оператора MERGE (СЛИТЬ) — 'MERGE OUTPUT' («ВЫВОД СЛИЯНИЯ»).

(20) Если выполнение отладочной секции вызвано передачей управления имени-процедуры-1 при выполнении оператора PERFORM (ВЫПОЛНИТЬ), то имеет место следующее:

а) DEBUG-LINE (СТРОКА-ОТЛАДКИ) идентифицирует этот оператор;

б) DEBUG-NAME (ИМЯ-ОТЛАДКИ) содержит имя-процедуры-1;

GOST EXPERT ГОСТ 22558-89,

в) DEBUG-CONTENTS (ЗНАЧЕНИЕ-ОТЛАДКИ) содержит 'PERFORM LOOP' («ЦИКЛ ВЫПОЛНИТЬ»).

(21) Если имя-процедуры-1 относится к процедуре, выполнение которой управляется оператором USE (ИСПОЛЬЗОВАТЬ), и наступили условия ее выполнения, то имеет место следующее:

а) DEBUG-LINE (СТРОКА-ОТЛАДКИ) идентифицирует оператор, выполнение которого повлекло выполнение имени-процедуры-1;

б) DEBUG-NAME (ИМЯ-ОТЛАДКИ) содержит имя-процедуры-1;

в) DEBUG-CONTENTS (ЗНАЧЕНИЕ-ОТЛАДКИ) содержит значение 'USE PROCEDURE' («ПРОЦЕДУРА ИСПОЛЬЗОВАТЬ»).

(22) Если неявная передача управления из предыдущего последовательного параграфа к имени-процедуры-1 вызывает выполнение отладочной секции, имеет место следующее:

а) DEBUG-LINE (СТРОКА-ОТЛАДКИ) идентифицирует предыдущий оператор;

б) DEBUG-NAME (ИМЯ-ОТЛАДКИ) содержит имя-процедуры-1;

в) DEBUG-CONTENTS (ЗНАЧЕНИЕ-ОТЛАДКИ) содержит 'FALL THROUGN' («ПРОХОДИТ ЧЕРЕЗ»).

(23) Если выполнение отладочной секции вызвано ссылкой на имя-файла-1 или имя-коммуникации-1, то имеет место следующее:

а) DEBUG-LINE (СТРОКА-ОТЛАДКИ) идентифицирует исходный оператор, ссылающийся на имя-файла-1 или имя-коммуникации-1;

б) DEBUG-NAME (ИМЯ-ОТЛАДКИ) содержит имя-файла-1 или имя-коммуникации-1;

в) для оператора READ (ЧИТАТЬ) DEBUG-CONTENTS (ЗНАЧЕНИЕ-ОТЛАДКИ) содержит прочитанную запись;

г) для остальных операторов, ссылающихся на имя-файла-1, DEBUG-CONTENTS (ЗНАЧЕНИЕ-ОТЛАДКИ) содержит пробелы;

д) для любого оператора, ссылающегося на имя-коммуникации-1, DEBUG-CONTENTS (ЗНАЧЕНИЕ-ОТЛАДКИ) содержит значение связанной с ним области.

(24) Если выполнение отладочной секции вызвано ссылкой на идентификатор-1, то имеет место следующее:

а) DEBUG-LINE (СТРОКА-ОТЛАДКИ) идентифицирует оператор, ссылающийся на идентификатор-1;

б) DEBUG-NAME (ИМЯ-ОТЛАДКИ) содержит идентификатор-1;

в) DEBUG-CONTENTS (ЗНАЧЕНИЕ-ОТЛАДКИ) содержит

GOST EXPERT ГОСТ 22558-89,

значение, представленное идентификатором-1 в момент передачи управления отладочной секции (см. общие правила 5 и 6).

Часть 16. МОДУЛЬ СЕГМЕНТАЦИИ

1. ВВЕДЕНИЕ В МОДУЛЬ СЕГМЕНТАЦИИ

1.1. **Назначение**

Модуль сегментации предоставляет средства, которые позволяют пользователю взаимодействовать с компилятором для задания требуемых перекрытий в объектной программе.

Модуль сегментации рассматривается в настоящем стандарте Кобола как устаревший элемент и будет устранен из последующих редакций.

1.2. **Характеристика уровней**

Уровень 1 сегментации обеспечивает средства для определения фиксированных и независимых сегментов (п. 1.4.1 настоящей части). Все секции, имеющие одинаковый номер сегмента, должны быть смежными в исходной программе.

Все сегменты, специфицированные как фиксированные, должны быть смежными в исходной программе.

Уровень 2 сегментации допускает смешивание секций с различными номерами сегментов и разрешает фиксированной части исходной программы содержать сегменты, которые могут быть перекрыты (п. 1.4.1 настоящей части).

1.3. **Область действия**

Модуль сегментации в Коболе касается только сегментации процедур. Следовательно, при определении требований к сегментации объектной программы рассматриваются только раздел процедур и раздел оборудования исходной программы.

1.4. **Организация**

1.4.1. Сегменты программы

Раздел процедур исходной программы, хотя это и необязательно, обычно записывается как группа последовательных секций, каждая из которых составлена из ряда операций, предназначенных в целом для выполнения некоторой конкретной функции. Однако, когда применяется сегментация, раздел процедур должен быть разделен на секции. Кроме того, каждая секция должна быть отнесена либо к фиксированной части, либо к одному из независимых сегментов объектной программы. Сегментация не освобождает от необходимости уточнения имен процедур.

1.4.2. Фиксированная часть программы

Фиксированная часть определяется как часть объектной программы, которая логически рассматривается, как если бы она пол-

GOST EXPERT ГОСТ 22558-89,

ностью и постоянно находилась в памяти. Эта часть программы составляется из двух типов сегментов: фиксированных постоянных сегментов и фиксированных перекрываемых сегментов.

Фиксированный постоянный сегмент — это сегмент в фиксированной части, который не может перекрываться никакой другой частью программы. Фиксированный перекрываемый сегмент — это сегмент фиксированной части, который, хотя и обрабатывается логически так, как если бы он постоянно находился в памяти, может быть перекрыт любым другим сегментом с целью оптимизации использования памяти. Число фиксированных постоянных сегментов в фиксированной части может быть изменено указанием фразы SEGMENT-LIMIT (ГРАНИЦА СЕГМЕНТОВ) (п. 2.3 настоящей части).

Если такой сегмент вызван программой, он всегда становится доступным в его последнем использованном состоянии.

1.4.3. Независимые сегменты

Независимый сегмент определяется как часть объектной программы, которая может перекрывать фиксированные перекрываемые сегменты или любые другие независимые сегменты или перекрываться ими. Независимый сегмент находится в своем начальном состоянии, когда в ходе выполнения программы управление впервые передано (явно или неявно) этому сегменту. При последующих передачах управления этому сегменту он также будет находиться в своем начальном состоянии при следующих условиях:

(1) когда рассматриваемому сегменту управление передается в результате неявной передачи управления из сегмента с номером сегмента, отличным от рассматриваемого;

(2) когда рассматриваемому сегменту передается управление в результате неявной передачи управления между операторами SORT (СОРТИРОВАТЬ) или MERGE (СЛИТЬ) в сегменте с номером сегмента, отличным от рассматриваемого, и процедурой ввода или вывода в рассматриваемом независимом сегменте;

(3) когда рассматриваемому сегменту управление передается явно из сегмента с номером сегмента, отличным от рассматриваемого, за исключением передачи управления от оператора EXIT (ВЫЙТИ).

При последующих передачах управления независимому сегменту он находится в своем последнем использованном состоянии при следующих условиях:

(1) когда рассматриваемому сегменту управление передается неявно из сегмента с номером сегмента, отличным от рассматриваемого, за исключением, отмеченным выше в (1) и (2);

(2) когда управление рассматриваемому сегменту передается

GOST EXPERT ГОСТ 22558-89,

явно в результате выполнения оператора EXIT PROGRAM (ВЫЙТИ ИЗ ПРОГРАММЫ) (см. ч. 4 п. 4.4.2).

1.5. **Классификация сегментации**

Сегментируемые секции классифицируются системой номеров сегментов (п. 3.2 настоящей части) и следующими правилами.

(1) Логические требования. Секции, обращения к которым осуществляются постоянно или очень часто, обычно относятся к одному из постоянных сегментов; секции, используемые реже, обычно относят либо к одному из перекрываемых фиксированных сегментов, либо к одному из независимых сегментов, определяемых логическими требованиями.

(2) Частота использования. Обычно чаще используемым секциям присваиваются меньшие номера сегментов, реже — большие номера сегментов.

(3) Отношения с другими секциями. Секциям, которые часто обращаются друг к другу, следует присваивать один и тот же номер сегмента.

1.6. **Управление сегментацией**

Логическая последовательность программы совпадает с физической, за исключением специальных передач управления. Если для управления переходами от сегмента к сегменту (в соответствии с правилами п. 3.2 настоящей части) требуется переупорядочение объектной программы, то реализация должна обеспечивать передачи управления так, чтобы осуществить логический поток, определенный в исходной программе. Реализация должна обеспечивать также все необходимые передачи управления для обрабатываемого сегмента при любом его использовании. Управление может быть передано внутри исходной программы любому параграфу в секции; таким образом, необязательно передавать управление в начало секции.

2. РАЗДЕЛ ОБОРУДОВАНИЯ В МОДУЛЕ СЕГМЕНТАЦИИ

2.1. **Секция конфигурации**

Информация, связанная с секцией конфигурации, помещена в ч. 6.

2.2. **Параграф OBJECT-COMPUTER (РАБОЧАЯ-МАШИНА)**

2.2.1. Назначение

Параграф OBJECT-COMPUTER (РАБОЧАЯ-МАШИНА) обеспечивает средства для описания машины, на которой должна выполняться программа.

2.2.2. Общий формат

OBJECT-COMPUTER [имя-машины

GOST EXPERT ГОСТ 22558-89,

[MEMORY SIZE целое-1 { WORDS / CHARACTERS / MODULES }]
[PROGRAM COLLATING SEQUENCE IS имя-алфавита-1]
[SEGMENT-LIMIT IS номер-сегмента] .]

РАБОЧАЯ-МАШИНА. [имя-машины

[РАЗМЕР ПАМЯТИ целое-1 { СЛОВ / ЛИТЕР / МОДУЛЕЙ }]
[ПРОГРАММНЫЙ АЛФАВИТ имя-алфавита-1]
[ГРАНИЦА СЕГМЕНТОВ номер-сегмента] .]

2.2.3. Синтаксическое правило

(1) Имя-машины — это системное имя.

2.2.4. Общие правила

(1) Все фразы параграфа OBJECT-COMPUTER (РАБОЧАЯ-МАШИНА) применяются к программе, в которой они явно или неявно заданы, и к любой программе, содержащейся в этой программе.

(2) Общие правила для имени-машины, фраз MEMORY SIZE (РАЗМЕР ПАМЯТИ) и PROGRAM COLLATING SEQUENCE (ПРОГРАММНЫЙ АЛФАВИТ) приводятся в ч. 6.

(3) Фраза SEGMENT-LIMIT (ГРАНИЦА СЕГМЕНТОВ) описывается в п. 2.3 настоящей части.

2.3. **Фраза SEGMENT-LIMIT (ГРАНИЦА СЕГМЕНТОВ)**

2.3.1. Назначение

В идеальном случае все программные сегменты, имеющие номера-сегментов от 0 до 49, специфицируются как постоянные сегменты. Однако, когда доступная память недостаточна для того, чтобы разместить все постоянные сегменты плюс максимальный перекрываемый сегмент, становится необходимым сократить число постоянных сегментов. Фраза SEGMENT-LIMIT (ГРАНИЦА СЕГМЕНТОВ) дает пользователю средство, которым он может уменьшить число постоянных сегментов в своей программе с сохранением логических свойств сегментов фиксированной части (номера-сегментов от 0 до 49).

2.3.2. Общий формат

SEGMENT-LIMIT IS номер-сегмента
ГРАНИЦА СЕГМЕНТОВ номер-сегмента

GOST EXPERT ГОСТ 22558-89,

2.3.3. Синтаксическое правило

Номер-сегмента должен быть целым, принимающим значения от 0 до 49.

2.3.4. Общие правила

(1) Если задана фраза SEGMENT-LIMIT (ГРАНИЦА СЕГМЕНТОВ), то в качестве постоянных сегментов объектной программы рассматриваются только сегменты, имеющие номера от 0 до (но не включая) номера, специфицированного фразой SEGMENT-LIMIT (ГРАНИЦА СЕГМЕНТОВ).

(2) Сегменты, имеющие номера от номера, специфицированного фразой SEGMENT-LIMIT (ГРАНИЦА СЕГМЕНТОВ), до 49, рассматриваются как перекрываемые фиксированные сегменты.

(3) Если фраза SEGMENT-LIMIT (ГРАНИЦА СЕГМЕНТОВ) опущена, все сегменты с номерами-сегментов от 0 до 49 будут рассматриваться как постоянные сегменты объектной программы.

3. РАЗДЕЛ ПРОЦЕДУР В МОДУЛЕ СЕГМЕНТАЦИИ

3.1. Общее описание

При использовании модуля сегментации в исходной Кобол-программе раздел процедур содержит секции с номерами сегментов.

Ниже приводится общий формат раздела процедур, в котором задаются секции и номера сегментов.

```
PROCEDURE DIVISION.
[DECLARATIVES.

{имя-секции SECTION [номер-сегмента].
  Оператор USE.
 [имя-параграфа.
   [предложение] ...] ...}...
END DECLARATIVES.
 {имя-секции SECTION [номер-сегмента].
  [имя-параграфа.
    [предложение] ...] ...}...
 РАЗДЕЛ ПРОЦЕДУР.
 [ДЕКЛАРАТИВЫ.
 {СЕКЦИЯ имя-секции [номер-сегмента] .
   Оператор ИСПОЛЬЗОВАТЬ.
  [имя-параграфа.
    [предложение] ...] ...}...
```


GOST EXPERT ГОСТ 22558-89,

КОНЕЦ ДЕКЛАРАТИВ.

{СЕКЦИЯ имя-секции [номер-сегмента] .

[имя-параграфа.

[предложение] ...] ...}...

3.2. **Номера сегментов**

3.2.1. Назначение

Классификация секций выполняется с помощью системы номеров сегментов. Номер сегмента включается в заголовок секции.

3.2.2. Общий формат

имя-секции SECTION [номер-сегмента] .

СЕКЦИЯ имя-секции [номер-сегмента] .

3.2.3. Синтаксические правила

(1) Номер-сегмента — целое, принимающее значения от 0 до 99.

(2) Если номер-сегмента в заголовке секции опущен, то по умолчанию он предполагается равным 0.

(3) Секции декларатив должны иметь номера-сегментов, меньшие 50.

3.2.4. Общие правила

(1) Все секции, имеющие один и тот же номер-сегмента, составляют единый программный сегмент. В уровне 1 все секции, имеющие один и тот же номер-сегмента, должны быть смежными.

В уровне 2 смежность секций, имеющих в исходной программе одинаковые номера-сегментов, необязательна.

(2) Сегменты с номерами-сегментов от 0 до 49 принадлежат фиксированной части программы. На уровне 1 все секции с номерами-сегментов от 0 до 49 должны быть в исходной программе смежны.

(3) Сегменты с номерами-сегментов от 50 до 99 — независимые сегменты.

3.3. **Ограничения на программный поток**

При использовании сегментации на операторы ALTER (ИЗМЕНИТЬ), PERFORM (ВЫПОЛНИТЬ), MERGE (СЛИТЬ) и SORT (СОРТИРОВАТЬ) накладываются следующие ограничения.

3.3.1. Оператор ALTER (ИЗМЕНИТЬ)

Оператор GO TO (ПЕРЕЙТИ) в секции с номером сегмента, большим или равным 50, не должен быть объектом оператора ALTER (ИЗМЕНИТЬ) в секции с другим номером сегмента.

Все другие использования оператора ALTER (ИЗМЕНИТЬ) допустимы и выполняются даже в том случае, когда оператор GO TO (ПЕРЕЙТИ), на который ссылается оператор ALTER (ИЗМЕНИТЬ), находится в фиксированном перекрываемом сегменте.

3.3.2. Оператор PERFORM (ВЫПОЛНИТЬ)

Оператор PERFORM (ВЫПОЛНИТЬ), указанный в секции, не принадлежащей независимому сегменту, может иметь в своей об-

GOST EXPERT ГОСТ 22558-89,

ласти действия кроме декларативных секций только одну из следующих процедур:

(1) секции и (или) параграфы, полностью содержащиеся в одном или более сегментах, не являющихся независимыми;

(2) секции и (или) параграфы, полностью содержащиеся в одном независимом сегменте.

Оператор PERFORM (ВЫПОЛНИТЬ), указанный в независимом сегменте, может иметь в своей области действия кроме декларативных секций только одну из следующих процедур:

(1) секции и (или) параграфы, полностью содержащиеся в одном или более сегментах, не являющихся независимыми;

(2) секции и (или) параграфы, полностью содержащиеся в том же независимом сегменте, что и рассматриваемый оператор PERFORM (ВЫПОЛНИТЬ).

3.3.3. Оператор MERGE (СЛИТЬ)

Если оператор MERGE (СЛИТЬ) появляется в секции, не принадлежащей независимому сегменту, то процедура вывода, указанная в операторе MERGE (СЛИТЬ) должна содержаться:

(1) полностью внутри сегментов, не являющихся независимыми, или

(2) целиком содержаться в одном независимом сегменте.

Если оператор MERGE (СЛИТЬ) появляется в независимом сегменте, то процедура вывода, на которую ссылается оператор MERGE (СЛИТЬ), должна содержаться:

(1) полностью внутри сегментов, не являющихся независимыми, или

(2) полностью внутри того же независимого сегмента, в котором находится оператор MERGE (СЛИТЬ).

3.3.4. Оператор SORT (СОРТИРОВАТЬ)

Если оператор SORT (СОРТИРОВАТЬ) появляется в секции, не принадлежащей независимому сегменту, то процедура ввода и процедура вывода, на которую ссылается оператор SORT (СОРТИРОВАТЬ), должны появляться:

1) полностью внутри сегментов, не являющихся независимыми, или

(2) целиком содержаться в одном независимом сегменте.

Если оператор SORT (СОРТИРОВАТЬ) указан в независимом сегменте, то процедура ввода и процедура вывода, на которые ссылается оператор SORT (СОРТИРОВАТЬ), должны содержаться:

(1) полностью внутри сегментов, не являющихся независимыми, или

(2) полностью внутри того же независимого сегмента, в котором находится оператор SORT (СОРТИРОВАТЬ).

GOST EXPERT ГОСТ 22558-89,

Часть 17. ПРИЛОЖЕНИЯ

ПРИЛОЖЕНИЕ 1
Справочное

ОТЛИЧИЯ МЕЖДУ ПРЕДЫДУЩИМ И НАСТОЯЩИМ СТАНДАРТОМ

1. Перечень отличий

Приложение содержит перечень всех элементов ГОСТ 22558 и настоящего стандарта. Элементы упорядочены соответственно разделам Кобола.

Литера «—» в столбце обозначает отсутствие указанного элемента. Наличие элемента определяется трехбуквенным обозначением модуля согласно следующей таблице.

Сокращение	Значение
ЯДР	Ядро
ТАБ	Обработка таблиц
ПОД	Последовательный ввод-вывод
ОТД	Относительный ввод-вывод
ИПД	Индексный ввод-вывод
МПС	Межпрограммные связи
СРТ	Сортировка-слияние
ОИТ	Обработка исходных текстов
БИБ	Библиотека
ГОТ	Генератор отчетов
КОМ	Коммуникации
ОТЛ	Отладка
СЕГ	Сегментация

Уровень, на котором встречается элемент в модуле, указан цифрой, предшествующей трехбуквенному сокращению имени модуля. Например, 2 ЯДР указывает, что элемент принадлежит второму уровню ядра, а 1 ИПД — что элемент принадлежит первому уровню модуля индексного ввода-вывода. Литера «-», следующая за сокращением имени модуля, обозначает, что элемент является устаревшим элементом в данной редакции стандарта Кобола и будет удален в следующей редакции.

1.1. Перечень отличий в понятиях языка

Элемент	ГОСТ 22558	Настоящий стандарт
Понятия языка		
Набор литер		
Литеры, используемые в словах для английской нотации:		
0—9, A—Z, — (дефис) (для русской нотации 0—9, A—Z, А—Я — (дефис))	1 ЯДР	1 ЯДР

GOST EXPERT ГОСТ 22558-89,

Элемент	ГОСТ 22558	Настоящий стандарт
Литеры, используемые в пунктуации: « ( ). пробел	1 ЯДР	1 ЯДР
Литеры, используемые в пунктуации: , (запятая) ; (точка с запятой)	2 ЯДР	1 ЯДР
Литеры, используемые в пунктуации, : (двоеточие)	—	2 ЯДР
Литеры, используемые в пунктуации, =	2 БИБ	2 ОИТ
Литеры, используемые в редактировании, B + — . , Z (П) * $ 0 CR (КР) DB (ДБ)	1 ЯДР	1 ЯДР
Литеры, используемые в арифметических операциях, + — * / **	2 ЯДР	2 ЯДР
Литеры, используемые в условиях отношения, = > <	2 ЯДР	1 ЯДР
Литеры, используемые в условиях отношения, > = < =	—	1 ЯДР
Литеры, используемые при индексировании, + —	2 ТАБ	1 ЯДР
Разрешена замена двумя литерами	1 ЯДР	1 ЯДР
Разрешена замена одной литерой	—	1 ЯДР
Замена одной литерой должна быть сделана для недостающих литер Кобола	1 ЯДР	—
Разделители		
» ( ) . пробел	1 ЯДР	1 ЯДР
, (запятая) ; (точка с запятой)	2 ЯДР	1 ЯДР
: (двоеточие)	—	2 ЯДР
— —	2 БИБ	2 ОИТ
Один или больше пробелов, являющихся частью разделителя	—	1 ЯДР
Строки-литер		
Слова Кобола		
Максимум 30 литер	1 ЯДР	1 ЯДР
Системные имена и слова, определенные пользователем, должны образовывать непересекающиеся множества	1 ЯДР	—
Системные имена и слова, определенные пользователем, образуют пересекающиеся множества	—	1 ЯДР
Слова, определенные пользователем,		
Имя-алфавита	1 ЯДР	1 ЯДР
Имя-коммуникации	1 КОМ	1 КОМ
Имя-класса	—	1 ЯДР
Имя-условия	2 ЯДР	2 ЯДР
Имя-данного	1 ЯДР	1 ЯДР
Должно начинаться буквой	1 ЯДР	—
Не обязательно начинается буквой	2 ЯДР	1 ЯДР
Имя-файла	1 ПОД	1 ПОД
	1 ОТД	1 ОТД
	1 ИПД	1 ИПД
	1 СРТ	1 СРТ
	1 ГОТ	1 ГОТ

GOST EXPERT ГОСТ 22558-89,

Элемент	ГОСТ 22558	Настоящий стандарт
Имя-индекса	1 ТАБ	1 ЯДР
Номер-уровня	1 ЯДР	1 ЯДР
Имя-библиотеки	2 БИБ	2 ОИТ
Мнемоническое-имя	1 ЯДР	1 ЯДР
Имя-параграфа	1 ЯДР	1 ЯДР
Имя-программы	1 ЯДР	1 ЯДР
Имя-записи	1 ПОД	1 ПОД
	1 ОТД	1 ОТД
	1 ИПД	1 ИПД
	1 СРТ	1 СРТ
Имя-отчета	1 ГОТ	1 ГОТ
Имя-программного-модуля	1 ЯДР	1 ЯДР
Имя-секции	1 ЯДР	1 ЯДР
Номер-сегмента	1 СЕГ	1 СЕГ
Символическая-литера	—	2 ЯДР
Имя-текста	1 БИБ	1 ОИТ
Системные-имена		
Имя-машины	1 ЯДР	1 ЯДР
Имя-реализации	1 ЯДР	1 ЯДР
Имя-языка	1 ЯДР	1 ЯДР
Зарезервированные слова		
Обязательные слова	1 ЯДР	1 ЯДР
Ключевые слова	1 ЯДР	1 ЯДР
Слова-специальные-литеры		
Знаки арифметических операций + — */**	2 ЯДР	2 ЯДР
Знаки арифметических операций, используемые при индексировании именем-данного, + —	—	1 ЯДР
Знаки арифметических операций, используемые при индексировании именем-индекса + —	2 ТАБ	1 ЯДР
Литеры отношения = > <	2 ЯДР	1 ЯДР
Литеры отношения >= <=	—	1 ЯДР
Необязательные слова	1 ЯДР	1 ЯДР
Связки	2 ЯДР	—
Слова специального назначения		
Стандартные константы: ZERO (НУЛЬ), SPACE (ПРОБЕЛ), HIGH-VALUE (НАИБОЛЬШЕЕ-ЗНАЧЕНИЕ), LOW-VALUE (НАИМЕНЬШЕЕ-ЗНАЧЕНИЕ), QUOTE (КАВЫЧКА)	1 ЯДР	1 ЯДР
Стандартные константы: ZEROES, ZEROS (НУЛИ), SPACES (ПРОБЕЛЫ), HIGH-VALUES (НАИБОЛЬШИЕ-ЗНАЧЕНИЯ), LOW-VALUES (НАИМЕНЬШИЕ-ЗНАЧЕНИЯ), QUOTES (КАВЫЧКИ)	2 ЯДР	1 ЯДР
Стандартные константы: ALL литерал (ВСЕ литерал)	2 ЯДР	2 ЯДР
Стандартные константы: символическая-литера, ALL стандартная-константа (ВСЕ стандартная-константа), ALL символическая-литера (ВСЕ символическая-литера)	—	2 ЯДР

GOST EXPERT ГОСТ 22558-89,

Элемент	ГОСТ 22558	Настоящий стандарт
Специальные регистры		
LINAGE-COUNTER (СЧЕТЧИК-ВЕРСТКИ)	2 ПОД	2 ПОД
LINE-COUNTER (СЧЕТЧИК-СТРОК)	1 ГОТ	1 ГОТ
PAGE-COUNTER (СЧЕТЧИК-СТРАНИЦ)	1 ГОТ	1 ГОТ
DEBUG ITEM (ДАННЫЕ-ОТЛАДКИ)	1 ОТЛ	1 ОТЛ +
Литералы		
Числовые литералы: от 1 до 18 цифр	1 ЯДР	1 ЯДР
Нечисловые литералы от 1 до 120 литер	1 ЯДР	—
Нечисловые литералы: от 1 до 160 литер	—	1 ЯДР
Нечисловые литералы: длина зависит от представления в объектной программе	—	1 ЯДР
PICTURE строка-литер (ШАБЛОН строка-литер)	1 ЯДР	1 ЯДР
Статья-комментарий	1 ЯДР	1 ЯДР +
Однозначность ссылки		
Однозначность ссылки, требуемая во время ссылки	—	1 ЯДР
Однозначность ссылки, требуемая во время спецификации	2 ЯДР	—
Уточнение		
Уточнение не разрешается	1 ЯДР	1 ЯДР
Уточнение разрешается	2 ЯДР	2 ЯДР
Должно быть разрешено по крайней мере 5 уровней уточнения	2 ЯДР	—
50 уточнителей	—	2 ЯДР
Индексирование (имя-данного/литерал)	1 ТАБ	1 ЯДР
3 уровня	1 ТАБ	1 ЯДР
7 уровней	—	2 ЯДР
Индексирование (имя-индекса)	1 ТАБ	1 ЯДР
3 уровня	1 ТАБ	1 ЯДР
7 уровней	—	2 ЯДР
Относительное индексирование именем-данного	—	1 ЯДР
Относительное индексирование именем-индекса	1 ТАБ	1 ЯДР
Модификация ссылки	—	2 ЯДР
Формат представления		
Порядковый номер	1 ЯДР	1 ЯДР
Должен быть цифровым	1 ЯДР	—
Может содержать любую литеру из набора литер машины	—	1 ЯДР
Продолжение строк		
Продолжение нечисловых литералов	1 ЯДР	1 ЯДР
Продолжение слов Кобола, числовых литералов	2 ЯДР	1 ЯДР
Продолжение строки-литер шаблона	—	2 ЯДР
Внутри продолжения допускаются строки комментария	1 ЯДР	1 ЯДР
Внутри продолжения допускаются строки пробелов	—	1 ЯДР
Строки пробелов (пустые строки)	0 ЯДР	1 ЯДР
Строки комментария		

GOST EXPERT ГОСТ 22558-89,

Элемент	ГОСТ 22558	Настоящий стандарт
Строка комментария со * (звездочкой)	1 ЯДР	1 ЯДР
Строка комментария с / (наклонной чертой)	1 ЯДР	1 ЯДР
Строка отладочная D(T) в поле индикатора	1 ОТЛ	1 ЯДР
Структура исходной программы		
Раздел идентификации обязателен	1 ЯДР	1 ЯДР
Раздел оборудования необязателен	—	1 ЯДР
Раздел данных необязателен	—	1 ЯДР
Раздел процедур необязателен	—	1 ЯДР
Заголовок конца программы	—	2 ЯДР
Вложенные исходные программы	—	2 МПС

1.2. **Перечень отличий в разделе идентификации**

РАЗДЕЛ ИДЕНТИФИКАЦИИ		
Параграф PROGRAM-ID (ПРОГРАММА)		
Имя-программы	1 ЯДР	1 ЯДР
Идентифицирует исходную-программу и листинги	1 ЯДР	1 ЯДР
Идентифицирует объектную программу	—	1 ЯДР
Фраза COMMON (ОБЩАЯ)	—	2 МС
Фраза INITIAL (НАЧАЛЬНАЯ)	—	2 МПС
Параграф AUTHOR (АВТОР)	1 ЯДР	1 ЯДР +
Параграф INSTALLATION (ПРЕДПРИЯТИЕ)	1 ЯДР	1 ЯДР +
Параграф DATE-WRITTEN (ДАТА-НАПИСАНИЯ)	1 ЯДР	1 ЯДР +
Параграф DATE-COMPILED (ДАТА-ТРАНСЛЯЦИИ)	2 ЯДР	2 ЯДР +
Параграф SECURITY (ПОЛНОМОЧИЯ)	1 ЯДР	1 ЯДР +

1.3. **Перечень отличий в разделе оборудования**

РАЗДЕЛ ОБОРУДОВАНИЯ		
Раздел оборудования обязателен	1 ЯДР	—
Раздел оборудования необязателен	—	1 ЯДР
Секция конфигурации		
Секция конфигурации обязательна	1 ЯДР	—
Секция конфигурации необязательна	—	1 ЯДР
Параграф SOURCE-COMPUTER (ИСХОДНАЯ-МАШИНА)		
Параграф SOURCE-COMPUTER (ИСХОДНАЯ МАШИНА) обязателен	1 ЯДР	—
Параграф SOURCE-COMPUTER (ИСХОДНАЯ-МАШИНА) необязателен	—	1 ЯДР
Может быть определен пустой параграф	—	1 ЯДР
Имя-машины	1 ЯДР	1 ЯДР
Фраза WITH DEBUGGING MODE (В РЕЖИМЕ ОТЛАДКИ) для отладочных строк	1 ОТЛ	1 ЯДР
Фраза WITH DEBUGGING MODE (В РЕЖИМЕ ОТЛАДКИ) для отладочных секций	1 ОТЛ	1 ОТЛ +

GOST EXPERT ГОСТ 22558-89,

Элемент	ГОСТ 22558	Настоящий стандарт
Параграф OBJECT-COMPUTER (РАБОЧАЯ МАШИНА)		
Параграф OBJECT-COMPUTER (РАБОЧАЯ МАШИНА) обязателен	1 ЯДР	—
Параграф OBJECT-COMPUTER (РАБОЧАЯ МАШИНА) необязателен	—	1 ЯДР
Может быть определен пустой параграф	—	1 ЯДР
Имя-машины	1 ЯДР	1 ЯДР
Фраза MEMORY SIZE (РАЗМЕР ПАМЯТИ)	1 ЯДР	1 ЯДР +
Фраза PROGRAM COLLATING SEQUENCE (ПРОГРАММНЫЙ АЛФАВИТ)	1 ЯДР	1 ЯДР
Фраза SEGMENT-LIMIT (ГРАНИЦА СЕГМЕНТОВ)	2 СЕГ	2 СЕГ +
Параграф SPECIAL-NAMES (СПЕЦИАЛЬНЫЕ-ИМЕНА)		
Фраза ALPHABET (АЛФАВИТ)	1 ЯДР	1 ЯДР
Вариант STANDARD-1 (СТАНДАРТ-А)	1 ЯДР	1 ЯДР
Вариант STANDARD-2 (СТАНДАРТ-М)	—	1 ЯДР
Вариант СТАНДАРТ-Р	1 ЯДР	1 ЯДР
Вариант NATIVE (ВНУТРЕННИЙ)	1 ЯДР	1 ЯДР
Вариант имя-реализации	1 ЯДР	1 ЯДР
Вариант литерал	2 ЯДР	2 ЯДР
Фраза CLASS (КЛАСС)	—	1 ЯДР
Фраза CURRENCY SIGN (ВАЛЮТНЫЙ ЗНАК)	1 ЯДР	1 ЯДР
Литерал может быть стандартной константой	1 ЯДР	—
Фраза DECIMAL-POINT (ДЕСЯТИЧНАЯ ТОЧКА)	1 ЯДР	1 ЯДР
Фраза имя-реализации	1 ЯДР	1 ЯДР
Вариант мнемоническое-имя	1 ЯДР	1 ЯДР
Если имя-реализации является переключателем, должно быть указано имя-условия	1 ЯДР	—
Если имя-реализации является переключателем, имя-условия может быть не указано	—	1 ЯДР
Вариант ON STATUS IS имя-условия (ВКЛЮЧЕНО имя-условия)	1 ЯДР	1 ЯДР
Вариант OFF STATUS IS имя-условия (ВЫКЛЮЧЕНО имя-условия)	1 ЯДР	1 ЯДР
Фраза SYMBOLIC CHARACTERS (СИМВОЛИЧЕСКАЯ ЛИТЕРА)	—	2 ЯДР
Секция ввода-вывода	1 ПОД 1 ОТД 1 ИПД 1 СРТ 1 ГОТ	1 ПОД 1 ОТД 1 ИПД 1 СРТ 1 ГОТ
Параграф FILE-CONTROL (УПРАВЛЕНИЕ-ФАЙЛАМИ)	1 ПОД 1 ОТД 1 ИПД 1 СРТ 1 ГОТ	1 ПОД 1 ОТД 1 ИПД 1 СРТ 1 ГОТ

GOST EXPERT ГОСТ 22558-89,

Элемент	ГОСТ 22558	Настоящий стандарт
Статья управления файлом	1 ПОД 1 ОТД 1 ИПД 1 СРТ 1 ГОТ	1 ПОД 1 ОТД 1 ИПД 1 СРТ 1 ГОТ
Фраза SELECT (ДЛЯ)	1 ПОД 1 ОТД 1 ИПД 1 СРТ 1 ГОТ	1 ПОД 1 ОТД 1 ИПД 1 СРТ 1 ГОТ
Вариант OPTIONAL (НЕОБЯЗАТЕЛЬНОГО)	2 ПОД	2 ПОД 2 ОТД 2 ИПД 1 ГОТ
Входной файл	2 ПОД	2 ПОД 2 ОТД 2 ИПД
Входной-выходной файл	—	2 ПОД 2 ОТД 2 ИПД
Дополняемый файл	—	2 ПОД 2 ОТД 2 ИПД 2 ГОТ
Имя-файла ссылается на определитель файла	—	1 ПОД 1 ОТД 1 ИПД 1 СРТ 1 ГОТ
Фраза ACCESS MODE (ДОСТУП) SEQUENTIAL (ПОСЛЕДОВАТЕЛЬНЫЙ)	1 ПОД 1 ОТД 1 ИПД 1 ГОТ	1 ПОД 1 ОТД 1 ИПД 1 ГОТ
RANDOM (ПРОИЗВОЛЬНЫЙ)	1 ОТД 1 ИПД	1 ОТД 1 ИПД
DYNAMIC (ДИНАМИЧЕСКИЙ)	2 ОТД 2 ИПД	2 ОТД 2 ИПД
Вариант RELATIVE KEY (ОТНОСИТЕЛЬНЫЙ КЛЮЧ)	1 ОТД	1 ОТД
Фраза ALTERNATE RECORD KEY (ДОПОЛНИТЕЛЬНЫЙ КЛЮЧ ЗАПИСИ)	2 ИПД	2 ИПД
Вариант WITH DUPLICATES (С ДУБЛИРОВАНИЕМ)	2 ИПД	2 ИПД
Фраза ASSIGN (НАЗНАЧИТЬ)	1 ПОД 1 ОТД 1 ИПД 1 СРТ 1 ГОТ	1 ПОД 1 ОТД 1 ИПД 1 СРТ 1 ГОТ
Имя-реализации	1 ПОД 1 ОТД 1 ИПД 1 СРТ	1 ПОД 1 ОТД 1 ИПД 1 СРТ

