ГОСТ Р ИСО/МЭК 8824—93

ГОСУДАРСТВЕННЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

ИНФОРМАЦИОННАЯ ТЕХНОЛОГИЯ

ВЗАИМОСВЯЗЬ ОТКРЫТЫХ СИСТЕМ СПЕЦИФИКАЦИЯ АБСТРАКТНО-СИНТАКСИЧЕСКОЙ НОТАЦИИ ВЕРСИИ ОДИН (АСН.1)

БЗ 12—92/1179 7600

ГОССТАНДАРТ РОССИИ
Москва

gost.e-ecolog.ru

ГОСТ Р ИСО/МЭК 8824—93

ГОСУДАРСТВЕННЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

ИНФОРМАЦИОННАЯ ТЕХНОЛОГИЯ

ВЗАИМОСВЯЗЬ ОТКРЫТЫХ СИСТЕМ СПЕЦИФИКАЦИЯ АБСТРАКТНО-СИНТАКСИЧЕСКОЙ НОТАЦИИ ВЕРСИИ ОДИН (АСН.1)

МОСКВА 1994

Предисловие

1 ПОДГОТОВЛЕН И ВНЕСЕН Техническим комитетом по стандартизации «Информационная технология» (ТК 22)

2 УТВЕРЖДЕН И ВВЕДЕН В ДЕЙСТВИЕ Постановлением Госстандарта России от 20.12.93 № 263
Настоящий стандарт подготовлен на основе применения аутентичного текста международного стандарта ИСО/МЭК 8824—90 «Информационная технология. Взаимосвязь открытых систем. Спецификация абстрактно-синтаксической нотации версии один (АСН.1)»

3 ВВЕДЕН ВПЕРВЫЕ

© Издательство стандартов, 1994

Настоящий стандарт не может быть полностью или частично воспроизведен, тиражирован и распространен в качестве официального издания без разрешения Госстандарта России

СОДЕРЖАНИЕ

Введение . . . 1
1 Область применения . . . 5
2 Нормативные ссылки . . . 6
3 Определения . . . 6
4 Сокращения . . . 11
5 Система обозначений, используемая в настоящем стандарте . . . 11
5.1 Продукции . . . 11
5.2 Совокупности-варианты . . . 12
5.3 Пример продукции . . . 12
5.4 Расположение текста . . . 13
5.5 Рекурсия . . . 13
5.6 Ссылки на совокупность последовательностей . . . 13
5.7 Ссылки на элемент . . . 13
5.8 Теги . . . 13
6 Использование нотации АСН.1 . . . 14
7 Набор знаков нотации АСН.1 . . . 15
8 Элементы АСН.1 . . . 16
8.1 Общие правила . . . 16
8.2 Ссылки на тип . . . 16
8.3 Идентификаторы . . . 16
8.4 Ссылки на значение . . . 17
8.5 Ссылка на модуль . . . 17
8.6 Комментарий . . . 18
8.7 Пустой элемент . . . 18
8.8 Элемент «число» . . . 18
8.9 Элемент «двоичная строка» . . . 18
8.10 Элемент «шестнадцатеричная строка» . . . 18
8.11 Элемент «строка знаков» . . . 19
8.12 Элемент «присвоение» . . . 19
8.13 Элементы, состоящие из одного знака . . . 19
8.14 Элементы — ключевые слова . . . 19
9 Определение модуля . . . 21
10 Ссылки на определения типов и значений . . . 24
11 Присвоение типов и значений . . . 25
12 Определение типов и значений . . . 25
13 Обозначения для булевского типа . . . 28
14 Обозначения для целочисленного типа . . . 29
15 Обозначения для перечислительного типа . . . 30
16 Обозначения для типа «действительное число» . . . 30
17 Обозначения для типа «строка битов» . . . 31
18 Обозначения для типа «строка октетов» . . . 34
19 Обозначения для вырожденного типа . . . 34
20 Обозначения для типов «последовательность» . . . 34
21 Обозначения для типов «последовательность из» . . . 36
22 Обозначения для типов «множество» . . . 37
23 Обозначения для типов «множество из» . . . 38
24 Обозначения для выборочных типов . . . 39
25 Обозначения для селективных типов . . . 40
26 Обозначения для тегированных типов . . . 41
27 Обозначения для произвольного типа . . . 43
28 Обозначения для типа «идентификатор объекта» . . . 44
29 Обозначения для типов «строка знаков» . . . 45
30 Обозначения для типов, определенных в разделах 32—35 . . . 47
31 Определение типов «строка знаков» . . . 47
32 Общая форма записи времени . . . 49

33 Всемирное время . . . 51
34 Внешний тип . . . 52
35 Тип «описатель объекта» . . . 55
36 Обозначения для подтипов . . . 56
37 Подмножества значений подтипа . . . 57
37.1 Одно значение . . . 57
37.2 Вложенный подтип . . . 57
37.3 Диапазон значений . . . 58
37.4 Ограничение размера . . . 59
37.5 Разрешенный алфавит . . . 59
37.6 Образование внутренних подтипов . . . 60
Приложение А МАКРОНОТАЦИЯ . . . 63
А.1 Введение . . . 63
А.2 Дополнения к набору знаков и множеству элементов АСН.1 . . . 64
А.2.1 Макроссылка . . . 64
А.2.2 Ссылка на продукцию . . . 64
А.2.3 Локальная ссылка на тип . . . 64
А.2.4 Локальная ссылка на значение . . . 64
А.2.5 Элемент разграничения вариантов . . . 64
А.2.6 Элемент, обозначающий конец определения . . . 64
А.2.7 Синтаксический терминальный элемент . . . 64
А.2.8 Элементы — ключевые слова синтаксических категорий . . . 65
А.2.9 Дополнительные элементы — ключевые слова . . . 65
А.3 Обозначения для записи макроопределений . . . 66
А.4 Использование новой нотации . . . 69
Приложение В ОПРЕДЕЛЯЕМЫЕ ИСО ЗНАЧЕНИЯ КОМПОНЕНТОВ OBJECT IDENTIFIER . . . 70
Приложение С ОПРЕДЕЛЯЕМЫЕ МККТТ ЗНАЧЕНИЯ КОМПОНЕНТОВ OBJECT IDENTIFIER . . . 72
Приложение D СОВМЕСТНО ОПРЕДЕЛЯЕМЫЕ ЗНАЧЕНИЯ КОМПОНЕНТОВ OBJECT IDENTIFIER . . . 73
Приложение Е ПРИМЕРЫ И РЕКОМЕНДАЦИИ . . . 74
Е.1 Пример записи учета кадров . . . 74
Е.1.1 Неформальное описание записи учета кадров . . . 74
Е.1.2 Описание структуры записи с помощью АСН.1 . . . 74
Е.1.3 Описание значения записи с помощью АСН.1 . . . 75
Е.2 Рекомендации по использованию нотации . . . 75
Е.2.1 Булевский тип . . . 75
Е.2.2 Целочисленный тип . . . 75
Е.2.3 Перечислительный тип . . . 76
Е.2.4 Тип «действительное число» . . . 76
Е.2.5 Тип «строка битов» . . . 76
Е.2.6 Тип «строка октетов» . . . 77
Е.2.7 Вырожденный тип . . . 77
Е.2.8 Типы «последовательность» и «последовательность из» . . . 77
Е.2.9 Тип «множество» . . . 78
Е.2.10 Тегированный тип . . . 79
Е.2.11 Выборочный тип . . . 80
Е.2.12 Селективный тип . . . 81
Е.2.13 Произвольный тип . . . 82
Е.2.14 Внешний тип . . . 82
Е.3 Пример использования макронотации . . . 82
Е.4 Использование нотаций для определения абстрактных синтаксисов . . . 84
Е.5 Подтипы . . . 85
Приложение F. СВОДКА СИНТАКСИСА НОТАЦИИ АСН.1 . . . 87

На нижних уровнях базовой эталонной модели (см. ГОСТ 28906) каждый параметр данных пользователя, задаваемый в сервисном примитиве, определяется как двоичное значение некоторой последовательности октетов.