GOST EXPERT ГОСТ 22558-89,

Элемент	ГОСТ 22558	Настоящий стандарт
	1 ГОТ	1 ГОТ
литерал	—	1 ПОД
		1 ОТД
		1 ИПД
		1 СРТ
		1 ГОТ
Фраза FILE STATUS (СОСТОЯНИЕ ФАЙЛА)	1 ПОД	1 ПОД
	1 ОТД	1 ОТД
	1 ИПД	1 ИПД
	1 ГОТ	1 ГОТ
Фраза ORGANIZATION (ОРГАНИЗАЦИЯ)		
SEQUENTIAL (ПОСЛЕДОВАТЕЛЬНАЯ)	1 ПОД	1 ПОД
	1 ГОТ	1 ГОТ
RELATIVE (ОТНОСИТЕЛЬНАЯ)	1 ОТД	1 ОТД
INDEXED (ИНДЕКСНАЯ)	1 ИПД	1 ИПД
Фраза PADDING CHARACTER (ЛИТЕРА ЗАПОЛНИТЕЛЬ)	—	2 ПОД
		1 ГОТ
Фраза RECORD DELIMITER (ОГРАНИЧИТЕЛЬ ЗАПИСИ)	—	2 ПОД
		1 ГОТ
Фраза RECORD KEY (КЛЮЧ ЗАПИСИ)	1 ИПД	1 ИПД
Фраза RESERVE AREA (РЕЗЕРВИРОВАТЬ ОБЛАСТЕЙ)	2 ПОД	2 ПОД
	2 ОТД	2 ОТД
	2 ИПД	2 ИПД
	1 ГОТ	1 ГОТ
Параграф I-O-CONTROL (УПРАВЛЕНИЕ-ВВОДОМ-ВЫВОДОМ)	2 ПОД	1 ПОД
	2 ОТД	1 ОТД
	2 ИПД	1 ИПД
	2 СРТ	1 СРТ
		1 ГОТ
Порядок фраз несущественен	—	1 ПОД
		1 ОТД
		1 ИПД
		1 СРТ
		1 ГОТ
Фраза MULTIPLE FILE TAPE (НА ОДНОЙ КАТУШКЕ)	2 ПОД	2 ПОД +
		1 ГОТ +
Фраза RERUN (ПЕРЕПРОГОН)	1 ПОД	1 ПОД +
	1 ОТД	1 ОТД +
	1 ИПД	1 ИПД +
Фраза SAME AREA (ОБЩАЯ ОБЛАСТЬ)	1 ПОД	1 ПОД
	1 ОТД	1 ОТД
	1 ИПД	1 ИПД
	1 ГОТ	1 ГОТ
Фраза SAME RECORD AREA (ОБЩАЯ ОБЛАСТЬ ЗАПИСИ)	2 ПОД	2 ПОД
	2 ОТД	2 ОТД
	2 ИПД	2 ИПД
	2 СРТ	1 СРТ

GOST EXPERT ГОСТ 22558-89,

Элемент	ГОСТ 22558	Настоящий стандарт
Фраза SAME SORT/SORT-MERGE AREA (ОБЩАЯ ОБЛАСТЬ СОРТИРОВКИ/СОРТИРОВКИ-СЛИЯНИЯ)	2 СРТ	1 СРТ

1.4. Перечень отличий в разделе данных

РАЗДЕЛ ДАННЫХ

Раздел данных обязателен	1 ЯДР	—
Раздел данных необязателен	—	1 ЯДР
Секция файлов	1 ПОД	1 ПОД
	1 ОТД	1 ОТД
	1 ИПД	1 ИПД
	1 СРТ	1 СРТ
	1 ГОТ	1 ГОТ
Статья описания файла	1 ПОД	1 ПОД
	1 ОТД	1 ОТД
	1 ИПД	1 ИПД
	1 ГОТ	1 ГОТ
Индикатор уровня FD (ОФ)	1 ПОД	1 ПОД
	1 ОТД	1 ОТД
	1 ИПД	1 ИПД
	1 ГОТ	1 ГОТ
Фраза BLOCK CONTAINS (В БЛОКЕ) Целое RECORDS/CHARACTERS (целое ЗАПИСЕЙ/ЛИТЕР)	1 ПОД	1 ПОД
	1 ОТД	1 ОТД
	1 ИПД	1 ИПД
	1 ГОТ	1 ГОТ
Целое-1 TO целое-2 RECORDS/CHARACTERS (ОТ целое-1 ДО целое-2 ЗАПИСЕЙ/ЛИТЕР)	2 ПОД	2 ПОД
	2 ОТД	2 ОТД
	2 ИПД	2 ИПД
	1 ГОТ	1 ГОТ
Фраза CODE-SET (АЛФАВИТ)	1 ПОД	1 ПОД
	1 ГОТ	1 ГОТ
Фраза DATA RECORDS (ЗАПИСИ ДАННЫХ)	1 ПОД	1 ПОД +
	1 ОТД	1 ОТД +
	1 ИПД	1 ИПД +
Фраза EXTERNAL (ВНЕШНЕЕ)	—	2 МПС
Фраза GLOBAL (ГЛОБАЛЬНОЕ)	—	2 МПС
Фраза LABEL RECORDS (МЕТКИ)	1 ПОД	1 ПОД +
	1 ОТД	1 ОТД +
	1 ИПД	1 ИПД +
	1 ГОТ	1 ГОТ +
Фраза LINAGE (ВЕРСТКА)	2 ПОД	2 ПОД
Вариант FOOTING (КОНЦОВКА)	2 ПОД	2 ПОД
Вариант TOP (ВЕРХНЕЕ ПОЛЕ)	2 ПОД	2 ПОД
Вариант BOTTOM (НИЖНЕЕ ПОЛЕ)	2 ПОД	2 ПОД
Фраза RECORD (В ЗАПИСИ) целое-1 CHARACTERS (целое-1 ЛИТЕР)	1 ПОД	1 ПОД
	1 ОТД	1 ОТД

GOST EXPERT ГОСТ 22558-89,

Элемент	ГОСТ 22558	Настоящий стандарт	
	1 ИПД	1 ИПД	
	1 ГОТ	1 ГОТ	
Вариант VARYING IN SIZE (ПЕРЕМЕННОЕ ЧИСЛО)	—	2 ПОД	
		2 ОТД	
		2 ИПД	
целое-4 TO целое-5 CHARACTERS (целое-4 ДО целое-5 ЛИТЕР	1 ПОД	1 ПОД	
	1 ОТД	1 ОТД	
	1 ИПД	1 ИПД	
	1 ГОТ	1 ГОТ	
Фраза REPORT (ОТЧЕТ)	1 ГОТ	1 ГОТ	
Фраза VALUE (ЗНАЧЕНИЕ)			
Имя-реализации литерал-1	1 ПОД	1 ПОД	+
	1 ОТД	1 ОТД	+
	1 ИПД	1 ИПД	+
	1 ГОТ	1 ГОТ	+
Имя-реализации несколько-литералов	1 ПОД	1 ПОД	+
	1 ОТД	1 ОТД	+
	1 ИПД	1 ИПД	+
	1 ГОТ	1 ГОТ	+
Имя-реализации имя-данного	2 ПОД	2 ПОД	+
	2 ОТД	2 ОТД	+
	2 ИПД	2 ИПД	+
	1 ГОТ	1 ГОТ	+
Имя-реализации несколько имен-данных	2 ПОД	2 ПОД	+
	2 ОТД	2 ОТД	+
	2 ИПД	2 ИПД	+
	1 ГОТ	1 ГОТ	+
Статья описания сортируемого-сливаемого файла	1 СРТ	1 СРТ	
Индикатор уровня SD (ОС)	1 СРТ	1 СРТ	
Фраза DATA RECORDS (ЗАПИСИ ДАННЫХ)	1 СРТ	1 СРТ	+
Фраза RECORD (В ЗАПИСИ)			
Целое-1 CHARACTERS (целое-1 ЛИТЕР)	1 СРТ	1 СРТ	
Фраза VARYING IN SIZE (ПЕРЕМЕННОЕ ЧИСЛО)	—	1 СРТ	
Целое-4 TO целое-5 CHARACTERS (целое-4 ДО целое-5 ЛИТЕР)	1 СРТ	1 СРТ	
Статья описания записи в секции файлов	1 ПОД	1 ПОД	
	1 ОТД	1 ОТД	
	1 ИПД	1 ИПД	
	1 СРТ	1 СРТ	
Секция рабочей памяти	1 ЯДР	1 ЯДР	
Статья описания записи	1 ЯДР	1 ЯДР	
Статья описания уровня 77	1 ЯДР	1 ЯДР	
Секция связи	1 МПС	1 МПС	
Статья описания записи	1 МПС	1 МПС	
Статья описания уровня 77	1 МПС	1 МПС	

GOST EXPERT ГОСТ 22558-89,

Элемент	Том 2 ГОСТ 22558	Настоящий стандарт
Секция коммуникаций	1 КОМ	1 КОМ
Статья описания коммуникации	1 КОМ	1 КОМ
Индикатор уровня CD (ОК)	1 КОМ	1 КОМ
Фраза FOR INPUT (ДЛЯ ВВОДА)	1 КОМ	1 КОМ
Фраза INITIAL (НАЧАЛЬНОГО)	2 КОМ	2 КОМ
Фраза END KEY (КЛЮЧ КОНЦА)	1 КОМ	1 КОМ
Фраза MESSAGE COUNT (ЧИСЛО СООБЩЕНИЙ)	1 КОМ	1 КОМ
Фраза MESSAGE DATE (ДАТА СООБЩЕНИЯ)	1 КОМ	1 КОМ
Фраза MESSAGE TIME (ВРЕМЯ СООБЩЕНИЯ)	1 КОМ	1 КОМ
Фраза SYMBOLIC QUEUE (СИМВОЛИЧЕСКАЯ ОЧЕРЕДЬ)	1 КОМ	1 КОМ
Фраза SYMBOLIC SOURCE (СИМВОЛИЧЕСКИЙ ИСТОЧНИК)	1 КОМ	1 КОМ
Фраза SYMBOLIC SUB-QUEUE-1 (СИМВОЛИЧЕСКАЯ ПОДОЧЕРЕДЬ-1)	1 КОМ	2 КОМ
Фраза SYMBOLIC SUB-QUEUE-2 (СИМВОЛИЧЕСКАЯ ПОДОЧЕРЕДЬ-2)	1 КОМ	2 КОМ
Фраза SYMBOLIC SUB-QUEUE-3 (СИМВОЛИЧЕСКАЯ ПОДОЧЕРЕДЬ-3)	1 КОМ	2 КОМ
Фраза STATUS KEY (КЛЮЧ СОСТОЯНИЯ)	1 КОМ	1 КОМ
Фраза TEXT LENGTH (ДЛИНА ТЕКСТА)	1 КОМ	1 КОМ
Несколько имен-данных	1 КОМ	2 КОМ
Фраза FOR OUTPUT (ДЛЯ ВЫВОДА)	1 КОМ	1 КОМ
Фраза DESTINATION COUNT (ЧИСЛО АДРЕСАТОВ)	1 КОМ	1 КОМ
Должен быть один	1 КОМ	1 КОМ
Должен быть один или больше	2 КОМ	2 КОМ
Фраза DESTINATION TABLE (ТАБЛИЦА АДРЕСАТОВ)	1 КОМ	2 КОМ
Фраза INDEXED BY (ИНДЕКСИРУЕТСЯ)	1 КОМ	2 КОМ
Фраза ERROR KEY (КЛЮЧ ОШИБКИ)	1 КОМ	1 КОМ
Фраза SYMBOLIC DESTINATION (СИМВОЛИЧЕСКИЙ АДРЕСАТ)	1 КОМ	1 КОМ
Фраза STATUS KEY (КЛЮЧ СОСТОЯНИЯ)	1 КОМ	1 КОМ
Фраза TEXT LENGTH (ДЛИНА ТЕКСТА)	1 КОМ	1 КОМ
Фраза FOR I-O (ДЛЯ ВВОДА-ВЫВОДА)	—	1 КОМ
Фраза INITIAL (НАЧАЛЬНОГО)	—	2 КОМ
Фраза END KEY (КЛЮЧ КОНЦА)	—	1 КОМ
Фраза MESSAGE DATE (ДАТА СООБЩЕНИЯ)	—	1 КОМ
Фраза MESSAGE TIME (ВРЕМЯ СООБЩЕНИЯ)	—	1 КОМ
Фраза STATUS KEY (КЛЮЧ СОСТОЯНИЯ)	—	1 КОМ
Фраза SYMBOLIC TERMINAL (СИМВОЛИЧЕСКИЙ ТЕРМИНАЛ)	—	1 КОМ
Фраза TEXT LENGTH (ДЛИНА ТЕКСТА)	—	1 КОМ
Несколько имен-данных	—	2 КОМ
Статья описания записи	1 КОМ	1 КОМ

GOST EXPERT ГОСТ 22558-89,

Элемент	ГОСТ 22557	Настоящий стандарт
Секция отчетов	1 ГОТ	1 ГОТ
Статья описания отчета	1 ГОТ	1 ГОТ
Индикатор уровня FD (00)	1 ГОТ	1 ГОТ
Фраза CODE (С КОДОМ)	1 ГОТ	1 ГОТ
Фраза CONTROL (УПРАВЛЕНИЕ)	1 ГОТ	1 ГОТ
Фраза GLOBAL (ГЛОБАЛЬНОЕ)	—	1 МПС
Фраза PAGE (РАЗМЕР СТРАНИЦЫ)	1 ГОТ	1 ГОТ
Статья описания группы отчета	1 ГОТ	1 ГОТ
Следующие фразы появляются в статье описания записи, статье описания данного, статье описания уровня 77 или в статье описания группы отчета.		
Фраза BLANK WHEN ZERO (ПРОБЕЛ КОГДА НУЛЬ)	1 ЯДР 1 ГОТ	1 ЯДР 1 ГОТ
Фраза COLUMN NUMBER (НОМЕР СТОЛБЦА)	1 ГОТ	1 ГОТ
Фраза имя-данного	1 ЯДР 1 ГОТ	1 ЯДР 1 ГОТ
Фраза EXTERNAL (ВНЕШНЕЕ)	—	2 МПС
Фраза FILLER (ЗАПОЛНИТЕЛЬ)	1 ЯДР	1 ЯДР
Фраза FILLER (ЗАПОЛНИТЕЛЬ) необязательна	—	1 ЯДР
Элементарное данное	1 ЯДР	1 ЯДР
Групповое данное	—	1 ЯДР
Фраза GLOBAL (ГЛОБАЛЬНОЕ)	—	2 МПС
Фраза JUSTIFIED (СДВИНУТО)	1 ЯДР 1 ГОТ	1 ЯДР 1 ГОТ
Фраза номер-уровня	1 ЯДР	1 ЯДР
От 01 до 10; представление двумя цифрами	1 ЯДР	—
От 01 до 49; представление одной или двумя цифрами	2 ЯДР 1 ГОТ	1 ЯДР 1 ГОТ
66	2 ЯДР	2 ЯДР
77	1 ЯДР	1 ЯДР
88	2 ЯДР	2 ЯДР
Фраза LINE NUMBER (НОМЕР СТРОКИ)	1 ГОТ	1 ГОТ
Фраза NEXT GROUP (СЛЕДУЮЩАЯ ГРУППА)	1 ГОТ	1 ГОТ
Фраза OCCURS (ПОВТОРЯЕТСЯ)	1 ТАБ	1 ЯДР
Целое TIMES (целое РАЗ)	1 ТАБ	1 ЯДР
Фраза ASCENDING/DESCENDING KEY (ПО ВОЗРАСТАНИЮ/УБЫВАНИЮ КЛЮЧА)	2 ТАБ	2 ЯДР
Фраза INDEXED BY (ИНДЕКСИРУЕТСЯ)	1 ТАБ	1 ЯДР
Фраза целое-1 TO целое-2 TIMES DEPENDING ON (ОТ целое-1 ДО целое-2 РАЗ В ЗАВИСИМОСТИ ОТ)	2 ТАБ	2 ЯДР
Целое-1 может быть нулем	—	2 ЯДР
Имя-данного в фразе DEPENDING ON (В ЗАВИСИМОСТИ ОТ) должно быть положительным целым	2 ТАБ	2 ЯДР
Фраза PICTURE (ШАБЛОН)	1 ЯДР 1 ГОТ	1 ЯДР 1 ГОТ
Строка литер содержит максимум 30 литер	1 ЯДР 1 ГОТ	1 ЯДР 1 ГОТ

GOST EXPERT ГОСТ 22558-89,

Элемент	ГОСТ 22558	Настоящий стандарт
Литеры данных: X 9 A	1 ЯДР 1 ГОТ	1 ЯДР 1 ГОТ
Операционные символы: S V P (З Т М)	1 ЯДР 1 ГОТ	1 ЯДР 1 ГОТ
Литеры фиксированной вставки: B + — . ,		
$ (¤) 0 CR (КР) DB (ДБ) /	1 ЯДР 1 ГОТ	1 ЯДР 1 ГОТ
Литера B разрешается в буквенном данном	1 ЯДР 1 ГОТ	— —
Литеры замены или плавающей вставки		
$ (¤) ' + — Z (П) *	1 ЯДР 1 ГОТ	1 ЯДР 1 ГОТ
Замена валютного знака	1 ЯДР 1 ГОТ	1 ЯДР 1 ГОТ
Замена десятичной точки	1 ЯДР 1 ГОТ	1 ЯДР 1 ГОТ
Фраза REDEFINES (ПЕРЕОПРЕДЕЛЯЕТ)	1 ЯДР	1 ЯДР
Не может быть вложенной	1 ЯДР	1 ЯДР
Может быть вложенной	2 ЯДР	2 ЯДР
Переопределение уровней 01 может быть больше размера поля оригинала	1 ЯДР	1 ЯДР
Переопределение уровней, отличных от 01, должно равняться размеру поля оригинала	1 ЯДР	—
Переопределение уровней, отличных от 01, должно быть меньше или равно размеру поля оригинала	—	1 ЯДР
Фраза RENAMES (ПЕРЕИМЕНОВЫВАЕТ)	2 ЯДР	2 ЯДР
Фраза SIGN (ЗНАК)	1 ЯДР —	1 ЯДР 1 ГОТ
Фраза SOURCE (ИСТОЧНИК)	1 ГОТ	1 ГОТ
Фраза SUM (СУММА)	1 ГОТ	1 ГОТ
Фраза SYNCHRONIZED (ВЫДЕЛЕНО)	1 ЯДР	1 ЯДР
Фраза TYPE (ТИП)	1 ГОТ	1 ГОТ
Фраза USAGE (об использовании)	1 ЯДР 1 ГОТ	1 ЯДР 1 ГОТ
BYNARY (ДВОИЧНОЕ)	—	1 ЯДР
COMPUTATIONAL (ДЛЯ ВЫЧИСЛЕНИЙ)	1 ЯДР	1 ЯДР
DISPLAY (ДЛЯ ВЫДАЧИ)	1 ЯДР 1 ГОТ	1 ЯДР 1 ГОТ
INDEX (ДЛЯ ИНДЕКСА)	1 ТАБ	1 ЯДР
PACKED-DECIMAL (ДЕСЯТИЧНОЕ)	—	1 ЯДР
Фраза VALUE (ЗНАЧЕНИЕ)	1 ЯДР 1 ГОТ	1 ЯДР 1 ГОТ
Литерал	1 ЯДР 1 ГОТ	1 ЯДР 1 ГОТ
Несколько литералов	2 ЯДР	2 ЯДР
Литерал-1 THROUGH литерал-2 (литерал-1 ПО литерал-2)	2 ЯДР	2 ЯДР
Несколько диапазонов литералов	2 ЯДР	2 ЯДР

GOST EXPERT ГОСТ 22558-89,

Элемент	ГОСТ 22558	Настоящий стандарт
1.5. Перечень отличий в разделе процедур		
РАЗДЕЛ ПРОЦЕДУР		
Раздел процедур обязателен	1 ЯДР	—
Раздел процедур необязателен	—	1 ЯДР
Заголовок раздела процедур	1 ЯДР	1 ЯДР
Фраза USING (ИСПОЛЬЗУЯ)	1 МПС	1 МПС
Разрешается по крайней мере 5 операндов	—	1 МПС
Для числа операндов нет ограничения	1 МПС	2 МПС
Декларативные процедуры	1 ПОД	1 ПОД
	1 ОТД	1 ОТД
	1 ИПД	1 ИПД
	1 ГОТ	1 ГОТ
	1 ОТД	1 ОТЛ +
Вариант DECLARATIVES (ДЕКЛАРАТИВЫ)	1 ПОД	1 ПОД
	1 ОТД	1 ОТД
	1 ИПД	1 ИПД
	1 ГОТ	1 ГОТ
	1 ОТЛ	1 ОТЛ +
Вариант END DECLARATIVES (КОНЕЦ ДЕКЛАРАТИВ)	1 ПОД	1 ПОД
	1 ОТД	1 ОТД
	1 ИПД	1 ИПД
	1 ГОТ	1 ГОТ
	1 ОТЛ	1 ОТЛ +
Арифметические выражения	2 ЯДР	2 ЯДР
Знаки бинарных арифметических операций + — * / **	2 ЯДР	2 ЯДР
Знаки унарных арифметических операций	2 ЯДР	2 ЯДР
Условные выражения	1 ЯДР	1 ЯДР
Простое условие	1 ЯДР	1 ЯДР
Условие отношения	1 ЯДР	1 ЯДР
Знаки условий отношения		
[NOT] GREATER THAN ([НЕ] БОЛЬШЕ)	1 ЯДР	1 ЯДР
[NOT] > ([НЕ] >)	2 ЯДР	1 ЯДР
[NOT] LESS THAN ([НЕ] МЕНЬШЕ)	1 ЯДР	1 ЯДР
[NOT] < ([НЕ] <)	2 ЯДР	1 ЯДР
[NOT] EQUAL TO ([НЕ] РАВНО)	1 ЯДР	1 ЯДР
[NOT] = ([НЕ] =)	2 ЯДР	1 ЯДР
GREATER THAN OR EQUAL TO (БОЛЬШЕ ИЛИ РАВНО)	—	1 ЯДР
>=	—	1 ЯДР
LESS THAN OR EQUAL TO (МЕНЬШЕ ИЛИ РАВНО)	—	1 ЯДР
<=	—	1 ЯДР
Сравнение числовых операндов	1 ЯДР	1 ЯДР
Сравнение нечисловых операндов		
Операнды должны быть одинакового размера	1 ЯДР	—
Операнды могут быть разного размера	2 ЯДР	1 ЯДР
Сравнение имен-индексов и (или) индексных данных	1 ТАБ	1 ЯДР

GOST EXPERT ГОСТ 22558-89,

Элемент	ГОСТ 22558	Настоящий стандарт
Условие класса	1 ЯДР	1 ЯДР
NUMERIC (ЧИСЛОВОЕ)	1 ЯДР	1 ЯДР
ALPHABETIC (БУКВЕННОЕ) (прописные буквенные литеры)	1 ЯДР	—
ALPHABETIC (БУКВЕННОЕ) (прописные и строчные буквенные литеры)	—	1 ЯДР
ALPHABETIC—LOWER (СТРОЧНЫЕ)	—	1 ЯДР
ALPHABETIC—UPPER (ПРОПИСНЫЕ)	—	1 ЯДР
Имя-класса	—	1 ЯДР
Условие имени-условия	2 ЯДР	2 ЯДР
Условие знака	2 ЯДР	2 ЯДР
Условие состояния-переключателя	1 ЯДР	1 ЯДР
Сложное условие	2 ЯДР	2 ЯДР
Знаки логических операций AND (И) OR (ИЛИ) NOT (НЕ)	2 ЯДР	2 ЯДР
Отрицание условия	2 ЯДР	2 ЯДР
Комбинированное условие	2 ЯДР	2 ЯДР
Условия в скобках	2 ЯДР	1 ЯДР
Сокращенные комбинированные условия отношения	2 ЯДР	2 ЯДР
Арифметические операторы	1 ЯДР	1 ЯДР
Арифметические операнды ограничены 18 цифрами	1 ЯДР	1 ЯДР
Размер операндов ограничен 18 цифрами	1 ЯДР	1 ЯДР
Оператор ACCEPT (ПРИНЯТЬ)	1 ЯДР	1 ЯДР
Идентификатор	1 ЯДР	1 ЯДР
Только одна передача данных	1 ЯДР	1 ЯДР
Нет ограничения на число передач данных	2 ЯДР	2 ЯДР
Фраза FROM мнемоническое-имя (С мнемоническое-имя)	2 ЯДР	2 ЯДР
Фраза FROM DATE/DAY/TIME (ДАТУ/ДЕНЬ/ВРЕМЯ)	2 ЯДР	2 ЯДР
Фраза FROM DAY-OF-WEEK (ДЕНЬ-НЕДЕЛИ)	—	2 ЯДР
Оператор ACCEPT MESSAGE COUNT (ПРИНЯТЬ ЧИСЛО СООБЩЕНИЙ)	1 КОМ	1 КОМ
Оператор ADD (СЛОЖИТЬ)	1 ЯДР	1 ЯДР
Идентификатор/литерал	1 ЯДР	1 ЯДР
Несколько идентификаторов/литералов	1 ЯДР	1 ЯДР
ТО идентификатор (С идентификатор)	1 ЯДР	1 ЯДР
ТО несколько идентификаторов (С несколько идентификаторов)	2 ЯДР	1 ЯДР
ТО идентификатор/литерал GIVING идентификатор (С идентификатор/литерал ПОЛУЧАЯ идентификатор)	—	1 ЯДР
ТО идентификатор/литерал GIVING несколько идентификаторов (С идентификатор/литерал ПОЛУЧАЯ несколько идентификаторов	—	1 ЯДР
GIVING идентификатор (ПОЛУЧАЯ идентификатор)	1 ЯДР	1 ЯДР
GIVING несколько идентификаторов (ПОЛУЧАЯ несколько идентификаторов)	2 ЯДР	1 ЯДР
Вариант ROUNDED (ОКРУГЛЯЯ)	1 ЯДР	1 ЯДР

GOST EXPERT ГОСТ 22558-89,

Элемент	ГОСТ 22555	Настоящий стандарт
Вариант ON SIZE ERROR (ПРИ ПЕРЕПОЛНЕНИИ)	1 ЯДР	1 ЯДР
Вариант NOT ON SIZE ERROR (БЕЗ ПЕРЕПОЛНЕНИЯ)	—	1 ЯДР
Вариант END-ADD (КОНЕЦ-СЛОЖИТЬ)	—	1 ЯДР
Вариант CORRESPONDING (СООТВЕТСТВЕННО)	2 ЯДР	2 ЯДР
Оператор ALTER (ИЗМЕНИТЬ)	1 ЯДР	1 ЯДР +
Только одно имя-процедуры	1 ЯДР	1 ЯДР +
Несколько имен-процедур	2 ЯДР	2 ЯДР +
Оператор CALL (ВЫЗВАТЬ)	1 МПС	1 МПС
Литерал	1 МПС	1 ПС
Идентификатор	2 МПС	2 МПС
Вариант USING (ИСПОЛЬЗУЯ)	1 МПС	1 МПС
Идентификатор	1 МПС	1 МПС
Разрешается по крайней мере 5 операндов	—	1 МПС
Нет ограничений на число допустимых операндов	1 МПС	2 МПС
Элементарное данное с уровнем, отличным от 01	—	1 МПС
Вариант BY REFERENCE (ССЫЛКУ НА)	—	2 МПС
Вариант BY CONTENT (ЗНАЧЕНИЕ)	—	2 МПС
Вариант ON OVERFLOW (ПРИ ПЕРЕПОЛНЕНИИ)	—	2 МПС
Вариант NOT ON OVERFLOW (БЕЗ ПЕРЕПОЛНЕНИЯ)	—	2 МПС
Вариант END-CALL (КОНЕЦ-ВЫЗВАТЬ) (формат 1)	—	1 МПС
Вариант END-CALL (КОНЕЦ-ВЫЗВАТЬ) (формат 2)	—	2 МПС
Оператор CANCEL (ОСВОБОДИТЬ)	2 МПС	2 МПС
Литерал	2 МПС	2 МПС
Идентификатор	2 МПС	2 МПС
Оператор CLOSE (ЗАКРЫТЬ)	1 ПОД	1 ПОД
	1 ОТД	1 ОТД
	1 ИПД	1 ИПД
	1 ГОТ	1 ГОТ
Имя-файла	1 ПОД	1 ПОД
	1 ОТД	1 ОТД
	1 ИПД	1 ИПД
	1 ГОТ	1 ГОТ
Несколько имен-файлов	2 ПОД	1 ПОД
	1 ОТД	1 ОТД
	1 ИПД	1 ИПД
Вариант REEL/UNIT (КАТУШКУ/ТОМ)	1 ПОД	1 ПОД
	1 ГОТ	1 ГОТ
Вариант FOR REMOVAL (С УДАЛЕНИЕМ)	2 ПОД	2 ПОД
	1 ГОТ	1 ГОТ
Вариант WITH NO REWIND (БЕЗ ПЕРЕМОТКИ)	2 ПОД	2 ПОД
	1 ГОТ	1 ГОТ

GOST EXPERT ГОСТ 22558-89,

Элемент	ГОСТ 22558	Настоящий стандарт
Вариант WITH LOCK (С ЗАМКОМ)	2 ПОД 1 ОТД 1 ИПД 1 ГОТ	2 ПОД 2 ОТД 2 ИПД 1 ГОТ
Оператор COMPUTE (ВЫЧИСЛИТЬ)	2 ЯДР	2 ЯДР
Арифметическое выражение	2 ЯДР	2 ЯДР
Несколько идентификаторов	2 ЯДР	2 ЯДР
Вариант ROUNDED (ОКРУГЛЯЯ)	2 ЯДР	2 ЯДР
Вариант ON SIZE ERROR (ПРИ ПЕРЕПОЛНЕНИИ)	2 ЯДР	2 ЯДР
Вариант NOT ON SIZE ERROR (БЕЗ ПЕРЕПОЛНЕНИЯ)	—	2 ЯДР
Вариант END-COMPUTE (КОНЕЦ-ВЫЧИСЛИТЬ)	—	2 ЯДР
Оператор CONTINUE (ПРОДОЛЖИТЬ)	—	1 ЯДР
Оператор DELETE (УДАЛИТЬ)	1 ОТД 1 ИПД	1 ОТД 1 ИПД
Вариант INVALID KEY (ПРИ ОШИБКЕ КЛЮЧА)	1 ОТД 1 ИПД	1 ОТД 1 ИПД
Вариант NOT INVALID KEY (БЕЗ ОШИБКИ КЛЮЧА)	—	1 ОТД 1 ИПД
Вариант END-DELETE (КОНЕЦ-УДАЛИТЬ)	—	1 ОТД 1 ИПД
Оператор DISABLE (ЗАПРЕТИТЬ)	1 КОМ	2 КОМ
Вариант INPUT (ВВОД)	1 КОМ	2 КОМ
Вариант TERMINAL (С ТЕРМИНАЛА)	2 КОМ	2 КОМ
Вариант I-O TERMINAL (ВВОД-ВЫВОД С ТЕРМИНАЛА)	—	2 КОМ
Вариант OUTPUT (ВЫВОД)	1 КОМ	2 КОМ
Вариант KEY (КЛЮЧ)	1 КОМ	2 КОМ +
Оператор DISPLAY (ВЫДАТЬ)	1 ЯДР	1 ЯДР
Только одна передача данных	1 ЯДР	1 ЯДР
Нет ограничений на число передач данных	2 ЯДР	2 ЯДР
Идентификатор/литерал	1 ЯДР	1 ЯДР
Несколько идентификаторов/литералов	1 ЯДР	1 ЯДР
Вариант UPON мнемоническое имя (НА мнемоническое имя)	2 ЯДР	2 ЯДР
Вариант WITH NO ADVANCING (БЕЗ ПРОДВИЖЕНИЯ)	—	2 ЯДР
Оператор DIVIDE (РАЗДЕЛИТЬ)	1 ЯДР	1 ЯДР
BY идентификатор/литерал (НА идентификатор/литерал)	1 ЯДР	1 ЯДР
INTO идентификатор (идентификатор НА)	1 ЯДР	1 ЯДР
INTO несколько идентификаторов (несколько идентификаторов НА)	2 ЯДР	1 ЯДР
GIVING идентификатор (ПОЛУЧАЯ идентификатор)	1 ЯДР	1 ЯДР
GIVING несколько идентификаторов (ПОЛУЧАЯ несколько идентификаторов)	2 ЯДР	1 ЯДР
Вариант ROUNDED (ОКРУГЛЯЯ)	1 ЯДР	1 ЯДР

GOST EXPERT ГОСТ 22558-89,

Элемент	ГОСТ 22558	Настоящий стандарт
Вариант REMAINDER (ОСТАТОК)	2 ЯДР	2 ЯДР
Вариант ON SIZE ERROR (ПРИ ПЕРЕПОЛНЕНИИ)	1 ЯДР	1 ЯДР
Вариант NOT ON SIZE ERROR (БЕЗ ПЕРЕПОЛНЕНИЯ)	—	1 ЯДР
Вариант END-DIVIDE (КОНЕЦ-РАЗДЕЛИТЬ)	—	1 ЯДР
Оператор ENABLE (РАЗРЕШИТЬ)	1 КОМ	2 КОМ
Вариант INPUT (ВВОД)	1 КОМ	2 КОМ
Вариант TERMINAL (С ТЕРМИНАЛА)	2 КОМ	2 КОМ
Вариант I-O TERMINAL (ВВОД-ВЫВОД С ТЕРМИНАЛА)	—	2 КОМ
Вариант OUTPUT (ВЫВОД)	1 КОМ	2 КОМ
Вариант KEY (КЛЮЧ)	1 КОМ	2 КОМ +
Оператор ENTER (ВОЙТИ)	0 ЯДР	1 ЯДР +
Оператор EVALUATE (ОЦЕНИТЬ)	—	2 ЯДР
Идентификатор/литерал	—	2 ЯДР
Арифметическое выражение	—	2 ЯДР
Условное выражение	—	2 ЯДР
TRUE/FALSE (ИСТИНА/ЛОЖЬ)	—	2 ЯДР
Вариант ALSO (ТАКЖЕ)	—	2 ЯДР
Вариант WHEN (КОГДА)	—	2 ЯДР
Вариант ALSO (ТАКЖЕ)	—	2 ЯДР
Вариант WHEN OTHER (ИНАЧЕ)	—	2 ЯДР
Вариант END-EVALUATE (КОНЕЦ-ОЦЕНИТЬ)	—	2 ЯДР
Оператор EXIT (ВЫЙТИ)	1 ЯДР	1 ЯДР
Оператор EXIT PROGRAM (ВЫЙТИ ИЗ ПРОГРАММЫ)	1 МПС	1 МПС
Оператор GENERATE (ГЕНЕРИРОВАТЬ)	1 ГОТ	1 ГОТ
Имя-данного	1 ГОТ	1 ГОТ
Имя-отчета	1 ГОТ	1 ГОТ
Оператор GO TO (ПЕРЕЙТИ К)	1 ЯДР	1 ЯДР
Имя-процедуры обязательно	1 ЯДР	1 ЯДР
Имя-процедуры необязательно	2 ЯДР	2 ЯДР +
Вариант DEPENDING ON (В ЗАВИСИМОСТИ ОТ)	1 ЯДР	1 ЯДР
Оператор IF (ЕСЛИ)	1 ЯДР	1 ЯДР
Только повелительные операторы	1 ЯДР	1 ЯДР
Повелительные и(или) условные операторы	2 ЯДР	2 ЯДР
Вложенные операторы IF (ЕСЛИ)	2 ЯДР	1 ЯДР
Необязательное слово THEN (ТО)	—	1 ЯДР
Вариант NEXT SENTENCE (СЛЕДУЮЩЕЕ ПРЕДЛОЖЕНИЕ)	1 ЯДР	1 ЯДР
Вариант ELSE (ИНАЧЕ)	1 ЯДР	1 ЯДР
Вариант END-IF (КОНЕЦ-ЕСЛИ)	—	1 ЯДР
Оператор INITIALIZE (ИНИЦИИРОВАТЬ)	—	2 ЯДР
Несколько идентификаторов	—	2 ЯДР
Вариант REPLACING (ЗАМЕНЯЯ)	—	2 ЯДР
Несколько REPLACING (ЗАМЕНЯЯ)	—	2 ЯДР
Оператор INITIATE (НАЧАТЬ)	1 ГОТ	1 ГОТ
Оператор INSPECT (ПРОСМОТРЕТЬ)	1 ЯДР	1 ЯДР
Только однолитерные данные	1 ЯДР	1 ЯДР