На уровне представления характер параметров данных пользователя становится иным. Спецификации по прикладному уровню требуют, чтобы данные пользователя услуг представления (см. ГОСТ 34.971) могли принимать значения весьма сложных типов, возможно, включающих строки знаков из различных знаковых наборов. Эти спецификации требуют наличия определенной нотации, которая определяла бы передаваемое значение, не задавая конкретного способа его представления. Способ представления определяется заданием одного или нескольких алгоритмов, называемых правилами кодирования. Правила кодирования определяют конкретный вид октетов на сеансовом уровне, с помощью которых передаются значения данных прикладного уровня (синтаксис передачи). Протокол уровня представления (см. ГОСТ 34.972) позволяет взаимодействующим объектам договариваться о том, какие синтаксисы передачи будут ими использоваться.

Цель задания значения состоит в том, чтобы можно было отличать его от всех возможных других значений. Объединение некоторого значения с другими значениями, отличными от данного, называется типом, и каждый конкретный элемент этой совокупности является значением этого типа. С более общих позиций значение или тип часто могут рассматриваться как образования, состоящие из более простых значений или типов, между которыми установлены взаимосвязи. В качестве синонима термина «тип» часто используют термин «тип данных»

Для правильной интерпретации конкретного представления значения (например, с помощью пометок на бумаге или бит, передаваемых по линии связи) необходимо знать тип передаваемых данных (обычно определяемый из контекста). Таким образом, указание типов является одним из важных вопросов, рассматриваемых в настоящем стандарте.

Одним из общих методов определения сложных типов является определение небольшого количества «простых типов» путем задания всех возможных значений этих простых типов и последующего образования различных комбинаций из этих простых типов. Ниже перечислены некоторые из способов определения новых типов:

а) имея список (упорядоченный) существующих типов, можно сформировать новое значение в виде последовательности, в которую входит по одному из значений каждого типа, входящего в список; совокупность всех возможных значений, полу-

чаемых таким образом, образует новый тип (если все уже определенные типы, входящие в список, являются различными, то можно расширить этот способ, допуская пропуск некоторых значений из списка);

b) имея список существующих типов, можно сформировать новое значение в виде множества (неупорядоченного) значений, в которое входит по одному из значений каждого типа, входящего в список; совокупность всех значений, получаемых таким образом, образует новый тип (этот способ можно также расширить, допуская пропуск некоторых значений);

c) имея один существующий тип, можно сформировать новое значение в виде последовательности (упорядоченной) или множества (неупорядоченного), состоящего из пустого множества значений и одного или нескольких значений исходного типа; совокупность (неограниченная) всех возможных значений, полученных таким образом, образует новый тип;

d) имея список различных существующих типов, можно выбрать какое-либо значение одного из этих типов; множество всех возможных значений, получаемых таким образом, образует новый тип;

e) имея некоторый тип, можно сформировать новый тип в виде подмножества исходного типа, используя некоторую структурную или порядковую взаимосвязь между элементами исходного множества.

Типы, определенные с помощью описанных выше методов, называются структурированными типами.

Каждому типу, определенному посредством нотации, описываемой настоящим стандартом, назначается «тег». Тег либо определен в самом стандарте, либо определяется пользователем описываемой здесь нотации.

Часто один и тот же тег назначается нескольким разным типам, при этом конкретный тип идентифицируется контекстом, в котором используется тег.

Пользователь настоящей нотации может присвоить новый тег какому-либо существующему типу, создавая тем самым новый тип, отличный от первоначального. Это может потребоваться в тех случаях, когда необходимо определить, какой выбор был сделан в ситуации типа описанной выше в п. d).

Настоящей нотацией определяется четыре класса тегов.

Первый класс называется универсальным. Использование тегов универсального класса определяется спецификациями настоящего стандарта, причем каждый тег либо:

a) присвоен какому-то одному типу, либо

b) присвоен некоторому способу построения типов.

Второй класс тегов называется прикладным. Теги прикладного класса присваиваются типам данных в других стандартах. В рамках одного стандарта тег прикладного класса присваивается только какому-то одному типу.

Третий класс тегов называется пользовательским. Теги пользовательского класса не присваиваются в стандартах ИСО. Порядок их использования может быть различным в различных организациях.

Последний класс тегов называется контекстно-зависимым. Теги этого класса могут свободно назначаться при любом использовании настоящей нотации, и интерпретируются в соответствии с контекстом, в котором они используются.

Теги ориентированы главным образом на машинную обработку данных и не несут существенной информации при использовании нотации человеком. Однако в тех случаях, когда необходимо потребовать, чтобы все типы были различными, это выражается требованием различия тегов, присвоенных вышеупомянутым типам. Таким образом, присвоение тегов является важным аспектом использования настоящей нотации.

Примечания

1 Все типы, которые могут быть определены с использованием нотации, описываемой настоящим стандартом, имеют теги. Пользователь настоящей нотации может определять новый тип, присваивая новый тег ранее определенному типу.

2 Правила кодирования гарантируют явное или неявное задание тега типа вместе со значением данного типа при любом определении этого значения. Ограничения, налагаемые на использование настоящей нотации, обеспечивают достаточность тега для идентификации типа, при условии доступности определений соответствующих типов.

Настоящий стандарт определяет нотацию, которая служит одновременно и средством определения сложных типов данных, и средством задания конкретных значений этих типов. Это осуществляется без указания конкретного способа представления (в виде последовательности октетов) значений данного типа при их передаче. Нотация, обладающая этим свойством, называется нотацией для определения абстрактного синтаксиса.

Назначение настоящего стандарта состоит в описании нотации для определения абстрактного синтаксиса, называемой абстрактно-синтаксической нотацией версии 1, или АСН.1. Абстрактно-синтаксическая нотация версии 1 является полуформальным средством, используемым при определении протоколов. Использование настоящей нотации само по себе не исключает неоднозначности соответствующих спецификаций. Обеспечение однозначности спецификаций должно осуществляться пользователем нотации.

Настоящий стандарт поддерживается другими стандартами, определяющими правила кодирования. Применение правил кодирования к значениям некоторого типа, определенным в АСН.1,

полностью определяет представление этих значений при передаче (т. е. синтаксис передачи).

С технической и редакционной точек зрения настоящий стандарт согласуется с рекомендацией X.208 (1988) МККТТ.

Часть 1 настоящего стандарта содержит определения простых типов, используемых в АСН.1, и определяет обозначения, которые следует использовать для ссылок на простые типы и для описания структурированных типов. В части 1 также описываются обозначения, которые следует использовать при задании значений типов, определенных с помощью АСН.1.

В части 2 настоящего стандарта определяются дополнительные типы (типы «строка знаков»), которые применением правил кодирования могут быть приравнены к типу «строка октетов».

В части 3 настоящего стандарта определяются некоторые структурированные типы, которые являются общеупотребительными, и в то же время не требуют введения новых правил кодирования, отличных от правил кодирования ранее определенных типов.

В части 4 настоящего стандарта определяются обозначения, позволяющие определять подтипы из значений порождающего типа.

Приложение А является обязательной частью настоящего стандарта и определяет нотацию, расширяющую базовую нотацию АСН.1. Эти расширения называются макросредствами.

Приложение В является обязательной частью настоящего стандарта и содержит определение логического дерева идентификаторов объектов для источников идентификации, поддерживаемых ИСО

ГОСУДАРСТВЕННЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

Информационная технология.

ВЗАИМОСВЯЗЬ ОТКРЫТЫХ СИСТЕМ. СПЕЦИФИКАЦИЯ АБСТРАКТНО-СИНТАКСИЧЕСКОЙ НОТАЦИИ ВЕРСИИ ОДИН (АСН.1)

Information technology. Open Systems Interconnection, Specification Abstract Syntax Notation One (ASN.1)

Дата введения 1994—07—01

1 ОБЛАСТЬ ПРИМЕНЕНИЯ

Настоящий стандарт описывает нотацию для определения абстрактного синтаксиса, называемую абстрактно-синтаксической нотацией версии один (АСН.1).

Настоящий стандарт определяет ряд простых типов вместе с присвоенными им тегами и обозначения, которые следует использовать для ссылок на эти типы, а также для задания значений этих типов.