GOST EXPERT ГОСТ 22558-89,

Элемент	ГОСТ 22558	Настоящий стандарт
Многолитерные данные	2 ЯДР	2 ЯДР
Вариант TALLYING (СЧИТАЯ)	1 ЯДР	1 ЯДР
Вариант BEFORE/AFTER (ДО/ПОСЛЕ)	1 ЯДР	1 ЯДР
Несколько BEFORE/AFTER (ДО/ПОСЛЕ)	—	1 ЯДР
ALL/LEADING (ВСЕ/ВЕДУЩИЕ) несколько идентификаторов/литералов	—	2 ЯДР
Несколько фраз TALLYING (СЧИТАЯ)	2 ЯДР	2 ЯДР
Вариант REPLACING (ЗАМЕНЯЯ)	1 ЯДР	1 ЯДР
Вариант BEFORE/AFTER (ДО/ПОСЛЕ)	1 ЯДР	1 ЯДР
Несколько фраз BEFORE/AFTER (ДО/ПОСЛЕ)	—	2 ЯДР
Несколько ALL/LEADING/FIRST (ВСЕ/ВЕДУЩИЕ/ПЕРВЫЙ) идентификатор/литерал	2 ЯДР	2 ЯДР
Несколько фраз REPLACING (ЗАМЕНЯЯ)	—	2 ЯДР
Варианты TALLYING (СЧИТАЯ) и REPLACING (ЗАМЕНЯЯ)	1 ЯДР	1 ЯДР
Вариант CONVERTING (ПРЕВРАЩАЯ)	—	2 ЯДР
Оператор MERGE (СЛИТЬ)	2 СРТ	1 СРТ
Вариант ASCENDING/DESCENDING KEY (ПО ВОЗРАСТАНИЮ/УБЫВАНИЮ КЛЮЧА)	2 СРТ	1 СРТ
Вариант COLLATING SEQUENCE (АЛФАВИТ)	2 СРТ	1 СРТ
Вариант USING (ИСПОЛЬЗУЯ)	2 СРТ	1 СРТ
Вариант OUTPUT PROCEDURE (ПРОЦЕДУРА ВЫВОДА)	2 СРТ	1 СРТ
Имя-секции	2 СРТ	—
Имя-процедуры	—	1 СРТ
Вариант GIVING (ПОЛУЧАЯ)	2 СРТ	1 СРТ
Несколько фраз GIVING (ПОЛУЧАЯ)	—	1 СРТ
Файл в вариантах USING/GIVING (ИСПОЛЬЗУЯ/ПОЛУЧАЯ) должен быть последовательным файлом	2 СРТ	—
Файл в вариантах USING/GIVING (ИСПОЛЬЗУЯ/ПОЛУЧАЯ) может быть последовательным, относительным или индексным	—	1 СРТ
Оператор MOVE (ПОМЕСТИТЬ)	1 ЯДР	1 ЯДР
ТО идентификатор (В идентификатор)	1 ЯДР	1 ЯДР
ТО несколько идентификаторов (В несколько идентификаторов)	1 ЯДР	1 ЯДР
Вариант CORRESPONDING (СООТВЕТСТВЕННО)	2 ЯДР	2 ЯДР
Дередактирование числовых редактируемых данных	—	2 ЯДР
Оператор MULTIPLY (УМНОЖИТЬ)	1 ЯДР	1 ЯДР
BY идентификатор (НА идентификатор)	1 ЯДР	1 ЯДР
BY несколько идентификаторов (НА несколько идентификаторов)	2 ЯДР	1 ЯДР
GIVING идентификатор (ПОЛУЧАЯ идентификатор)	1 ЯДР	1 ЯДР
GIVING несколько идентификаторов (ПОЛУЧАЯ несколько идентификаторов)	2 ЯДР	1 ЯДР
Вариант ROUNDED (ОКРУГЛЯЯ)	1 ЯДР	1 ЯДР

GOST EXPERT ГОСТ 22558-89,

Элемент	ГОСТ 22558	Настоящий стандарт
Вариант ON SIZE ERROR (ПРИ ПЕРЕПОЛНЕНИИ)	1 ЯДР	1 ЯДР
Вариант NOT ON SIZE ERROR (БЕЗ ПЕРЕПОЛНЕНИЯ)	—	1 ЯДР
Вариант END-MULTIPLY (КОНЕЦ-УМНОЖИТЬ)	—	1 ЯДР
Оператор OPEN (ОТКРЫТЬ)	1 ПОД	1 ПОД
	1 ОТД	1 ОТД
	1 ИПД	1 ИПД
	1 ГОТ	1 ГОТ
Имя-файла	1 ПОД	1 ПОД
	1 ОТД	1 ОТД
	1 ИПД	1 ИПД
	1 ГОТ	1 ГОТ
Несколько имен-файлов	2 ПОД	1 ПОД
	1 ОТД	1 ОТД
	1 ИПД	1 ИПД
	1 ГОТ	1 ГОТ
Вариант INPUT (ВХОДНОЙ)	1 ПОД	1 ПОД
	1 ОТД	1 ОТД
	1 ИПД	1 ИПД
Вариант WITH NO REWIND (БЕЗ ПЕРЕМОТКИ)	2 ПОД	2 ПОД
Вариант REVERSED (РЕВЕРСНО)	2 ПОД	2 ПОД +
Вариант OUTPUT (ВЫХОДНОЙ)	1 ПОД	1 ПОД
	1 ОТД	1 ОТД
	1 ИПД	1 ИПД
	1 ГОТ	1 ГОТ
Вариант WITH NO REWIND (БЕЗ ПЕРЕМОТКИ)	2 ПОД	2 ПОД
	1 ГОТ	1 ГОТ
Вариант I-O (ВХОДНОЙ-ВЫХОДНОЙ)	1 ПОД	1 ПОД
	1 ОТД	1 ОТД
	1 ИПД	1 ИПД
Вариант EXTEND (ДОПОЛНЯЕМЫЙ)	2 ПОД	2 ПОД
		2 ОТД
		2 ИПД
		1 ГОТ
Несколько INPUT, OUTPUT, I-O (ВХОДНОЙ, ВЫХОДНОЙ, ВХОДНОЙ-ВЫХОДНОЙ)	1 ОТД	1 ОТД
	1 ИПД	1 ИПД
		1 ПОД
Несколько EXTEND (ДОПОЛНЯЕМЫЙ)	2 ПОД	2 ПОД
		2 ОТД
		2 ИПД
Оператор PERFORM (ВЫПОЛНИТЬ)	1 ЯДР	1 ЯДР
Имя-процедуры обязательно	1 ЯДР	—
Имя-процедуры необязательно	—	1 ЯДР
Вариант THROUGH имя-процедуры (ПО имя-процедуры)	1 ЯДР	1 ЯДР
Вариант повелительного-оператора	—	1 ЯДР
Вариант END-PERFORM (КОНЕЦ-ВЫПОЛНИТЬ)	—	1 ЯДР

GOST EXPERT ГОСТ 22558-89,

Элемент	ГОСТ 22558	Настоящий стандарт
Вариант TIMES (РАЗ)	1 ЯДР	1 ЯДР
Вариант UNTIL (ДО)	2 ЯДР	1 ЯДР
Вариант TEST BEFORE/AFTER (С ПРОВЕРКОЙ В НАЧАЛЕ/В КОНЦЕ)	—	2 ЯДР
Вариант VARYING (МЕНЯЯ)	2 ЯДР	2 ЯДР
Вариант TEST BEFORE/AFTER (С ПРОВЕРКОЙ В НАЧАЛЕ/В КОНЦЕ)	—	2 ЯДР
Вариант AFTER (ЗАТЕМ)	2 ЯДР	2 ЯДР
Максимум два AFTER (ЗАТЕМ)	2 ЯДР	—
Разрешено по крайней мере AFTER (ЗАТЕМ)	—	2 ЯДР
Идентификатор-2 увеличивается до установки идентификатора-5	—	2 ЯДР
Идентификатор-5 устанавливается до увеличения идентификатора-2	2 ЯДР	—
Оператор PURGE (ОЧИСТИТЬ)	—	2 КОМ
Оператор READ (ЧИТАТЬ)	1 ПОД	1 ПОД
	1 ОТД	1 ОТД
	1 ИПД	1 ИПД
Вариант NEXT (СЛЕДУЮЩУЮ)	2 ОТД	2 ОТД
	2 ИПД	2 ИПД
		2 ПОД
Вариант INTO (В)	1 ПОД	1 ПОД
	1 ОТД	1 ОТД
	1 ИПД	1 ИПД
Вариант AT END (В КОНЦЕ)	1 ПОД	1 ПОД
	1 ОТД	1 ОТД
	1 ИПД	1 ИПД
Вариант NOT AT END (НЕ В КОНЦЕ)	—	1 ПОД
		1 ОТД
		1 ИПД
Вариант KEY (КЛЮЧ)	2 ИПД	2 ИПД
Вариант INVALID KEY (ПРИ ОШИБКЕ КЛЮЧА)	1 ОТД	1 ОТД
	1 ИПД	1 ИПД
Вариант NOT INVALID KEY (БЕЗ ОШИБКИ КЛЮЧА)	—	1 ОТД
		1 ИПД
Вариант END-READ (КОНЕЦ-ЧИТАТЬ)	—	1 ПОД
		1 ОТД
		1 ИПД
Оператор RECEIVE (ПОЛУЧИТЬ)	1 КОМ	1 КОМ
Вариант MESSAGE (СООБЩЕНИЕ)	1 КОМ	1 КОМ
Вариант SEGMENT (СЕГМЕНТ)	2 КОМ	2 КОМ
	1 КОМ	1 КОМ
Вариант INTO (В)	1 КОМ	1 КОМ
Вариант NO DATA (НЕТ ДАННЫХ)	1 КОМ	1 КОМ
Вариант WITH DATA (ЕСТЬ ДАННЫЕ)	—	1 КОМ
Вариант END-RECEIVE (КОНЕЦ-ПОЛУЧИТЬ)	—	1 КОМ
Оператор RELEASE (ПЕРЕДАТЬ)	1 СРТ	1 СРТ
Вариант FROM (ИЗ ПОЛЯ)	1 СРТ	1 СРТ
Оператор RETURN (ВЕРНУТЬ)	1 СРТ	1 СРТ

GOST EXPERT ГОСТ 22558-89,

Элемент	ГОСТ 22558	Настоящий стандарт
Вариант INTO (В)	1 СРТ	1 СРТ
Вариант AT END (В КОНЦЕ)	1 СРТ	1 СРТ
Вариант NOT AT END (НЕ В КОНЦЕ)	—	1 СРТ
Вариант END-RETURN (КОНЕЦ-ВЕРНУТЬ)	—	1 СРТ
Оператор REWRITE (ОБНОВИТЬ)	1 ПОД	1 ПОД
	1 ОТД	1 ОТД
	1 ИПД	1 ИПД
Вариант FROM (ИЗ ПОЛЯ)	1 ПОД	1 ПОД
	1 ОТД	1 ОТД
	1 ИПД	1 ИПД
Вариант INVALID KEY (ПРИ ОШИБКЕ КЛЮЧА)	1 ОТД	1 ОТД
	1 ИПД	1 ИПД
Вариант NOT INVALID KEY (БЕЗ ОШИБКИ КЛЮЧА)	—	1 ОТД
		1 ИПД
Вариант END-REWRITE (КОНЕЦ-ОБНОВИТЬ)	—	1 ОТД
		1 ИПД
Оператор SEARCH (ИСКАТЬ)	2 ТАБ	2 ЯДР
Вариант VARYING (МЕНЯЯ)	2 ТАБ	2 ЯДР
Вариант AT END (В КОНЦЕ)	2 ТАБ	2 ЯДР
Вариант WHEN (КОГДА)	2 ТАБ	2 ЯДР
Несколько WHEN (КОГДА)	2 ТАБ	2 ЯДР
Вариант END-SEARCH (КОНЕЦ-ИСКАТЬ)	—	2 ЯДР
Оператор SEARCH ALL (ИСКАТЬ ОСОБО)	2 ТАБ	2 ЯДР
Вариант AT END (В КОНЦЕ)	2 ТАБ	2 ЯДР
Вариант WHEN (КОГДА)	2 ТАБ	2 ЯДР
Вариант END-SEARCH (КОНЕЦ-ИСКАТЬ)	—	2 ЯДР
Оператор SEND (ПОСЛАТЬ)	1 КОМ	1 КОМ
Вариант FROM идентификатор (ИЗ ПОЛЯ идентификатор) (часть сообщения)	2 КОМ	2 КОМ
Вариант FROM идентификатор (ИЗ ПОЛЯ идентификатор) (полное сообщение)	1 КОМ	1 КОМ
Вариант WITH идентификатор (С идентификатор)	2 КОМ	2 КОМ
Вариант WITH ESI (С ИКС)	2 КОМ	2 КОМ
Вариант WITH EMI (С ИКЩ)	1 КОМ	1 КОМ
Вариант WITH EGI (С ИКГ)	1 КОМ	1 КОМ
Вариант BEFORE/AFTER ADVANCING (ДО/ПОСЛЕ ПРОДВИЖЕНИЯ)	1 КОМ	1 КОМ
Целое LINE/LINES (целое СТРОК)	1 КОМ	1 КОМ
Идентификатор LINE/LINES (идентификатор СТРОК)	1 КОМ	1 КОМ
Мнемоническое-имя	1 КОМ	2 КОМ
PAGE (СТРАНИЦЫ)	1 КОМ	1 КОМ
Вариант REPLACING LINE (ЗАМЕНЯЯ СТРОКУ)	—	2 КОМ
Оператор SET (УСТАНОВИТЬ)	1 ТАБ	1 ЯДР
Имя-индекса/идентификатор TO (НА)	1 ТАБ	1 ЯДР
Имя-индекса UP BY/DOWN BY (имя-индекса ПРИБАВЛЯЯ/ВЫЧИТАЯ)	1 ТАБ	1 ЯДР

GOST EXPERT ГОСТ 22558-89,

Элемент	ГОСТ 22558	Настоящий стандарт
Мнемоническое имя TO ON/OFF (мнемоническое имя НА ВКЛЮЧЕНО/ВЫКЛЮЧЕНО)	—	1 ЯДР
Имя-условия TO TRUE (НА ИСТИНА)	—	2 ЯДР
Оператор SORT (СОРТИРОВАТЬ)	1 СРТ	1 СРТ
Вариант ASCENDING/DESCENDING KEY (ПО ВОЗРАСТАНИЮ/УБЫВАНИЮ КЛЮЧА)	1 СРТ	1 СРТ
Вариант DUPLICATES (С ДУБЛИРОВАНИЕМ)	—	1 СРТ
Вариант COLLATING SEQUENCE (АЛФАВИТ)	2 СРТ	1 СРТ
Вариант INPUT PROCEDURE (ПРОЦЕДУРА ВВОДА)	1 СРТ	1 СРТ
Имя-секции	1 СРТ	—
Имя-процедуры	—	1 СРТ
Вариант USING (ИСПОЛЬЗУЯ)	1 СРТ	1 СРТ
Несколько имен-файлов	2 СРТ	1 СРТ
Вариант OUTPUT PROCEDURE (ПРОЦЕДУРА ВЫВОДА)	1 СРТ	1 СРТ
Имя-секции	1 СРТ	—
Имя-процедуры	—	1 СРТ
Вариант GIVING (ПОЛУЧАЯ)	1 СРТ	1 СРТ
Несколько имен-файлов	—	1 СРТ
Оператор START (ПОДВЕСТИ)	2 ОТД	2 ОТД
	2 ИПД	2 ИПД
Вариант KEY (КЛЮЧ)	2 ОТД	2 ОТД
	2 ИПД	2 ИПД
EQUAL TO (РАВНО)	2 ОТД	2 ОТД
	2 ИПД	2 ИПД
=	2 ОТД	2 ОТД
	2 ИПД	2 ИПД
GREATER THAN (БОЛЬШЕ)	2 ОТД	2 ОТД
	2 ИПД	2 ИПД
>	2 ОТД	2 ОТД
	2 ИПД	2 ИПД
NOT LESS THAN (НЕ МЕНЬШЕ)	2 ОТД	2 ОТД
	2 ИПД	2 ИПД
НЕ <	2 ОТД	2 ОТД
	2 ИПД	2 ИПД
GREATER THAN OR EQUAL TO (БОЛЬШЕ ИЛИ РАВНО)	—	2 ОТД
		2 ИПД
>=	—	2 ОТД
		2 ИПД
Вариант INVALID KEY (ПРИ ОШИБКЕ КЛЮЧА)	2 ОТД	2 ОТД
	2 ИПД	2 ИПД
Вариант NOT INVALID KEY (БЕЗ ОШИБКИ КЛЮЧА)	—	2 ОТД
		2 ИПД
Вариант END-START (КОНЕЦ-ПОДВЕСТИ)	—	2 ОТД
		2 ИПД
Оператор STOP (ОСТАНОВИТЬ)	1 ЯДР	1 ЯДР

GOST EXPERT ГОСТ 22558-89,

Элемент	ГОСТ 22558	Настоящий стандарт
RUN (РАБОТАТЬ)	1 ЯДР	1 ЯДР
Литерал	1 ЯДР	1 ЯДР +
*Оператор STRING (СОБРАТЬ)	2 ЯДР	2 ЯДР
Несколько DELIMITED BY (ОГРАНИЧИВАЯСЬ)	2 ЯДР	2 ЯДР
Вариант WITH POINTER (УКАЗАТЕЛЬ)	2 ЯДР	2 ЯДР
Вариант ON OVERFLOW (ПРИ ПЕРЕПОЛНЕНИИ)	2 ЯДР	2 ЯДР
Вариант NOT ON OVERFLOW (БЕЗ ПЕРЕПОЛНЕНИЯ)	—	2 ЯДР
Вариант END-STRING (КОНЕЦ-СОБРАТЬ)	—	2 ЯДР
*Оператор SUBTRACT (ОТНЯТЬ)	1 ЯДР	1 ЯДР
Идентификатор/литерал	1 ЯДР	1 ЯДР
Несколько идентификаторов/литералов	1 ЯДР	1 ЯДР
FROM идентификатор (ОТ идентификатор)	1 ЯДР	1 ЯДР
FROM несколько идентификаторов (ОТ несколько идентификаторов)	2 ЯДР	1 ЯДР
GIVING идентификатор (ПОЛУЧАЯ идентификатор)	1 ЯДР	1 ЯДР
GIVING несколько идентификаторов (ПОЛУЧАЯ несколько идентификаторов)	2 ЯДР	1 ЯДР
Вариант ROUNDED (ОКРУГЛЯЯ)	1 ЯДР	1 ЯДР
Вариант ON SIZE ERROR (ПРИ ПЕРЕПОЛНЕНИИ)	1 ЯДР	1 ЯДР
Вариант NOT ON SIZE ERROR (БЕЗ ПЕРЕПОЛНЕНИЯ)	—	1 ЯДР
Вариант END-SUBTRACT (КОНЕЦ-ОТНЯТЬ)	—	1 ЯДР
Вариант CORRESPONDING (СООТВЕТСТВЕННО)	2 ЯДР	2 ЯДР
*Оператор SUPPRESS (ПОДАВИТЬ)	1 ГОТ	1 ГОТ
*Оператор TERMINATE (ЗАКОНЧИТЬ)	2 ЯДР	2 ЯДР
*Оператор UNSTRING (РАЗОБРАТЬ)	2 ЯДР	2 ЯДР
Вариант DELIMITED BY (ОГРАНИЧИВАЯСЬ)	2 ЯДР	2 ЯДР
Вариант DELIMITER IN (ОГРАНИЧИТЕЛЬ В)	2 ЯДР	2 ЯДР
Вариант COUNT IN (СЧЕТ В)	2 ЯДР	2 ЯДР
Вариант WITH POINTER (УКАЗАТЕЛЬ)	2 ЯДР	2 ЯДР
Вариант TALLYING (СЧИТАЯ)	2 ЯДР	2 ЯДР
Вариант ON OVERFLOW (ПРИ ПЕРЕПОЛНЕНИИ)	2 ЯДР	2 ЯДР
Вариант NOT ON OVERFLOW (БЕЗ ПЕРЕПОЛНЕНИЯ)	—	2 ЯДР
Вариант END-UNSTRING (КОНЕЦ-РАЗОБРАТЬ)	—	2 ЯДР
*Оператор USE (ИСПОЛЬЗОВАТЬ)	1 ПОД 1 ОТД 1 ИПД 1 ГОТ 1 ОТЛ	1 ПОД 1 ОТД 1 ИПД 1 ГОТ 1 ОТЛ +
Вариант EXCEPTION/ERROR PROCEDURE (ПРОЦЕДУРЫ ОШИБКИ)	1 ПОД 1 ОТД 1 ИПД	1 ПОД 1 ОТД 1 ИПД

GOST EXPERT ГОСТ 22558-89,

Элемент	ГОСТ 22558	Настоящий стандарт
	1 ГОТ	1 ГОТ
Вариант GLOBAL (ГЛОБАЛЬНО)	—	2 МПС
Вариант ON имя-файла (ДЛЯ имя-файла)	1 ПОД	1 ПОД
	1 ОТД	1 ОТД
	1 ИПД	1 ИПД
	1 ГОТ	1 ГОТ
Вариант ON несколько имен-файлов (ДЛЯ несколько имен-файлов)	2 ПОД	2 ПОД
	2 ОТД	2 ОТД
	2 ИПД	2 ИПД
	1 ГОТ	1 ГОТ
ON INPUT (ДЛЯ ВХОДНЫХ)	1 ПОД	1 ПОД
	1 ОТД	1 ОТД
	1 ИПД	1 ИПД
ON OUTPUT (ДЛЯ ВЫХОДНЫХ)	1 ПОД	1 ПОД
	1 ОТД	1 ОТД
	1 ИПД	1 ИПД
	1 ГОТ	1 ГОТ
ON I-O (ДЛЯ ВХОДНЫХ-ВЫХОДНЫХ)	1 ПОД	1 ПОД
	1 ОТД	1 ОТД
	1 ИПД	1 ИПД
ON EXTEND (ДЛЯ ДОПОЛНЯЕМЫХ)	2 ПОД	2 ПОД
		2 ОТД
		2 ИПД
		1 ГОТ
Вариант BEFORE REPORTING (ДО ВЫДАЧИ)	1 ГОТ	1 ГОТ
Вариант GLOBAL (ГЛОБАЛЬНО)	—	2 МПС
Вариант FOR DEBUGGING (ДЛЯ ОТЛАДКИ)	1 ОТЛ	1 ОТЛ +
Имя-процедуры	1 ОТЛ	1 ОТЛ +
ALL PROCEDURES (ВСЕХ ПРОЦЕДУРАХ)	1 ОТЛ	1 ОТЛ +
ALL REFERENCES OF идентификатор-1 (ВСЕХ ССЫЛКАХ НА идентификатор-1)	2 ОТЛ	2 ОТЛ +
Имя-коммуникации	2 ОТЛ	2 ОТЛ +
Имя-файла	2 ОТЛ	2 ОТЛ +
Оператор WRITE (ПИСАТЬ)	1 ПОД	1 ПОД
	1 ОТД	1 ОТД
	1 ИПД	1 ИПД
Вариант FROM (ИЗ ПОЛЯ)	1 ПОД	1 ПОД
	1 ОТД	1 ОТД
	1 ИПД	1 ИПД
Вариант BEFORE/AFTER ADVANCING (ДО/ПОСЛЕ ПРОДВИЖЕНИЯ)	1 ПОД	1 ПОД
Целое LINE/LINES (СТРОК)	1 ПОД	1 ПОД
Идентификатор LINE/LINES (СТРОК)	2 ПОД	1 ПОД
Мнемоническое-имя	2 ПОД	2 ПОД
PAGE (СТРАНИЦЫ)	1 ПОД	1 ПОД
Вариант AT END-OF-PAGE/EOP (В КОНЦЕ СТРАНИЦЫ)	2 ПОД	2 ПОД
Вариант NOT AT END-OF-PAGE/EOP (НЕ В КОНЦЕ СТРАНИЦЫ)	—	2 ПОД
Вариант INVALID KEY (ПРИ ОШИБКЕ КЛЮЧА)	1 ОТД	1 ОТД
	1 ИПД	1 ИПД

GOST EXPERT ГОСТ 22558-89,

Элемент	ГОСТ 22558	Настоящий стандарт
Вариант NOT INVALID KEY (БЕЗ ОШИБКИ КЛЮЧА)	—	1 ОТД 1 ИПД
Вариант END-WRITE (КОНЕЦ-ПИСАТЬ)	—	2 ПОД 1 ОТД 1 ИПД

1.6. Дополнительный список отличий

СЕГМЕНТАЦИЯ			
Номер-сегментов от 0 до 49 для постоянных сегментов	1 СЕГ	1 СЕГ	+
Номера-сегментов от 50 до 99 для независимых сегментов	1 СЕГ	1 СЕГ	+
Все секции с одинаковыми номерами сегментов должны быть записаны подряд в исходной программе	1 СЕГ	1 СЕГ	+
Секции с одинаковыми номерами-сегментов не обязательно должны физически следовать одна за другой в исходной программе	2 СЕГ	2 СЕГ	+
ОБРАБОТКА ИСХОДНЫХ ТЕКСТОВ			
Оператор COPY (КОПИРОВАТЬ)	1 БИБ	1 ОИТ	
Вариант OF/IN имя-библиотеки (ИЗ имя библиотеки)	2 БИБ	2 ОИТ	
Вариант REPLACING (ЗАМЕНЯЯ)	2 БИБ	2 ОИТ	
Псевдотекст	2 БИБ	2 ОИТ	
Идентификатор	2 БИБ	2 ОИТ	
Литерал	2 БИБ	2 ОИТ	
Слово	2 БИБ	2 ОИТ	
Оператор REPLACE (ЗАМЕНИТЬ)	—	2 ОИТ	
Псевдотекст BY псевдотекст (псевдотекст НА псевдотекст)	—	2 ОИТ	
OFF (ОТКЛЮЧИТЬ ЗАМЕНИТЬ)	—	2 ОИТ	

2. СУЩЕСТВЕННЫЕ ИЗМЕНЕНИЯ

2.1. Существенные изменения, не влияющие на имеющиеся программы

Приведенный ниже список изменений расширяет средства ГОСТ 22558. Эти изменения являются новыми средствами, не влияющими на имеющиеся программы, например, новые глаголы или дополнительные возможности для старого глагола.

(1) Строчные буквы (1 ЯДР). Если набор литер ЭВМ включает строчные буквы, их можно использовать в строке-литер. Каждая такая литера эквивалентна прописной литере, за исключением случая, когда строчные буквы используются в нечисловом литерале.

(2) Двоеточие (:) (2 ЯДР). Набор литер Кобола расширен включением двоеточия, которое используется при модификации ссылок.

GOST EXPERT ГОСТ 22558-89,

(3) Литеры пунктуации (1 ЯДР). Разделители запятая, точка с запятой и пробел взаимозаменяемы в исходной программе.

(4) Слова, определенные пользователем, и системные имена (1 ЯДР). Одно и то же слово Кобола в исходной программе можно использовать и как системное имя и как слово, определенное пользователем; чем является данное слово — определяется по контексту.

(5) Символические литеры (2 ЯДР). Символическая литера является определенным пользователем словом, определяющим стандартную константу.

(6) Нечисловой литерал (1 ЯДР). Верхняя граница длины нечислового литерала 160 литер.

В предыдущем стандарте Кобола верхняя граница была 120 литер.

(7) Стандартная константа ZERO (НУЛЬ) (2 ЯДР). Стандартная константа ZERO (НУЛЬ) допустима в арифметических выражениях.

(8) Единственность ссылки (1 ЯДР). Определенное пользователем слово не обязательно должно быть уникальным или иметь возможность быть уточненным, если на него не ссылаются.

(9) Уточнения (2 ЯДР). Реализация должна обеспечивать возможность обработки 50 уровней уточнения. В предыдущем стандарте Кобола разрешалось 5 уровней уточнения.

(10) Индексирование (2 ЯДР). Верхняя граница размерности таблицы — семь.

(11) Относительное индексирование (1 ЯДР). Относительное индексирование допускает, чтобы за индексом следовал знак + или , за которым следует целое.

(12) Использование индексов и индексных данных (1 ЯДР). И индексы, и индексные данные могут быть записаны одновременно в одном наборе индексов, используемом для ссылки на отдельное вхождение в многомерной таблице.

(13) Модификация ссылки (2 ЯДР). Модификация ссылки является новым методом ссылки на данное посредством указания самой левой литеры и длины данного.

(14) Порядковый номер (1 ЯДР). Порядковый номер может содержать любую литеру из набора литер машины. В ГОСТ 22558 порядковый номер мог содержать только цифры.

(15) Формат представления раздела данных (1 ЯДР). Слово, следующее за указателем уровня, номером уровня 01 или номером уровня 77 на одной и той же строке, может начинаться в поле А.

(16) Заголовок конца программы (2 ЯДР). Заголовок конца программы указывает конец названной исходной Кобол-программы; за заголовком конца программы может следовать Кобол-программа, которая должна компилироваться отдельно при одном и том же вызове компилятора.

(17) Вложенные исходные программы (2 МПС). Программы могут содержаться в других программах.

(18) Фраза INITIAL (НАЧАЛЬНАЯ) в параграфе PROGRAM-ID (ПРОГРАММА) (2 МПС). Фраза INITIAL (НАЧАЛЬНАЯ) определяет программу, состояние которой инициировано при каждом вызове программы в то же состояние, как если бы программа вызывалась впервые в единице исполнения.

(19) Фраза COMMON (ОБЩАЯ) в параграфе PROGRAM-ID (ПРОГРАММА) (2 МПС). Фраза COMMON (ОБЩАЯ) определяет программу, которая, хотя и содержится непосредственно в другой программе, может быть вызвана из любой программы, прямо или косвенно содержащейся в этой другой программе.

(20) Раздел оборудования (1 ЯДР). Раздел оборудования необязателен. В разделе оборудования секция конфигурации необязательна. Параграф SOURCE-

GOST EXPERT ГОСТ 22558-89,

COMPUTER (ИСХОДНАЯ-МАШИНА), параграф OBJECT-COMPUTER (РАБОЧАЯ-МАШИНА), статьи в параграфе FILE-CONTROL (УПРАВЛЕНИЕ-ФАЙЛАМИ) и статьи в параграфе I—O-CONTROL (УПРАВЛЕНИЕ-ВВОДОМ-ВЫВОДОМ) также необязательны.

(21) Параграф SPECIAL-NAMES (СПЕЦИАЛЬНЫЕ ИМЕНА) (1 ЯДР). Если имя-реализации есть переключатель, нет необходимости указывать имя-условия.

(22) Параграф SPECIAL-NAMES (СПЕЦИАЛЬНЫЕ-ИМЕНА) (1 ЯДР). Зарезервированное слово IS (ЕСТЬ) сделано необязательным в параграфе SPECIAL-NAMES (СПЕЦИАЛЬНЫЕ-ИМЕНА) в соответствии с использованием слова IS (ЕСТЬ) в спецификациях Кобола.

(23) Вариант STANDARD-2 (СТАНДАРТ-М) (1 ЯДР). Вариант STANDARD-2 (СТАНДАРТ-М) фразы ALPHABET (АЛФАВИТ) параграфа SPECIAL-NAMES (СПЕЦИАЛЬНЫЕ-ИМЕНА) допускает спецификацию набора 7-битовых литер ISO для алфавита или основной последовательности.

(24) Фраза ASSIGN (НАЗНАЧИТЬ) (1 ПОД, 1 ОТД, 1 ИПД, 1 СРТ, 1 ГОТ). Во фразе ASSIGN (НАЗНАЧИТЬ) может быть указан нечисловой литерал.

(25) Фраза OPTIONAL (НЕОБЯЗАТЕЛЬНОГО) (2 ПОД, 2 ОТД, 2 ИПД). Фраза OPTIONAL (НЕОБЯЗАТЕЛЬНОГО) в статье управления файлом относится к последовательным файлам, относительным файлам и индексным файлам, открытым в режиме ввода, ввода-вывода или дополнения. В предыдущем стандарте Кобола фраза OPTIONAL (НЕОБЯЗАТЕЛЬНОГО) в статье управления файлом применялась к последовательным файлам, открытым в режиме ввода.

(26) Фраза ORGANIZATION (ОРГАНИЗАЦИЯ) (1 ПОД, 1 ОТД, 1 ИПД). Во фразе ORGANIZATION (ОРГАНИЗАЦИЯ) статьи управления файлом слова ORGANIZATION IS (ОРГАНИЗАЦИЯ) необязательны.

(27) Фраза PADDING CHARACTER (ЛИТЕРА ЗАПОЛНИТЕЛЬ) (2 ПОД, 1 ГОТ). Фраза PADDING CHARACTER (ЛИТЕРА ЗАПОЛНИТЕЛЬ) в статье управления файлом определяет литеру, которая должна быть использована для заполнения блока в последовательных файлах.

(28) Фраза RECORD DELIMITER (ОГРАНИЧИТЕЛЬ ЗАПИСИ) (2 ПОД, 1 ГОТ). Фраза RECORD DELIMITER (ОГРАНИЧИТЕЛЬ ЗАПИСИ) в статье управления файлом указывает способ определения длины записи переменной длины во внешней среде.

(29) Параграф I-O-CONTROL (УПРАВЛЕНИЕ-ВВОДОМ-ВЫВОДОМ) (1 ПОД, 1 ОТД, 1 ИПД, 1 ГОТ). В параграфе I-O-CONTROL (УПРАВЛЕНИЕ ВВОДОМ-ВЫВОДОМ) порядок фраз несущественен.

(30) Раздел данных (1 ЯДР). Раздел данных необязателен.

(31) Фраза BLOCK CONTAINS (В БЛОКЕ) (1 ПОД, 1 ОТД, 1 ИПД). Отсутствие фразы BLOCK CONTAINS (В БЛОКЕ) разрешается, если число записей, содержащихся в блоке, указывается операционным окружением. В предыдущем стандарте Кобола отсутствие фразы BLOCK CONTAINS (В БЛОКЕ) обозначало стандартный размер физической записи, определяемый реализацией.

(32) Фраза CODE-SET (АЛФАВИТ) (1 ПОД, 1 ГОТ). Фраза CODESET (АЛФАВИТ) может быть указана для всех файлов с последовательной организацией. В предыдущем стандарте Кобола была ограничена для файлов не массовой памяти.

(33) Фраза LABEL RECORDS (МЕТКИ) (1 ПОД, 1 ОТД, 1 ИПД, 1 ГОТ). Фраза LABEL RECORDS (МЕТКИ) необязательна; если она не указана, подразумевается фраза LABEL RECORDS ARE STANDARD (МЕТКИ СТАНДАРТНЫ).

GOST EXPERT ГОСТ 22558-89,

(34) Фраза LINAGE (ВЕРСТКА) (2 ПОД). Имена-данных во фразе LINAGE (ВЕРСТКА) могут быть уточнены.

(35) Фраза EXTERNAL (ВНЕШНЕЕ) (2 МПС). Фраза EXTERNAL (ВНЕШНЕЕ) указывает, что данное или определитель файла внешние и могут быть доступны и обработаны в любой программе единицы исполнения.

(36) Фраза GLOBAL (ГЛОБАЛЬНОЕ) (2 МПС). Фраза GLOBAL (ГЛОБАЛЬНОЕ) указывает, что имя-данного или имя-файла является глобальным именем и доступно любой программе, содержащейся в программе, объявляющей это имя.

(37) Фраза FILLER (ЗАПОЛНИТЕЛЬ) (1 ЯДР). Использование слоза FILLER (ЗАПОЛНИТЕЛЬ) необязательно для статей описания данных. Слово FILLER (ЗАПОЛНИТЕЛЬ) может появиться в статье описания данного, содержащей фразу REDEFINES (ПЕРЕОПРЕДЕЛЯЕТ). Слово FILLER (ЗАПОЛНИТЕЛЬ) может быть использовано в статье описания данного или группы данных.

(38) Фраза OCCURS (ПОВТОРЯЕТСЯ) (2 ЯДР). Данное, указанное в фразе DEPENDING ON (В ЗАВИСИМОСТИ ОТ), может иметь нулевое значение. Таким образом, минимальное число вхождений может равняться нулю.

(39) Строка-литер шаблона (2 ЯДР, 1 ГОТ). Строка литер шаблона может быть продолжена на следующих строках кодирования.

(40) Фраза PICTURE (ШАБЛОН) (1 ЯДР). Литера вставки «.» (точка) или «,» (запятая) может быть использована как последняя литера строки-литер шаблона, если за ней непосредственно следует разделитель точка, заканчивающая статью описания данного.

(41) Фраза RECORD (В ЗАПИСИ) (2 ПОД, 2 ОТД, 2 ИПД, 1 СРТ). Вариант VARYING (ПЕРЕМЕННОЕ ЧИСЛО) в фразе RECORD (В ЗАПИСИ) используется для указания записей переменной длины. Слова DEPENDING ON (В ЗАВИСИМОСТИ ОТ), связанные с вариантом VARYING (ПЕРЕМЕННОЕ ЧИСЛО), определяют данное, содержащее число позиций литер в записи.

(42) Фраза REDEFINES (ПЕРЕОПРЕДЕЛЯЕТ) (1 ЯДР). Размер данного, связанного с фразой REDEFINES (ПЕРЕОПРЕДЕЛЯЕТ), может быть меньше или равен размеру переопределяемого данного. В ГОСТ 22558 оба данных должны были иметь одинаковое число позиций литер.

(43) Фраза SIGN (ЗНАК) (1 ЯДР). В иерархии статьи описания данных может быть указано несколько фраз SIGN (ЗНАК); спецификация на подчиненном уровне имеет приоритет над спецификацией уровня содержащей группы.

(44) Фраза SIGN (ЗНАК) (1 ГОТ). Фраза SIGN (ЗНАК) разрешена в статье описания группы отчета.

(45) Фраза USAGE (фраза об использовании) (1 ЯДР). Во фразе об использовании введены два новых варианта BINARY (ДВОИЧНОЕ) и PACKED-DECIMAL (ДЕСЯТИЧНОЕ).

(46) Фраза VALUE (ЗНАЧЕНИЕ) (1 ЯДР). Фраза VALUE (ЗНАЧЕНИЕ) может быть указана в статье описания данного, содержащей фразу OCCURS (ПОВТОРЯЕТСЯ). Фраза VALUE (ЗНАЧЕНИЕ) может быть указана в статье описания данного, подчиненной статье, содержащей фразу OCCURS (ПОВТОРЯЕТСЯ). В ГОСТ 22558 фраза VALUE (ЗНАЧЕНИЕ) не разрешалась в статье описания данного, содержащей фразу OCCURS, или в статье описания данного, подчиненной статье, содержащей фразу OCCURS (ПОВТОРЯЕТСЯ).

(47) Статья описания коммуникации (1 КОМ). Порядок фраз в статье описания коммуникации несущественен.

(48) Вариант FOR I-O (ДЛЯ ВВОДА-ВЫВОДА) в статье описания коммуникации (1 КОМ). Вариант FOR I-O (ДЛЯ ВВОДА-ВЫВОДА) в статье описания коммуникации обеспечивает функции ввода и вывода одной статьей ОК.

GOST EXPERT ГОСТ 22558-89,

(49) Фраза LINE NUMBER (НОМЕР СТРОКИ) (1 ГОТ). В качестве относительного номера строки в варианте PLUS (ПЛЮС) фразы LINE NUMBER (НОМЕР СТРОКИ) может быть указано целое 0.

(50) Раздел процедур (1 ЯДР). Раздел процедур не обязателен.

(51) Заголовок раздела процедур (1 МПС). На данное секции связи, которое переопределяет или подчинено переопределяющему данному, и на данное, появляющееся в заголовке раздела процедур, можно ссылаться в разделе процедур.

(52) Ограничители области действия (1 ЯДР, 1 ПОД, 1 ОТД, 1 ИНД, 2 МПС, 1 СРТ, 1 КОМ). Ограничители области действия служат для указания границы действия определенных процедурных операторов. Ограничители области действия следующие: END-ADD (КОНЕЦ-СЛОЖИТЬ), END-CALL (КОНЕЦ-ВЫЗВАТЬ), END-COMPUTE (КОНЕЦ-ВЫЧИСЛИТЬ), END-DELETE (КОНЕЦ-УДАЛИТЬ), END-DIVIDE (КОНЕЦ-РАЗДЕЛИТЬ), END-EVALUATE (КОНЕЦ-ОЦЕНИТЬ), END-IF (КОНЕЦ-ЕСЛИ), END-MULTIPLY (КОНЕЦ-УМНОЖИТЬ), END-PERFORM (КОНЕЦ-ВЫПОЛНИТЬ), END-READ (КОНЕЦ-ЧИТАТЬ), END-RECEIVE (КОНЕЦ-ПОЛУЧИТЬ), END-RETURN (КОНЕЦ-ВЕРНУТЬ), END-REWRITE (КОНЕЦ-ОБНОВИТЬ), END-SEARCH (КОНЕЦ-ИСКАТЬ), END-START (КОНЕЦ-ПОДВЕСТИ), END-STRING (КОНЕЦ-СОБРАТЬ), END-SUBTRACT (КОНЕЦ-ОТНЯТЬ), END-UNSTRING (КОНЕЦ-РАЗОБРАТЬ), END-WRITE (КОНЕЦ-ПИСАТЬ).

(53) Знаки операций отношения (1 ЯДР). Знак операции отношения IS GREATER THAN OR EQUAL TO (>=) эквивалентен знаку операции отношения IS NOT LESS THAN (НЕ МЕНЬШЕ). Знак операции отношения IS LESS THAN OR EQUAL TO (МЕНЬШЕ ИЛИ РАВНО) эквивалентен знаку операции отношения IS NOT GREATER THAN (НЕ БОЛЬШЕ).

(54) Условие класса (1 ЯДР). Имя-класса связано с набором литер, определенным пользователем во фразе CLASS (КЛАСС) в параграфе SPECIAL-NAMES (СПЕЦИАЛЬНЫЕ-ИМЕНА).

(55) Вариант DAY-OF-WEEK (ДЕНЬ-НЕДЕЛИ) оператора ACCEPT (ПРИНЯТЬ) (2 ЯДР). Вариант DAY-OF-WEEK (ДЕНЬ-НЕДЕЛИ) оператора ACCEPT (ПРИНЯТЬ) обеспечивает доступ к целому, представляющему день недели; например, 1 представляет понедельник, 2 — вторник, и 7 — воскресенье.

(56) Оператор ADD (СЛОЖИТЬ) (1 ЯДР). Слово TO (С) является необязательным словом в формате:

ADD идентификатор/литерал TO идентификатор-литерал GIVING идентификатор

СЛОЖИТЬ идентификатор/литерал С идентификатор-литерал ПОЛУЧАЯ идентификатор.

(57) Вариант NOT ON SIZE ERROR (БЕЗ ПЕРЕПОЛНЕНИЯ) оператора ADD (СЛОЖИТЬ) (1 ЯДР). Вариант NOT ON SIZE ERROR (БЕЗ ПЕРЕПОЛНЕНИЯ) предоставляет программисту возможность указывать процедуры, которые должны быть выполнены, когда не выполняется условие переполнения для оператора ADD (СЛОЖИТЬ).

(58) Оператор CALL (ВЫЗВАТЬ) (2 МПС). Вариант BY CONTENT (ЗНАЧЕНИЕ) указывает, что вызываемая программа не может изменять значение параметра фразы USING (ИСПОЛЬЗУЯ) в операторе CALL (ВЫЗВАТЬ), но вызываемая программа может изменять значение соответствующего данного в заголовке раздела процедур программы. Вариант BY REFERENCE (ССЫЛКУ) означает, что параметр фразы USING (ИСПОЛЬЗУЯ) оператора CALL (ВЫЗВАТЬ) должен рассматриваться так же, как и параметр в ГОСТ 22558.

(59) Оператор CALL (ВЫЗВАТЬ) (1 МПС). Параметры, передаваемые в операторе CALL (ВЫЗВАТЬ), могут быть данными, имеющими уровни, отлич-

GOST EXPERT ГОСТ 22558-89,

ные от 01 или 77. Параметры, передаваемые в операторе CALL (ВЫЗВАТЬ), могут быть индексированы или указываться модификацией ссылки.

(60) Варианты ON EXCEPTION (ПРИ ОШИБКЕ) и NOT ON EXCEPTION (БЕЗ ОШИБКИ) оператора CALL (ВЫЗВАТЬ) (2 МПС). Вариант ON EXCEPTION (ПРИ ОШИБКЕ) оператора CALL (ВЫЗВАТЬ) эквивалентен варианту ON OVERFLOW (ПРИ ПЕРЕПОЛНЕНИИ) оператора CALL (ВЫЗВАТЬ). Вариант NOT ON EXCEPTION (БЕЗ ОШИБКИ) обеспечивает программисту возможность указать процедуры, которые должны выполняться, когда программа, указанная в операторе CALL (ВЫЗВАТЬ), может быть доступна для выполнения.

(61) Вариант REEL/UNIT (КАТУШКУ/ТОМ) оператора CLOSE (ЗАКРЫТЬ) (1 ПОД, 1 ГОТ). Вариант REEL/UNIT (КАТУШКУ/ТОМ) оператора CLOSE (ЗАКРЫТЬ) может относиться к файлам на одной катушке (томе) и разрешена в данном стандарте для файла отчетов.

(62) Вариант FOR REMOVAL (С УДАЛЕНИЕМ) оператора CLOSE (ЗАКРЫТЬ) (2 ПОД, 1 ГОТ). Вариант FOR REMOVAL (С УДАЛЕНИЕМ) оператора CLOSE (ЗАКРЫТЬ) разрешается для последовательного файла на одной катушке/томе.

(63) Вариант NOT ON SIZE ERROR (БЕЗ ПЕРЕПОЛНЕНИЯ) оператора COMPUTE (ВЫЧИСЛИТЬ) (2 ЯДР). Вариант NOT ON SIZE ERROR (БЕЗ ПЕРЕПОЛНЕНИЯ) оператора COMPUTE (ВЫЧИСЛИТЬ) предоставляет программисту возможность указывать процедуры, которые должны быть выполнены, когда для оператора COMPUTE (ВЫЧИСЛИТЬ) не выполняется условие переполнения.

(64) Оператор CONTINUE (ПРОДОЛЖИТЬ) (1 ЯДР). Оператор CONTINUE (ПРОДОЛЖИТЬ) указывает, что нет в наличии выполнимого оператора, и приводит к неявной передаче управления следующему выполнимому оператору.

(65) Вариант NOT INVALID KEY (БЕЗ ОШИБКИ КЛЮЧА) оператора DELETE (УДАЛИТЬ) (1 ОТД, 1 ИПД). Вариант NOT INVALID KEY (БЕЗ ОШИБКИ КЛЮЧА) оператора DELETE (УДАЛИТЬ) предоставляет программисту возможность указать процедуры, которые должны быть выполнены, когда для оператора DELETE (УДАЛИТЬ) не выполняется условие ошибки ключа.

(66) Оператор DISPLAY (ВЫДАТЬ) (1 ЯДР). Для оператора DISPLAY (ВЫДАТЬ) разрешена стандартная константа ALL литерал (ВСЕ литерал).

В ГОСТ 22558 стандартная константа ALL литерал (ВСЕ литерал) не разрешалась в операторе DISPLAY (ВЫДАТЬ).

(67) Вариант NOT ON SIZE ERROR (БЕЗ ПЕРЕПОЛНЕНИЯ) оператора DIVIDE (РАЗДЕЛИТЬ) (1 ЯДР). Вариант NOT ON SIZE ERROR (БЕЗ ПЕРЕПОЛНЕНИЯ) оператора DIVIDE (РАЗДЕЛИТЬ) обеспечивает программисту возможность указывать процедуры, которые должны быть выполнены, когда для оператора DIVIDE (РАЗДЕЛИТЬ) не выполняется условие переполнения.

(68) Вариант WITH NO ADVANCING (БЕЗ ПРОДВИЖЕНИЯ) оператора DISPLAY (ВЫДАТЬ) (2 ЯДР). Вариант WITH NO ADVANCING (БЕЗ ПРОДВИЖЕНИЯ) оператора DISPLAY (ВЫДАТЬ) обеспечивает программисту возможность взаимодействия с устройством, обладающим вертикальным позиционированием.

(69) Оператор EVALUATE (ОЦЕНИТЬ) (2 ЯДР). Оператор EVALUATE (ОЦЕНИТЬ) описывает многоветвящуюся, многосвязную структуру, в которой оценивается несколько условий для определения последующих действий объектной программы.

(70) Оператор EXIT PROGRAM (ВЫЙТИ ИЗ ПРОГРАММЫ) (1 МПС). Оператор EXIT PROGRAM (ВЫЙТИ ИЗ ПРОГРАММЫ) не обязательно должен быть единственным оператором в параграфе.

GOST EXPERT ГОСТ 22558-89,

(71) Оператор GO TO DEPENDING ON (ПЕРЕЙТИ Ь ЗАВИСИМОСТИ) ЯДР). Число имен-процедур, требуемых в операторе GO TO DEPENDING (ПЕРЕЙТИ В ЗАВИСИМОСТИ), снижено до единицы.

(72) Оператор IF (ЕСЛИ) (1 ЯДР). Необязательное слово THEN (ТО) добавлено в общий формат оператора IF (ЕСЛИ).

(73) Оператор INITIALIZE (ИНИЦИИРОВАТЬ) (2 ЯДР). Оператор INITIALIZE (ИНИЦИИРОВАТЬ) обеспечивает возможность устанавливать выбранные типы полей данных в предопределенные значения.

(74) Оператор INSPECT (ПРОСМОТРЕТЬ) (2 ЯДР). Многократные повторения варианта BEFORE/AFTER (ДО/ПОСЛЕ) позволяют начать операцию подсчета/замены после того, как начался просмотр начал данных, и/или закончить до конца просмотра концов данных.

(75) Оператор INSPECT (ПРОСМОТРЕТЬ) (2 ЯДР). Слова ALL/LEADING (ВСЕ/ВЕДУЩИЕ) могут быть размещены среди многих идентификаторов/литералов и возможны многократные повторения варианта REPLACING CHARACTERS (ЗАМЕНЯЯ ЛИТЕРЫ).

(76) Оператор INSPECT CONVERTING (ПРОСМОТРЕТЬ ПРЕВРАЩАЯ) (2 ЯДР). Вариант CONVERTING (ПРЕВРАЩАЯ) обеспечивает новую разновидность оператора INSPECT (ПРОСМОТРЕТЬ).

(77) Оператор MERGE (СЛИТЬ) (1 СРТ). В варианте GIVING (ПОЛУЧАЯ) оператора MERGE (СЛИТЬ) разрешается несколько имен-файлов. Файлы, указанные в операторе MERGE (СЛИТЬ), могут содержать записи переменной длины. Файл, указанный в варианте USING (ИСПОЛЬЗУЯ) или GIVING (ПОЛУЧАЯ) оператора MERGE (СЛИТЬ), может быть относительным или индексным файлом.

(78) Оператор MOVE (ПОМЕСТИТЬ) (2 ЯДР). Числовое редактируемое данное может быть помещено в числовое или числовое редактируемое данное; таким образом, имеет место доредактирование.

(79) Вариант NOT ON SIZE ERROR (БЕЗ ПЕРЕПОЛНЕНИЯ) оператора MULTIPLY (УМНОЖИТЬ) (1 ЯДР). Вариант NOT ON SIZE ERROR (БЕЗ ПЕРЕПОЛНЕНИЯ) обеспечивает программисту возможность указать процедуры, которые должны быть выполнены, когда для оператора MULTIPLY (УМНОЖИТЬ) не выполняется условие переполнения.

(80) Вариант EXTEND (ДОПОЛНЯЕМЫЙ) оператора OPEN (ОТКРЫТЬ) (2 ОГД, 2 ИПД). Вариант EXTEND (ДОПОЛНЯЕМЫЙ) оператора OPEN (ОТКРЫТЬ) может быть использован для относительного или индексного файла.

(81) Оператор PURGE (ОЧИСТИТЬ) (2 КОМ). Оператор PURGE (ОЧИСТИТЬ) побуждает систему управления сообщениями ликвидировать любое несвершенное сообщение, которое было передано одним или несколькими операторами SEND (ПОСЛАТЬ).

(82) Оператор PERFORM (ВЫПОЛНИТЬ) (1 ЯДР). Имя-процедуры может быть опущено, что равнозначно последовательному выполнению повелительных операторов, предшествующих фразе END-PERFORM (КОНЕЦ-ВЫПОЛНИТЬ), заканчивающей этот оператор PERFORM (ВЫПОЛНИТЬ).

(83) Оператор PERFORM (ВЫПОЛНИТЬ) (2 ЯДР). Вариант TEST AFTER (С ПРОВЕРКОЙ ПОСЛЕ) означает, что проверка условия должна проводиться после выполнения указанного набора операторов. Вариант TEST BEFORE (С ПРОВЕРКОЙ ДО) означает, что проверка условий должна проводиться до выполнения указанного набора операторов.

(84) Оператор PERFORM (ВЫПОЛНИТЬ) (2 ЯДР). В варианте VARYING (МЕНЯЯ) оператора PERFORM (ВЫПОЛНИТЬ) должно быть разрешено по крайней мере шесть фраз AFTER (ЗАТЕМ). В предыдущем стандарте разрешалось максимум две фразы AFTER (ЗАТЕМ).

GOST EXPERT ГОСТ 22558-89,

(85) Оператор READ (ЧИТАТЬ) (2 ПОД, 2 ОТД, 2 ИПД). Если оператор READ (ЧИТАТЬ) используется с вариантом INTO (В), разрешаются записи переменной длины. Вариант NEXT (СЛЕДУЮЩУЮ) разрешен в операторе READ (ЧИТАТЬ), ссылающемся на файл с последовательной организацией.

(86) Вариант NOT AT END (НЕ В КОНЦЕ) оператора READ (ЧИТАТЬ (1 ПОД, 1 ОТД, 1 ИПД). Вариант NOT AT END (НЕ В КОНЦЕ) оператора READ (ЧИТАТЬ) обеспечивает программисту возможность указывать процедуры, которые должны быть выполнены, если для оператора READ (ЧИТАТЬ) не выполняется условие «в конце».

(87) Вариант NOT INVALID KEY (БЕЗ ОШИБКИ КЛЮЧА) оператора READ (ЧИТАТЬ) (1 ОТД, 1 ИПД). Вариант NOT INVALID KEY (БЕЗ ОШИБКИ КЛЮЧА) оператора READ (ЧИТАТЬ) обеспечивает программисту возможность указывать процедуры, которые должны быть выполнены, когда условие ошибки ключа для оператора READ (ЧИТАТЬ) не выполняется.

(88) Вариант WITH DATA (ЕСТЬ ДАННЫЕ) оператора RECEIVE (ПОЛУЧИТЬ) (1 КОМ). Вариант WITH DATA (ЕСТЬ ДАННЫЕ) оператора RECEIVE (ПОЛУЧИТЬ) предоставляет программисту возможность указать процедуры, которые должны быть выполнены, когда система управления сообщениями делает данные доступными во время выполнения оператора RECEIVE (ПОЛУЧИТЬ).

(89) Оператор REPLACE (ЗАМЕНИТЬ) (2 ОИТ). Оператор REPLACE (ЗАМЕНИТЬ) приводит к замене каждого вхождения указанного текста в исходной программе на соответствующий текст, указанный в операторе REPLACE (ЗАМЕНИТЬ).

(90) Оператор RETURN (ВЕРНУТЬ) (1 СРТ). Записи переменной длины разрешены, если в операторе RETURN (ВЕРНУТЬ) имеется вариант INTO (В).

(91) Вариант NOT AT END (НЕ В КОНЦЕ) оператора RETURN (ВЕРНУТЬ) (1 СРТ). Вариант NOT AT END (НЕ В КОНЦЕ) оператора RETURN (ВЕРНУТЬ) предоставляет программисту возможность указывать процедуры, которые должны быть выполнены, если для оператора RETURN (ВЕРНУТЬ) не выполняется условие «в конце».

(92) Оператор REWRITE (ОБНОВИТЬ) (2 ОТД, 2 ИПД). Запись в относительном или индексном файле может заменяться записью другой длины.

(93) Вариант NOT INVALID KEY (БЕЗ ОШИБКИ КЛЮЧА) оператора REWRITE (ОБНОВИТЬ) (1 ОТД, 1 ИПД). Вариант NOT INVALID KEY (БЕЗ ОШИБКИ КЛЮЧА) оператора REWRITE (ОБНОВИТЬ) предоставляет программисту возможность указывать процедуры, которые должны быть выполнены, если условие ошибки ключа для оператора REWRITE (ОБНОВИТЬ) не выполняется.

(94) Оператор SEND (ПОСЛАТЬ) (2 КОМ). Вариант REPLACING LINE (ЗАМЕНЯЯ СТРОКУ) предоставляет новую возможность оператора SEND (ПОСЛАТЬ).

(95) Оператор SET (УСТАНОВИТЬ) (1 ЯДР). В ряде операндов, предшествующих слову TO (В) оператора SET (УСТАНОВИТЬ), теперь могут одновременно встречаться и имена-индексов и идентификаторы. Два новых варианта оператора SET (УСТАНОВИТЬ) позволяют изменять установку внешнего переключателя и значение условной переменной.

(96) Оператор SORT (СОРТИРОВАТЬ) (1 СРТ). В варианте GIVING (ПОЛУЧАЯ) оператора SORT (СОРТИРОВАТЬ) разрешается несколько имен-файлов. Файл, указанный в операторе SORT (СОРТИРОВАТЬ), может содержать записи переменной длины. Файл, указанный в варианте USING (ИСПОЛЬЗУЯ) или GIVING (ПОЛУЧАЯ) оператора SORT (СОРТИРОВАТЬ), может быть относительным или индексным файлом. Файлы, указанные в вариантах

GOST EXPERT ГОСТ 22558-89,

USING (ИСПОЛЬЗУЯ) и GIVING (ПОЛУЧАЯ), могут находиться на одной и той же физической катушке. Если указан вариант DUPLICATES (С ДУБЛИРОВАНИЕМ), записи, значения ключей которых идентичны, остаются после завершения сортировки в том же порядке, в котором они были при вводе в процесс сортировки.

(97) Операторы SORT (СОРТИРОВАТЬ) и MERGE (СЛИТЬ) (1 СРТ). Процедуры ввода и вывода операторов SORT (СОРТИРОВАТЬ) и MERGE (СЛИТЬ) могут содержать явные передачи управления в точки вне процедуры ввода или вывода. Остальная часть раздела процедур может содержать передачи управления в точки внутри процедуры ввода или вывода. Имя-параграфа может быть указано в варианте INPUT PROCEDURE (ПРОЦЕДУРА ВВОДА) или OUTPUT PROCEDURE (ПРОЦЕДУРА ВЫВОДА).

(98) Вариант NOT INVALID KEY (БЕЗ ОШИБКИ КЛЮЧА) оператора START (ПОДВЕСТИ) (1 ОТД, 1 ИПД). Вариант NOT INVALID KEY (БЕЗ ОШИБКИ КЛЮЧА) предоставляет программисту возможность указывать процедуры, подлежащие выполнению, если для оператора START (ПОДВЕСТИ) не выполняется условие ошибки ключа.

(99) Оператор STRING (СОБРАТЬ) (2 ЯДР). Идентификатор в варианте INTO (В) оператора STRING (СОБРАТЬ) может быть групповым данным.

(100) Вариант NOT ON OVERFLOW (БЕЗ ПЕРЕПОЛНЕНИЯ) оператора STRING (СОБРАТЬ) (2 ЯДР). Вариант NOT ON OVERFLOW (БЕЗ ПЕРЕПОЛНЕНИЯ) обеспечивает программисту возможность указывать процедуры, подлежащие выполнению, если для оператора STRING (СОБРАТЬ) выполняется условие переполнения.

(101) Вариант NOT ON SIZE ERROR (БЕЗ ПЕРЕПОЛНЕНИЯ) оператора SUBTRACT (ОТНЯТЬ) (1 ЯДР). Вариант NOT ON SIZE ERROR (БЕЗ ПЕРЕПОЛНЕНИЯ) обеспечивает программисту возможность указывать процедуры, подлежащие выполнению, если для оператора SUBTRACT (ОТНЯТЬ) не выполняется условие переполнения.

(102) Вариант NOT ON OVERFLOW (БЕЗ ПЕРЕПОЛНЕНИЯ) оператора UNSTRING (РАЗОБРАТЬ) (2 ЯДР). Вариант NOT ON OVERFLOW (БЕЗ ПЕРЕПОЛНЕНИЯ) обеспечивает программисту возможность указывать процедуры, подлежащие выполнению, если для оператора UNSTRING (РАЗОБРАТЬ) не выполняется условие переполнения.

(103) Оператор USE (ИСПОЛЬЗОВАТЬ) (1 ПОД, 1 ОТД, 1 ИПД). Декларативный оператор USE AFTER EXCEPTION/ERROR (ИСПОЛЬЗОВАТЬ ПОСЛЕ ОШИБКИ), указывающий имя файла, имеет приоритет над декларативным оператором, указывающим режим открытия файла.

(104) Оператор USE (ИСПОЛЬЗОВАТЬ) (2 МПС). Вариант GLOBAL (ГЛОБАЛЬНО) указывает, что соответствующие декларативные процедуры вызываются во время выполнения любой программы, содержащейся в программе, включающей в себя оператор USE (ИСПОЛЬЗОВАТЬ).

(105) Оператор USE BEFORE REPORTING (ИСПОЛЬЗОВАТЬ ДО ВЫДАЧИ) (2 МПС). Вариант GLOBAL (ГЛОБАЛЬНО) указывает, что соответствующие декларативные процедуры вызываются во время выполнения любой программы, содержащейся в программе, включающей в себя оператор USE BEFORE REPORTING (ИСПОЛЬЗОВАТЬ ДО ВЫДАЧИ).

(106) Вариант NOT END-OF-PAGE (НЕ В КОНЦЕ СТРАНИЦЫ) оператора WRITE (ПИСАТЬ). Вариант NOT END-OF-PAGE (НЕ В КОНЦЕ СТРАНИЦЫ) обеспечивает программисту возможность указывать процедуры, подлежащие выполнению, если для оператора WRITE (ПИСАТЬ) не выполняется условие конца страницы.

GOST EXPERT ГОСТ 22558-89,

(107) Вариант NOT INVALID KEY (БЕЗ ОШИБКИ КЛЮЧА) оператора WRITE (ПИСАТЬ) (1 ОТД, 1 ИПД). Вариант NOT INVALID KEY (БЕЗ ОШИБКИ КЛЮЧА) обеспечивает программисту возможность указывать процедуры, подлежащие выполнению, если для оператора WRITE (ПИСАТЬ) не выполняется условие ошибки ключа.

2.2. **Существенные изменения, потенциально влияющие на имеющиеся программы**

Ниже приводится список изменений, которые могли бы воздействовать на существующие программы, например, добавление правила для ранее не определенных ситуаций или изменение правила для существующего глагола. Для каждого элемента этого списка приводится обоснование введения такого изменения.

Основные изменения вызваны стремлением повысить переносимость программ и облегчить написание новых программ, а также дать уточнение неясных или двусмысленных правил. Добавления новых средств также преследуют цель снизить стоимость разработки программ. Ожидается, что затраты на изменение существующих программ должны окупиться экономией при разработке и сопровождении программ. В этой части содержится список изменений, потенциально влияющих на имеющиеся программы. В тех случаях, где в ГОСТ 22558 имелись неясности, сделаны уточнения в соответствии с фактическим промышленным стандартом, если таковой имелся. В любом случае уточняющие разъяснения не приводят к несовместимости стандартов; они могут только привести к возможности несовместимости между некоторой отдельной реализацией и настоящим стандартом. Обоснования, включенные в следующий список, относятся прежде всего к воздействию изменений на Кобол-программы, соответствующие правилам ГОСТ 22558. Остается неясным влияние изменений на программы, в которых:

(1) нарушены правила ГОСТ 22558 или

(2) используются средства, для которых правила ГОСТ 22558 были определены недостаточно четко, и поэтому зависят от определяемых реализацией расширений или интерпретации правил.

(1) Длина константы ALL литерал (ВСЕ литерал) (2 ЯДР). Когда стандартная константа ALL литерал (ВСЕ литерал) не связана с другим данным, длиной строки является длина литерала.

Обоснование

Правила в стандарте Кобола для размера стандартной константы ALL литерал (ВСЕ литерал) различны в зависимости от того, где использована стандартная константа в программе. Если стандартная константа ALL литерал (ВСЕ литерал) используется в параграфе SPECIAL-NAMES (СПЕЦИАЛЬНЫЕ-ИМЕНА), ее длина равна единице, а ее значением является самая левая литера литерала. Рассмотрим следующий пример.

```
IDENTIFICATION DIVISION.
PROGRAM-ID. EXAMPLE.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
OBJECT-COMPUTER.
  PROGRAM COLLATING SEQUENCE IS COL-SEQ.
SPECIAL-NAMES.
  COL-SEQ IS ALL «0123456789»
DATA DIVISION.
01 FIELD1 PIC X(80).
PROCEDURE DIVISION.
START-PROGRAM.
  IF FIELD-1=ALL «ABCDEF»
  DISPLAY «TEXT IS TRUE».
РАЗДЕЛ ИДЕНТИФИКАЦИИ.
ПРОГРАММА. ПРИМЕР.
```


GOST EXPERT ГОСТ 22558-89,

```
РАЗДЕЛ ОБОРУДОВАНИЯ.
СЕКЦИЯ КОНФИГУРАЦИИ.
ОБЪЕКТНАЯ-МАШИНА
    ПРОГРАММНЫЙ АЛФАВИТ ПР-АЛФ.
СПЕЦИАЛЬНЫЕ-ИМЕНА.
    ПР-АЛФ ВСЕ «0123456789».
РАЗДЕЛ ДАННЫХ.
01  ПОЛЕ1  ПИХ(80).
РАЗДЕЛ ПРОЦЕДУР.
НАЧАЛО-ПРОГРАММЫ.
    ЕСЛИ ПОЛЕ1 = ВСЕ «АБВГДЕ»
      ВЫДАТЬ «ТЕКСТ ВЕРЕН».
```

В приведенном выше примере, когда стандартная константа ALL литерал (ВСЕ литерал) используется во фразе имя-алфавита параграфа SPECIAL-NAMES (СПЕЦИАЛЬНЫЕ-ИМЕНА), используется только первая литера литерала, независимо от количества литер в литерале. Во втором случае IF FIELD1 = «ABCDEF» (ЕСЛИ ПОЛЕ1 = ВСЕ «АБВГДЕ») размером литерала считаются все литеры, входящие в литерал.

Эта противоречивость в правиле для размера константы ALL литерал (ВСЕ литерал) приводит к неясности поведения программы. По новым правилам исключаются противоречия между спецификациями программы и ее поведением, в частности, указывается, что в случае фразы имя-алфавита со стандартной константой ALL литерал (ВСЕ литерал) длина строки равняется длине литерала.

(2) Фраза имя-алфавита (1 ЯДР). Ключевое слово ALPHABET (АЛФАВИТ) должно предшествовать имени-алфавита во фразе имени-алфавита параграфа SPECIAL-NAMES (СПЕЦИАЛЬНЫЕ ИМЕНА).

Обоснование

Имена реализации являются системными именами; имена-алфавитов и мнемонические-имена являются словами, определенными пользователем. В настоящем стандарте системные имена и слова, определенные пользователем, образуют пересекающиеся множества и поэтому могут совпадать. Допустима нижеследующая фраза:

```
SPECIAL-NAMES. WORD-1 IS WORD-2.
СПЕЦИАЛЬНЫЕ-ИМЕНА. СЛОВО-1 ЕСТЬ СЛОВО-2.
```

Если WORD-1 (СЛОВО-1) является именем-реализации и именем-алфавита, а WORD-2 (СЛОВО-2) было и мнемоническим-именем и именем-реализации, то невозможно установить, что предполагается в вышеприведенной фразе — фраза имени-реализации или фраза имени-алфавита. Введение ключевого слова ALPHABET (АЛФАВИТ) во фразе имени-алфавита разрешает эту неоднозначность.

Этой проблемы не было в ГОСТ 22558, поскольку системные имена и слова, определенные пользователем, входили в непересекающиеся множества, поэтому вышеприведенная конструкция не допускалась.

Разрешение совпадения системных-имен и слов, определенных пользователем, способствует облегчению переноса программ с одной реализации на другую: системные имена больше не нуждаются в замене. Для модификации имеющихся программ перед фразой имени-алфавита необходимо вставить ключевое слово ALPHABET (АЛФАВИТ).

(3) Программный алфавит (1 ИПД). Программный алфавит (основная последовательность), используемый для доступа к индексному файлу, является алфавитом, связанным с внутренним набором литер, действовавшим для файла во время создания файла.

Обоснование

В ГОСТ 22558 правила не устанавливали, какой именно алфавит используется для извлечения и занесения записей при доступе к индексному файлу. Были возможны две различные интерпретации

GOST EXPERT ГОСТ 22558-89,

а) Внутренний алфавит.

б) Алфавит, определенный фразой PROGRAM COLLATING SEQUENCE (ПРОГРАММНЫЙ АЛФАВИТ).

Новое правило стандарта Кобола явно указывает, что для извлечения и занесения записей при доступе к индексному файлу будет использоваться внутренний алфавит. Большинство из реализаций используют внутренний алфавит для извлечения и занесения записей при доступе к индексному файлу.

(4) Фраза CURRENCY SIGN (ВАЛЮТНЫЙ ЗНАК) (1 ЯДР). Литерал, указанный в фразе CURRENCY SIGN (ВАЛЮТНЫЙ ЗНАК), не может быть стандартной константой.

Обоснование

В ГОСТ 22558 допускалось использование стандартной константы во фразе CURRENCY SIGN (ВАЛЮТНЫЙ ЗНАК), но не было правил, определяющих смысл использования в этом контексте литералов HIGN-VALUE (НАИБОЛЬШЕЕ-ЗНАЧЕНИЕ), LOW-VALUE (НАИМЕНЬШЕЕ-ЗНАЧЕНИЕ) или ALL литерал (ВСЕ литерал).

Можно было бы добавить правила для разъяснения значения различных случаев, но польза от этого кажется мнимой. Таким образом, использование стандартной константы во фразе CURRENCY SIGN (ВАЛЮТНЫЙ ЗНАК) было запрещено. Предполагается, что эти изменения коснутся некоторых имеющихся программ.

(5) Фраза RELATIVE KEY (ОТНОСИТЕЛЬНЫЙ КЛЮЧ) (1 ОТД). Данное относительного ключа, указанное во фразе RELATIVE KEY (ОТНОСИТЕЛЬНЫЙ КЛЮЧ), не должно содержать символ шаблона P (M).

Обоснование

В ГОСТ 22558 допускается, чтобы относительный ключ содержал в строке литер шаблона символ P (M). Если бы относительный ключ был бы так описан, не все записи в файле были бы доступны программе. Например, данное с шаблоном 9P (9M) может иметь только значения 00, 10, 20, 30, 40, 50, 60, 70, 80 и 90. Это значит, что доступны только записи с этими номерами. Использование такого описания ключа возможно является ошибкой и может диагностироваться как таковое в соответствии с настоящим стандартом.

(6) Фраза LINAGE (ВЕРСТКА) (2 ПОД). Файлы, для которых указана фраза LINAGE (ВЕРСТКА), не могут быть открыты в режиме дополнения.

Обоснование

Поведение файла, имеющего соответствующую фразу LINAGE (ВЕРСТКА) и открытого в режиме дополнения, определено недостаточно четко в ГОСТ 22558. Например, указано, что значение LINAGE-COUNTER (СЧЕТЧИК-ВЕРСТКИ) при выполнении оператора OPEN (ОТКРЫТЬ) устанавливается в единицу. Кроме того, в ГОСТ 22558 не определяются значения для файла, имеющего соответствующую фразу LINAGE (ВЕРСТКА) и открываемого в режиме дополнения.

Возможность режима дополнения при открытии файла, имеющего связанную с ним фразу LINAGE (ВЕРСТКА), определяется техникой, используемой для реализации таких файлов. Некоторые разработчики реализовали режим дополнения для открытия файла, имеющего соответствующую фразу LINAGE (ВЕРСТКА). Другие разработчики запрещают режим дополнения при открытии файла, имеющего соответствующую фразу LINAGE (ВЕРСТКА).

В настоящем стандарте определено, что вариант EXTEND (ДОПОЛНЯЕМЫЙ) может использоваться только для файлов, для которых фраза LINAGE (ВЕРСТКА) не указана. Предполагается, что пользователи, у которых реализовано OPEN EXTEND (ОТКРЫТЬ ДОПОЛНЯЕМЫЙ) для файлов, имеющих в описании фразу LINAGE (ВЕРСТКА), будут продолжать поддерживать эту функцию. Независимо от того, будет ли еще какая-либо реализация делать такой вариант, введенные изменения отразятся на малом количестве имеющихся программ.

GOST EXPERT ГОСТ 22558-89,

(7) Вариант FOOTING (КОНЦОВКА) (2 ПОД). Если вариант FOOTING (КОНЦОВКА) не указан, не выполняется никакое условие конца страницы, независимо от условия переполнения страницы.

Обоснование

В ГОСТ 22558 спецификации поля концовки во фразе LINAGE (ВЕРСТКА) и операторе WRITE (ПИСАТЬ) противоречивы. Некоторые имеющиеся реализации обеспечивают для поля концовки одну строку, в то время как другие реализации не обеспечивают поле концовки, если фраза FOOTING (КОНЦОВКА) не указана. Это несоответствие нельзя разрешить, не затрагивая некоторые имеющиеся реализации. Новое правило базируется на принципе:

если поле концовки не указано, значит оно нежелательно.

Таким образом, если фраза FOOTING (КОНЦОВКА) не указана во фразе LINAGE (ВЕРСТКА), то не существует поля концовки и не выполняется условие конца страницы. Это изменение повлияет только на те программы, в которых для файла не указан вариант FOOTING (КОНЦОВКА) во фразе LINAGE и которые используют оператор WRITE (ПИСАТЬ) с вариантом END-OF-PAGE (В КОНЦЕ СТРАНИЦЫ) для этого файла и используют имеющуюся реализацию, обеспечивающую поле концовки.

(8) Фраза OCCURS (ПОВТОРЯЕТСЯ) (2 ЯДР). Если принимающее данное является данным переменной длины и содержит объект варианта DEPENDING ON (В ЗАВИСИМОСТИ ОТ), будет использоваться максимальная длина данного.

Обоснование

В ГОСТ 22558 длина вычислялась по значению данного фразы DEPENDING ON (В ЗАВИСИМОСТИ ОТ) до выполнения оператора. Использование правил ГОСТ 22558 с оператором MOVE (ПОМЕСТИТЬ) или READ INTO (ЧИТАТЬ В) могло привести в результате к потере данного, если значение данного во фразе DEPENDING ON (В ЗАВИСИМОСТИ ОТ) не было установлено для указания длины посылаемого данного до выполнения оператора MOVE (ПОМЕСТИТЬ).

```
FD INPUT-FILE.
01 A.
  02 A-TABLE.
    03 A-ODO PIC 99.
    03 A-ITEM OCCURS 1 TO 10 TIMES DEPENDING ON A-ODO.
WORKING-STORAGE SECTION.
01 B.
  02 B-TABLE.
    03 B-ODO PIC 99.
    03 B-ITEM OCCURS 1 TO 10 TIMES DEPENDING ON B-ODO.
ОФ ВХОДНОЙ-ФАЙЛ.
01 А.
  02 А-ТАБЛИЦА.
    03 А-ОДО Ш 99.
    03 А-ДАННОЕ ПОВТОРЯЕТСЯ 1 ДО 10 РАЗ В ЗАВИСИМОСТИ ОТ
    А-ОДО.
СЕКЦИЯ РАБОЧЕЙ-ПАМЯТИ.
01 Б.
  02 Б-ТАБЛИЦА.
    03 Б-ОДО Ш 99.
    03 Б-ДАННОЕ ПОВТОРЯЕТСЯ 1 ДО 10 РАЗ В ЗАВИСИМОСТИ ОТ
    Б-ОДО.
```

Предположим, что в приведенном фрагменте программы A-ODO (А-ОДО) установлено в 10 и B-ODO (Б-ОДО) установлено в 5. Согласно стандарту Кобола для того, чтобы поместить все повторения A-ITEM (А-ДАННОЕ) в B-TABLE

GOST EXPERT ГОСТ 22558-89,

(Б-ТАБЛИЦА), нужно было бы сначала поместить A-ODO (А-ОДО) в B-ODO (Б-ОДО). Таким образом, следующие последовательности операторов Кобола эквивалентны:

По ГОСТ 22558	По настоящему стандарту Кобола
MOVE A-ODO TO B-ODO. MOVE A TO B. READ INPUT-FILE. MOVE A-ODO TO B-ODO. MOVE A TO B.	MOVE A TO B. READ INPUT-FILE INTO B.
ПОМЕСТИТЬ А-ОДО В Б-ОДО. ПОМЕСТИТЬ А В Б. ЧИТАТЬ ВХОДНОЙ-ФАЙЛ ПОМЕСТИТЬ А-ОДО В Б-ОДО. ПОМЕСТИТЬ А В Б.	ПОМЕСТИТЬ А В Б. ЧИТАТЬ ВХОДНОЙ-ФАЙЛ В Б.