Настоящий стандарт определяет методы построения новых типов из более простых типов и обозначения, которые следует использовать при определении такого рода структурированных типов и присвоении им тегов, а также при задании значений этих типов.

Настоящий стандарт определяет знаковые наборы, используемые в АСН.1.

Настоящий стандарт определяет (с помощью АСН.1) ряд общеупотребительных типов, на которые могут ссылаться пользователи АСН.1.

Нотация АСН.1 может применяться во всех случаях, где требуется определять абстрактный синтаксис информации. Одной из сфер ее применения (но не единственной) являются протоколы прикладного уровня.

Ссылки на нотацию АСН.1 содержатся в других стандартах по уровню представления, в которых определяются правила кодирования для простых типов, структурированных типов, различных типов «строка знаков» и общеупотребительных типов, определенных в нотации АСН.1.

Издание официальное

2 НОРМАТИВНЫЕ ССЫЛКИ

ГОСТ 28906—91 (ИСО 7498) Системы обработки информации. Взаимосвязь открытых систем. Базовая эталонная модель (см. также Рекомендацию МККТТ X.200)

ГОСТ 34.971—91 (ИСО 8822) Системы обработки информации. Взаимосвязь открытых систем. Спецификация услуг уровня представления для режима с установлением соединения

ГОСТ 34.972—91 (ИСО 8823) Системы обработки информации. Взаимосвязь открытых систем. Спецификация протокола уровня представления для режима с установлением соединения

ГОСТ Р ИСО/МЭК 8825—93 Системы обработки информации. Взаимосвязь открытых систем. Спецификация базовых правил кодирования для нотации абстрактного синтаксиса версии один (АСН 1)

ИСО 2375—85 Обработка информации. Процедуры регистрации последовательностей расширения*

ИСО 3166—88 Коды для представления названий стран*

ИСО 6523—84 Обмен информацией. Структура идентификаторов организаций*

Рекомендация МККТТ X.208 : 1988 Спецификация нотации абстрактного синтаксиса версии 1 (АСН 1)

3 ОПРЕДЕЛЕНИЯ

В настоящем стандарте используются определения, содержащиеся в ГОСТ 28906.

3.1 **Значение** — отличный от других элементов множества значений.

3.2 **Тип** — множество значений, которому присвоено имя.

3.3 **Простой тип** — тип, определяемый прямым заданием множества составляющих его значений.

3.4. **Структурированный тип** — тип, при определении которого используются ссылки на другие типы.

3.5 **Тип-компонент** — один из типов, используемых при определении структурированного типа.

3.6 **Тег** — обозначение типа, присваиваемое каждому типу в АСН.1.

3.7 **Тегирование** — изменение существующего (возможно, заданного по умолчанию) тега некоторого типа новым тегом.

---

* До прямого применения данного документа в качестве государственного стандарта распространение его осуществляет секретариат ТК 22 «Информационная технология»

3.8 **Набор знаков АСН.1** — набор знаков, определенный в разделе 7, который используют в обозначениях нотации АСН.1.

3.9 **Элементы** — поименованные последовательности знаков из набора знаков АСН.1, определенные в разделе 8, используемые для формирования обозначений нотации АСН.1.

3.10 **Ссылка на тип (или значение)** — имя, однозначно идентифицирующее тип (или значение) в рамках некоторого контекста.

Примечание. Ссылочные имена присваивают типам, определяемым в настоящем стандарте; эти имена могут применяться при любом использовании АСН.1. Другие ссылочные имена определены в других стандартах, они применимы только в контексте соответствующих стандартов.

3.11 **Правила кодирования АСН.1** — правила, определяющие конкретное представление (при передаче) значений любых типов, определенных в АСН.1; правила кодирования АСН.1 позволяют получателю распознать переданную информацию как конкретное значение конкретного типа, определенного в АСН.1.

3.12 **Тип «строка знаков»** – тип, значения которого являются строками знаков из некоторого набора знаков.

3.13 **Булевский тип** — простой тип с двумя различными значениями.

3.14 **Истинно** — одно из двух различных значений булевского типа.

3.15 **Ложно** – другое значение булевского типа.

3.16 **Целочисленный тип** – простой тип с различными значениями, являющимися положительными или отрицательными целыми числами, включая ноль (рассматриваемый как одно значение).

Примечание. Конкретные правила кодирования ограничивают диапазон, в котором могут находиться числа, однако рамки этих ограничений выбраны достаточно широкими для того, чтобы быть практически ощутимыми для любого из пользователей АСН.1.

3.17 **Перечислительный тип** — простой тип, значениям которого назначаются отличные друг от друга идентификаторы, являющиеся частью обозначения для этого типа

3.18 **Тип «действительное число»** — простой тип, различными значениями которого (описанными в 16.2) являются элементы множества действительных чисел.

3.19 **Тип «строка бит»** – простой тип, различными значениями которого являются упорядоченные последовательности из пустого множества бит, одного или более бит.

Примечание. Правила кодирования не ограничивают число бит в строке бит.

3.20 **Тип «строка октетов»** — простой тип, различные значения которого являются упорядоченными последовательностями из нуг

того множества октетов, одного или более октетов, где октет — это упорядоченная последовательность из восьми бит.

Примечание. Правила кодирования не ограничивают число октетов в строке октетов.

3.21 **Вырожденный тип (тип NULL)** — простой тип, состоящий из единственного значения, также называемого вырожденным значением («NULL»).

Примечание. Вырожденное значение обычно используют в тех случаях, когда возможно несколько различных вариантов, но ни один из них не имеет места.

3.22 **Тип «последовательность»** — структурированный тип, определяемый ссылкой на фиксированный, упорядоченный список типов (некоторые из которых могут быть объявлены необязательными); каждое значение нового типа является упорядоченной последовательностью значений типов-компонентов, по одному из каждого из них.

Примечание. Если тип-компонент объявлен необязательным, то значение нового типа может не включать в себя значение этого типа-компонента.

3.23 **Тип «последовательность из»** — структурированный тип, определяемый ссылкой на один из существующих типов; каждое значение нового типа является упорядоченной последовательностью из пустого множества значений, одного или более значений существующего типа.

Примечание. Правила кодирования не ограничивают число значений — компонентов в значении «последовательность из».

3.24 **Тип «множество»** — структурированный тип, определяемый ссылкой на фиксированный неупорядоченный список различных типов (некоторые из которых могут быть объявлены необязательными); каждое значение нового типа является неупорядоченным списком значений, по одному из каждого типа-компонента.

Примечание. Если тип-компонент объявлен необязательным, то новый тип не обязательно содержит значение этого типа-компонента.

3.25 **Тип «множество из»** — структурированный тип, определяемый ссылкой на единственный существующий тип; каждое значение нового типа является неупорядоченным списком из нуля, одного или более значений существующего типа.

Примечание. Правила кодирования не ограничивают число значений-компонентов в значении «множество из».

3.26 **Тегированный тип** — тип, определяемый указанием одного из существующих типов и тега; новый тип является изоморфным существующему типу, но отличен от него.

3.27 **Выборочный тип** — структурированный тип, определяемый указанием фиксированного неупорядоченного списка различных типов; каждое значение нового типа является значением одного из типов-компонентов.

3.28 **Селективный тип** — структурированный тип, определяемый указанием одного из типов-компонентов выборочного типа.

3.29 **Произвольный тип** — выборочный тип, типы-компоненты которого не заданы, но ограничены множеством типов, которые могут быть определены с помощью АСН.1.

3.30 **Внешний тип** — тип, различные значения которого не могут быть установлены только лишь по той информации, что они являются значениями внешнего типа, однако значения этого типа могут быть установлены по их кодовому представлению; эти значения могут быть (но это не обязательно) описаны с помощью АСН.1, и соответственно их кодовое представление может (но не обязательно) соответствовать правилам кодирования для АСН.1.

3.31 **Информационный объект** — некоторая точно определенная совокупность информации, определение или спецификация, использование которой в конкретном взаимодействии идентифицируется соответствующим именем.

3.32 **Идентификатор объекта** — значение (отличающееся от других подобных значений), которое связывается с информационным объектом.

3.33 **Тип «идентификатор объекта»** — тип, различные значения которого составляют множество всех идентификаторов объектов, присвоенных в соответствии с правилами настоящего международного стандарта.