Некоторые реализации позволяют, чтобы за таблицами переменной длины в записи следовали другие данные. В следующем примере A-TRAILER (А-ОСТАТОК) и B-TRAILER (Б-ОСТАТОК) в некоторых реализациях динамически размещаются во время выполнения программы в соответствии со значениями A-ODO (А-ОДО) и B-ODO (Б-ОДО).

```
FD INPUT-FILE.
01 A.
  02 A-TABLE.
    03 A-ODO PIC 99.
    03 A-ITEM OCCURS 1 TO 10 TIMES DEPENDING ON A-ODO.
  02 A-TRAILER PIC XX.
WORKING-STORAGE SECTION.
01 B.
  02 B-TABLE.
    03 B-ODO PIC 99.
    03 B-ITEM OCCURS 1 TO 10 TIMES DEPENDING ON B-ODO.
  02 B-TRAILER PIC XX.
ОФ ВХОДНОЙ-ФАЙЛ.
01 А.
  02 А-ТАБЛИЦА.
    03 А-ОДО Ш 99.
    03 А-ДАННОЕ ПОВТОРЯЕТСЯ 1 ДО 10 РАЗ В ЗАВИСИМОСТИ ОТ
    А-ОДО.
  02 А-ОСТАТОК Ш ХХ.
СЕКЦИЯ РАБОЧЕЙ-ПАМЯТИ.
01 Б.
  02 Б-ТАБЛИЦА.
    03 Б-ОДО Ш 99.
    03 Б-ДАННОЕ ПОВТОРЯЕТСЯ 1 ДО 10 РАЗ В ЗАВИСИМОСТИ ОТ
    Б-ОДО.
  02 Б-ОСТАТОК Ш ХХ.
```

Если значение A-ODO (А-ОДО) равно 10 и значение B-ODO (Б-ОДО) равно 5, то, согласно ГОСТ 22558, поместить A-TABLE (А-ТАБЛИЦА) в B-TABLE (Б-ТАБЛИЦА) значило бы поместить только пять вхождений A-ITEM (А-ДАННОЕ), а B-TRAILER (Б-ОСТАТОК) остался бы без изменения. По правилам настоящего стандарта вхождения A-ITEM (А-ДАННОЕ) от 6 до 10 было бы тоже помещено, и B-TRAILER (Б-ОСТАТОК) было бы перекрыто.

Если значение A-ODO (А-ОДО) равно 5 и значение B-ODO (Б-ОДО) равно 5, то согласно ГОСТ 22558 поместить A-TABLE (А-ТАБЛИЦА) в B-TABLE (Б-ТАБЛИЦА) значило бы поместить только пять вхождений A-ITEM (А-ДАН-

GOST EXPERT ГОСТ 22558-89,

НОЕ), а B-TRAILER (Б-ОСТАТОК) осталось бы без изменения. По правилам настоящего стандарта вхождения B-ITEM (Б-ДАННОЕ) от 6 до 10 будут заполнены пробелами, и B-TRAILER (Б-ОСТАТОК) изменится.

На программы, соответствующие ГОСТ 22558, это изменение правила пересылки не повлияет.

Для изменения имеющихся программ, на которые влияют предложенные изменения, нужно перестроить соответствующие записи данных так, чтобы в записи не было данных, следующих за данными переменной длины.

(9) Символ P (M) шаблона (1 ЯДР). При ссылке на данное, описанное шаблоном, содержащим символ P(M), цифровые позиции, указанные символом P(M), означают нули в следующих операциях: (1) любой операции, требующей числовое посылаемое данное; (2) операторе MOVE (ПОМЕСТИТЬ), в котором посылаемый операнд является числовым и строка литер его шаблона содержит символ P(M); (3) операторе MOVE (ПОМЕСТИТЬ), где посылаемый операнд является числовым редактируемым и строка литер его шаблона содержит символ P(M) и принимающий операнд является числовым или числовым редактируемым; (4) операции сравнения, в которой оба операнда являются числовыми.

Обоснование

В ГОСТ 22558 считалось, что цифровые позиции, описанные символом P(M), содержат нули при использовании в операциях, включающих преобразование данных из одной формы представления в другую. При этом не указывалось, что происходит в операциях, не включающих в себя преобразование данных, или когда преобразование является необходимым. Настоящий стандарт определяет, когда цифровые позиции, описанные символом P(M), будут рассматриваться как содержащие нули.

Это разъяснение согласуется с текущими реализациями для общих применений литеры P(M) в шаблоне в цифровых контекстах и дает согласующиеся результаты для числовых и буквенно-цифровых пересылок, где посылаемое данное является числовым. Например, в результате пересылки данного, описание которого PICTURE 9P VALUE IS 10 (ШАБЛОН 9M ЗНАЧЕНИЕ 10), в данные с PICTURE 99 (ШАБЛОН 99) и PICTURE XX (ШАБЛОН XX) в принимающих полях будет 10 в обоих случаях. В более непонятных случаях, где числовые данные не требуются, как и в случае, когда данное сравнивается с буквенно-цифровым данным, будет использоваться значение литеры. Таким образом, данное, описание которого PICTURE 9P (ШАБЛОН 9M) и VALUE IS 10 (ЗНАЧЕНИЕ 10), при сравнении равно данному с PICTURE XX (ШАБЛОН XX) и VALUE IS «1» (ЗНАЧЕНИЕ «1») (за цифрой 1 следует пробел).

Предполагается, что эти изменения стандарта коснутся немногих программ.

(10) Заголовок раздела процедур (1 МПС). Данное, появляющееся во фразе USING (ИСПОЛЬЗУЯ) заголовка раздела процедур, не может иметь фразу REDEFINES (ПЕРЕОПРЕДЕЛЯЕТ) в своей статье описания данного.

Обоснование

Согласно ГОСТ 22558 во фразе USING (ИСПОЛЬЗУЯ) заголовка раздела процедур могло быть указано данное, описанное фразой REDEFINES (ПЕРЕОПРЕДЕЛЯЕТ). Таким образом, следующий пример был верен:

```
LINKAGE SECTION.
01 A PIC X(10).
01 B REDEFINES A PIC 9(10).
PROCEDURE DIVISION USING A, B.
СЕКЦИЯ СВЯЗИ.
01 А Ш X(10).
01 Б ПЕРЕОПРЕДЕЛЯЕТ А Ш 9(10).
РАЗДЕЛ ПРОЦЕДУР ИСПОЛЬЗУЯ А, Б.
```

Если в вызывающей программе указаны два различных параметра, результаты не определены. Разрешение указывать во фразе USING (ИСПОЛЬЗУЯ) заголовка раздела процедур данное со статьей REDEFINES (ПЕРЕОПРЕДЕ-

GOST EXPERT ГОСТ 22558-89,

ЛЯЕТ) могло бы привести к программистским ошибкам, остающимся невыявленными, приводящим к неверным результатам, и не обеспечивает никаких дополнительных функций. В большинстве случаев программы, в заголовке раздела процедур которых во фразе USING (ИСПОЛЬЗУЯ) указаны переопределяющие данные, могут быть преобразованы подстановкой переопределяемых данных.

(11) Возведение в степень (2 ЯДР). В настоящем стандарте определены следующие специальные случаи возведения в степень:

а) если значение, меньшее или равное нулю, возводится в нулевую степень, возникает условие переполнения;

б) если возведение в степень даст положительное или отрицательное действительное число, возвращается положительное число;

в) если результатом вычисления является не действительное число, возникает условие переполнения.

Обоснование

Поскольку ГОСТ 22558 не устанавливал, что может произойти в этих специальных случаях возведения в степень, выбор обработки предоставлялся реализации. Это изменение приводит к устранению неопределенной ситуации и поможет обеспечению переносимости программ. Поскольку в двух из этих случаев вырабатывается условие ошибки и третий случай совпадает с большинством реализаций, эти изменения затронут немногие программы.

(12) Порядок выполнения условных выражений (2 ЯДР). Два или больше условий, связанных только знаком логической операции AND (И) или только знаком логической операции OR (ИЛИ) на одном иерархическом уровне, вычисляются в порядке слева направо и вычисление этого иерархического уровня заканчивается, как только определено значение истинности, независимо от того, вычислялись ли все составные части связанных условий на этом иерархическом уровне.

Обоснование

Поскольку согласно ГОСТ 22558 порядок вычисления условных выражений определяется реализацией, одни и те же программы, использующие одни и те же данные для ввода, приводят к получению определенных результатов на одних реализациях и неопределенных — на других реализациях. Определение порядка вычисления усиливает переносимость программ.

Это изменение позволит программам проверить, принадлежит ли индекс интервалу, непосредственно до использования его как индекса в операторе; например:

IF INDEX-A IS LESS THAN 5 AND TABLE-A (INDEX-A) IS EQUAL TO 25.
ЕСЛИ ИНДЕКС-А МЕНЬШЕ 5 И ТАБЛИЦА-А (ИНДЕКС-А) РАВНО 25.

Изменение может повлиять на выполнение декларатив для фразы ALL REFERENCES (ПРИ ВСЕХ ССЫЛКАХ) в некоторых компиляторах. Изменение не оказывает никакого другого эффекта на имеющиеся программы.

(13) Условие класса (1 ЯДР). Проверка на ALPHABETIC (БУКВЕННОЕ) дает значение истины для прописных букв, строчных букв и литеры пробела. Проверка ALPHABETIC-UPPER (ПРОПИСНЫЕ) дает значение истины для прописных букв и литеры пробела. Проверка ALPHABETIC-LOWER (СТРОЧНЫЕ) дает значение истины для строчных букв и литеры пробела.

Обоснование

В старых реализациях Кобола буквенные литеры, принятые в большинстве наборов литер, были только прописными. Поэтому в ГОСТ 22558 проверка ALPHABETIC (БУКВЕННОЕ) давала значение истины для прописных литер и литеры пробела. В настоящее время наборы литер включают и прописные и строчные литеры. Согласно изменению в технологии проверка ALPHABETIC (БУКВЕННОЕ) сейчас следует логическому значению термина и воспринимает все буквенные литеры — и прописные, и строчные.

Обеспечиваются две дополнительные проверки для подклассов буквенных литер. В частности, изменение ALPHABETIC (БУКВЕННОЕ) в новую проверку

GOST EXPERT ГОСТ 22558-89,

ALPHABETIC-UPPER (ПРОПИСНЫЕ) в соответствующих ГОСТ 22558 исходных программах даст возможность программам выполняться в соответствии с правилами ГОСТ 22558.

Некоторые реализации уже сделали эти изменения. Поэтому это изменение касается некоторых программ, использующих проверку класса ALPHABETIC (БУКВЕННОЕ) в двух случаях: (1) на реализациях, допускающих только прописные буквы и пробел, или (2) там, где исходные программы не могут разрешать восприятие строчных букв. Многие исходные программы используют проверку класса на ALPHABETIC (БУКВЕННОЕ), поэтому изменение из ALPHABETIC (БУКВЕННОЕ) в ALPHABETIC-UPPER (ПРОПИСНЫЕ) должно быть выполнено программой автоматического преобразования кода.

(14) Оператор CANCEL (ОСВОБОДИТЬ) (2 МПС).

Оператор CANCEL (ОСВОБОДИТЬ) закрывает все открытые файлы.

Обоснование

В ГОСТ 22558 не определено состояние файлов, остающихся в режиме открытия, когда освобождается программа. Изменение в настоящем стандарте вырабатывает предсказуемый результат при выполнении этого оператора. Это повлияет только на те программы, которые освобождены и предполагают, что файлы освобожденной программы должны остаться открытыми после выполнения оператора CANCEL (ОСВОБОДИТЬ).

(15) Оператор CLOSE (ЗАКРЫТЬ) (2 ПОД). Вариант NO REWIND (БЕЗ ПЕРЕМОТКИ) не может быть указан в операторе CLOSE (ЗАКРЫТЬ), имеющем фразу REEL/UNIT (КАТУШКУ/ТОМ).

Обоснование

В ГОСТ 22558 правила для вариантов NO REWIND (БЕЗ ПЕРЕМОТКИ) и REEL/UNIT (КАТУШКУ/ТОМ) иногда были противоречивы. Противоречие заключалось в том, что правила для варианта NO REWIND (БЕЗ ПЕРЕМОТКИ) указывают, что катушка/том остаются в текущем положении, в то время как правила варианта REEL/UNIT (КАТУШКУ/ТОМ) указывают, что должна иметь место перемотка катушки/тома.

Изменение в настоящем стандарте повлияет на очень немногие программы, поскольку оператор CLOSE (ЗАКРЫТЬ) с обоими вариантами NO REWIND (БЕЗ ПЕРЕМОТКИ) и REEL/UNIT (КАТУШКУ/ТОМ) не мог быть правильно обработан.

(16) Оператор COPY (КОПИРОВАТЬ) (1 ОИТ). Если слово COPY (КОПИРОВАТЬ) появляется в статье-комментарии или месте, где может появиться статья-комментарий, оно рассматривается как часть статьи-комментария.

Обоснование

В стандарте Кобола появление слова COPY (КОПИРОВАТЬ) в статье-комментарии являлось неопределенной ситуацией. Определение этой ситуации в настоящем стандарте улучшит программную переносимость.

(17) Оператор COPY (КОПИРОВАТЬ) (1 ОИТ). Если в параграфе SOURCE-COMPUTER (ИСХОДНАЯ-МАШИНА) не указана фраза WITH DEBUGGING MODE (В РЕЖИМЕ ОТЛАДКИ), после обработки всех операторов COPY (КОПИРОВАТЬ) отладочная строка будет рассматриваться как имеющая все характеристики строки-комментария.

Обоснование

В ГОСТ 22558 не рассматривалась ситуация оператора COPY (КОПИРОВАТЬ) или части оператора COPY (КОПИРОВАТЬ), появляющегося в строке отладки. Рассмотрим следующий оператор COPY (КОПИРОВАТЬ):

```
    COPY XYZ
D   REPLACING 1 BY 2.
    КОПИРОВАТЬ АБВ
Т   ЗАМЕНЯЯ 1 НА 2.
```


GOST EXPERT ГОСТ 22558-89,

Если программа компилируется без фразы WITH DEBUGGING MODE (В РЕЖИМЕ ОТЛАДКИ), ГОСТ 22558 не определяет, будет ли выполняться вариант REPLACING (ЗАМЕНЯЯ). По правилам настоящего стандарта вариант REPLACING (ЗАМЕНЯЯ) выполняется.

Имеет место несовместимость, если при реализации ГОСТ 22558 оговорено, что отладочная строка рассматривается как строка комментария. Если же при реализации предполагается, что отладочная строка не рассматривается как строка комментария, несоответствие не имеет места.

Изменение в настоящем стандарте определяет, как обрабатывать такую ситуацию, усиливая тем самым степень программной совместимости. Это изменение повлияет на очень небольшое количество имеющихся программ.

(18) Оператор COPY (КОПИРОВАТЬ) (2 ОИТ). Псевдотекст-1 не может состоять полностью из разделителя запятая или разделителя точки с запятой.

Обоснование

В ГОСТ 22558 допускалось, чтобы псевдотекст-1 в операторе COPY (КОПИРОВАТЬ) состоял полностью из разделителя запятая или разделителя точка с запятой, но не определял, при каких условиях происходит замена. Любая попытка определить семантику в этой ситуации привела бы к потенциальной несовместимости.

Поскольку нет явной пользы от замены одинойной запятой или точки с запятой, это средство удалено из настоящего стандарта.

(19) Оператор DISPLAY (ВЫДАТЬ) (1 ЯДР). После выдачи последнего операнда на устройство, устройство будет установлено в самую левую позицию следующей строки устройства.

Обоснование

В ГОСТ 22558 позиционирование устройства после последнего операнда не определено. Новое правило в настоящем стандарте необходимо для полного определения варианта NO ADVANCING (БЕЗ ПРОДВИЖЕНИЯ). Большинство реализаций уже функционируют соответственно новому правилу.

(20) Оператор DIVIDE (РАЗДЕЛИТЬ) (2 ЯДР).

Любые индексы для идентификатора-4 варианта REMAINDER (ОСТАТОК) вычисляются после запоминания результата операции DIVIDE (РАЗДЕЛИТЬ) в идентификаторе-3 варианта GIVING (ПОЛУЧАЯ).

Обоснование

В ГОСТ 22558 не определена точка, в которой определяется любой индекс в варианте REMAINDER (ОСТАТОК) во время обработки оператора DIVIDE (РАЗДЕЛИТЬ).

Это изменение может повлиять на имеющиеся программы, если:

(1) частное используется как индекс для остатка, и

(2) вычисление индекса в реализации Кобола еще не вычисляет индекс по методике, определенной настоящим стандартом. Например:

```
01 DD PIC 99 VALUE IS 50.
01 DR PIC 99 VALUE IS 2.
01 QU PIC 99.
01 REMAIN.
   02 RM  PIC 99 OCCURS 100 TIMES.
PROCEDURE DIVISION.
   DIVIDE DD BY DR GIVING QU REMAINDER RM (QU).
```

```
01 ДД Ш 99 ЗНАЧЕНИЕ 50.
01 ДР Ш 99 ЗНАЧЕНИЕ 2.
01 КУ Ш 99.
01 ОСТАТ.
   02  ОШ Ш 99 ПОВТОРЯЕТСЯ 100 РАЗ.
РАЗДЕЛ ПРОЦЕДУР.
   РАЗДЕЛИТЬ ДД НА ДР ПОЛУЧАЯ КУ ОСТАТОК ОШ (КУ).
```


GOST EXPERT ГОСТ 22558-89,

Это изменение повлияет только на некоторые программы, если таковые имеются.

(21) Оператор EXIT PROGRAM (ВЫЙТИ ИЗ ПРОГРАММЫ) (1 МПС).

Если в вызываемой программе нет следующего выполнимого оператора, выполняется неявный оператор EXIT PROGRAM (ВЫЙТИ ИЗ ПРОГРАММЫ).

Обоснование

В ГОСТ 22558 эта ситуация была не определена. Определение этой ситуации в настоящем стандарте обеспечивает большую переносимость программ. Это изменение повлияет только на программы, зависящие от некоторых других действий реализации, когда оператор EXIT PROGRAM (ВЫЙТИ ИЗ ПРОГРАММЫ) опущен.

(22) Оператор EXIT PROGRAM (ВЫЙТИ ИЗ ПРОГРАММЫ) (1 МПС).

Для оператора EXIT PROGRAM (ВЫЙТИ ИЗ ПРОГРАММЫ) появляется следующее новое правило: «... концы диапазонов всех операторов PERFORM (ВЫПОЛНИТЬ), выполняемых вызываемой программой, считаются достигнутыми». Эта ситуация в ГОСТ 22558 не определена.

Обоснование

В ГОСТ 22558 общее правило 3 оператора CALL (ВЫЗВАТЬ) гласит: «При всех последующих входах в вызываемую программу состояние программы остается неизменным после последнего выхода из нее. Состояние программы включает все ее поля данных, состояние и позиционирование всех файлов и все установки переключателей». Не ясно, рассматривается или нет активация оператора PERFORM (ВЫПОЛНИТЬ) как часть состояния программы. В настоящем стандарте эта неоднозначность разрешена добавлением правила о том, что состояние программы не изменяется за исключением того, что границы диапазонов всех операторов PERFORM (ВЫПОЛНИТЬ) считаются достигнутыми.

Потенциальная несовместимость остается, если при реализации ГОСТ 22558 было принято решение не считать достигнутыми концы диапазонов оператора PERFORM (ВЫПОЛНИТЬ). Если же при реализации было принято рассматривать концы диапазонов оператора PERFORM (ВЫПОЛНИТЬ) достигнутыми, то никакой несовместимости не возникнет.

Данное изменение определяет путь обработки такой ситуации, что повысит степень переносимости программ. Изменение повлияет на небольшое количество имеющихся программ.

(23) Оператор INSPECT (ПРОСМОТРЕТЬ) (2 ЯДР). Определяется порядок вычисления индексов в операторе INSPECT (ПРОСМОТРЕТЬ). Индексирование, связанное с любым индикатором, выполняется только один раз как первая операция при выполнении оператора INSPECT (ПРОСМОТРЕТЬ).

Обоснование

Порядок вычисления индексов в операторе INSPECT (ПРОСМОТРЕТЬ) в ГОСТ 22558 не определен. Несовместимость имеет место, если реализация ГОСТ 22558 выполняет вычисление индексов в операторе INSPECT (ПРОСМОТРЕТЬ) не первой операцией.

Изменение в настоящем стандарте определяет порядок вычисления индексов в операторе INSPECT (ПРОСМОТРЕТЬ). Таким образом, такой оператор как

INSPECT X TALLYING I FOR ALL A(I)

ПРОСМОТРЕТЬ X СЧИТАЯ В I ВСЕ A(I)

который был неясен по ГОСТ 22558, становится определенным в настоящем стандарте Кобола. Определение этой ситуации в стандарте увеличивает степень переносимости программ.

(24) Оператор MERGE (СЛИТЬ) (1 СРТ).

Два файла оператора MERGE (СЛИТЬ) не могут быть указаны во фразе SAME AREA (ОБЩАЯ ОБЛАСТЬ) или SAME SORT/MERGE AREA (ОБЩАЯ ОБЛАСТЬ СОРТИРОВКИ-СЛИЯНИЯ). Только те файлы оператора MERGE

GOST EXPERT ГОСТ 22558-89,

(СЛИТЬ) могут быть указаны в фразе SAME RECORD AREA (ОБЩАЯ ОБЛАСТЬ), которые связаны с вариантом GIVING (ПОЛУЧАЯ).

Обоснование

Это правило является разъяснением взаимодействия фразы SAME (ОБЩАЯ) и оператора MERGE (СЛИТЬ), которого в ГОСТ 22558 не было. Если это правило, даже не установленное, было нарушено в реализации ГОСТ 22558, оператор MERGE (СЛИТЬ), возможно, не выполнялся должным образом.

Рассмотрим следующее правило оператора MERGE (СЛИТЬ) в настоящем стандарте: «Никакие два имени-файла в операторе MERGE (СЛИТЬ) не могут быть указаны в одной и той же фразе SAME AREA (ОБЩАЯ ОБЛАСТЬ), SAME SORT AREA (ОБЩАЯ ОБЛАСТЬ СОРТИРОВКИ) или SAME SORT-MERGE AREA (ОБЩАЯ ОБЛАСТЬ СОРТИРОВКИ-СЛИЯНИЯ)». Соответственно фразе SAME AREA (ОБЩАЯ ОБЛАСТЬ) оператор MERGE (СЛИТЬ) может потребовать, чтобы оба файла были открыты одновременно, однако, фраза SAME AREA (ОБЩАЯ ОБЛАСТЬ) не разрешает, чтобы два файла, указанные в фразе SAME (ОБЩАЯ), были открыты одновременно. Соответственно фразе SAME SORT-MERGE AREA (ОБЩАЯ ОБЛАСТЬ СОРТИРОВКИ-СЛИЯНИЯ), оператор MERGE (СЛИТЬ) может потребовать область памяти, используемую для одного из файлов, но может потребоваться, чтобы тот файл был открыт; правила фразы SAME SORT-MERGE AREA (ОБЩАЯ ОБЛАСТЬ СОРТИРОВКИ-СЛИЯНИЯ) не разрешили бы открыть тот файл.

Соответственно правилу, гласящему: «Исключение составляют только имена файлов, относящиеся к фразе GIVING (ПОЛУЧАЯ)», стандартный алгоритм слияния требует доступность в одно и то же время одной записи из каждого сливаемого файла. Поскольку оператор MERGE (СЛИТЬ) определен в терминах ввода-вывода стандарта Кобола, сливаемые файлы не могли использовать одну и ту же область записи. Единственный способ, чтобы в ГОСТ 22558 оператор MERGE (СЛИТЬ) мог работать соответствующим образом — это игнорирование фразы SAME RECORD AREA (ОБЩАЯ ОБЛАСТЬ ЗАПИСИ).

Это новое правило добавляет синтаксические ограничения в случаях, вызывающих затруднения. Следовательно, такие ситуации, возможно, имеют место в немногих имеющихся программах.

(25) Оператор PERFORM (ВЫПОЛНИТЬ) (2 ЯДР). Определен порядок инициации (установки начального значения) идентификаторов в нескольких вариантах VARYING (МЕНЯЯ) оператора PERFORM (ВЫПОЛНИТЬ).

Обоснование

Порядок установки начального значения идентификаторов в нескольких вариантах VARYING (МЕНЯЯ) в ГОСТ 22558 не определен. В ГОСТ 22558 общее правило 6 г оператора PERFORM (ВЫПОЛНИТЬ) в частности гласит: «когда изменяются два идентификатора, идентификатор-2 и идентификатор-5 устанавливаются...». Правило настоящего стандарта гласит: «... идентификатор-2 затем идентификатор-5 устанавливаются ...», определяя таким образом порядок установки начального значения.

В случае, когда установка одного идентификатора определяет значение другого, и реализация обеспечивает установку идентификатора-5 первой, может возникнуть несовместимость. Например:

```
MOVE 2 TO X.
PERFORM PARA1 VARYING X FROM 1 BY 1 UNTIL X=3
  AFTER Y FROM X BY 1 UNTIL Y=3.

ПОМЕСТИТЬ 2 В X.
ВЫПОЛНИТЬ ПАРА1 МЕНЯЯ X ОТ 1 НА 1 ДО X=3
  ЗАТЕМ Y ОТ X НА 1 ДО Y=3.
```

Если сначала устанавливается Y, оно будет установлено в 2; если первым устанавливается X, Y будет установлено в 1. По правилам настоящего стандарта сначала устанавливается X, поэтому Y будет установлено в 1.

GOST EXPERT ГОСТ 22558-89,

Это изменение разрешает неоднозначность и поможет обеспечить переносимость программ. Вероятность несовместимости мала: реализация должна устанавливать первым идентификатор-5, что возможно, но маловероятно, и одна переменная варианта VARYING (МЕНЯЯ) должна зависеть от другой.

(26) Оператор PERFORM (ВЫПОЛНИТЬ) (2 ЯДР).

В варианте VARYING...AFTER (МЕНЯЯ...ЗАТЕМ) оператора PERFORM (ВЫПОЛНИТЬ) идентификатор-2 увеличивается до установки идентификатора-5. В ГОСТ 22558 идентификатор-5 устанавливался до увеличения идентификатора-2.

Обоснование

В ГОСТ 22558 общее правило н устанавливает, что при изменении двух переменных в промежуточном состоянии, когда внутреннее условие истинно, внутренняя переменная (идентификатор-5) устанавливается в ее текущее значение FROM (ОТ) до того, как увеличится внешняя переменная его текущим значением BY (НА).

По правилу настоящего стандарта идентификатор-2 увеличивается до установки идентификатора-5..

При этом изменении возникает несовместимость, если имеется зависимость между идентификатором-2 и идентификатором-5.

Рассмотрим следующий пример.

PERFORM PARA3 VARYING X FROM 1 BY 1 UNTIL X IS GREATER THAN 3
AFTER Y FROM X BY 1 UNTIL Y IS GREATER THAN 3.

ВЫПОЛНИТЬ ПАРАЗ МЕНЯЯ X ОТ 1 НА 1 ДО X БОЛЬШЕ 3
ЗАТЕМ Y ОТ X НА 1 ДО Y БОЛЬШЕ 3.

По стандарту Кобола PARA3 (ПАРАЗ) будет выполняться 8 раз со следующими значениями.

X: 1 1 1 2 2 2 3 3
Y: 1 2 3 1 2 3 2 3

По правилам настоящего стандарта PARA3 (ПАРАЗ) будет выполняться 6 раз со следующими значениями.

X: 1 1 1 2 2 3
Y: 1 2 3 2 3 3

Можно ожидать, что выше приведенный пример выполняется так же, как и следующий пример:

PERFORM PARA2 VARYING X FROM 1 BY 1 UNTIL X IS GREATER THAN 3. PARA2.
PERFORM PARA3 VARYING Y FROM X BY 1 UNTIL Y IS GREATER THAN 3.

ВЫПОЛНИТЬ ПАРА2 МЕНЯЯ X ОТ 1 НА 1 ДО X БОЛЬШЕ 3.
ПАРА2.
ВЫПОЛНИТЬ ПАРАЗ МЕНЯЯ Y ОТ X НА 1 ДО Y БОЛЬШЕ 3.

По ГОСТ 22558 PARA3 (ПАРАЗ) будет выполняться 8 раз, как показано выше. По правилам настоящего стандарта PARA3 (ПАРАЗ) будет выполняться 6 раз.

Это изменение повлияет на малое количество имеющихся программ. Ситуация, когда одна переменная в варианте VARYING (МЕНЯЯ) зависит от другой переменной, удобна для обработки полуматриц вдоль главной диагонали; правила в настоящем стандарте определяют эту функцию должным образом, в то время как правила ГОСТ 22558 не определяют ее.

(27) Оператор PERFORM (ВЫПОЛНИТЬ) (2 ЯДР).

Определен порядок вычисления индексов в операторе PERFORM VARYING (ВЫПОЛНИТЬ МЕНЯЯ). Эта ситуация не определена в ГОСТ 22558.

Обоснование

По правилам настоящего стандарта Кобола индексы в операторе PERFORM VARYING (ВЫПОЛНИТЬ МЕНЯЯ) вычисляются следующим образом:

GOST EXPERT ГОСТ 22558-89,

а) для идентификатора (идентификаторов) VARYING (МЕНЯЯ) индексирование выполняется каждый раз при установке или изменении идентификатора;

б) для идентификаторов FROM (ОТ) и BY (НА) индексирование выполняется каждый раз, когда идентификатор используется в операциях установки или приращения;

в) для любых идентификаторов, включенных в условие UNTIL (ДО), индексирование выполняется каждый раз при проверке условия.

ГОСТ 22558 не устанавливал, когда вычисляются индексы в цикле PERFORM (ВЫПОЛНИТЬ). Поэтому реализация была свободна в выборе, когда вычислять индексы. Изменение в настоящем стандарте приводит к несовместимости только в случаях, если программа:

а) использует индексируемые идентификаторы в операторе PERFORM VARYING (ВЫПОЛНИТЬ МЕНЯЯ);

б) изменяет значение (значения) индекса (индексов) в то время, когда оператор PERFORM (ВЫПОЛНИТЬ) активен, и

в) выполняется на реализации, которая осуществляет вычисление индексов, отличное от того, которое определяется в правилах настоящего стандарта.

Это изменение в стандарте Кобола является разрешением неоднозначности и поможет обеспечить переносимость программ. Это изменение повлияет на небольшое количество имеющихся программ.

(28) Оператор READ (ЧИТАТЬ) (1 ПОД, 1 ОТД, 1 ИПД). Вариант INTO (В) не может быть указан:

(а) если не все записи, связанные с файлом, и данное, указанное в варианте INTO (В), являются групповыми данными или элементарными буквенно-цифровыми данными, или (б) если не одно описание записи подчинено статье описания файла.

Обоснование

В ГОСТ 22558 не определена семантика перемещения записи в идентификатор, указанный в варианте INTO (В) оператора READ (ЧИТАТЬ). Для файла с несколькими элементарными записями нет утверждения относительно того, имеет ли место преобразование данных, либо выполняется групповая пересылка данных. Таким образом, в следующем примере:

```
FD FILEA...
01 RECA PIC S9(18).
01 RECB PIC 9(9)V9(9).
01 RECC PIC X(18).
WORKING-STORAGE SECTION.
01 A PIC S9(10)V9(8).
PROCEDURE DIVISION.
  READ FILEA INTO A.
ОФ ФАЙЛА...
01 ЗАПА Ш 9(18).
01 ЗАПБ Ш 9(9)Т9(9).
01 ЗАПВ Ш Х(18).
СЕКЦИЯ РАБОЧЕЙ-ПАМЯТИ.
01 А Ш 9(10)Т9(8).
РАЗДЕЛ ПРОЦЕДУР.
  ЧИТАТЬ ФАЙЛА В А.
```

перемещение записи в А не определено в ГОСТ 22558. Поэтому разные реализации могут выдавать разные результаты. Новые правила в настоящем стандарте исключают неоднозначность в ситуации, приведенной выше. Это изменение отразится только на программах, выполняющих оператор READ INTO (ЧИТАТЬ В) для файлов, имеющих в описании несколько элементарных записей, среди которых имеется по крайней мере одна числовая запись.

Это изменение повлияет только на некоторые имеющиеся программы.

(29) Оператор RECEIVE (ПОЛУЧИТЬ) (2 КОМ). Если размер сообщения больше области, на которую ссылаются, сообщение заполняет эту область слева

GOST EXPERT ГОСТ 22558-89,

направо, начиная с самой левой литеры сообщения. Для передачи остатка сообщения в ту же область должны быть выполнены дальнейшие операторы RECEIVE (ПОЛУЧИТЬ) со ссылкой на ту же очередь, подочередь и т. д.

Обоснование

По ГОСТ 22558, если получена часть сообщения и используется последующий оператор RECEIVE (ПОЛУЧИТЬ), ссылающийся на меньшую структуру определенной очереди, реализация определяет, передается или нет оставшаяся часть сообщения.

В настоящем стандарте разъяснено, что последующие операторы RECEIVE (ПОЛУЧИТЬ), ссылающиеся на полностью определенную структуру очереди, должны быть выполнены для получения остатка сообщения.

Это изменение повлияет только на некоторые программы, если таковые имеются.

(30) Оператор RETURN (ВЕРНУТЬ) (1 СРТ). Вариант INTO (В) не может быть указан, если:

(а) не все записи, связанные с файлом, и не все данные, указанные в варианте INTO (В), являются групповыми или элементарными буквенно-цифровыми данными, или

(б) не одно описание записи подчинено статье описания сортируемого-сливаемого файла.

Обоснование

В ГОСТ 22558 не определена семантика для перемещения записи в идентификатор, указанный в варианте INTO (В) оператора RETURN (ВЕРНУТЬ). Для файла с несколькими элементарными записями нет утверждения относительно того, имеет ли место преобразование данных или выполняется групповая пересылка данных.

```
SD FILEA . . . .
01 RECA PIC S9(18).
01 RECB PIC 9(9)V9(9).
01 RECC PIC X(18).
WORKING-STORAGE SECTION.
01 A PIC S9(10)V9(8).
PROCEDURE DIVISION.
  RETURN FILEA INTO A.
ОС ФАЙЛА . . . .
01 ЗАПА Ш 39(18).
01 ЗАПБ Ш 9(9)Т9(9).
01 ЗАПВ Ш Х(18).
СЕКЦИЯ РАБОЧЕЙ-ПАМЯТИ.
01 А Ш 39(10)Т9(8).
РАЗДЕЛ ПРОЦЕДУР.
  ВЕРНУТЬ ФАЙЛА В А.
```

перемещение записи в А не определено в ГОСТ 22558. Поэтому разные реализации могут выдавать разные результаты. Новые правила в настоящем стандарте Кобола устраняют неоднозначность приведенной выше ситуации. Это изменение повлияет только на программы, выполняющие оператор RETURN INTO (ВЕРНУТЬ В) для файла, содержащего несколько элементарных записей, включающих хотя бы одну числовую запись.

Это изменение повлияет только на некоторые программы, если таковые имеются.

(31) Оператор STOP RUN (ОСТАНОВИТЬ РАБОТУ) (1 ЯДР). Оператор STOP RUN (ОСТАНОВИТЬ РАБОТУ) закрывает все файлы.

Обоснование

В ГОСТ 22558 не определено состояние файлов, остающихся в режиме открытия при завершении работы. В некоторых случаях эта ситуация может привести к ошибкам.

GOST EXPERT ГОСТ 22558-89,

В настоящем стандарте Кобола оператор STOP RUN (ОСТАНОВИТЬ РАБОТУ) закрывает все открытые файлы. Многие реализации уже это делают и только некоторых, если таковые имеются, программ коснется это изменение.

(32) Оператор STOP RUN (ОСТАНОВИТЬ РАБОТУ) (1 ЯДР). Если единице исполнения доступны сообщения, оператор STOP RUN (ОСТАНОВИТЬ РАБОТУ) побуждает систему управления сообщениями удалить из очереди любое сообщение, частично полученное этой единицей исполнения.

Обоснование

В ГОСТ 22558 не определено, что происходит с частично полученным сообщением, если единица исполнения выполняет оператор STOP RUN (ОСТАНОВИТЬ РАБОТУ). Имеются три возможности, которые могли бы быть реализованы по стандарту Кобола:

а) система управления сообщениями делает частично полученные сообщения недоступными для любой последующей единицы исполнения посредством:

(1) игнорирования этих сообщений, или

(2) очистки очереди от них непосредственно или как части содержимого общей очереди. На программы, использующие эту реализацию, измененные спецификации не повлияют;

б) система управления сообщениями возвращает сообщение в целом, включая «полученную» часть входной очереди, для обработки некоторой последующей единицей исполнения. Вероятно, это удалось в предположении, что если программа выполняет оператор STOP RUN (ОСТАНОВИТЬ РАБОТУ) без окончания обработки входного сообщения, программа, возможно, потерпела неудачу (произошел сбой) и ее выполнение, возможно, будет возобновлено. На программы, использующие этот тип реализации, измененные спецификации повлияли бы;

в) система управления сообщениями оставляет во входных очередях фрагменты сообщений, которые были частично получены, и эти фрагменты становятся доступными для последующих единиц исполнения. Непохоже, чтобы какие-либо программы полагались на такую реализацию, так как вероятность ошибок в обработке была бы очень большой.

Уверены, что большинство реализаций предприняли первый путь, и поэтому похоже, что это изменение повлияет только на некоторые программы.

(33) Оператор STRING (СОБРАТЬ) (2 ЯДР). Определен порядок вычисления индексов в операторе STRING (СОБРАТЬ).

Обоснование

В ГОСТ 22558 порядок вычисления индексов не определен, поэтому определяется реализацией. В частности, не определен относительный порядок вычисления индексов и модификации указателя.

Рассмотрим следующий пример:

```
01 A PIC X(1000).
01 B PIC XXX.
01 CC.
   02 C1 PIC 9(4) OCCURS 100 TIMES.
PROCEDURE DIVISION.
   MOVE 1 TO PTR.
   STRING A DELIMITED BY SPACE INTO B POINTER C1(PTR).
01 А Ш X(1000).
01 Б Ш XXX
01 ПТР ПЕРЕОПРЕДЕЛЯЕТ Б Ш 999.
01 СС.
   02 С1 Ш 9(4) ПОВТОРЯЕТСЯ 100 РАЗ.
РАЗДЕЛ ПРОЦЕДУР.
   ПОМЕСТИТЬ 1 В ПТР.
   СОБРАТЬ А ОГРАНИЧИВАЯСЬ ПРОБЕЛ В Б УКАЗАТЕЛЬ С1 (ПТР).
```

В ГОСТ 22558 не определено, вычисляется ли C1 (PTR) (С1 (ПТР)) (а) один раз, или (б) до или после запоминания в B (Б) при каждой итерации. Но-

GOST EXPERT ГОСТ 22558-89,

вые правила в настоящем стандарте Кобола устанавливают, что C1 (PTR) (C1 (ПТР)) вычисляется один раз, непосредственно перед выполнением оператора STRING (СОБРАТЬ)

Это изменение влияет на программу, если идентификатор варианта INTO (В) оператора STRING (СОБРАТЬ) перекрывал индекс ограничителя или идентификатора в варианте WITH POINTER (УКАЗАТЕЛЬ). Такое программирование неудобно и это изменение повлияет на небольшое количество программ, если таковые имеются.

(34) Оператор UNSTRING (РАЗОБРАТЬ) (2 ЯДР).

В операторе UNSTRING (РАЗОБРАТЬ) любое индексирование, связанное с идентификатором DELIMITED BY (ОГРАНИЧИВАЯСЬ), идентификатором INTO (В) идентификатором DELIMITER IN (ОГРАНИЧИТЕЛЬ В) или идентификатором COUNT IN (СЧЕТ В), определяется один раз, непосредственно перед просмотром пересылаемых полей в поисках ограничителя.

Обоснование

Рассмотрим следующий пример:

```
01 A PIC X(30).
01 BB.
   02 PTR PIC 99.
01 CC.
   02 C1 PIC XX OCCURS 10 TIMES.
01 D PIC XX.
01 E PIC X(30).
PROCEDURE DIVISION.
   UNSTRING A  DELIMITED BY C1 (PTR) INTO BB, E WITH POINTER
       PTR.
```

```
01 А Ш X(30).
01 ВВ.
   02 ПТР Ш 99.
01 СС.
   02 С1 Ш XX ПОВТОРЯЕТСЯ 10 РАЗ.
01 Д Ш XX.
01 Е Ш X(30).
РАЗДЕЛ ПРОЦЕДУР.
   РАЗОБРАТЬ А ОГРАНИЧИВАЯСЬ С1 (ПТР) В ВВ, Е
       УКАЗАТЕЛЬ ПТР.
```

Согласно правилам ГОСТ 22558 ограничитель C1 (PTR) (C1 (ПТР)) будет опять рассматриваться перед перемещением второй строки в Е, в то время как по новым правилам в настоящем стандарте Кобола C1 (PTR) (C1 (ПТР)) рассматривается только один раз перед просмотром пересылаемых полей. Таким образом, ограничители никогда не меняются во время всего процесса разбора.

Хотя по ГОСТ 22558 любое индексирование, связанное с ограничителем, вычисляется непосредственно перед пересылкой данного в соответствующее данное, это приводит к ошибке, поскольку ограничитель должен быть определен до просмотра пересылаемого поля, и поэтому не может вычисляться непосредственно перед пересылкой. Таким образом, это изменение в установленных правилах позволяет вычисление ограничителей в соответствующее время так, как это делают некоторые имеющиеся реализации при обработке оператора UNSTRING (РАЗОБРАТЬ).

Для того чтобы эти изменения повлияли на программу, идентификатор в варианте INTO (В) оператора UNSTRING (РАЗОБРАТЬ) должен перекрывать индекс ограничителя. Такое программирование некорректно и это изменение отразится только на немногих программах, если таковые имеются.

(35) Оператор WRITE (ПИСАТЬ) (2 НОД).

В одном и том же операторе WRITE (ПИСАТЬ) не могут быть одновременно определены фразы ADVANCING PAGE (ДО/ПОСЛЕ ПРОДВИЖЕНИЯ СТРАНИЦЫ) и END-OF-PAGE (В КОНЦЕ СТРАНИЦЫ).

GOST EXPERT ГОСТ 22558-89,

Обоснование

В ГОСТ 22558 допускается указание обеих фраз в одном и том же операторе WRITE (ПИСАТЬ). Тем не менее нет правил, определяющих порядок их обработки. Следовательно, обработка определяется реализацией.

Обе фразы ADVANCING PAGE (ДО/ПОСЛЕ ПРОДВИЖЕНИЯ СТРАНИЦЫ) и END-OF-PAGE (В КОНЦЕ СТРАНИЦЫ) позволяют управлять вертикальным позиционированием печатаемой страницы. Продвижение страницы средствами фразы ADVANCING PAGE (ДО/ПОСЛЕ ПРОДВИЖЕНИЯ СТРАНИЦЫ) осуществляется в соответствии с техникой, определяемой реализацией. В то же время продвижение страницы средствами фразы END-OF-PAGE (В КОНЦЕ СТРАНИЦЫ) является техникой, определяемой пользователем. Поэтому было решено разделить эти разные техники.

Хотя допускалось одновременное использование этих вариантов соответственно ГОСТ 22558, этим пользовались мало реализаций. Поэтому на имеющиеся программы это изменение повлияет в минимальной степени.

(36) Указатель позиции файла (1 ПОД, 1 ОТД, 1 ИПД). Понятие указателя текущей записи в ГОСТ 22558 изменено на указатель позиции файла.

Обоснование

В настоящем стандарте Кобола правила, основанные на указателе позиции файла, усилены и облегчены для понимания. Кроме того, для сочетания обновления и операторов READ NEXT (ЧИТАТЬ СЛЕДУЮЩУЮ) указатель текущей записи в ГОСТ 22558 сложен и не всегда приводил к интуитивно ожидаемому результату. Правила указателя текущей записи также недостаточно определены в отдельных случаях, когда указываемая запись становится недоступной.

Это изменение в понятиях может повлиять на программы только в двух случаях, описанных в пунктах (37 и 38). Они могут встретиться только в очень необычной последовательности операций над файлами при динамическом доступе.

(37) Указатель позиции файла (2 ОТН, 2 ИПД). Для относительного или индексного файла при динамическом доступе выполнение оператора OPEN I-O (ОТКРЫТЬ ВХОДНОЙ-ВЫХОДНОЙ), за которым следуют один или более операторов WRITE (ПИСАТЬ) с последующим оператором READ NEXT (ЧИТАТЬ СЛЕДУЮЩУЮ), приведет к тому, что оператору READ (ЧИТАТЬ) будет доступна первая запись в файле во время выполнения оператора READ (ЧИТАТЬ).

Обоснование

В ГОСТ 22558 эта последовательность приводит к тому, что оператору READ (ЧИТАТЬ) доступна первая запись во время выполнения оператора OPEN (ОТКРЫТЬ). Если один из операторов WRITE (ПИСАТЬ) вставляет запись с ключом или относительным номером записи ниже, чем любые записи, имеющиеся в файле, оператору READ (ЧИТАТЬ) была бы доступна запись, отличная от первой.

Считается более логичным, чтобы при выполнении первого оператора READ NEXT (ЧИТАТЬ СЛЕДУЮЩУЮ) после оператора OPEN (ОТКРЫТЬ) доступной записью была бы первая запись в файле во время выполнения оператора READ (ЧИТАТЬ).

Семантика в настоящем стандарте ставит на один уровень ситуацию, возникающую после оператора OPEN (ОТКРЫТЬ), и ситуацию, возникающую после оператора READ (ЧИТАТЬ). В последнем случае, если оператор WRITE (ПИСАТЬ) вставляет запись с ключом таким, что она следует непосредственно за последней прочитанной записью, следующий оператор READ NEXT (ЧИТАТЬ СЛЕДУЮЩУЮ) получит вставленную запись.

Особенно затруднительна семантика в ГОСТ 22558, когда в дополнение к вставкам начальная первая запись удаляется между оператором OPEN (ОТКРЫТЬ) и оператором READ NEXT (ЧИТАТЬ СЛЕДУЮЩУЮ).

(38) Указатель позиции файла (2 ИПД). Если дополнительный ключ является ключом ссылки и дополнительный ключ изменяется оператором REWRITE

GOST EXPERT ГОСТ 22558-89,

(ОБНОВИТЬ) в значение между текущим значением и следующим значением в файле, следующий оператор READ NEXT (ЧИТАТЬ СЛЕДУЮЩУЮ) получит ту же запись.

Обоснование

В ГОСТ 22558 следующий оператор READ (ЧИТАТЬ) получил бы запись со следующим значением для того дополнительного ключа до оператора REWRITE (ОБНОВИТЬ).

Логически согласовано, что последующий оператор READ (ЧИТАТЬ) получает «ту же» запись, поскольку эта запись в тот момент является первой имеющейся записью в файле, значение ключа которой больше значения ключа записи, ставшей доступной посредством последнего оператора READ (ЧИТАТЬ). В сущности это не «та же» запись, которая была доступна последнему оператору READ (ЧИТАТЬ), поскольку значение дополнительного ключа и возможно другие значения уже изменены.

Семантика стандарта Кобола для этой ситуации была предметом требований для разъяснения и приведена в документах интерпретации X3J4.

(39) Зарезервированные слова (1 ЯДР). Добавлены следующие зарезервированные слова:

ALPHABET	АЛФАВИТ
ALPHABETIC-LOWER	СТРОЧНЫЕ
ALPHABETIC-UPPER	ПРОПИСНЫЕ
ALPHANUMERIC	БЦ
ALPHANUMERIC EDITED	БЦР
ANY	ЛЮБОЕ
BINARY	ДВОИЧНОЕ
CLASS	КЛАСС
COMMON	ОБЩАЯ
CONTENT	
CONTINUE	ПРОДОЛЖИТЬ
CONVERTING	ПРЕВРАЩАЯ
DAY-OF-WEEK	ДЕНЬ-НЕДЕЛИ
END-ADD	КОНЕЦ-СЛОЖИТЬ
END-CALL	КОНЕЦ-ВЫЗВАТЬ
END-COMPUTE	КОНЕЦ-ВЫЧИСЛИТЬ
END-DELETE	КОНЕЦ-УДАЛИТЬ
END-DIVIDE	КОНЕЦ-РАЗДЕЛИТЬ
END-EVALUATE	КОНЕЦ-ОЦЕНИТЬ
END-IF	КОНЕЦ-ЕСЛИ
END-MULTIPLY	КОНЕЦ-УМНОЖИТЬ
END-PERFORM	КОНЕЦ-ВЫПОЛНИТЬ
END-READ	КОНЕЦ-ЧИТАТЬ
END-RECEIVE	КОНЕЦ-ПОЛУЧИТЬ
END-RETURN	КОНЕЦ-ВЕРНУТЬ
END-REWRITE	КОНЕЦ-ОБНОВИТЬ
END-SEARCH	КОНЕЦ-ИСКАТЬ
END-START	КОНЕЦ-ПОДВЕСТИ
END-STRING	КОНЕЦ-СОБРАТЬ
END-SUBTRACT	КОНЕЦ-ОТНЯТЬ
END-UNSTRING	КОНЕЦ-РАЗОБРАТЬ
END-WRITE	КОНЕЦ-ПИСАТЬ
EVALUATE	ОЦЕНИТЬ
EXTERNAL	ВНЕШНЕЕ
FALSE	ЛОЖЬ
GLOBAL	ГЛОБАЛЬНОЕ, ГЛОБАЛЬНО
INITIALIZE	ИНИЦИИРОВАТЬ
NUMERIC-EDITED	ЧР

GOST EXPERT ГОСТ 22558-89,

ORDER	
OTHER	
PACKED-DECIMAL	ДЕСЯТИЧНОЕ
PADDING	
PURGE	ОЧИСТИТЬ
REFERENCE	ССЫЛКУ
REPLACE	ЗАМЕНИТЬ
STANDARD-2	СТАНДАРТ-М
TEST	ПРОВЕРКОЙ
THEN	ЗАТЕМ
TRUE	ИСТИНА

Обоснование

В каждом случае предполагается, что польза, получаемая от дополнительных возможностей, обеспечиваемых добавлением каждого зарезервированного слова, перевешивает неудобства, возникающие из переноса этого слова из сферы слов, определенных пользователем. Предполагается, что применение нового оператора REPLACE (ЗАМЕНИТЬ) смягчит неудобства для имеющихся программ, которые могут использовать любое из новых зарезервированных слов как слово пользователя.

Коснемся некоторых вопросов относительно необходимости того или иного зарезервированного слова. Зарезервированные слова облегчают создание эффективных компиляторов, облегчая синтаксический анализ исходной программы. Синтаксическое распознавание Кобола было бы затруднительным без зарезервированных слов. Рассмотрим следующий фрагмент программы:

ADD A TO B C CONTINUE

СЛОЖИТЬ А С Б Е ПРОДОЛЖИТЬ

Предполагая, что нет зарезервированных слов, невозможно определить, является ли CONTINUE (ПРОДОЛЖИТЬ) принимающим полем для ADD (СЛОЖИТЬ) или оператором CONTINUE (ПРОДОЛЖИТЬ). Также невозможно определить, является ли TO (С) принимающим полем, и оператор синтаксически неверен.

Если бы в синтаксисе Кобола были ограничители операторов, которые необходимы, приведенный выше пример можно было бы переписать так:

ADD A TO B C; CONTINUE

СЛОЖИТЬ А С Б Е; ПРОДОЛЖИТЬ

где ясно, что CONTINUE (ПРОДОЛЖИТЬ) не является частью оператора ADD (СЛОЖИТЬ). Тем не менее в синтаксисе Кобола нет ограничителей оператора; требование ограничителей добавило бы значительно большую несовместимость.

Другим возможным решением было бы, чтобы компилятор проверял до синтаксического разбора оператора каждое зарезервированное слово, не использовано ли оно как определенное пользователем. Однако это привело бы к большим потерям в терминах сложности компилятора и скорости компилирования.

Добавление нескольких зарезервированных слов для новых возможностей является значительно менее серьезной несовместимостью, чем несовместимость, вызываемая устранением всех зарезервированных слов из Кобола.

Итак, зарезервированные слова продолжают использоваться в новейших языках. Например, Паскаль и Ада, оба развивающиеся и возникшие после Кобола, также используют зарезервированные слова.

(40) Состояние ввода-вывода (1 ПОД, 1 ОТД, 1 ИПД). Добавлены новые значения состояния ввода-вывода.

Обоснование

В стандарте Кобола определены только несколько кодовых условий состояния ввода-вывода. В результате пользователь не мог установить среди многих различных ошибочных условий те, которые он хотел бы обрабатывать разными способами и/или каждая реализация определяла свое множество кодов состояния, которое покрывало разные ситуации разными способами. Кроме того, ГОСТ

GOST EXPERT ГОСТ 22558-89,

22558 оставлял результаты многих ситуаций ввода-вывода неопределенными; это значит, что ГОСТ 22558 устанавливал, что определенные критерии должны выполняться, но не устанавливал, что произойдет, если они не выполнены; таким образом, выполнение программы становилось неопределенным.

Настоящий стандарт Кобола определяет коды состояния для таких неопределенных ситуаций ввода-вывода. Таким образом, пользователь может проверить по этим условиям ошибки стандартным путем и предпринять корректирующее действие для определенных условий ошибки, где это возможно.

В общем, добавления могут повлиять на программы в следующих случаях:

а) если они проверяют специальные значения состояний, определенные реализацией, для выявления условий, определяемых в настоящее время;

б) если они относятся к состоянию успешного завершения для любого из условий, определяемых в настоящее время. (В случаях новых значений состояния ввода-вывода 04, 05 и 07 это влияет только на программы, которые просматривают обе позиции литеры состояния ввода-вывода для проверки на успешное выполнение;

в) если они относятся к некоторым действиям, зависящим от реализации, таким как ненормальное завершение программы, когда возникает некоторое из заново определенных условий.

Это изменение может иметь существенное влияние на те программы, которые проверяют специфические значения состояния ввода вывода.

Нужно отметить, что предыдущий стандарт Кобола не обеспечивал всех кодов состояния.

Отдельные часто употребляемые значения состояния ввода-вывода описаны в следующих пунктах.

а) Состояние ввода вывода-04. Оператор READ (ЧИТАТЬ) выполнен успешно, но длина обрабатываемой записи не соответствует фиксированным свойствам этого файла.

Обоснование

ГОСТ 22558 не определял последствия, если оператору READ (ЧИТАТЬ) доступна запись, содержащая соответственно больше или меньше литер, чем максимум или минимум для этого файла. Поэтому результат чтения такой записи не определен. Новое значение 04 состояния ввода-вывода сообщает пользователю о такой ситуации.

Поскольку настоящий стандарт предупреждает попытку занесения или обновления записи слишком большой или слишком малой, эта ситуация не может встретиться для записей, заносимых программой при реализации настоящего стандарта.

б) Состояние ввода-вывода = 05. Оператор OPEN (ОТКРЫТЬ) выполнен успешно, но необязательного файла, на который ссылаются, нет в наличии во время выполнения оператора OPEN (ОТКРЫТЬ).

Обоснование

Соответственно ГОСТ 22558 об отсутствии необязательного файла программе не сообщается до первого выполнения оператора READ (ЧИТАТЬ) для этого файла. Новое значение состояния ввода-вывода 05 дает определенную и доступную информацию при ссылке на файл в операторе OPEN (ОТКРЫТЬ), позволяя программе предпринимать более конкретные действия соответственно этому условию.

Это может повлиять только на программы, которые используют второй уровень фразы OPTIONAL (НЕОБЯЗАТЕЛЬНОГО) в модуле последовательного ввода-вывода и которые просматривают обе позиции литеры данного состояния ввода-вывода для проверки на успешное завершение оператора OPEN INPUT (ОТКРЫТЬ ВХОДНОЙ).

в) Состояние ввода-вывода = 07. Оператор ввода-вывода выполнен успешно. Тем не менее файл, на который ссылается оператор CLOSE (ЗАКРЫТЬ) с вариантом WITH NO REWIND (БЕЗ ПЕРЕМОТКИ), REEL/UNIT (КАТУШ-

GOST EXPERT ГОСТ 22558-89,

КУ/ТОМ) или FOR REMOVAL (С УДАЛЕНИЕМ), или оператор OPEN (ОТКРЫТЬ) с вариантом NO REWING (БЕЗ ПЕРЕМОТКИ), находится на носителе данных, к которому неприменимо понятие катушка (том).

Обоснование

Соответственно ГОСТ 22558 оператор OPEN (ОТКРЫТЬ) с вариантом NO REWIND (БЕЗ ПЕРЕМОТКИ) может быть использован только для последовательных файлов на одной катушке (томе), и оператор CLOSE (ЗАКРЫТЬ) с вариантом NO REWIND (БЕЗ ПЕРЕМОТКИ), REEL/UNIT (КАТУШКУ/ТОМ) или FOR REMOVAL (С УДАЛЕНИЕМ) неприемлем для файлов, находящихся на носителе, к которому неприменимы понятия катушки и тома. Поэтому для файлов массовой памяти эти образцы операторов OPEN (ОТКРЫТЬ) и CLOSE (ЗАКРЫТЬ) могут рассматриваться в сущности как успешные, если пересмотрены аномалии фраз NO REWIND (БЕЗ ПЕРЕМОТКИ), REEL/UNIT (КАТУШКУ/ТОМ) или FOR REMOVAL (С УДАЛЕНИЕМ). Новое значение 07 состояния ввода-вывода делает возможным успешное завершение, сохраняя в то же время информацию для пользователя, и предоставляя возможность предпринять определенные действия.

г) Состояние ввода-вывода = 14. Попытка выполнения последовательного оператора READ (ЧИТАТЬ) для относительного файла и число значащих цифр в относительном номере записи больше данного относительный ключ, описанного для этого файла.

Обоснование

ГОСТ 22558 устанавливает, что успешное выполнение оператора READ (ЧИТАТЬ) формата 1 для относительного файла обновляет содержимое данного относительный ключ (если он указан) и оно становится равным относительному номеру записи, доступной в этот момент. ГОСТ 22558 не определяет результат, если номер, определенный значащими цифрами относительного номера записи, больше данного относительный ключ. Новое значение 14 состояния ввода-вывода определяет этот результат.

д) Состояние ввода-вывода = 24. Производится попытка писать за пределами внешне определенных границ относительного или индексного файла, или сделана попытка последовательного оператора WRITE (ПИСАТЬ) и число значащих цифр в относительном номере записи больше размера данного относительный ключ, описанного для этого файла.

Обоснование

В ГОСТ 22558 значение состояния ввода-вывода 24 означало только попытку писать за пределами внешне определенных границ относительного или индексного файла. В настоящем стандарте Кобола значение 24 состояния ввода-вывода также включает последовательный оператор WRITE (ПИСАТЬ) для относительного файла, когда число значащих цифр в относительном номере записи больше размера данного относительный ключ, описанного для этого файла.

ГОСТ 22558 устанавливает, что при успешном выполнении оператора WRITE (ПИСАТЬ) для относительного файла относительный номер переданной записи будет помещен в данное относительный ключ (если он указан). Результат не определен, если номер, определяемый значащими цифрами относительного номера записи, больше данного относительный ключ.

Это изменение может повлиять только на те программы, которые последовательно записывают больше записей, чем максимальное значение, допускаемое шаблоном данного относительный ключ.

е) Состояние ввода-вывода = 35. Попытка выполнения оператора OPEN (ОТКРЫТЬ) с вариантом INPUT (ВХОДНОЙ) для файла, не определенного как необязательный, которого нет в наличии.

Обоснование

ГОСТ 22558 требует обязательного указания фразы OPTIONAL (НЕОБЯЗАТЕЛЬНОГО) для входных файлов, которые могут отсутствовать в определен-

ные моменты выполнения объектной программы. При этом не определено, что может случиться, когда файл, не объявленный необязательным, отсутствует. Новое значение 35 состояния ввода-вывода позволяет пользователю проверять это условие.

ж) Состояние ввода-вывода = 37. Попытка выполнения оператора OPEN (ОТКРЫТЬ) для файла, который должен быть файлом массовой памяти, но не является таковым.

Обоснование

Это новое значение состояния ввода-вывода сообщает, что: (1) предпринята попытка выполнения оператора OPEN I-O (ОТКРЫТЬ ВХОДНОЙ-ВЫХОДНОЙ) для файла не массовой памяти, или (2) предпринята попытка выполнения оператора OPEN (ОТКРЫТЬ) для файла не-массовой памяти, объявленного в программе как относительный или индексный файл.

ГОСТ 22558 не определяет, что происходит при таких обстоятельствах. Новое значение 37 состояния ввода-вывода позволяет пользователю проверять это условие ошибки.

Некоторые реализации обеспечивают расширение использования операторов OPEN I-O (ОТКРЫТЬ ВХОДНОЙ-ВЫХОДНОЙ) для доступа к терминалам. Такое расширение может быть устранено новым значением 37 состояния ввода-вывода.

з) Состояние ввода-вывода = 38. Попытка выполнения оператора OPEN (ОТКРЫТЬ) для файла, предварительно закрытого с замком.

Обоснование

ГОСТ 22558 указывает, что реализация должна обеспечить, чтобы файл, закрытый с замком, не мог быть открыт опять во время текущего выполнения единицы исполнения, но не определяет, что случится, если сделана попытка вновь открыть этот файл. Новое значение 38 состояния ввода-вывода позволяет пользователю проверять это условие.

и) Состояние ввода-вывода = 39. Оператор OPEN (ОТКРЫТЬ) неуспешен из-за конфликтных свойств файла (несоответствие между фиксированными свойствами файла и свойствами файла, указанными в программе).

Обоснование

Фиксированные свойства файла — это свойства, отмеченные при создании файла и которые не могут изменяться на протяжении жизненного цикла файла. К этим свойствам (атрибутам) относятся организация, кодовый набор, минимальный и максимальный размер логической записи, тип записи (фиксированной или переменной длины) коэффициент блокирования, литера заполнитель и ограничитель записи. Для индексных файлов (только для них) дополнительными фиксированными свойствами файла являются основной ключ записи, дополнительный ключ записи и основная последовательность ключей.

ГОСТ 22558 указывает, что организация файла устанавливается во время создания файла и в дальнейшем не может быть изменена. Определяется также для оператора OPEN INPUT (ОТКРЫТЬ ВХОДНОЙ), OPEN I-O (ОТКРЫТЬ ВХОДНОЙ-ВЫХОДНОЙ) или OPEN EXTEND (ОТКРЫТЬ ДОПОЛНЯЕМЫЙ), что описание файла, включающее фразы CODE-SET (АЛФАВИТ), RECORD CONTAINS (В ЗАПИСИ) и BLOCK CONTAINS (В БЛОКЕ), должно быть эквивалентным тому, что используется при создании файла. Возможность указывать литеру заполнитель и ограничитель записи — новые возможности Кобола. Для индексных файлов ГОСТ 22558 указывал, что описания данных и относительное расположение в записи данных ключ записи и дополнительный ключ записи, и число дополнительных ключей записи должны быть такими же, как и при создании файла. ГОСТ 22558 обеспечивает возможность воздействия на основную последовательность, используемую для ключей индексного файла.

ГОСТ 22558 не определяет что произойдет, если фиксированные свойства файла не согласуются со свойствами файла, определяемыми в программе. Новое

GOST EXPERT ГОСТ 22558-89,

значение 39 состояния ввода-вывода позволяет пользователю проверять это условие.

к) Состояние ввода-вывода = 42. Попытка выполнения оператора CLOSE (ЗАКРЫТЬ) для файла, который не был открыт.

Обоснование

ГОСТ 22558 не разрешает ссылаться в операторе CLOSE (ЗАКРЫТЬ) на файл, который не был открыт, но не определяет последствий такой ссылки. Новое значение 42 состояния ввода-вывода позволяет пользователю проверять это условие.

л) Состояние ввода-вывода = 43. Для файла массовой памяти при последовательном доступе последним оператором ввода-вывода, выполненным для соответствующего файла до выполнения оператора DELETE (УДАЛИТЬ) или REWRITE (ОБНОВИТЬ), не был успешно выполненный оператор READ (ЧИТАТЬ).

Обоснование

ГОСТ 22558 указывает, что для файла при последовательном доступе последним оператором ввода-вывода, выполненным для файла перед выполнением оператора DELETE (УДАЛИТЬ) или REWRITE (ОБНОВИТЬ), должен быть успешно выполненный оператор READ (ЧИТАТЬ), но не указывает, что случится, если требование не удовлетворено. Новое значение 43 состояния ввода-вывода позволяет пользователю проверять это условие.

м) Состояние ввода-вывода = 44. Имеется нарушение границ из-за попытки обновить запись: (1) или в последовательном файле, (2) или в относительном файле на уровне 1 модуля относительного ввода-вывода, (3) или в индексном файле на уровне 1 модуля индексного ввода-вывода и размер записи не совпадает с размером заменяемой записи.

Обоснование

ГОСТ 22558 указывает, что для оператора REWRITE (ОБНОВИТЬ) число позиций литер в новой записи должно быть равно числу позиций литер в заменяемой записи, но не определяет, что происходит, если это требование не удовлетворено.

Новое значение 44 состояния ввода-вывода позволяет пользователю проверять эти условия.

н) Состояние ввода-вывода = 46. Сделана попытка выполнения последовательного оператора READ (ЧИТАТЬ) для файла, открытого в режиме ввода или ввода-вывода, и не была установлена действительная следующая запись по одной из следующих причин: (1) предшествующий оператор START (ПОДВЕСТИ) неуспешен, (2) предшествующий оператор READ (ЧИТАТЬ) неуспешен, но не привел к условию «в конце», или (3) предшествующий оператор READ (ЧИТАТЬ) привел к условию «в конце».

Обоснование

ГОСТ 22558 указывает, что при этих обстоятельствах выполнение оператора READ (ЧИТАТЬ) неправомочно или его выполнение неуспешно, но не определяет код состояния для указания такой ситуации. Новое значение 46 состояния ввода-вывода обеспечивает пользователю возможность проверять это условие. Состояние ввода-вывода 46 может встретиться только тогда, когда не предпринято никаких корректирующих действий, следующих за предыдущим оператором READ (ЧИТАТЬ) или START (ПОДВЕСТИ).

о) Состояние ввода-вывода = 47. Имеется попытка выполнения оператора READ (ЧИТАТЬ) или START (ПОДВЕСТИ) для файла, не открытого как входной или входной-выходной.

Обоснование

ГОСТ 22558 требует, чтобы ко времени выполнения оператора READ (ЧИТАТЬ) или START (ПОДВЕСТИ) файл был открыт как входной или входной-выходной, но не указывает, что происходит, если требование не удовлетворено,

GOST EXPERT ГОСТ 22558-89,

Новое значение 47 состояния ввода-вывода дает возможность пользователю проверять это условие.

п) Состояние ввода-вывода = 48. Попытка выполнения оператора WRITE (ПИСАТЬ) : (1) или для последовательного файла, не открытого для вывода или дополнения, (2) или для относительного или индексного файла, не открытого для ввода-вывода, вывода или дополнения.

Обоснование

ГОСТ 22558 требует, чтобы файл был открыт в одном из определенных режимов, но не определяет, что происходит, если требование не удовлетворено. Новое значение 48 состояния ввода-вывода позволяет пользователю проверять это условие.

р) Состояние ввода-вывода = 49. Попытка выполнения оператора DELETE (УДАЛИТЬ) или REWRITE (ОБНОВИТЬ) для файла, не открытого для ввода-вывода.

Обоснование

ГОСТ 22558 требует, чтобы файл был открыт для ввода-вывода, но не определяет, что происходит, если требование не удовлетворено. Новое значение 49 состояния ввода-вывода дает пользователю возможность проверять это условие.

(45) Ключ состояния коммуникации (1 КОМ).

Добавлены новые значения ключа состояния коммуникации.

Обоснование

ГОСТ 22558 оставляют результаты некоторых ситуаций коммуникации неопределенными. Настоящий стандарт определяет новые значения ключа состояния коммуникации для этих ситуаций с тем, чтобы пользователь мог выявить эти условия ошибки стандартным путем и таким образом предпринять корректирующие действия, подходящие для данного случая.

Новые значения ключа состояния коммуникации влияют только на те существующие программы, которые располагают некоторыми другими действиями, имеющими место, когда встречаются вновь определяемые условия ошибки.

Конкретные добавленные значения ключа состояния коммуникации описаны ниже.

а) Ключ состояния коммуникации — 15. Символический источник, или одна или несколько очередей или адресаты уже запрещены/разрешены.

Обоснование

Если ко времени выполнения оператора DISABLE (ЗАПРЕТИТЬ) или ENABLE (РАЗРЕШИТЬ) источник, очередь или адресат уже запрещен или разрешен соответственно, спецификации ГОСТ 22558 подразумевают, что значение ключа состояния коммуникации полагается равным 00 (нулю). Новое значение 15 ключа состояния коммуникации обеспечивает пользователя этой информацией.

б) Ключ состояния коммуникации — 21. Символический источник неизвестен.

Обоснование

В ГОСТ 22558 для того, чтобы определить, известно ли символическое имя исходного терминала системе управления сообщениями при выполнении оператора RECEIVE (ПОЛУЧИТЬ), пользователь должен сравнивать данное символический-источник с пробелами. ГОСТ 22558 не определяет, что произойдет, если символический источник в CD (ОК), на который имеется ссылка в операторе ENABLE (РАЗРЕШИТЬ) или DISABLE (ЗАПРЕТИТЬ), неизвестен. Новое значение 21 ключа состояния коммуникации предоставляет такую информацию.

в) Ключ состояния коммуникации = 65. Превышена вместимость выходной очереди.

Обоснование

ГОСТ 22558 не определяет, что произойдет, если превышена вместимость выходной очереди при выполнении оператора SEND (ПОСЛАТЬ). Такая ситуация теперь определяется заданием нового значения 65 ключа состояния комму-

GOST EXPERT ГОСТ 22558-89,

никации.

г) Ключ состояния коммуникации =70. Один или несколько адресатов не содержат соотнесенных им порций.

Обоснование

Это значение ключа состояния коммуникации возникает только при новом операторе PURGE (ОЧИСТИТЬ). Таким образом оно не может встретиться в программах, соответствующих ГОСТ 22558.

д) Ключ состояния коммуникации =80. Встретилась комбинация по крайней мере двух условий ключа состояния 10, 15 и 20.

Обоснование

Если использована возможность нескольких адресатов и один из адресатов запрещен, в то время как другой адресат неизвестен, ГОСТ 22558 не определяет, будет ли ключ состояния установлен оператором SEND (ПОСЛАТЬ) на значение 10 или 20. Новое значение 80 ключа состояния коммуникации установлено для определения такой ситуации. Новое значение 80 ключа состояния коммуникации также устанавливается в случае оператора ENABLE (РАЗРЕШИТЬ) или DISABLE (ЗАПРЕТИТЬ), где встречаются и новое условие ключа состояния коммуникации 15 и условие ключа состояния коммуникации 20.

е) Ключ состояния коммуникации = 9х. Состояние, определяемое реализацией.

Обоснование

Этот новый диапазон значений ключа состояния коммуникации позволяет реализации определять множество различных условий ошибки. Это предоставляет возможность пользователю проверять условия ошибки, определяемые реализацией, подобно возможности ГОСТ 22558 для проверки значений состояния ввода-вывода для ошибок ввода-вывода, определяемых реализацией.

(42) Ключ ошибки коммуникации (1 КОМ). Добавлены новые значения ключа ошибки коммуникации. Эти новые значения описаны ниже.

а) Значение ключа ошибки коммуникации = 2. Символический адресат запрещен.

Обоснование

Был выполнен оператор SEND (ПОСЛАТЬ) и адресат, к которому относится этот ключ ошибки, запрещен. В ГОСТ 22558 это условие не распознавалось пользователем.

б) Значение ключа ошибки коммуникации = 5. Символический адресат уже разрешен (запрещен).

Обоснование

Был выполнен оператор ENABLE (РАЗРЕШИТЬ) или DISABLE (ЗАПРЕТИТЬ) и адресат, к которому относится этот ключ ошибки, уже разрешен (запрещен). В ГОСТ 22558 это условие не распознавалось пользователем.

в) Значение ключа ошибки коммуникации = 6. Превышена вместимость выходной очереди.

Обоснование

Был выполнен оператор SEND (ПОСЛАТЬ) и система управления сообщениями не смогла включить в очередь сообщение, сегмент сообщения или порцию сообщения или сегмента сообщения, поскольку выходная очередь адресата, к которому относится это значение ключа ошибки коммуникации, заполнена. В ГОСТ 22558 это условие пользователем не распознавалось.

г) Значение ключа ошибки коммуникации = A до Z. Условие, определяемое реализацией.

GOST EXPERT ГОСТ 22558-89,

Обоснование

Система управления сообщениями получила определенное реализацией условие ошибки, не входящее в имеющиеся значения ключа ошибки коммуникации. Согласно ГОСТ 22558 реализация не могла предоставить пользователю такую информацию.