Примечание Правила настоящего международного стандарта предоставляют возможность целому ряду организаций — источников идентификации независимо друг от друга связывать идентификаторы объектов с информационными объектами.

3.34 **Тип «описатель объекта»** — тип, различные значения которого имеют вид текста, ориентированного на восприятие человеком, дающего краткое описание информационного объекта.

Примечание. Обычно (но не всегда) описатель объекта связывается с каким-то одним информационным объектом. Информационный объект полностью идентифицируется только с помощью идентификатора объекта

3.35 **Рекурсивные определения** — множество таких определений, допускающихся в нотации АСН.1, которые не могут быть переупорядочены таким образом, чтобы все типы, использующиеся в соответствующих конструкциях этих определений, были определены до использования этих конструкций.

Примечание. Рекурсивные определения разрешены в АСН.1; пользователь этой нотации должен позаботиться о том, что используемые значения получаемых в результате типов имели конечное представление.

3.36 **Модуль** — одно или несколько выражений нотации АСН.1 для определения типов и значений, оформленных как единое целое в соответствии с обозначениями определения модулей. (См. раздел 9).

3.37 **Продукция** — часть формальной системы обозначений, используемой при спецификации АСН.1, в которой допустимым последовательностям элементов присваивается имя, которое может быть использовано для последующих ссылок на эти последовательности при определении новых допустимых последовательностей.

3.38 **Всемирное согласованное время (UTC)** — временной эталон, поддерживаемый Международным бюро времени и служащий основой для согласованного распространения частот и сигналов времени.

Примечания

1 Источником этого определения является рекомендация 460—2 Международной консультативной комиссии по радио (CCIR). Акроним UTC для всемирного согласованного времени также был введен CCIR.

2 UTC также называют средним гринвичским временем; соответствующие временные сигналы регулярно транслируются

3.39 **Пользователь (нотации АСН.1)** — частное лицо или организация, определяющие абстрактный синтаксис некоторой информации, используя АСН.1.

3.40 **Подтип (порождающего типа)** — тип, значения которого задаются как подмножество значений другого (порождающего) типа.

3.41 **Порождающий тип (для подтипа)** — тип, используемый при определении подтипа.

Примечание. Порождающий тип может быть подтипом некоторого другого типа.

3.42 **Спецификация подтипа** — обозначение, которое может быть использовано в совокупности с обозначением некоторого типа для определения подтипа этого типа.

3.43 **Подмножество значений подтипа** — обозначение, образующее часть спецификации подтипа, задающее нехоторое множество значений, которое должно быть включено в подтип

В настоящем стандарте используют следующие термины, определенные в ГОСТ 34971:

a) **значение данных представления;**
b) **абстрактный синтаксис;**
c) **имя абстрактного синтаксиса;**
d) **имя синтаксиса передачи.**

В настоящем стандарте используют также следующие термины, определенные в ИСО 6523:

a) **организация — источник кодов;**

b) **код организации;**

c) **Международный кодовый определитель.**

4 СОКРАЩЕНИЯ

ACH.1 — абстрактно-синтаксическая нотация версии 1

UTC — Coordinated Universal Time (всемирное согласованное время)

ICD — International Code Designator (Международный кодовый определитель)

DCC — Data Country Code (цифровой код страны)

DNIC — Data Network Identification Code (идентификационный код сети передачи данных)

5 СИСТЕМА ОБОЗНАЧЕНИЙ, ИСПОЛЬЗУЕМАЯ В НАСТОЯЩЕМ СТАНДАРТЕ

Нотация ACH.1 строится из последовательностей знаков знакового набора ACH.1, определенного в разделе 7.

Каждый конкретный случай использования нотации ACH.1 содержит знаки из знакового набора ACH.1, сгруппированные в элементы. В разделе 8 определяются все последовательности знаков, образующие элементы нотации ACH.1, и каждому элементу присваивается имя.

Нотация ACH.1 определяется в 9 (и последующих разделах) путем описания совокупности последовательностей элементов, которые образуют правильные варианты использования нотации ACH.1, и определения семантики каждой такой последовательности.

Для того, чтобы описать эту совокупность последовательностей, в настоящем стандарте используют формальную систему обозначений, определяемую в последующих подразделах.

5.1 **Продукции**

Новая (более сложная) совокупность последовательностей, допустимых в ACH.1, определяется с помощью продукции. Продукция, используя имена совокупностей последовательностей, определенных в настоящем стандарте, строит новую совокупность последовательностей с помощью двух следующих способов:

a) образование новой совокупности из последовательностей, содержащихся в любой из исходных совокупностей, либо

b) образование новой совокупности из всех последовательностей, которые могут быть получены путем выбора одной последовательности из каждой исходной совокупности и соединения выбранных последовательностей в заданном порядке.

Каждая продукция состоит из следующих частей, образующих одну или несколько строк, в следующем порядке:

a) имени новой совокупности последовательностей;
b) знаков : : =
c) одной или нескольких совокупностей - вариантов последовательностей, определенных по правилам, описанным в 5.2, и разделенных знаком :

Последовательность присутствует в новой совокупности, если она присутствует в одной или нескольких совокупностях-вариантах. Ссылки на новую совокупность в настоящем стандарте осуществляются путем задания имени из a).

Примечание. Если одна и та же последовательность присутствует в нескольких совокупностях-вариантах, то любая семантическая неоднозначность полученного обозначения снимается с помощью других частей полной последовательности, определяемой данной продукцией.

5.2 **Совокупности-варианты**

Каждую из совокупностей-вариантов, упомянутых в 5.1c, задают с помощью списка имен. Каждое из этих имен является либо именем элемента, либо именем совокупности последовательностей, определенных с помощью продукции в настоящем стандарте.

Совокупность вариант состоит из всех последовательностей, получаемых выбором произвольной последовательности (или элемента) из совокупности, определяемой первым именем, за которой помещается произвольная последовательность (или элемент) из совокупности, определяемой вторым именем, за которой свою очередь помещается произвольная последовательность (или элемент) из совокупности, определяемой третьим именем, и так далее, вплоть до последнего имени, заданного для данного варианта.

5.3 **Пример продукции**

Значение СтрокаБит : : = b-строка |
h-строка |
{Список-Идентификаторов}

Приведенная выше запись является примером продукции, которая связывает имя Значение-СтрокаБит со следующими последовательностями:

a) произвольным элементом «b-строка»;
b) произвольным элементом «h-строка»;

c) произвольной последовательностью из совокупности с именем «Список-Идентификаторов», перед которой следует знак {и за которой следует знак}.

Примечание. Знаки {и} являются именами элементов, состоящими из одного знака, соответственно, {и}.

«Список-Идентификаторов» в данном примере должен быть определен другой продукцией, которая может быть приведена как до, так и после продукции «Значение-СтрокаБит».

5.4 **Расположение текста**

Каждая продукция, используемая в настоящем стандарте, предваряется пустой строкой; за продукцией также следует пустая строка. Внутри продукции пустые строки отсутствуют. Продукция может либо состоять из одной строки, либо быть разбитой на несколько строк. Расположению текста продукции не придается какого-либо значения.

5.5 **Рекурсия**

Продукции, используемые в настоящем стандарте, часто являются рекурсивными. Рекурсивные продукции применяются неограниченное число раз до тех пор, пока не будут исчерпаны все порождаемые ими последовательности.

Примечание. Во многих случаях такое повторное применение продукции приводит к неограниченной совокупности разрешенных последовательностей, некоторые из которых сами могут быть неограниченными. Это не является ошибкой.

5.6 **Ссылки на совокупность последовательностей**

В настоящем стандарте ссылки на совокупность последовательностей (часть нотации АСН.1) осуществляются указанием первого имени в продукции (перед : : =); если ссылка на имя не является частью продукции, имя обрамляется знаками " для того, чтобы отличить его от остального текста на естественном языке.

5.7 **Ссылки на элемент**

В настоящем стандарте ссылки на элемент осуществляются указанием имени элемента; если имя не является частью продукции, оно обрамляется знаками " для того, чтобы отличить его от остального текста на естественном языке

5.8 **Теги**

Тег задается указанием его класса и номера в классе. Определены следующие классы тегов:

универсальный (universal)

пользовательский (private)

контекстно-зависимый (context-spesific)

Номер задается в виде неотрицательного целого числа в десятичной системе счисления.