ПРИЛОЖЕНИЕ 2

СПИСКИ ЭЛЕМЕНТОВ ЯЗЫКА

1. Список устаревших элементов языка

Назначение категории устаревших элементов языка — ограничить влияние удаляемых средств, которые рассматриваются как устаревшие или не должным образом определенные и будут удалены в следующей редакции стандарта. Хотя элементы этой категории устаревшие, немедленное удаление их из стандарта принесло бы неудобства пользователям. Средства, отнесенные к категории устаревших элементов, имеют следующие характеристики:

* элементы языка, которые подлежат удалению из стандарта Кобола, определяются в настоящем стандарте как устаревшие элементы языка (до их удаления);

* взаимоотношение между устаревшими элементами и другими элементами языка не определено, если только не оговорено другое;

* устаревшие элементы языка будут удалены в следующей редакции стандарта;

* требуется, чтобы соответствующая стандарту реализация Кобола поддерживала устаревшие элементы языка для подмножеств и уровней необязательных модулей, для которых объявлена поддержка.

Далее следует список элементов языка, объявляемых в настоящем стандарте устаревшими. Для каждого элемента списка приводится обоснование отнесения этого элемента к категории устаревших элементов.

(1) Замена литеры двумя литерами (1 ЯДР). Если набор литер состоит менее чем из 51 (72) литер, вместо одной литеры могут представляться две литеры. Эта возможность отнесена к категории устаревших элементов.

Обоснование

Эта спецификация привнесена с тех времен, когда оборудование не могло обеспечить полный набор литер Кобола. Такое ограничение на количество литер, допускаемых оборудованием, больше не существует.

(2) ALL литерал (ВСЕ литерал) и числовое или числовое редактируемое данное (2 ЯДР). Стандартная константа ALL литерал (ВСЕ литерал), когда она связана с числовым или числовым редактируемым данным и когда длина литерала больше единицы, относится к категории устаревших элементов.

Обоснование

Причиной считать этот элемент устаревшим послужило то, что результат перемещения ALL литерал (ВСЕ литерал) в числовое данное часто непредсказуем. Например, соответственно интерпретации, операторы

01 A PIC 99V99.	01 A Ш 99Т99.
MOVE ALL «99» TO A.	ПОМЕСТИТЬ ВСЕ «99» В А.
MOVE ALL «123» TO A.	ПОМЕСТИТЬ ВСЕ «123» В А.

дадут соответственно значения 99.00 и 31.00.

GOST EXPERT ГОСТ 22558-89,

(3) Параграфы AUTHOR (АВТОР), INSTALLATION (ПРЕДПРИЯТИЕ), DATE-WRITTEN (ДАТА-НАПИСАНИЯ), DATE-COMPILED (ДАТА-ТРАНСЛЯЦИИ) и SECURITY (ПОЛНОМОЧИЯ) (1 ЯДР). Эти параграфы раздела оборудования отнесены к категории устаревших элементов.

Обоснование

Назначение параграфов AUTHOR (АВТОР), INSTALLATION (ПРЕДПРИЯТИЕ), DATE-WRITTEN (ДАТА-НАПИСАНИЯ), DATE-COMPILED (ДАТА-ТРАНСЛЯЦИИ) и SECURITY (ПОЛНОМОЧИЯ) может быть достигнуто посредством использования строк комментария в разделе идентификации, поскольку эти параграфы не влияют на работу Кобол-программы.

Чистка и регуляризация языка Кобол достигается объявлением устаревшими многих определяемых реализацией элементов. Форматы параграфов DATE-COMPILED (ДАТА-ТРАНСЛЯЦИИ) и SECURITY (ПОЛНОМОЧИЯ) являются примером параграфов статьи-комментария, определяемых реализацией.

Взаимодействие оператора COPY (КОПИРОВАТЬ) со статьями комментариев в параграфах AUTHOR (АВТОР), INSTALLATION (ПРЕДПРИЯТИЕ), DATE-WRITTEN (ДАТА-НАПИСАНИЯ), DATE-COMPILED (ДАТА-ТРАНСЛЯЦИИ) и SECURITY (ПОЛНОМОЧИЯ) часто неоднозначно, например, наличие слова COPY (КОПИРОВАТЬ) в статье-комментарии и использование оператора COPY (КОПИРОВАТЬ) в статье-комментарии.

(4) Фраза MEMORY SIZE (РАЗМЕР ПАМЯТИ) (1 ЯДР).

Фраза MEMORY SIZE (РАЗМЕР ПАМЯТИ) параграфа OBJECT-COMPUTER (РАБОЧАЯ-МАШИНА) отнесена к категории устаревших элементов языка.

Обоснование

Эта анахроническая возможность языка сохранилась с того времени, когда многие системы требовали определения размера памяти для загрузки единицы исполнения. Емкость памяти для семейства главных моделей часто колебалась от 8К до 64К максимум. Кобол-программы используют фразу MEMORY SIZE (РАЗМЕР ПАМЯТИ) для генерирования объектов для специальных моделей.

Эта возможность рассматривается как функция операционной системы в современной вычислительной среде. В стандарте Кобола фраза MEMORY SIZE (РАЗМЕР ПАМЯТИ) необязательна. Таким образом нет стандартных соответствующих реализаций Кобола, требующих использования фразы MEMORY SIZE (РАЗМЕР ПАМЯТИ) для определения размера памяти объектной машины.

(5) Фраза RERUN (ПЕРЕПРОГОН) (1 ПОД, 1 ОТД, 1 ИПД). Фраза RERUN (ПЕРЕПРОГОН) параграфа I-O-CONTROL (УПРАВЛЕНИЕ-ВВОДОМ-ВЫВОДОМ) отнесена к категории устаревших элементов.

Обоснование

Обеспечиваются семь форм фразы RERUN (ПЕРЕПРОГОН). Требуется, чтобы реализация поддерживала по крайней мере одну из них.

Эта возможность рассматривается как функция операционной системы в современной вычислительной среде.

Фраза RERUN (ПЕРЕПРОГОН) обеспечивает только наполовину возможности перепрогона/рестарта. То есть синтаксис и семантика для рестарта не определены. Из-за разнообразия форм фразы RERUN (ПЕРЕПРОГОН) нет гарантии, что программа, использующая эту фразу, будет переносимой (мобильной).

(6) Фраза MULTIPLE FILE TAPE (НА ОДНОЙ КАТУШКЕ) (2 ПОД, 1 ГОТ). Фраза MULTIPLE FILE TAPE (НА ОДНОЙ КАТУШКЕ) параграфа I-O-CONTROL (УПРАВЛЕНИЕ-ВВОДОМ-ВЫВОДОМ) раздела оборудования отнесена к категории устаревших элементов.

Обоснование

Фраза MULTIPLE FILE TAPE (НА ОДНОЙ КАТУШКЕ) могла бы быть функцией операционной системы, а не отдельной Кобол-программы. Поэтому фраза MULTIPLE FILE TAPE (НА ОДНОЙ КАТУШКЕ) отнесена к категории устаревших элементов.

(7) Фраза LABEL RECORDS (МЕТКИ) (1 ПОД, 1 ОТД, 1 ИПД, 1 ГОТ). Фраза LABEL RECORDS (МЕТКИ) статьи описания файла отнесена к категории устаревших элементов и рассматривается как необязательная фраза.

Обоснование

Фраза LABEL RECORDS (МЕТКИ) рассматривается как необязательная фраза категории устаревших элементов. Определение наличия меток файла рассматривается как функция операционной системы и поэтому не относится к Кобол-программе.

(8) Фраза VALUE OF (ЗНАЧЕНИЕ) (1 ПОД, 1 ОТД, 1 ИПД, 1 ГОТ). Фраза VALUE OF (ЗНАЧЕНИЕ) статьи описания файла отнесена к категории устаревших элементов.

Обоснование

Описание элементов метки файла рассматривается как функция операционной системы и не относится к Кобол-программе. Поэтому фраза VALUE OF (ЗНАЧЕНИЕ) отнесена к категории устаревших элементов.

(9) Фраза DATA RECORDS (ЗАПИСИ ДАННЫХ) (1 ПОД, 1 ОТД, 1 ИПД). Фраза DATA RECORDS (ЗАПИСИ ДАННЫХ) статьи описания файла отнесена к категории устаревших элементов.

Обоснование

Фраза DATA RECORDS (ЗАПИСИ ДАННЫХ) избыточна и может привести к несоответствию документации.

(10) Оператор ALTER (ИЗМЕНИТЬ) (1 ЯДР).

Оператор ALTER (ИЗМЕНИТЬ) отнесен к категории устаревших элементов.

Обоснование

Использование в программе оператора ALTER (ИЗМЕНИТЬ) приводит к затруднению понимания программы и ее поддержки. Оператор ALTER (ИЗМЕНИТЬ) не имеет уникального назначения, поскольку оператор GO TO DEPENDING (ПЕРЕЙТИ В ЗАВИСИМОСТИ ОТ) служит для той же цели.

(11) Фраза KEY (КЛЮЧ) оператора DISABLE (ЗАПРЕТИТЬ) (2 КОМ). Фраза KEY (КЛЮЧ) оператора DISABLE (ЗАПРЕТИТЬ) отнесена к категории устаревших элементов и рассматривается как необязательная фраза.

Обоснование

Фраза KEY (КЛЮЧ) оператора DISABLE (ЗАПРЕТИТЬ) используется как парольное средство доступа для оператора DISABLE (ЗАПРЕТИТЬ). Тем не менее не указаны правила для определения, когда значение в фразе KEY (КЛЮЧ) соответствует системному паролю, вследствие чего ситуация определяется реализацией. Таким образом, функция, обеспечиваемая фразой KEY (КЛЮЧ), немобильна.

(12) Фраза KEY (КЛЮЧ) оператора ENABLE (РАЗРЕШИТЬ) (2 КОМ). Фраза KEY (КЛЮЧ) оператора ENABLE (РАЗРЕШИТЬ) отнесена к категории устаревших элементов и рассматривается как необязательная фраза

Обоснование

Фраза KEY (КЛЮЧ) оператора ENABLE (РАЗРЕШИТЬ) используется как парольное средство доступа для оператора ENABLE (РАЗРЕШИТЬ). Тем не менее правила для определения, когда значение в фразе KEY (КЛЮЧ) соответствует системному паролю, не указаны, вследствие чего ситуация определяется реализацией. Таким образом, функция, обеспечиваемая фразой KEY (КЛЮЧ), немобильна.

(13) Оператор ENTER (ВОЙТИ) (1 ЯДР). Оператор ENTER (ВОЙТИ) отнесена к категория устаревших элементов.

Обоснование

Оператор ENTER (ВОЙТИ) был предвестником оператора CALL (ВЫЗВАТЬ) и вызова внешних подпрограмм. Оператор ENTER (ВОЙТИ) не обеспечивает мобильность, поскольку является необязательным и определяется реали-

GOST EXPERT ГОСТ 22558-89,

зацией; поэтому оператор ENTER (ВОЙТИ) не является хорошим объектом для стандартизации.

(14) Необязательность имени-процедуры-1 в операторе GO TO (ПЕРЕЙТИ К) (2 ЯДР). Необязательность имени-процедуры-1 в операторе GO TO (ПЕРЕЙТИ К) отнесено к категории устаревших элементов.

Обоснование

Необязательность имени-процедуры-1 в операторе GO TO (ПЕРЕЙТИ К) зависит от оператора ALTER (ИЗМЕНИТЬ). Если имя-процедуры-1 не указано в формате 1 оператора GO TO (ПЕРЕЙТИ К), то оператор ALTER (ИЗМЕНИТЬ), ссылающийся на этот оператор GO TO (ПЕРЕЙТИ К), должен быть выполнен до выполнения оператора GO TO (ПЕРЕЙТИ). Поскольку оператор ALTER (ИЗМЕНИТЬ) отнесен к категории устаревших элементов языка, необязательность имени-процедуры-1 в операторе GO TO (ПЕРЕЙТИ К) также отнесена к категории устаревших элементов.

(15) Фраза REVERSED (РЕВЕРСНО) оператора OPEN (ОТКРЫТЬ) (2 ПОД). Фраза REVERSED (РЕВЕРСНО) оператора OPEN (ОТКРЫТЬ) отнесена к категории устаревших элементов.

Обоснование

Последовательный файл может быть открыт для ввода с чтением в обратном порядке. Оборудование, необходимое для обеспечения таких действий, используется не очень широко. Следовательно, это средство редко реализуется и не является хорошим объектом для стандартизации. Поскольку это средство из списка средств, зависящих от оборудования, оно является необязательным и может быть по желанию реализовано или не реализовано.

(16) Оператор STOP литерал (ОСТАНОВИТЬ литерал) (1 ЯДР). Вариант литерал оператора STOP (ОСТАНОВИТЬ) отнесен к категории устаревших элементов.

Обоснование

Общее правило (4) оператора STOP (ОСТАНОВИТЬ) гласит: «Если указан вариант STOP литерал-1 (ОСТАНОВИТЬ литерал-1), выполнение единицы исполнения приостанавливается, а литерал-1 сообщается оператору. Продолжение функционирования единицы исполнения начинается со следующего выполнимого оператора в том случае, когда подключена зависящая от реализации процедура управления возобновлением единицы исполнения».

Назначение оператора STOP литерал (ОСТАНОВИТЬ литерал) по существу определяется реализацией и поэтому программы, использующие его, немобильны.

(17) Модуль сегментации. Модуль сегментации отнесен к категории устаревших элементов.

Обоснование

К настоящему моменту функция, обеспечиваемая модулем сегментации, обеспечивается на уровне операционной системы, внешнем по отношению к исходному коду Кобола. Поэтому средства сегментации остаются в настоящем стандарте как устаревший элемент, подлежащий удалению в следующей редакции стандарта.

Решение сделать модуль сегментации необязательным позволяет имеющимся реализациям по-прежнему предлагать это средство с целью совместимости, не вынуждая при этом новые реализации обеспечивать средства, базирующиеся на устаревшей технологии.

(18) Модуль отладки. Модуль отладки отнесен к категории устаревших элементов.

Обоснование

К настоящему моменту функция, обеспечиваемая модулем отладки, зачастую обеспечивается средством диалоговой отладки, не требующим исходных

GOST EXPERT ГОСТ 22558-89,

операторов Кобола. Поэтому средства отладки остаются в настоящем стандарте как устаревший элемент, подлежащий удалению в следующей редакции стандарта.

Решение сделать модуль отладки необязательным позволяет имеющимся реализациям по-прежнему предлагать это средство с целью совместимости, не вынуждая при этом новые реализации обеспечивать средства, базирующиеся на устаревшей технологии.

2. Список элементов языка, определяемых реализацией

Ниже следует список элементов языка настоящего стандарта, для которых спецификация синтаксиса или правил доопределяется в реализации.

(1) Системное-имя. Правила образования системного-имени определяются реализацией (см. ч. 4, п. 4.2.2.1.2).

(2) Представление данных. Выбор основания системы счисления обычно зависит от арифметических возможностей вычислительной машины (см. ч. 4, п. 4.3.4).

(3) Алгебраический знак. Если не использована фраза SIGN (ЗНАК), знаки числа будут представлены так, как это определяет реализация (см. ч. 4, п. 4.3.5).

(4) Выравнивание данных. Каждая реализация, обеспечивающая специальные типы выравнивания, должна описать действие неявного заполнителя и семантику операторов, ссылающихся на содержащие его группы (см. ч. 4, п. 4.3.7).

(5) Внешний переключатель. Внешний переключатель является устройством оборудования или программным средством, определяемым и именуемым реализацией (см. ч. 4, п. 4.5).

(6) Внешний переключатель. Реализация определяет область действия (программа, единица исполнения и т. п.) каждого внешнего переключателя и возможности (внешние по отношению к Коболу), используемые для изменения состояния внешнего переключателя (см. ч. 4, п. 4.5).

(7) Область B. Область B занимает конечное число позиций литер, определяемое реализацией (см. ч. 4, п. 7.2).

(8) Имя-машины в параграфе SOURCE-COMPUTER (ИСХОДНАЯ-МАШИНА). Имя-машины является системным-именем, поэтому правила образования имени-машины определяются реализацией (см. ч. 6, п. 4.3.3, синтаксическое правило (1)).

(9) Имя-машины в параграфе OBJECT-COMPUTER (РАБОЧАЯ-МАШИНА). Имя-машины является системным-именем, поэтому правила образования имени-машины определяется реализацией (см. ч. 6, п. 4.4.3, синтаксическое правило (1)).

(10) Фраза MEMORY SIZE (РАЗМЕР ПАМЯТИ).

Если подмножество, задаваемое пользователем, меньше минимальной конфигурации, требуемой для выполнения объектной программы, необходимые меры определяются реализацией (см. ч. 6, п. 4.4.4, общее правило (1)).

(11) Программная основная последовательность.

При отсутствии фразы PROGRAM COLLATING SEQUENCE (ПРОГРАММНЫЙ АЛФАВИТ) используется внутренняя основная последовательность (см. ч. 6, п. 4.4.4, общее правило (6)).

(12) Имя-реализации в параграфе SPECIAL-NAMES (СПЕЦИАЛЬНЫЕ-ИМЕНА). Имя-реализации является системным именем, поэтому правила образования имени-реализации определяются реализацией (см. ч. 6, п. 4.5.2).

(13) STANDARD-1 (СТАНДАРТ-А) во фразе ALPHABET (АЛФАВИТ). Если между отдельными литерами стандартного и внутреннего наборов отсутст-

GOST EXPERT ГОСТ 22558-89,

вует соответствие и оно никак не определено, это соответствие определяется реализацией (см. ч. 6, п. 4.5.4, общее правило 4 а).

(14) Имя-реализации во фразе ALPHABET (АЛФАВИТ).

Если задана фраза имя-реализации-2, набор кодов литер или соответствующая основная последовательность определяется реализацией. Реализация также определяет соответствие между литерами или литерой набора кодов, специфицированного фразой имя-реализации-2, и внутренним набором кодов литер (см. ч. 6, п. 4.5.4, общее правило 4в).

(15) Фраза RERUN (ПЕРЕПРОГОН). Реализация должна обеспечивать хотя бы один вариант фразы RERUN (ПЕРЕПРОГОН) (см. ч. 7, п. 2.12.4, общее правило (2)).

(16) Фраза RECORD (В ЗАПИСИ). Если в статье описания записи не указана фраза RECORD (В ЗАПИСИ) или если во фразе RECORD (В ЗАПИСИ) указан диапазон позиций литер, реализация определяет, какой получается тип записей — переменной длины или фиксированной длины (см. ч. 2, п. 2.1.4.3).

(17) Фраза INDEXED BY (ИНДЕКСИРУЕТСЯ). Имя-индекса, идентифицируемое в исходной программе посредством фразы INDEXED BY (ИНДЕКСИРУЕТСЯ), не определяется в программе, так как его размещение и формат зависят от машины и, не будучи данными, имена-индексов не могут быть связаны с какой-либо структурой данных (см. ч. 6, п. 5.8.3, синтаксическое правило (13)).

(18) Фраза SIGN (ЗНАК). Если статья описания числового данного не содержит необязательную фразу SIGN (ЗНАК), а строка литер его шаблона содержит символ S (З), то позиция и представление знака определяется реализацией (см. ч. 6, п. 5.12.4, общее правило (4)).

(19) Фраза SIGN (ЗНАК). Если вариант SEPARATE CHARACTER (ОТДЕЛЬНО) не указан, представление знака данного определяет реализация (см. ч. 6, п. 5.12.4, общее правило 5в).

(20) Фраза SYNCHRONIZED (ВЫДЕЛЕНО). Фраза SYNCHRONIZED (ВЫДЕЛЕНО), в которой не указано ни RIGHT (ВПРАВО), ни LEFT (ВЛЕВО), означает, что элементарное данное нужно расположить между естественными границами таким образом, чтобы достичь наиболее эффективного его использования. Особенности размещения определяются реализацией (см. ч. 6, п. 5.13.4, общее правило (2)).

(21) Фраза SYNCHRONIZED (ВЫДЕЛЕНО). Реализация должна указывать, как обрабатываются элементарные данные, связанные с фразой SYNCHRONIZED (ВЫДЕЛЕНО), в зависимости от: (а) формата внешнего представления содержащих их записей или групп, в описании которых указана фраза SYNCHRONIZED (ВЫДЕЛЕНО); (б) необходимого порождения неявных заполнителей, если элементарное данное, непосредственно предшествующее данному, описание которого содержит фразу SYNCHRONIZED (ВЫДЕЛЕНО), не оканчивается на естественной границе (см. ч. 6, п. 5.13.4, общее правило (8)).

(22) Фраза SYNCHRONIZED (ВЫДЕЛЕНО). Реализация может задавать автоматическое выравнивание внутри записи для любого внутреннего формата данных, за исключением данных, использование которых указано DISPLAY (ДЛЯ ВЫДАЧИ). Однако запись в целом может быть выделена (см. ч. 6, п. 5.13.4, общее правило (9)).

(23) Фраза об использовании USAGE IS BINARY (ДВОИЧНОЕ). Каждая реализация определяет точное воздействие фразы BINARY (ДВОИЧНОЕ) на выравнивание и представление данного в памяти машины, включая представление любого алгебраического знака. Реализацией должно быть распределено достаточно памяти для размещения десятичных значений максимального диапазона, определяемого строкой литер шаблона (см. ч. 6, п. 5.14.4, общее правило (3)).

GOST EXPERT ГОСТ 22558-89,

(24) Фраза об использовании USAGE IS COMPUTATIONAL (ДЛЯ ВЫЧИСЛЕНИЙ). Каждая реализация определяет точное воздействие фразы USAGE IS COMPUTATIONAL (ДЛЯ ВЫЧИСЛЕНИЙ) на выравнивание и представление данного в памяти машины, включая представление алгебраического знака и допустимый диапазон значений данных (см. ч. 6, п. 5.14.4, общее правило (4)).

(25) Фраза об использовании USAGE IS INDEX (ДЛЯ ИНДЕКСА). Каждая реализация определяет точное воздействие фразы USAGE IS INDEX (ДЛЯ ИНДЕКСА) на выравнивание и представление данного в памяти машины, включая действительные значения, присвоенные номеру вхождения элемента таблицы (см. ч. 6, п. 5.14.4, общее правило (7)).

(26) Фраза об использовании USAGE IS PACKED-DECIMAL (ДЕСЯТИЧНОЕ). Каждая реализация определяет точное воздействие фразы USAGE IS PACKED-DECIMAL (ДЕСЯТИЧНОЕ) на выравнивание и представление данного в памяти машины, включая представление алгебраического знака. Реализацией должен быть выделен достаточный объем памяти для размещения максимального значения, соответствующего десятичным позициям числа, определяемого строкой литер шаблона (см. ч. 6, п. 5.14.4, общее правило (9)).

(27) Арифметическое выражение. Каждая реализация указывает способы, используемые в обработке арифметических выражений (см. ч. 6, п. 6.2.3, правило (6)).

(28) Оператор ACCEPT (ПРИНЯТЬ). Мнемоническое имя в операторе ACCEPT (ПРИНЯТЬ) должно связываться с устройством оборудования (см. ч. 6, п. 6.5.3, синтаксическое правило (1)).

(29) Оператор ACCEPT (ПРИНЯТЬ). Любое преобразование данных, требуемое при их перемещении с внешнего устройства в данное, представленное идентификатором-1, определяется реализацией (см. ч. 6, п. 6.5.4, общее правило (1)).

(30) Оператор ACCEPT (ПРИНЯТЬ). Размер одной передачи для каждого устройства оборудования определяется реализацией (см. ч. 6, п. 6.5.4, общее правило (2)).

(31) Оператор ACCEPT (ПРИНЯТЬ). Если вариант FROM (С) не указан, то используется устройство, определяемое реализацией как стандартное (см. ч. 6, п. 6.5.4, общее правило (5)).

(32) Оператор ADD (СЛОЖИТЬ). Выделение достаточного поля памяти для выполнения вычислений без потери значащих цифр обеспечивается компилятором (см. ч. 6, п. 6.6.4, общее правило (4)).

(33) Оператор COMPUTE (ВЫЧИСЛИТЬ). Каждая реализация определяет приемы обработки арифметических выражений (см. ч. 6, п. 6.8.4, общее правило (3)).

(34) Оператор DISPLAY (ВЫДАТЬ). Мнемоническое-имя в операторе DISPLAY (ВЫДАТЬ) связывается с устройством оборудования (см. ч. 6, п. 6.10.3, синтаксическое правило (1)).

(35) Оператор DISPLAY (ВЫДАТЬ). Любое преобразование данных, которое может потребоваться при выводе литерала-1 или данного, соотнесенного идентификатору-1, на внешнее устройство, определяется реализацией (см. ч. 6, п. 6.10.4, общее правило (1)).

(36) Оператор DISPLAY (ВЫДАТЬ). Размер одной передачи для каждого устройства определяется реализацией (см. ч. 6, п. 6.10.4, общее правило (2)).

(37) Оператор DISPLAY (ВЫДАТЬ). Если не указана фраза UPON (НА), используется стандартное выводное устройство реализации (см. ч. 6, п. 6.10.4, общее правило (7)).

(38) Оператор ENTER (ВОЙТИ). Имя-языка-1 определяется реализацией (см. ч. 6, п. 6.12.3, синтаксическое правило (1)).

GOST EXPERT ГОСТ 22558-89,

(39) Оператор SEARCH ALL (ИСКАТЬ ОСОБО). Начальная установка имени-индекса для идентификатора-1 игнорируется и установка имени-индекса меняется в ходе операции поиска способом, определяемым реализацией (см. ч. 6, п. 6.22.4, общее правило (4)).

(40) Оператор SET (УСТАНОВИТЬ). Каждая реализация определяет, на какие внешние переключатели может ссылаться оператор SET (УСТАНОВИТЬ) (см. ч. 6, п. 6.23.3, синтаксическое правило (5)).

(41) Оператор STOP литерал (ОСТАНОВИТЬ литерал). Продолжение функционирования единицы исполнения начинается со следующего выполнимого оператора в том случае, когда подключена зависящая от реализации процедура управления возобновлением единицы исполнения (см. ч. 6. п. 6.24.4, общее правило (4)).

(42) Оператор SUBTRACT (ОТНЯТЬ). Отведение достаточного поля памяти для выполнения вычислений без потери значащих цифр обеспечивается компилятором (см. ч. 6, п. 6.26.4, общее правило (4)).

(43) Состояние ввода-вывода. Если значение состояния ввода-вывода для операции ввода-вывода указывает критическое состояние ошибки, реализацией определяются действия, которые предпринимаются после выполнения любой применимой по оператору USE AFTER STANDARD EXCEPTION (ИСПОЛЬЗОВАТЬ ПОСЛЕ СТАНДАРТНОЙ ОШИБКИ) процедуры, или, если ни одна такая процедура не применима, после завершения стандартной системной обработки ошибок ввода-вывода (см. ч. 7, п. 1.3.5; ч. 8, п. 1.3.4; ч. 9, п. 1.3.4).

(44) Состояние ввода-вывода. Условие постоянной ошибки остается действующим на все последующие операции ввода-вывода файла до тех пор, пока не будут вызваны определенные реализацией средства для устранения условия постоянной ошибки (см. ч. 7, п. 1.3.5; ч. 8, п. 1.3.4; ч. 9, п. 1.3.4).

(45) Состояние ввода-вывода. Если применимо более одного значения, значение, которое помещается в состояние ввода-вывода, определяется реализацией (см. ч. 7, п. 1.3.5; ч. 8, п. 1.3.4; ч. 9, п. 1.3.4).

(46) Состояние ввода-вывода = 24. Сделана попытка занесения записей в относительный или индексный файл за пределами внешне определенных его границ. Способ определения границ устанавливается реализацией (см. ч. 8, п. 1.3.4; ч. 9, п. 1.3.4).

(47) Состояние ввода-вывода = 34. Сделана попытка занесения записи в последовательный файл за пределами внешне определенных границ файла. Способ определения границ устанавливается реализацией (см. ч. 7, п. 1.3.5).

(48) Состояние ввода-вывода = 9x. Значение 9 состояния ввода-вывода указывает на существование условий, определяемых реализацией. Значение x определяется реализацией (см. ч. 7, п. 1.3.5; ч. 8, п. 1.3.4; ч. 9, п. 1.3.4).

(49) Фраза ASSIGN (НАЗНАЧИТЬ). Значение и правила для допустимого содержимого имени-реализации-1 и значение литерала-1 определяются реализацией (см. ч. 7, п. 2.3.3, синтаксическое правило (3); ч. 9, п 2.3.3, синтаксическое правило (3); ч. 8, п. 2.3.3, синтаксическое правило (3): ч. 13, п. 2.3.3, синтаксическое правило (3)).

(50) Фраза ASSIGN (НАЗНАЧИТЬ). Правила корректности имени-реализации-1 и литерала-1 определяются реализацией (см. ч. 7, 8, 9, 13, п. 2.3.4, общее правило 1б).

(51) Фраза ASSIGN (НАЗНАЧИТЬ). Реализация определяет связь между файлом и запоминающей средой, на которую ссылается имя-реализации или литерал (см. ч. 7, п. 2.3.4, общее правило (3); ч. 8, п. 2.3.4 общее правило (4); ч. 9, п. 2.3.4, общее правило (5); ч. 13, п. 2.3.4, общее правило (3)).

(52) Фраза PADDING CHARACTER (ЛИТЕРА ЗАПОЛНИТЕЛЬ). Если фраза PADDING CHARACTER (ЛИТЕРА ЗАПОЛНИТЕЛЬ) не задана, значе-

GOST EXPERT ГОСТ 22558-89,

ние, используемое для литеры заполнителя, определяется реализацией (см. ч. 7, п. 2.7.4, общее правило (5)).

(53) Имя-реализации во фразе RECORD DELIMITER (ОГРАНИЧИТЕЛЬ ЗАПИСИ). Имя-реализации является системным-именем; поэтому правила образования имени-реализации определяются реализацией (см. ч. 7, п. 2.8.2).

(54) Фраза RECORD DELIMITER (ОГРАНИЧИТЕЛЬ ЗАПИСИ). Правила корректности имени-реализации во фразе RECORD DELIMITER (ОГРАНИЧИТЕЛЬ ЗАПИСИ) определяются реализацией (см. ч. 7, п. 2.3.4, общее правило 1в).

(55) Фраза RECORD DELIMITER (ОГРАНИЧИТЕЛЬ ЗАПИСИ).

Если задано имя-реализации-1, для определения длины записей переменной длины используется метод, определяемый реализацией (см. ч. 7, п. 2.8.4, общее правило (3)).

(56) Фраза RESERVE (РЕЗЕРВИРОВАТЬ). Если фраза RESERVE (РЕЗЕРВИРОВАТЬ) не указана, количество распределяемых областей ввода-вывода определяется реализацией (см. ч. 7, п. 2.9.3, общее правило (1)).

(57) Фраза LABEL RECORDS (МЕТКИ). Вариант STANDARD (СТАНДАРТНЫ), указывает, что существуют метки файла или устройства, назначенного файлу, и эти метки соответствуют спецификациям меток, определяемым реализацией (см. ч. 7, п. 3.6.3, общее правило (2)).

(58) Фраза LABEL RECORDS (МЕТКИ). Если фраза LABEL RECORDS (МЕТКИ) не указана для файла, метки для этого файла должны соответствовать спецификациям меток, определяемых реализацией (см. ч. 7, п. 3.6.3, общее правило (3)).

(59) Фраза VALUE (ЗНАЧЕНИЕ). Имя-реализации является системным-именем, поэтому правила образования имени-реализации определяются реализацией (см. ч. 7, п. 3.9.2).

(60) Оператор CLOSE (ЗАКРЫТЬ). Метки обрабатываются в соответствии со стандартной процедурой обработки меток, определенной реализацией. Выполняются операции закрытия, определенные реализацией (см. ч. 7, п. 4.2.4, общее правило 3В; ч. 8, п. 4.2.4, общее правило 2А; ч. 9, п. 4.2.4, общее правило 2А; ч. 13, п. 4.2.4, общее правило 3В).

(61) Оператор OPEN (ОТКРЫТЬ). Проверка или запись меток осуществляется в соответствии с процедурами, определяемыми реализацией для обработки входных или выходных меток (см. ч. 7, п. 4.3.4, общее правило (7); ч. 8, п. 4.4.4, общее правило (7); ч. 9, п. 4.4.4, общие правила (7) и (14); ч. 13, п. 4.5.4, общее правило 5а).

(62) Оператор WRITE (ПИСАТЬ). Если указано мнемоническое-имя-1, имя связывается с особым свойством, определяемым реализацией (см. ч. 7, п. 4.7.3, синтаксическое правило (6)).

(63) Оператор WRITE (ПИСАТЬ). Если указано мнемоническое имя-1, печатаемая страница продвигается в соответствии с правилами, установленными реализацией для данного устройства (см. ч. 7, п. 4.7.4, общее правило 15г);

(64) Оператор WRITE (ПИСАТЬ). Если указано PAGE (СТРАНИЦЫ) фразы ADVANCING (ПРОДВИЖЕНИЯ) и в статье описания, связанной с выводимой записью файла, не указана фраза LINAGE (ВЕРСТКА), позиционирование устройства на следующую физическую страницу выполняется в соответствии с правилами, определенными реализацией (см. ч. 7, п. 4.7.4, общее правило 15а).

(65) Катушка. Размеры катушки определяются реализацией (см. ч. 3).

(66) Том (unit). Размер тома определяется реализацией (см. ч. 3).

(67) Том (volume). Размер тома определяется реализацией.

GOST EXPERT ГОСТ 22558-89,

(68) Оператор CALL (ВЫЗВАТЬ). Если вызываемая программа не является программой Кобола, правила формирования имени-программы определяются реализацией (см. ч. 10, п. 5.2.4, общее правило (1)).

(69) Оператор CALL (ВЫЗВАТЬ). Ресурсы вычислительной среды времени выполнения, которые должны проверяться с целью установления доступности вызываемой программы для выполнения, определяются реализацией (см. ч. 10, п. 5.2.4, общее правило (3)).

(70) Оператор CALL (ВЫЗВАТЬ). Если программа, заданная в операторе CALL (ВЫЗВАТЬ), не может стать доступной для выполнения в это время и фраза ON OVERFLOW (ПРИ ПЕРЕПОЛНЕНИИ) или ON EXCEPTION (ПРИ ОШИБКЕ) не задана, все остальные действия оператора CALL (ВЫЗВАТЬ) определяются реализацией (см. ч. 10, п. 5.2.4, общее правило (3)).

(71) Оператор CALL (ВЫЗВАТЬ). Если вызываемая программа не является программой на Коболе, использование фразы USING (ИСПОЛЬЗУЯ) определяется реализацией (см. ч. 10, п. 5.2.4, общее правило (8)).

(72) Фраза SAME SORT/SORT-MERGE AREA (ОБЩАЯ ОБЛАСТЬ СОРТИРОВКИ/СОРТИРОВКИ СЛИЯНИЯ). Особенности размещения не сортируемых и не сливаемых файлов определяются реализацией (см. ч. 11, п. 2.5.4, общее правило 2.6).

(73) Структура записи файла отчетов. В генераторе отчетов структура логической записи файла, соответствующего имени-файла-1, определяется реализацией (см. ч. 13, п. 3.2.4, общее правило (2)).

(74) Символическое имя. Символическое имя коммуникационного термина должно соответствовать правилам образования системных-имен, поэтому образование символического имени определяется реализацией (см. ч. 14, п. 2.2.4, правило (10)).

(75) Ключ состояния коммуникации = 9x. Значение 9x ключа состояния коммуникации указывает на наличие условия, определяемого реализацией (см. ч. 14, п. 2.2.5).

(76) Ключ ошибки коммуникации. Значения от A до Z ключа ошибки коммуникации указывают на условия, определяемые реализацией (см. ч. 14, п. 2.2.6).

(77) Вариант KEY (КЛЮЧ) оператора DISABLE (ЗАПРЕТИТЬ). Параль встраивается в систему (см. ч. 14, п. 3.2.4, общее правило (7)).

(78) Вариант KEY (КЛЮЧ) оператора ENABLE (РАЗРЕШИТЬ). Параль встраивается в систему (см. ч. 14, п. 3.3.4, общее правило (6)).

(79) Оператор SEND (ПОСЛАТЬ). Если исполняется запрос мнемонического имени, имя идентифицируется специальным средством, определяемым реализацией (см. ч. 14, п. 3.3.3, общее правило (9)).

(80) Оператор SEND (ПОСЛАТЬ). Если указано мнемоническое имя-откл-1, символы, определяемые как коммуникационное свойство, передаются в соответствии с правилом, определяемым реализацией для этого устройства (см. ч. 14, п. 3.3.3, общее правило 15.a).

8. Список элементов языка, зависящих от оборудования

Ниже приводится список элементов языка настоящего стандарта, зависящих от оборудования.

(1) Замена двумя литерами зависит от набора литер, допустимого машиной (1 ЯДР).

(2) Фраза об использовании USAGE IS BINARY (ДВОИЧНОЕ) зависит от наличия подходящей машинной архитектуры для двоичного формата данных (1 ЯДР).

GOST EXPERT ГОСТ 22558-89,

(3) Фраза об использовании USAGE IS PACKED-DECIMAL (ДЕСЯТИЧНОЕ) зависит от наличия подходящей машинной архитектуры для упакованного десятичного формата данных (1 ЯДР).

(4) Если позиционирование не применимо на устройстве оборудования, операционная система игнорирует позиционирование при выполнении оператора DISPLAY (ВЫДАТЬ) (1 ЯДР).

(5) Фраза PADDING CHARACTER (ЛИТЕРА ЗАПОЛНИТЕЛЬ) зависит от того, применимы ли литеры заполнители к типу устройства, предназначенного файлу (2 ПОД, 1 ГОТ).

(6) Вариант STANDARD-1 (СТАНДАРТ-А) фразы RECORD DELIMITER (ОГРАНИЧИТЕЛЬ ЗАПИСИ) зависит от типа катушки, устройства (2 ПОД, 1 ГОТ).