Ограничения на теги, присванваемые пользователем АСН.1, определены в разделе 24.

Теги универсального класса назначаются таким образом, чтобы для структурированных типов по тегу можно было определить структуру верхнего уровня, а для простых типов тег полностью определял тип. В таблице 1 перечислены теги универсального класса, назначенные в рамках настоящего стандарта.

Примечание. Для последующих версий настоящего стандарта резервируется дополнительное множество тегов.

Таблица 1 — **Присвоение тегов универсального класса**

UNIVERSAL 1	Булевский тип
UNIVERSAL 2	Целочисленный тип
UNIVERSAL 3	Тип «строка битов»
UNIVERSAL 4	Тип «строка октетов»
UNIVERSAL 5	Вырожденный тип
UNIVERSAL 6	Тип «идентификатор объекта»
UNIVERSAL 7	Тип «описатель объекта»
UNIVERSAL 8	Внешний тип
UNIVERSAL 9	Тип «действительное число»
UNIVERSAL 10	Перечислительный тип
UNIVERSAL 12—15	Зарезервированы для последующих версий настоящего стандарта
UNIVERSAL 16	Типы «последовательность» и «последовательность из»
UNIVERSAL 17	Типы «множество» и «множество из»
UNIVERSAL 18—22, 25—27	Типы «строка знаков»
UNIVERSAL 23—24	Типы «время»
UNIVERSAL 28—	Зарезервированы для последующих версий настоящего стандарта

6 ИСПОЛЬЗОВАНИЕ НОТАЦИИ АСН.1

6.1. Обозначения нотации АСН.1 для определения типа определяются продукцией «Тип» (см. 12.1).

6.2 Обозначения нотации АСН.1 для определения значения определяются продукцией «Значение» (см. 12.7).

Примечание Вообще говоря, невозможно интерпретировать значение, не зная типа, к которому оно относится.

6.4 Обозначения нотации АСН.1 для присвоения значения ссылочному имени значения определяются продукцией «Присвоение Значения» (см. 11.2).

6.5 Обозначения вида «ПрисвоениеТипа» и «ПрисвоениеЗначения» должны использоваться только внутри обозначения вида «ОпределениеМодуля» (см. 9.1).

7 НАБОР ЗНАКОВ НОТАЦИИ АСН.1

7.1 Каждый элемент АСН.1 состоит из последовательности знаков, перечисленных в таблице 2, за исключением случаев, описанных в 7.2 и 7.3.

Таблица 2 — **Набор знаков АСН.1**

от A до Z
от a до z
от 0 до 9
: : =, { } < .
( ) [ ] — ' "

Примечания

1 Дополнительно к знакам, перечисленным в таблице 2, в макронотации используют знаки | и >> (см. приложение А).

2 В стандартах, производных от данного стандарта, которые будут разработаны национальными организациями по стандартизации, знаки, отсутствующие в таблице 2, могут присутствовать в следующих элементах (пять последних элементов определяются в приложении А):

— СсылкаНаТип (8.2.1);
— Идентификатор (8.3);
СсылкаНаЗначение (8.4);
СсылкаНаМодуль (8.5);
— Макроссылка (А.2.1);
— СсылкаНаПродукцию (А.2.2);
— ЛокальнаяСсылкаНаТип (А.2.3);
— ЛокальнаяСсылкаНаЗначение (А.2.4);
— а-строка (А.2.7).

3 В тех случаях, когда используют дополнительные знаки национальных алфавитов, в которых отсутствует различие между строчными и прописными буквами, синтаксическое различие, достигаемое за счет использования строчных и прописных букв в некоторых из перечисленных выше элементов АСН.1, должно обеспечиваться каким-то другим способом.

7.2 При использовании нотации для определения значения типа «строка знаков» в обозначение нотации АСН.1 могут входить все знаки заданного знакового набора, обрамленные знаками» (см. 8.11).

7.3 Дополнительные символы (не включенные в таблицу 2) могут входить в состав элемента «комментарий» (см. 8.6).

7.4 Не следует придавать какого-либо значения типографскому стилю, размеру, цвету, яркости и другим характеристикам знаков нотации, связанным со способом их отображения.

7.5 Строчные и прописные буквы следует рассматривать как различные знаки.

8 ЭЛЕМЕНТЫ АСН.1

8.1 **Общие правила**

8.1.1 В следующих подразделах определяется, с помощью каких знаков образуются элементы АСН.1 В каждом случае дается имя элемента и определение знаковых последовательностей, из которых может состоять элемент.

Примечание. В приложении А определяются дополнительные элементы, использующиеся в макронотации.

8.1.2 Каждый из элементов, определяемых в последующих подразделах, должен располагаться в пределах одной строки и (за исключением элемента «комментарий») не должен содержать пробелов.

8.1.3 Длина строки не ограничивается.

8.1.4 Элементы в последовательностях, определяемых настоящим стандартом (в обозначениях нотации АСН.1), могут располагаться на одной или на нескольких строках и могут разделяться одним или несколькими пробелами или пустыми строками.

8.1.5 Каждый элемент должен отделяться от следующего за ним элемента пробелом или помещением его на следующую строку, если начальный знак (или знаки) следующего элемента являются разрешенными для включения в конец последовательности знаков предыдущего элемента.

8.2 **Ссылки на тип**

Имя элемента — ССылкаНаТип

8.2.1 Элемент «ССылкаНаТип» должен состоять из произвольного количества (большего или равного единице) букв, цифр и дефисов. Начальный знак должен быть прописной буквой. Последний знак не должен быть дефисом. Два дефиса не должны следовать друг за другом.

Примечание. Правила, относящиеся к использованию дефиса, вводятся во избежание неоднозначности, которая может быть вызвана наличием комментария, следующего за элементом.

8.2.2 Элемент «ССылкаНаТип» не должен совпадать с одной из зарезервированных знаковых последовательностей, перечисленных в таблице 3.

8.3 **Идентификаторы**

Наименование элемента — Идентификатор

Элемент «идентификатор» должен состоять из произвольного количества (большего или равного единице) букв, цифр и дефисов. Начальный знак должен быть строчной буквой. Последний знак не

должен быть дефисом. Два дефиса не должны следовать друг за другом.

Примечание. Правила, относящиеся к использованию дефиса, вводятся во избежание неоднозначности, которая может быть вызвана наличием комментария, следующего за элементом.

Таблица 3 — **Зарезервированные последовательности знаков**

BOOLEAN	BEGIN
INTEGER	END
BIT	DEFINITIONS
STRING	EXPLICIT
OCTET	ENUMERATED
NULL	EXPORTS
SEQUENCE	IMPORTS
OF	REAL
SET	INCLUDES
IMPLICIT	MIN
CHOICE	MAX
ANY	SIZE
EXTERNAL	FROM
OBJECT	WITH
IDENTIFIER	COMPONENT
OPTIONAL	PRESENT
DEFAULT	ABSENT
COMPONENTS	DEFINED
UNIVERSAL	BY
APPLICATION	PLUS-IFINITY
PRIVATE	MINUS-INFINITY
TRUE	TAGS
FALSE	

8.4 Ссылки на значение

Имя элемента — ССылкаНаЗначение

Элемент «ССылкаНаЗначение» должен состоять из последовательности знаков, удовлетворяющей определению элемента «идентификатор» в 8.3. Отличие элемента «ССылкаНаЗначение» от элемента «идентификатор» при анализе конкретного использования настоящей нотации достигается различием контекстов, в которых появляются эти элементы.

8.5 Ссылка на модуль

Имя элемента — ССылкаНаМодуль

Элемент «ССылкаНаМодуль» должен состоять из последовательности знаков, удовлетворяющей определению элемента «ССылкаНаТип» в 8.2. Отличие элемента «ССылкаНаМодуль» от элемента

«СсылкаНаТип» при анализе конкретного использования настоящей нотации обеспечивается различием контекстов, в которых появляются эти элементы.