(7) Фраза MULTIPLE FILE TAPE (НА ОДНОЙ КАТУШКЕ) зависит от типа катушки устройства (2 ПОД, 1 ГОТ).

(8) Фраза CODE-SET (АЛФАВИТ) зависит от того, может ли поддерживать устройство указанный код (1 ПОД, 1 ГОТ).

(9) Вариант REEL/UNIT (КАТУШКУ/ТОМ) оператора CLOSE (ЗАКРЫТЬ) зависит от типа (катушки или устройства массовой памяти) (1 ПОД, 1 ГОТ).

(10) Вариант FOR REMOVAL (С УДАЛЕНИЕМ) оператора CLOSE (ЗАКРЫТЬ) зависит от типа катушки или устройства массовой памяти (2 ПОД, 1 ГОТ).

(11) Вариант WITH NO REWIND (БЕЗ ПЕРЕМОТКИ) оператора CLOSE (ЗАКРЫТЬ) зависит от типа катушки или типа устройства массовой памяти (2 ПОД, 1 ГОТ).

(12) Оператор DELETE (УДАЛИТЬ) зависит от устройства массовой памяти (1 ОТД, 1 ИПД).

(13) Вариант I-O (ВХОДНОЙ-ВЫХОДНОЙ) оператора OPEN (ОТКРЫТЬ) зависит от типа устройства массовой памяти (1 ПОД, 1 ОТД, 1 ИПД).

(14) Вариант REVERSED (РЕВЕРСНО) оператора OPEN (ОТКРЫТЬ) зависит от типа катушки или устройства массовой памяти, имеющих возможность обеспечивать доступ к записям в обратном порядке (2 ПОД).

(15) Вариант WITH NO REWIND (БЕЗ ПЕРЕМОТКИ) оператора OPEN (ОТКРЫТЬ) зависит от типа катушки или устройства массовой памяти (2 ПОД, 1 ГОТ).

(16) Вариант EXTEND (ДОПОЛНЯЕМЫЙ) оператора OPEN (ОТКРЫТЬ) зависит от типа катушки или устройства массовой памяти (2 ПОД, 2 ОТД, 2 ИПД, 1 ГОТ).

(17) Оператор REWRITE (ОБНОВИТЬ) зависит от типа устройства массовой памяти (1 ПОД, 1 ОТД, 1 ИПД).

(18) Вариант I-O (ВХОДНЫХ-ВЫХОДНЫХ) оператора USE (ИСПОЛЬЗОВАТЬ) зависит от типа устройства массовой памяти (1 ПОД, 1 ОТД, 1 ИПД).

(19) Вариант BEFORE/AFTER (ДО/ПОСЛЕ) оператора WRITE (ПИСАТЬ) зависит от возможности вертикального позиционирования устройства или от действия, базирующегося на мнемоническом-имени (1 ПОД).

(20) Вариант BEFORE/AFTER ADVANCING (ДО/ПОСЛЕ ПРОДВИЖЕНИЯ) оператора SEND (ПОСЛАТЬ) зависит от возможности вертикального позиционирования устройства или от действия, базирующегося на мнемоническом-имени (1 КОМ).

4. Список неопределенных элементов языка

Ниже приводится список не определенных явно элементов языка настоящего стандарта.

(1) Явные и неявные передачи управления. Когда нет следующего выполнимого оператора и управление не передается за пределы Кобол-программы, передача управления в программе не определена, если только выполнение программы не происходит в недекларативной части процедур программы, вызванной

GOST EXPERT ГОСТ 22558-89,

оператором CALL (ВЫЗВАТЬ), где выполняется неявный оператор EXIT PROGRAM (ВЫЙТИ ИЗ ПРОГРАММЫ) (см. ч. 4, п. 4.4.2).

(2) Начальные значения данных. Начальное значение любого индексного данного или данного, не связанного с фразой VALUE (ЗНАЧЕНИЕ), не определено (см. ч. 6, п. 5.2.4).

(3) Вариант DEPENDING ON (В ЗАВИСИМОСТИ ОТ) фразы OCCURS (ПОВТОРЯЕТСЯ). Значения данных, номера вхождений которых превышают значение данного, на которое ссылается имя-данного-1, не определены (см. ч. 6, п. 5.8.4, общее правило 2 б).

(4) Фраза VALUE (ЗНАЧЕНИЕ) в секции файлов. Начальные значения данных в секции файлов не определено (см. ч. 6, п. 2.15.6, правило 1 а).

(5) Фраза VALUE (ЗНАЧЕНИЕ) в секции рабочей памяти и секции коммуникаций. Если фраза VALUE (ЗНАЧЕНИЕ) не указана для данного в секции рабочей памяти или в секции коммуникаций, значение этого данного не определено (см. ч. 6, п. 5.15.6, правило 1 в).

(6) Вариант ON SIZE ERROR (ПРИ ПЕРЕПОЛНЕНИИ). Если фраза ON SIZE ERROR (ПРИ ПЕРЕПОЛНЕНИИ) не задана и во время выполнения арифметической операции в арифметическом операторе возникает условие переполнения, значения результирующих идентификаторов не определены (см. ч. 6, п. 6.4.2).

(7) Перекрывающиеся операнды. Если посылаемое и принимающее данное в любом операторе имеют общую часть или всю область памяти, то даже если они не определены в одной статье описания данного, результат выполнения такого оператора не определен (см. ч. 6, п. 6.4.5).

(8) Несовместимые данные. Если значения данных, к которым происходит обращение в разделе процедур, не соответствуют классу, определенному при описании этих данных с фразой PICTURE (ШАБЛОН), результат обращения не определен. Исключение составляет условие класса (см. ч. 6, п. 6.4.7).

(9) Оператор SEARCH ALL (ИСКАТЬ ОСОБО). В операторе SEARCH ALL (ИСКАТЬ ОСОБО) результат операции поиска предсказуем только в следующих случаях: (а) данные в таблице упорядочены так же, как это описано в варианте KEY IS (ПО ВОЗРАСТАНИЮ/УБЫВАНИЮ КЛЮЧА) фразы OCCURS (ПОВТОРЯЕТСЯ), связанном с идентификатором-1 и (б) значения ключа (ключей), упомянутого во фразе WHEN (КОГДА), достаточны, чтобы однозначно идентифицировать элемент таблицы (см. ч. 6, п. 6.22.4, общее правило (3)).

(10) Оператор SEARCH ALL (ИСКАТЬ ОСОБО). Если для каждой установки индекса в разрешенном интервале ни одно условие, задаваемое во фразе WHEN (КОГДА), не может быть удовлетворено, то управление передается повелительному-оператору-1 фразы AT END (В КОНЦЕ), если она указана, или на конец оператора SEARCH (ИСКАТЬ), если эта фраза не указана. В любом случае конечная установка индекса не определена (см. ч. 6, п. 6.22.4, общее правило (4)).

(11) Оператор CLOSE (ЗАКРЫТЬ). Действия оператора CLOSE (ЗАКРЫТЬ) не определены, если метки специфицированы, но в файле отсутствуют, или когда они не специфицированы, но присутствуют (см. ч. 7, п. 4.2.4, общее правило 3 В; ч. 8; ч. 9, п. 4.2.4, общее правило 2А; ч. 13, п. 4.2.4, общее правило 3В).

(12) Оператор CLOSE (ЗАКРЫТЬ). В случае неуспешного выполнения оператора CLOSE (ЗАКРЫТЬ) доступность области записи не определена (см. ч. 7, п. 4.2.4, общее правило (6); ч. 8 и 9, п. 4.2.4, общее правило (5)).

(13) Оператор OPEN (ОТКРЫТЬ). Действия оператора OPEN (ОТКРЫТЬ) не определены, когда метки специфицированы, но отсутствуют, или не специфицированы, но присутствуют (см. ч. 7, п. 4.3.4, общее правило (7); ч. 8 и 9, п. 4.4.4, общее правило (7); ч. 13, п. 4.5.4, общее правило (5)).

GOST EXPERT ГОСТ 22558-89,

(14) Оператор READ (ЧИТАТЬ). По завершении оператора READ (ЧИТАТЬ) значения всех данных, находящихся вне диапазона текущей записи данных, не определены (см. ч. 7, п. 4.4.4, общее правило (6); ч. 8 и 9, п. 4.5.4, общее правило (6)).

(15) Оператор READ (ЧИТАТЬ). После неуспешного выполнения оператора READ (ЧИТАТЬ) содержимое соответствующей области записи не определено; для индексных файлов ключ ссылки также не определен (см. ч. 7, п. 4.4.4, общее правило (12); ч. 8 и 9, п. 4.5.4, общее правило (12)).

(16) Оператор START (ПОДВЕСТИ). После неуспешного выполнения оператора START (ПОДВЕСТИ) для индексного файла ключ ссылки для этого файла не определен (см. ч. 9, п. 4.7.4, общее правило (8)).

(17) Оператор WRITE (ПИСАТЬ) (последовательный файл). Если декларатива USE AFTER STANDARD EXCEPTION (ИСПОЛЬЗОВАТЬ ПОСЛЕ ПРОЦЕДУРЫ ОШИБКИ) не специфицирована явно или неявно для имени-файла, связанного с именем-записи-1, результат не определен при попытке записать запись за внешне определенными границами файла (см. ч. 7, п. 4.7.4, общее правило 13в).

(18) Вариант ADVANCING (ПРОДВИЖЕНИЯ) оператора WRITE (ПИСАТЬ). Если значение данного, указанного идентификатором-2, отрицательно, при использовании варианта ADVANCING (ПРОДВИЖЕНИЯ) результат не определен (см. ч. 17, п. 4.7.4, общее правило 15 б).

(19) Оператор CALL (ВЫЗВАТЬ) для программ, записанных не на Коболе. Оператор CALL (ВЫЗВАТЬ) может использоваться для вызова программ, записанных в языке, отличном от Кобола, но механизм возврата и передачи межпрограммных данных в настоящем документе не определяются (см. ч. 2, п. 6.4.1).

(20) Секция связи. Если к данному секции связи обращаются в программе, которая не является вызываемой, результат не определен (см. ч. 10, п. 4.1).

(21) Оператор MERGE (СЛИТЬ). Если записи файлов, представленных именем-файла-2 и именем-файла-3, не упорядочены в соответствии с фразами ASCENDING (ПО ВОЗРАСТАНИЮ КЛЮЧА) или DESCENDING (ПО УБЫВАНИЮ КЛЮЧА) оператора MERGE (СЛИТЬ), результат слияния не определен (см. ч. 11, п. 4.1.4, общее правило (6)).

(22) Оператор RETURN (ВЕРНУТЬ). При возникновении условия «в конце» выполнение оператора RETURN (ВЕРНУТЬ) считается неуспешным и содержимое области записи, соответствующей имени-файла-1, не определено (см. ч. 11, п. 4.3.4, общее правило (2)).

(23) Оператор SORT (СОРТИРОВАТЬ). Если фраза DUPLICATES (С ДУБЛИРОВАНИЕМ) не указана и содержимое всех ключей, связанных с одной записью данных, равно содержимому соответствующих ключей, связанных с одной или несколькими другими записями данных, то порядок возвращения этих записей не определен (см. ч. 11, п. 4.4.4, общее правило (4)).

(24) Оператор SORT (СОРТИРОВАТЬ). Для файла с относительной организацией, указанного именем-файла-2 во фразе GIVING (ПОЛУЧАЯ), значение данного, являющегося относительным ключом, после выполнения оператора SORT (СОРТИРОВАТЬ) не определено (см. ч. 11, п. 4.4.4, общее правило 9 б).

(25) Статья описания коммуникации. Если система управления сообщениями делает попытку диспетчировать программу, не имеющую фразу INITIAL (НАЧАЛЬНАЯ), результаты не определены (см. ч. 14, п. 2.2.4. общее правило (7)).

(26) Оператор SEND (ПОСЛАТЬ). При наличии в значении идентификатора-1 специальных символов управления результат выполнения оператора не определен (см. ч. 14, п. 3.6.4, общее правило (5)).

GOST EXPERT ГОСТ 22558-89,

(27) Оператор SEND (ПОСЛАТЬ). Во время выполнения единицы исполнения расположение части сообщения, которая не заканчивается EM1 (ИКЩ) или EG1 (ИКГ), или не уничтожена выполнением оператора PURGE (ОЧИСТИТЬ), не определено (см. ч. 14, п. 3.6.4, общее правило (7)).

(28) Оператор SEND (ПОСЛАТЬ). Если значение данного, указанного идентификатором-3, отрицательно, результат не определен (см. ч. 14, п. 3.6.4, общее правило 15 б).

ИНФОРМАЦИОННЫЕ ДАННЫЕ

1. **РАЗРАБОТАН И ВНЕСЕН Академией наук УССР**

   РАЗРАБОТЧИКИ

   Е. Л. Ющенко, член-корр. АН УССР (руководитель темы); **Л. П. Бабенко**, канд. физ.-мат. наук; **Г. А. Карпенко; Н. К. Лищитович; Л. А. Мельник; М. Р. Тарановский; Г. В. Пеледов; А. С. Марков; А. А. Севастюк; Л. М. Романовская; Л. К. Загузова**

2. **УТВЕРЖДЕН И ВВЕДЕН В ДЕЙСТВИЕ Постановлением Государственного комитета СССР по стандартам от 20.12.89 № 3894**

3. **Срок первой проверки 1996 г.; периодичность проверки — 5 лет.**

4. **Стандарт полностью соответствует СТ СЭВ 6184—88**

5. **Стандарт полностью соответствует международному стандарту ИСО 1989—85**

6. **ВЗАМЕН ГОСТ 22558—77**

7. **ССЫЛОЧНЫЕ НОРМАТИВНО-ТЕХНИЧЕСКИЕ ДОКУМЕНТЫ**

Обозначение НТД, на который дана ссылка	Номер приложения
ГОСТ 22558—77	Приложение 1

GOST EXPERT ГОСТ 22558-89,

СОДЕРЖАНИЕ

Часть 1. **ОСНОВНЫЕ ПОЛОЖЕНИЯ** 1
1. Введение к стандарту 1
1.1. Область действия и назначение 1
1.2. Структура спецификаций языка 1
1.3. Структура документа 4
1.4. Рекомендации по использованию текста стандарта . . . 4
1.5. Определение реализации стандарта языка Кобол . . . 6
1.6. Соответствие исходной программы стандарту 10
1.7. Сочетание соответствующей стандарту программы и соответствующей стандарту реализации 10
2. Список элементов языка по модулям 10
2.1. Общее описание 10
2.2. Список элементов в модуле ядра 11
2.3. Список элементов в модуле последовательного ввода-вывода . 20
2.4. Список элементов в модуле относительного ввода-вывода . . 23
2.5. Список элементов в модуле индексного ввода-вывода . . . 25
2.6. Список элементов в модуле межпрограммных связей . . . 28
2.7. Список элементов в модуле сортировки-слияния 30
2.8. Список элементов в модуле обработки исходных текстов . . 31
2.9. Список элементов в модуле генератора отчетов 31
2.10. Список элементов в модуле коммуникаций 34
2.11. Список элементов в модуле отладки 36
2.12. Список элементов в модуле сегментации 36
3. Список элементов по разделам Кобола 37
3.1. Общее описание 37
3.2. Список элементов понятий языка 37
3.3. Список элементов раздела идентификации 40
3.4. Список элементов раздела оборудования 40
3.5. Список элементов раздела данных 43
3.6. Список элементов раздела процедур 47
Часть 2. **КОНЦЕПЦИИ И СРЕДСТВА ЯЗЫКА** 58
1. Введение 58
2. Файлы 58
2.1. Свойства файла 58
2.2. Обработка файлов 61
3. Генератор отчетов 67
3.1. Секция отчетов 67
3.2. Структура отчета 68
3.3. Операторы раздела процедур генератора отчетов 70
4. Обработка таблиц 71
4.1. Определение таблиц 73
4.2. Начальные значения таблиц 73
4.3. Ссылки на табличные элементы 73
4.4. Индексирование 74
5. Общая область памяти 77
6. Организация программ и программных связей 77
6.1. Понятие программы и единицы исполнения 78
6.2. Доступные данные и файлы 79
6.3. Классы программ 82
6.4. Межпрограммные связи 83
6.5. Внутрипрограммные связи 87
6.6. Сегментация 88
7. Средства коммуникации 89
7.1. Система управления сообщениями 89

GOST EXPERT ГОСТ 22558-89,

7.2. Объектная программа Кобола 90
7.3. Связь программы на Коболе с системой управления сообщениями и коммуникационными устройствами 90
7.4. Понятие сообщений и сегментов сообщения 93
7.5. Понятие очередей 94
7.6. Понятие коммуникации транзакций 97
Часть 3. ГЛОССАРИЙ (СЛОВАРЬ ТЕРМИНОВ КОБОЛА) . . . 98
1. Введение 98
2. Определения 98
Часть 4. ОСНОВНЫЕ ПОНЯТИЯ 127
1. Введение 127
2. Обозначения, используемые в форматах и правилах . . . 127
3. Правила 129
4. Понятия языка 130
4.1. Набор литер 130
4.2. Структура языка 130
4.3. Понятие машинно-независимого описания данного . . . 139
4.4. Явные и неявные спецификации 152
4.5. Внешний переключатель 155
5. Исходная Кобол-программа 156
5.1. Введение 156
5.2. Организация 156
5.3. Структура 156
6. Разделы 156
6.1. Раздел идентификации 156
6.2. Раздел оборудования 157
6.3. Раздел данных 158
6.4. Раздел процедур 160
7. Формат представления 166
7.1. Общее описание 166
7.2. Описание формата представления 166
7.3. Форматы раздела, секции, параграфа 168
7.4. Статьи раздела данных 169
7.5. Декларативы 170
7.6. Заголовок конца программы 170
8. Зарезервированные слова Кобола 170
Часть 5. ФОРМАТЫ ЯЗЫКА 179
1. Сводка форматов английской нотации 179
2. Сводка форматов русской нотации 211
Часть 6. ЯДРО 244
1. Введение в модуль ядра 244
1.1. Назначение 244
1.2. Характеристика уровней 245
1.3. Ограничения по уровням, касающиеся общих характеристик языка 245
2. Исходная программа на Коболе 246
2.1. Общее описание 246
2.2. Организация 246
2.3. Структура 247
2.4. Заголовок конца программы 247
3. Раздел идентификации в ядре 248
3.1. Общее описание 248
3.2. Организация 249
3.3. Параграф PROGRAM-ID (ПРОГРАММА) 249
3.4. Параграф DATE-COMPILED (ДАТА-ТРАНСЛЯЦИИ) . . . 249
4. Раздел оборудования в ядре 250

GOST EXPERT ГОСТ 22558-89,

4.1. Общее описание 250
4.2. Секция конфигурации 250
4.3. Параграф SOURCE-COMPUTER (ИСХОДНАЯ-МАШИНА) . 251
4.4. Параграф OBJECT-COMPUTER (РАБОЧАЯ-МАШИНА) . . 252
4.5. Параграф SPECIAL-NAMES (СПЕЦИАЛЬНЫЕ-ИМЕНА) . . 253
5. Раздел данных в ядре 261
5.1. Общее описание 261
5.2. Секция рабочей памяти 261
5.3. Статья описания данного 262
5.4. Фраза BLANK WHEN ZERO (ПРОБЕЛ КОГДА НУЛЬ) . . 266
5.5. Фраза имя-данного или FILLER (ЗАПОЛНИТЕЛЬ) . . 267
5.6. Фраза JUSTIFIED (СДВИНУТО) 267
5.7. Номер-уровня 268
5.8. Фраза OCCURS (ПОВТОРЯЕТСЯ) 269
5.9. Фраза PICTURE (ШАБЛОН) 278
5.10. Фраза REDEFINES (ПЕРЕОПРЕДЕЛЯЕТ) 281
5.11. Фраза RENAMES (ПЕРЕИМЕНОВЫВАЕТ) 285
5.12. Фраза SIGN (ЗНАК) 286
5.13. Фраза SYNCHRONIZED (ВЫДЕЛЕНО) 287
5.14. Фраза об использовании 289
5.15. Фраза VALUE (ЗНАЧЕНИЕ) 292
6. Раздел процедур в ядре 294
6.1. Общее описание 294
6.2. Арифметические выражения 295
6.3. Условные выражения 298
6.4. Общие фразы и правила для форматов операторов . . 309
6.5. Оператор ACCEPT (ПРИНЯТЬ) 314
6.6. Оператор ADD (СЛОЖИТЬ) 316
6.7. Оператор ALTER (ИЗМЕНИТЬ) 319
6.8. Оператор COMPUTE (ВЫЧИСЛИТЬ) 320
6.9. Оператор CONTINUE (ПРОДОЛЖИТЬ) 321
6.10. Оператор DISPLAY (ВЫДАТЬ) 321
6.11. Оператор DIVIDE (РАЗДЕЛИТЬ) 323
6.12. Оператор ENTER (ВОЙТИ) 327
6.13. Оператор EVALUATE (ОЦЕНИТЬ) 327
6.14. Оператор EXIT (ВЫЙТИ) 333
6.15. Оператор GO TO (ПЕРЕЙТИ) 333
6.16. Оператор IF (ЕСЛИ) 334
6.17. Оператор INITIALIZE (ИНИЦИИРОВАТЬ) 336
6.18. Оператор INSPECT (ПРОСМОТРЕТЬ) 338
6.19. Оператор MOVE (ПОМЕСТИТЬ) 352
6.20. Оператор MULTIPLY (УМНОЖИТЬ) 357
6.21. Оператор PERFORM (ВЫПОЛНИТЬ) 358
6.22. Оператор SEARCH (ИСКАТЬ) 373
6.23. Оператор SET (УСТАНОВИТЬ) 378
6.24. Оператор STOP (ОСТАНОВИТЬ) 383
6.25. Оператор STRING (СОБРАТЬ) 384
6.26. Оператор SUBTRACT (ОТНЯТЬ) 387
6.27. Оператор UNSTRING (РАЗОБРАТЬ) 390
7. Отладка в ядре 395
7.1. Общее описание 395
7.2. Переключатель времени компиляции 395
7.3. Отладочные строки 396
Часть 7. **МОДУЛЬ ПОСЛЕДОВАТЕЛЬНОГО ВВОДА-ВЫВОДА** . 396
1. Введение в модуль последовательного ввода-вывода . . . 396
1.1. Назначение 396
1.2. Характеристика уровней 396

GOST EXPERT ГОСТ 22558-89,

1.3. Понятия языка . . . 397
2. Раздел оборудования в модуле последовательного ввода-вывода . . . 402
2.1. Секция ввода-вывода . . . 402
2.2. Параграф FILE-CONTROL (УПРАВЛЕНИЕ-ФАЙЛАМИ) . . . 402
2.3. Статья управления файлом . . . 403
2.4. Фраза ACCESS MODE (ДОСТУП) . . . 405
2.5. Фраза FILE STATUS (СОСТОЯНИЕ ФАЙЛА) . . . 405
2.6. Фраза ORGANIZATION IS SEQUENTIAL (ОРГАНИЗАЦИЯ ПОСЛЕДОВАТЕЛЬНАЯ) . . . 406
2.7. Фраза PADDING CHARACTER (ЛИТЕРА ЗАПОЛНИТЕЛЬ) . . . 406
2.8. Фраза RECORD DELIMITER (ОГРАНИЧИТЕЛЬ ЗАПИСИ) . . . 407
2.9. Фраза RESERVE (РЕЗЕРВИРОВАТЬ) . . . 408
2.10. Параграф I-O-CONTROL (УПРАВЛЕНИЕ-ВВОДОМ-ВЫВОДОМ) . . . 409
2.11. Фраза MULTIPLE FILE TAPE (НА ОДНОЙ КАТУШКЕ) . . . 410
2.12. Фраза RERUN (ПЕРЕПРОГОН) . . . 411
2.13. Фраза SAME AREA (ОБЩАЯ ОБЛАСТЬ) . . . 413
3 Раздел данных в модуле последовательного ввода-вывода . . . 415
3.1. Секция файлов . . . 415
3.2. Статья описания файла . . . 416
3.3. Фраза BLOCK CONTAINS (В БЛОКЕ) . . . 418
3.4. Фраза CODE-SET (АЛФАВИТ) . . . 419
3.5. Фраза DATA RECORDS (ЗАПИСИ ДАННЫХ) . . . 420
3.6. Фраза LABEL RECORDS (МЕТКИ) . . . 420
3.7. Фраза LINAGE (ВЕРСТКА) . . . 421
3.8. Фраза RECORD (В ЗАПИСИ) . . . 424
3.9. Фраза VALUE (ЗНАЧЕНИЕ) . . . 427
4. Раздел процедур в модуле последовательного ввода-вывода . . . 428
4.1. Общее описание . . . 428
4.2. Оператор CLOSE (ЗАКРЫТЬ) . . . 429
4.3. Оператор OPEN (ОТКРЫТЬ) . . . 433
4.4. Оператор READ (ЧИТАТЬ) . . . 439
4.5. Оператор REWRITE (ОБНОВИТЬ) . . . 443
4.6. Оператор USE (ИСПОЛЬЗОВАТЬ) . . . 444
4.7. Оператор WRITE (ПИСАТЬ) . . . 447
Часть 8. **МОДУЛЬ ОТНОСИТЕЛЬНОГО ВВОДА-ВЫВОДА** . . . 453
1. Введение в модуль относительного ввода вывода . . . 453
1.1. Назначение . . . 453
1.2. Характеристика уровней . . . 453
1.3. Понятия языка . . . 460
2. Раздел оборудования в модуле относительного ввода-вывода . . . 460
2.1. Секция ввода-вывода . . . 460
2.2. Параграф FILE-CONTROL (УПРАВЛЕНИЕ-ФАЙЛАМИ) . . . 460
2.3. Статья управления файлом . . . 463
2.4. Фраза ACCESS MODE (ДОСТУП) . . . 464
2.5. Фраза ORGANIZATION IS RELATIVE (ОРГАНИЗАЦИЯ ОТНОСИТЕЛЬНАЯ) . . . 465
2.6. Параграф I-O-CONTROL (УПРАВЛЕНИЕ-ВВОДОМ-ВЫВОДОМ) . . . 466
3. Раздел данных в модуле относительного ввода-вывода . . . 466
3.1. Секция файлов . . . 466
3.2. Статья описания файла . . . 468
4. Раздел процедур в модуле относительного ввода-вывода . . . 468
4.1. Общее описание . . . 468
4.2. Оператор CLOSE (ЗАКРЫТЬ) . . . 470
4.3. Оператор DELETE (УДАЛИТЬ) . . . 471
4.4. Оператор OPEN (ОТКРЫТЬ) . . . 476
4.5. Оператор READ (ЧИТАТЬ)

GOST EXPERT ГОСТ 22558-89,

4.6. Оператор REWRITE (ОБНОВИТЬ) 481
4.7. Оператор START (ПОДВЕСТИ) 484
4.8. Оператор USE (ИСПОЛЬЗОВАТЬ) 486
4.9. Оператор WRITE (ПИСАТЬ) 488

Часть 9. **МОДУЛЬ ИНДЕКСНОГО ВВОДА-ВЫВОДА** 492

1. Введение в модуль индексного ввода-вывода 492
1.1. Назначение 492
1.2. Характеристика уровней 492
1.3. Понятия языка 492
2. Раздел оборудования в модуле индексного ввода-вывода . . . 500
2.1. Секция ввода-вывода 500
2.2. Параграф FILE-CONTROL (УПРАВЛЕНИЕ-ФАЙЛАМИ) . . 500
2.3. Статья управления файлом 500
2.4. Фраза ACCESS MODE (ДОСТУП) 502
2.5. Фраза ALTERNATE RECORD KEY (ДОПОЛНИТЕЛЬНЫЙ КЛЮЧ ЗАПИСИ) 503
2.6. Фраза ORGANIZATION IS INDEXED (ОРГАНИЗАЦИЯ ИНДЕКСНАЯ) 505
2.7. Фраза RECORD KEY (КЛЮЧ ЗАПИСИ) 505
2.8. Параграф I-O-CONTROL (УПРАВЛЕНИЕ-ВВОДОМ-ВЫВОДОМ) 506
3. Раздел данных в модуле индексного ввода-вывода 507
3.1. Секция файлов 507
3.2. Статья описания файла 507
4. Раздел процедур в модуле индексного ввода-вывода 509
4.1. Общее описание 509
4.2. Оператор CLOSE (ЗАКРЫТЬ) 510
4.3. Оператор DELETE (УДАЛИТЬ) 512
4.4. Оператор OPEN (ОТКРЫТЬ) 513
4.5. Оператор READ (ЧИТАТЬ) 517
4.6. Оператор REWRITE (ОБНОВИТЬ) 523
4.7. Оператор START (ПОДВЕСТИ) 526
4.8. Оператор USE (ИСПОЛЬЗОВАТЬ) 529
4.9. Оператор WRITE (ПИСАТЬ) 532

Часть 10. **МОДУЛЬ МЕЖПРОГРАММНЫХ СВЯЗЕЙ** 536

1. Введение в модуль межпрограммных связей 536
1.1. Назначение 536
1.2. Характеристика уровней 536
1.3. Понятия языка 536
2. Вложенные исходные программы 543
2.1. Общее описание 543
2.2. Организация 543
2.3. Структура 544
2.4. Начальное состояние программы 544
2.5. Заголовок конца программы 545
3. Раздел идентификации в модуле межпрограммных связей . . . 546
3.1. Параграф PROGRAM-ID (ПРОГРАММА) и вложенные исходные программы 546
4. Раздел данных в модуле межпрограммных связей 547
4.1. Секция связи 547
4.2. Статья описания файла в модуле межпрограммных связей . . 548
4.3. Статья описания данного в модуле межпрограммных связей . 553
4.4. Статья описания отчета в модуле межпрограммных связей . . 556
4.5. Фраза EXTERNAL (ВНЕШНЕЕ) 557
4.6. Фраза GLOBAL (ГЛОБАЛЬНОЕ) 558
5. Раздел процедур в модуле межпрограммных связей 559
5.1. Заголовок раздела процедур 559

GOST EXPERT ГОСТ 22558-89,

5.2. Оператор CALL (ВЫЗВАТЬ) 560
5.3. Оператор CANCEL (ОСВОБОДИТЬ) 566
5.4. Оператор EXIT PROGRAM (ВЫЙТИ ИЗ ПРОГРАММЫ) . . 567
5.5. Оператор USE (ИСПОЛЬЗОВАТЬ) 568
5.6. Оператор USE BEFORE REPORTING (ИСПОЛЬЗОВАТЬ ПЕРЕД ВЫДАЧЕЙ) 569

Часть 11. **МОДУЛЬ СОРТИРОВКИ-СЛИЯНИЯ** 570

1. Введение в модуль сортировки-слияния 570
1.1. Назначение 570
1.2. Понятия языка 570
2. Раздел оборудования в модуле сортировки-слияния 571
2.1. Секция ввода-вывода 571
2.2. Параграф FILE-CONTROL (УПРАВЛЕНИЕ-ФАЙЛАМИ) . . 571
2.3. Статья управления файлом 571
2.4. Параграф I-O-CONTROL (УПРАВЛЕНИЕ-ВВОДОМ-ВЫВОДОМ) 572
2.5. Фразы SAME RECORD AREA (ОБЩАЯ ОБЛАСТЬ ЗАПИСИ), SAME SORT/SORT-MERGE AREA (ОБЩАЯ ОБЛАСТЬ СОРТИРОВКИ/СОРТИРОВКИ-СЛИЯНИЯ) 573
3. Раздел данных в модуле сортировки слияния 575
3.1. Секция файлов 575
3.2. Статья описания сортируемого-сливаемого файла 576
4. Раздел процедур в модуле сортировки-слияния 577
4.1. Оператор MERGE (СЛИТЬ) 577
4.2. Оператор RELEASE (ПЕРЕДАТЬ) 583
4.3. Оператор RETURN (ВЕРНУТЬ) 584
4.4. Оператор SORT (СОРТИРОВАТЬ) 586

Часть 12. **МОДУЛЬ ОБРАБОТКИ ИСХОДНЫХ ТЕКСТОВ** . . . 593

1. Введение в модуль обработки исходных текстов 593
2. Оператор COPY (КОПИРОВАТЬ) 593
3. Оператор REPLACE (ЗАМЕНИТЬ) 599

Часть 13. **МОДУЛЬ ГЕНЕРАТОРА ОТЧЕТОВ** 602

1. Введение в модуль генератора отчетов 602
1.1. Назначение 602
1.2. Понятия языка 602
2. Раздел оборудования в модуле генератора отчетов 603
2.1. Секция ввода-вывода 603
2.2. Параграф FILE CONTROL (УПРАВЛЕНИЕ-ФАЙЛАМИ) . . 603
2.3. Статья управления файлом 603
2.4. Параграф I-O-CONTROL (УПРАВЛЕНИЕ-ВВОДОМ-ВЫВОДОМ) 605
3. Раздел данных в модуле генератора отчетов 606
3.1. Секция файлов 606
3.2. Статья описания файла 607
3.3. Фраза REPORT (ОТЧЕТ) 608
3.4. Секция отчетов 609
3.5. Статья описания отчета 610
3.6. Фраза CODE (С КОДОМ) 612
3.7. Фраза CONTROL (УПРАВЛЕНИЕ) 613
3.8. Фраза PAGE (РАЗМЕР СТРАНИЦЫ) 614
3.9. Статья описания группы отчета 617
3.10. Таблицы правил представления 623
3.11. Фраза COLUMN NUMBER (НОМЕР СТОЛБЦА) . . . 640
3.12. Фраза «имя-данного» 640
3.13. Фраза GROUP INDICATE (ОПРЕДЕЛЯЕТ ГРУППУ) . . 641
3.14. Номер-уровня 642
3.15. Фраза LINE NUMBER (НОМЕР СТРОКИ) 642

GOST EXPERT ГОСТ 22558-89,

3.16. Фраза NEXT GROUP (СЛЕДУЮЩАЯ ГРУППА) . . . 644
3.17. Фраза SIGN (ЗНАК) . . . 645
3.18. Фраза SOURCE (ИСТОЧНИК) . . . 646
3.19. Фраза SUM (СУММА) . . . 647
3.20. Фраза TYPE (ТИП) . . . 650
3.21. Фраза USAGE (об использовании) . . . 657
3.22. Фраза VALUE (ЗНАЧЕНИЕ) . . . 657
4. Раздел процедур в модуле генератора отчетов . . . 658
4.1. Общее описание . . . 658
4.2. Оператор CLOSE (ЗАКРЫТЬ) . . . 659
4.3. Оператор GENERATE (ГЕНЕРИРОВАТЬ) . . . 662
4.4. Оператор INITIATE (НАЧАТЬ) . . . 665
4.5. Оператор OPEN (ОТКРЫТЬ) . . . 665
4.6. Оператор SUPPRESS (ПОДАВИТЬ) . . . 669
4.7. Оператор TERMINATE (ЗАКОНЧИТЬ) . . . 669
4.8. Оператор USE AFTER STANDARD EXCEPTION PROCEDURE (ИСПОЛЬЗОВАТЬ ПОСЛЕ СТАНДАРТНОЙ ПРОЦЕДУРЫ ОШИБКИ) . . . 671
4.9. Оператор USE BEFORE REPORTING (ИСПОЛЬЗОВАТЬ ДО ВЫДАЧИ) . . . 673
Часть 14. **МОДУЛЬ КОММУНИКАЦИЙ** . . . 674
1. Введение в модуль коммуникаций . . . 674
1.1. Назначение . . . 674
1.2. Характеристика уровней . . . 674
2. Раздел данных в модуле коммуникаций . . . 675
2.1. Секция коммуникаций . . . 675
2.2. Статья описания коммуникации . . . 676
3. Раздел процедур в модуле коммуникации . . . 697
3.1. Оператор ACCEPT MESSAGE COUNT (ПРИНЯТЬ ЧИСЛО СООБЩЕНИЙ) . . . 697
3.2. Оператор DISABLE (ЗАПРЕТИТЬ) . . . 697
3.3. Оператор ENABLE (РАЗРЕШИТЬ) . . . 699
3.4. Оператор PURGE (ОЧИСТИТЬ) . . . 701
3.5. Оператор RECEIVE (ПОЛУЧИТЬ) . . . 702
3.6. Оператор SEND (ПОСЛАТЬ) . . . 705
Часть 15. **МОДУЛЬ ОТЛАДКИ** . . . 711
1. Введение в модуль отладки . . . 711
1.1. Назначение . . . 711
1.2. Характеристика уровней . . . 711
1.3. Понятия языка . . . 711
2. Раздел оборудования в модуле отладки . . . 712
2.1. Фраза WITH DEBUGGING MODE (В РЕЖИМЕ ОТЛАДКИ) . . . 712
3. Раздел процедур в модуле отладки . . . 713
3.1. Общее описание . . . 713
3.2. Оператор USE FOR DEBUGGING (ИСПОЛЬЗОВАТЬ ДЛЯ ОТЛАДКИ) . . . 713
Часть 16. **МОДУЛЬ СЕГМЕНТАЦИИ** . . . 721
1. Введение в модуль сегментации . . . 721
1.1. Назначение . . . 721
1.2. Характеристика уровней . . . 721
1.3. Область действия . . . 721
1.4. Организация . . . 721
1.5. Классификация сегментации . . . 723
1.6. Управление сегментацией . . . 723
2. Раздел оборудования в модуле сегментации . . . 723
2.1. Секция конфигурации . . . 723

GOST EXPERT ГОСТ 22558-89,