8.6 **Комментарий**

Имя элемента — Комментарий

8.6.1 Ссылки на «комментарий» не используют в рамках определения нотации АСН.1. «Комментарий», тем не менее, может присутствовать в любом месте между другими элементами нотации АСН.1, формально не имея никакого значения.

8.6.2 «Комментарий» начинается с пары следующих друг за другом дефисов и заканчивается другой парой следующих друг за другом дефисов, либо концом текущей строки, в зависимости от того, что последует раньше. Комментарий не может содержать в себе пару следующих друг за другом дефисов, кроме той, с которой он начинается и, если она имеется, той, которая его завершает. Комментарий может содержать знаки, не являющиеся элементами знакового набора, определенного в 7.1 (см. 7.3).

8.7 **Пустой элемент**

Имя элемента — пусто

Элемент «пусто» не содержит никаких знаков. Он используется в обозначениях 5 при задании совокупностей-вариантов в тех случаях, когда требуется указать, что возможно отсутствие всех приведенных альтернатив.

8.8. **Элемент «число»**

Имя элемента — число

Элемент «число» состоит из одной или более цифр. Первая цифра не может быть нулем, за исключением случая, когда «число» состоит из одной цифры.

8.9 **Элемент «двоичная строка»**

Имя элемента — b-строка

Элемент «двоичная строка» состоит из произвольного количества (от нуля и более) нулей и единиц, перед которыми следует знак ′ и за которыми следует пара знаков:

′B

Пример — ′01101100′B

8.10 **Элемент «шестнадцатеричная строка»**

Имя элемента — h-строка

8.10.1 Элемент «шестнадцатеричная строка» состоит из произвольного количества (от нуля и более) знаков

A B C D E F 0 1 2 3 4 5 6 7 8 9,

перед которыми следует знак ′ и за которыми следует пара знаков:

′H

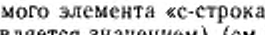
Пример ′AB0186′H

8.10.2 Каждый знак используется для обозначения значения полуоктета в шестнадцатеричном представлении.

8.11 **Элемент «строка знаков»**

Имя элемента — с-строка

Элемент «с-строка» состоит из произвольного количества (от нуля и более) знаков из набора знаков, который определяется типом рассматриваемой строки знаков, перед которыми и вслед за которыми следует знак ". Если набор знаков содержит знак ", то он представляется в элементе «с-строка» парой знаков ''. Набор знаков, о котором идет речь, совсем необязательно ограничивается знаками, перечисленными в таблице 2, он определяется типом рассматриваемого элемента «с-строка» (типом, для которого «СтрокаЗнаков» является значением) (см 7.2).

Пример 

8.12 **Элемент «присвоение»**

Имя элемента — «::=»

Этот элемент состоит из последовательности знаков

::=

Примечание. Эта последовательность не содержит пробелов (см. 8.1.2)

8.13 **Элементы, состоящие из одного знака**

Имена элементов —

{
}
<
,
.
(
)
[
]
- (дефис)
:

Элемент, имеющий имя, входящее в приведенный выше перечень, состоит из одного символа, образующего это имя.

8.14 **Элементы — ключевые слова**

Имена элементов —

BOOLEAN
INTEGER
BIT
STRING

OCTET
NULL
SEQUENCE
OF
SET
IMPLICIT
CHOICE
ANY
EXTERNAL
OBJECT
IDENTIFIER
OPTIONAL
DEFAULT
COMPONENTS
UNIVERSAL
APPLICATION
PRIVATE
TRUE
FALSE
BEGIN
END
DEFINITIONS
EXPLICIT
ENUMERATED
EXPORTS
IMPORTS
REAL
INCLUDES
MIN
MAX
SIZE
FROM
WITH
COMPONENT
PRESENT
ABSENT
DEFINED
BY
PLUS-INFINITY
MINUS-INFINITY
TAGS

Элементы с именем, входящим в приведенный выше перечень, состоят из последовательности знаков, составляющей это имя.

Примечания

1 В приведенных выше последовательностях знаков отсутствуют пробелы.

2 Последовательности из этого перечня, не являющиеся зарезервированными (перечисленными в 8.2.2), могут быть отличены от других элементов, содержащих такие же знаки, с помощью контекста, в котором они присутствуют.

9 ОПРЕДЕЛЕНИЕ МОДУЛЯ

9.1 «ОпределениеМодуля» задается следующими продукциями:

```
ОпределениеМодуля : : =
    ИдентификаторМодуля
    DEFINITIONS
ТегиПоУмолчанию
    « : : = »
    BEGIN
    ТелоМодуля
    END
ИдентификаторМодуля : : =
    ССылкаНаМодуль
    ПрисвоенныйИдентификатор
ПрисвоенныйИдентификатор : : =
    Значение-ИдентификаторОбъекта |
    пусто
ТегиПоУмолчанию : : =
    EXPLICIT TAGS |
    IMPLICIT TAGS |
    пусто
ТелоМодуля : : =
    ВыходныеССылки ВходныеССылки СписокПрисвоений |
    пусто
ВыходныеССылки : : =
    EXPORTS ВыходныеСимволы |
    пусто
ВыходныеСимволы : : =
    СписокСимволов |
    пусто
ВходныеССылки : : =
    IMPORTS ВходныеСимволы |
    пусто
ВходныеСимволы : : =
    СписокСимволовИзМодуля |
    пусто
СписокСимволовИзМодуля : : =
    СимволыИзМодуля СписокСимволовИзМодуля |
```

```
        СимволыИзМодуля
    СимволыИзМодуля : : =
        СписокСимволов FROM ИдентификаторМодуля
    СписокСимволов : : =
        Символ, СписокСимволов | Символ
    Символ : : =
        ССылкаНаТип | ССылкаНаЗначение
    СписокПрисвоений : : =
        СписокПрисвоений Присвоение |
        Присвоение
    Присвоение : : =
        ПрисвоениеТипа | ПрисвоениеЗначения
```

Примечания

1 В приложении А определяются последовательности вида «МакроОпределение», которые также могут присутствовать в последовательности вида «СписокПрисвоений». Выражения, определяемые с помощью макроопределений, могут быть приведены как до, так и после самого макроопределения, в рамках того же модуля.

2 В отдельных (исключительных) случаях, для использования в качестве примеров, а также для определения типов с тегами универсального класса, последовательность вида «ТелоМодуля» может использоваться самостоятельно, а не как часть последовательности «ОпределениеМодуля».

3 Продукции «ПрисвоениеТипа» и «ПрисвоениеЗначения» определены в разделе 11.

4 Объединение типов данных, описываемых с помощью нотации АСН.1, в модули необязательно определяет способ формирования из значений данных представления поименованных абстрактных синтаксисов, в целях определения контекста представления.

5 Значение, выбранное для последовательности «ТегиПоУмолчанию», влияет только на те типы, которые явно определены в модуле. Оно не влияет на интерпретацию типов, определенных входными ссылками.

6 В качестве «символа» может использоваться «макроссылка» (см. приложение А).

9.2 Если для последовательности «ТегиПоУмолчанию» было выбрано «пусто», то подразумевается вариант «EXPLICIT TAGS».

Примечание. Смысл вариантов «EXPLICIT TAGS» и «IMPLICIT TAGS» определяется в разделе 26.

9.3 «ССылкаНаМодуль», присутствующая в продукции «ОпределениеМодуля», называется именем модуля. Имена модулей выбирают таким образом, чтобы обеспечить полноту и согласованность всех последовательностей вида «присвоение», присутствующих в составных частях «ТелоМодуля» всех последовательностей вида «ОпределениеМодуля», в которых используется данное имя модуля. Множество последовательностей вида «присвоение» считают полным и согласованным, если для каждой содержащейся в ней последовательности вида «ССылкаНаТип» или «ССылкаНаЗначе-

ние», соответствует в точности одна последовательность вида «ПрисвоениеТипа» или «ПрисвоениеЗначения» соответственно, которые устанавливают связь между именем и, соответственно, типом или значением.

9.4 Имена модулей должны использоваться только один раз в сфере применения определения модуля, за исключением случаев, определенных в 9.10.

Примечание. Рекомендуется следующая форма имен модулей, определенных в стандартах ИСО:

ISOxxxx—yyyy

где xxxx — номер стандарта ИСО, а yyyy — подходящая аббревиатура для названия этого международного стандарта (например, JTM, FTAM или CCR). Аналогичное соглашение может использоваться другими организациями, выпускающими стандарты.

9.5 Если «ПрисвоенныйИдентификатор» включает «Значение-ИдентификаторОбъекта», то последний однозначно идентифицирует модуль.

Примечание. Рекомендуется назначать идентификатор объекта, чтобы позволить однозначные ссылки на данный модуль из других модулей.

9.6 «ИдентификаторМодуля» последовательности «СимволыИзМодуля» должен присутствовать в «ОпределенииМодуля» для некоторого другого модуля, однако, если он включает «Значение-ИдентификаторОбъекта», то «ССылкиНаМодуль» могут различаться в этих двух определениях.

Примечания

1 «ССылкаНаМодуль», отличная от используемой в другом модуле, может быть использована только в том случае, когда входные символы заимствуются из двух разных модулей с одинаковыми именами (модулей, имена которым были присвоены с нарушением 9.4. Использование альтернативных, отличных от исходных, имен позволяет сделать эти модули доступными для использования в теле определяемого модуля (см. 9.8).

2 Если ссылка на модуль осуществляется с использованием как «Ссылки-НаМодуль», так и «Значения-ИдентификаторОбъекта», то последнее является решающим.

9.7 Каждый «ВыходнойСимвол» должен быть определен в данном (определяемом) модуле.

Примечание. Рекомендуется все символы, ссылки на которые допустимы из других модулей, включать в «ВыходныеСимволы». Если таких символов нет, то следует выбрать альтернативный вариант «пусто» (но не «ВыходныеССылки»).

9.8 Каждый символ из конструкции «СимволыИзМодуля» должен быть определен в модуле, задаваемом «ИдентификаторомМодуля» конструкции «СимволыИзМодуля». Если в определении этого модуля используют конструкцию «ВыходныеССылки», то каждый символ должен присутствовать в ее «ВыходныхСимволах».

9.9 Любой «символ» из конструкции «СимволыИзМодуля» может присутствовать в «ТелеМодуля» в конструкции «ОпределенныйТип» (если символ — «ССылкаНаТип») или «Определенное Значение» (если символ — «ССылкаНаЗначение»). Смысл, связанный с этим «символом», — тот же, что и в модуле с соответствующим «ИдентификаторомМодуля». Если «символ» присутствует в одном из «присвоений» из «СпискаПрисвоений» (что не рекомендуется) или присутствует в одной или нескольких конструкциях «СимволыИзМодуля», то он должен использоваться только в конструкциях «ВнешняяССылкаНаТип» или «ВнешняяССылкаНа Значение», в которых «ССылкаНаМодуль» совпадает с ссылкой на модуль в конструкции «СимволыИзМодуля» (см. 9.10). Если он не присутствует в таких конструкциях, то он может непосредственно входить в конструкции «ОпределенныйТип» или «Определенное Значение».

9.10 За исключением случаев, отмеченных в 9.9, для обращения к элементам «ССылкаНаТип» и «ССылкаНаЗначение» в модулях, отличных от тех, где вышеупомянутые элементы были определены, используют последовательности вида «ВнешняяССылкаНаТип» и «ВнешняяССылкаНаЗначение», определяемые следующими продукциями:

```
ВнешняяССылкаНаТип ::=
                    ССылкаНаМодуль
                    .
                    ССылкаНаТип
ВнешняяССылкаНаЗначение ::=
                    ССылкаНаМодуль
                    .
                    ССылкаНаЗначение
```

10 ССЫЛКИ НА ОПРЕДЕЛЕНИЯ ТИПОВ И ЗНАЧЕНИЙ

10.1 Для задания ссылок на определения типов и значений используют последовательности, определяемые следующими продукциями:

```
ОпределенныйТип ::=
                    ВнешняяССылкаНаТип |
                    ССылкаНаТип
ОпределенноеЗначение ::=
                    ВнешняяССылкаНаЗначение |
                    ССылкаНаЗначение
```

10.2 За исключением случаев, отмеченных в 9.10, варианты «ССылкаНаТип» и «ССылкаНаЗначение» в приведенных выше про-

дукциях должны использоваться только в тех же модулях, в которых данной ссылке на тип или значение, соответственно, был присвоен некоторый тип или некоторое значение (см. 11.1 и 11.2).

10.3 «ВнешняяСсылкаНаТип» и «ВнешняяСсылкаНаЗначение» должны использоваться только в тех случаях, когда в модуле, определяемом элементом «СсылкаНаМодуль», данной ссылке на тип или значение был присвоен, соответственно, некоторый тип или некоторое значение (см. 11.1 и 11.2).

11 ПРИСВОЕНИЕ ТИПОВ И ЗНАЧЕНИЙ

11.1 Присвоение ссылке на тип некоторого типа описывается продукцией «ПрисвоениеТипа»:

ПрисвоениеТипа : : =
ССылкаНаТип
« : : = »
Тип

Элемент «ССылкаНаТип» не должен совпадать с каким-либо именем, используемым для ссылки на один из типов «строка знаков», определенных в части 2, а также не должен совпадать с каким-либо именем, используемым для ссылки на типы, определенные в части 3.

11.2 Присвоение ссылке на значение некоторого значения описывается продукцией «ПрисвоениеЗначения»:

ПрисвоениеЗначения : : =
ССылкаНаЗначение
Тип
« : : = »
Значение

Последовательность «Значение», присваиваемая ссылке на значение, должна быть допустимым выражением (см. 12.7), описывающем некоторое значение, имеющее тип, определяемый последовательностью «Тип».

12 ОПРЕДЕЛЕНИЕ ТИПОВ И ЗНАЧЕНИЙ

12.1 Тип определяется одной из последовательностей вида «Тип»:

Тип : : = ВстроенныйТип | ОпределенныйТип | Подтип
(см. 10.1) (см. 37)

ВстроенныйТип : : =
БулевскийТип
ЦелочисленныйТип

Тип-СтрокаБитов
Тип-СтрокаОктетов
ВырожденныйТип
Тип-Последовательность
Тип-ПоследовательностьИз
Тип-Множество
Тип-МножествоИз
ВыборочныТип
СелективныйТип
ТегированныйТип
ПроизвольныйТип
Тип-ИдентификаторОбъекта
Тип-СтрокаЗнаков
ОбщеупотребительныйТип
ПеречисленныйТип
Тип-ДействительноеЧисло

Примечания

1 В качестве последовательности вида «тип» также может использоваться обозначение типа, определенное с помощью макронотации.

2 В последующих версиях настоящего стандарта могут быть определены другие встроенные типы.

12.2 Обозначение «ВстроенныйТип» определяется в последующих разделах.

12.3 Обозначение «подтип» определяется в разделе 4.

12.4 Тип, на который осуществляется ссылка, либо определен обозначением «ВстроенныйТип», либо присвоен ссылке «ОпределенныйТип».

12.5 В некоторых обозначениях нотации, в которых имеется ссылка на какой-либо тип, этот тип может быть поименован. В этих случаях настоящий стандарт предписывает использование обозначения вида «ПоименованныйТип»:

ПоименованныйТип ::=
Идентификатор Тип
Тип
СелективныйТип

Обозначения вида «СелективныйТип» и обозначения для записи значений селективного типа определены в разделе 25.

Примечание. Обозначение вида «СелективныйТип» содержит «Идентификатор». Если в качестве «ПоименованногоТипа» используют «СелективныйТип» (см. 25.1), то этот идентификатор может быть частью обозначения для записи значения.

12.6 «Идентификатор» не является составной частью типа и никак не влияет на тип. Тип, на который дается ссылка последова-

тельностью «ПоименованныйТип», является тем же самым, что и тип, на который дается ссылка последовательностью «Тип», вложенной в последовательность «ПоименованныйТип».

12.7 Значение какого-либо типа должно определяться одной из последовательностей вида «Значение»:

Значение : : = ВстроенноеЗначение | ОпределенноеЗначение

ВстроенноеЗначение : : =

БулевскоеЗначение |
ЦелочисленноеЗначение |
Значение-СтрокаБитов |
Значение-СтрокаОктетов |
ВырожденноеЗначение |
Значение-Последовательность |
Значение-ПоследовательностьИз |
Значение-Множество |
ЗначениеМножествоИз |
ВыборочноеЗначение |
СелективноеЗначение |
ТегированноеЗначение |
ПроизвольноеЗначение |
Значение-ИдентификаторОбъекта |
Значение-СтрокаЗнаков |
ПеречислительноеЗначение |
Значение-ДействительноеЧисло .

Примечание. В качестве последовательности вида «Значение» также может использоваться обозначение значения, определенное с помощью макронотации (см. приложение А).

12.8 Если тип определяется с использованием одного из обозначений, приведенных слева в приведенном ниже перечне, то значение должно задаваться с использованием обозначения, приведенного справа.

Обозначение типа	Обозначение значения
БулевскийТип	БулевскоеЗначение
ЦелочисленныйТип	ЦелочисленноеЗначение
Тип-СтрокаБит	Значение-СтрокаБит
Тип-СтрокаОктетов	Значение-СтрокаОктетов
ВырожденныйТип	ВырожденноеЗначение
Тип-Последовательность	Значение-Последовательность
Тип-ПоследовательностьИз	Значение-ПоследовательностьИз
Тип-Множество	Значение-Множество

Тип-МножествоИз	Значение-множествоИз
ВыборочныйТип	ВыборочноеЗначение
ТегированныйТип	ТегированноеЗначение
ПроизвольныйТип	ПроизвольноеЗначение
Тип-ИдентификаторОбъекта	Значение-ИдентификаторОбъекта
Тип-СтрокаЗнаков	Значение-СтрокаЗнаков
ПеречислительныйТип	ПеречислительноеЗначение
Тип-ДействительноеЧисло	Значение-ДействительноеЧисло

Примечание. В последующих версиях настоящего стандарта могут быть определены дополнительные обозначения для записи значений.

В случаях, когда тип является «ОпределеннымТипом», обозначения, используемые для записи значения, должны соответствовать обозначениям, применяемым для типа, породившего «ОпределенныйТип».

12.9 Обозначения для записи значений типов, определяемых обозначением «ОбщеупотребительныйТип», приведены в разделе 3.

12.10 Обозначение «ВстроенноеЗначение» определяется в следующих разделах

12.11 Значение типа, ссылка на который задается обозначением вида «ПоименованныйТип», определяется обозначением вида «ПоименованноеЗначение»:

ПоименованноеЗначение ::=

Идентификатор Значение ,
Значение

где «Идентификатор» (если он задан) — тот же самый идентификатор, который использовался в обозначении «ПоименованныйТип». В 25.2 указаны дополнительные ограничения на «ПоименованноеЗначение» для случая, когда «ПоименованныйТип» является «СелективнымТипом».

Примечание. «Идентификатор» является составной частью обозначения, но не является составной частью значения.

12.12 «Идентификатор» является составной частью последовательности «ПоименованноеЗначение» в том и только том случае, когда он является составной частью последовательности «ПоименованныйТип».

Примечание. В том случае, когда используется вариант «СелективныйТип», «идентификатор» всегда включается в запись значения.

13 ОБОЗНАЧЕНИЯ ДЛЯ БУЛЕВСКОГО ТИПА

13.1 Для ссылки на булевский тип (см. 3.13) должно использоваться обозначение «БулевскийТип»:

БулевскийТип : : = BOOLEAN

13.2 Типы, определенные с помощью этого обозначения, имеют тег универсального класса 1.

13.3 Значение булевского типа (см. 3.14 и 3.15) должно определяться обозначением «БулевскоеЗначение»:

БулевскоеЗначение : : = TRUE | FALSE

14 ОБОЗНАЧЕНИЯ ДЛЯ ЦЕЛОЧИСЛЕННОГО ТИПА

14.1 Для ссылки на целочисленный тип (см. 3.16) должно использоваться обозначение «ЦелочисленныйТип»:

ЦелочисленныйТип : : =
INTEGER |
INTEGER {СписокПоименованныхЧисел}

СписокПоименованныхЧисел : : =
ПоименованноеЧисло |
СписокПоименованныхЧисел, ПоименованноеЧисло

ПоименованноеЧисло : : =
Идентификатор (ЧислоСоЗнаком) |
Идентификатор (ОпределенноеЗначение)

ЧислоСоЗнаком : : = Число | – Число

14.2 Вариант «число со знаком» не должен использоваться, если «число» является нулем.

14.3 «СписокПоименованныхЧисел» не является существенной составной частью определения типа. Эту последовательность используют только в обозначениях для записи значений, описанных в 14.9.

14.4 «ОпределенноеЗначение» является ссылкой на значение целочисленного типа или тегированного типа, производного от целочисленного типа.

14.5 Значение каждой из последовательностей «ЧислоСоЗнаком» и «ОпределенноеЗначение», входящих в «СписокПоименованныхЧисел», должны быть попарно различными и быть выделенными значениями целочисленного типа.

14.6 Элементы «идентификатор», входящие в «СписокПоименованныхЧисел», должны быть различными.

14.7 Порядок следования последовательностей «ПоименованноеЧисло» в последовательности «СписокПоименованныхЧисел» не существенен.

14.8 Типы, определенные с помощью этого обозначения, имеют тег универсального класса 2.

14.9 Значение целочисленного типа должно определяться обозначением вида «Целочисленное Значение»:

ЦелочисленноеЗначение : : =
ЧислоСоЗнаком |
Идентификатор

14.10 «Идентификатор» в продукции «ЦелочисленноеЗначение» должен совпадать с одной из последовательностей «Идентификатор», входящих в состав последовательности «ЦелочисленныйТип», которая определяет тип данного значения; «ЦелочисленноеЗначение» в этом случае определяется числом, соответствующим заданному идентификатору в описании целочисленного типа.

Примечание. При определении целочисленного значения, для которого был задан «Идентификатор» в определении типа, задание «ЦелочисленногоЗначения» в форме «Идентификатор» является предпочтительным.

15 ОБОЗНАЧЕНИЯ ДЛЯ ПЕРЕЧИСЛИТЕЛЬНОГО ТИПА

15.1 Перечислительный тип (см. 3.17) должен определяться обозначением «ПеречислительныйТип»:

ПеречислительныйТип : : =
ENUMERATED {Перечисление}
Перечисление : : =
ПоименованноеЧисло |
ПоименованноеЧисло, Перечисление

Примечания

1 Каждое значение имеет идентификатор, который связывается в этом обозначении с определенным числом. Это позволяет контролировать представление значений данного типа и облегчает согласованное расширение типов, но не предполагает наличие целочисленной семантики у значений данного типа.

2 Численные значения в обозначениях «ПоименованноеЧисло», входящих в «перечисление», не обязательно должны быть упорядоченными или смежными.

15.2 Для каждого «ПоименованногоЧисла» «Идентификатор» и «ЧислоСоЗнаком» должны отличаться от «Идентификаторов» и «ЧиселСоЗнаком» других «ПоименованныхЧисел», входящих в «Перечисление».

15.3 Перечислительный тип имеет тег универсального класса 10.

15.4 Значение перечислительного типа должно определяться обозначением «ПеречислительноеЗначение»:

ПеречислительноеЗначение : : = Идентификатор

16 ОБОЗНАЧЕНИЯ ДЛЯ ТИПА «ДЕЙСТВИТЕЛЬНОЕ ЧИСЛО»

16.1 Тип «действительное число» (см. 3.18) определяется обозначением «Тип-ДействительноеЧисло»:

Тип-ДействительноеЧисло : : = REAL

16.2 Значения типа «действительное число» включают значения PLUS-INFINITY и MINUS-INFINITY вместе с действительными числами, которые могут быть представлены следующей формулой, содержащей три целочисленных аргумента, M, B и E:

$$M \times B^E,$$

где M называется мантиссой, B — основанием, а E — порядком. M и E могут принимать любые целочисленные значения, как положительные, так и отрицательные, а B принимает значения 2 или 10. Допускаются любые комбинации M, B и E.

Примечания

1 Этот тип может передавать точное значение любого числа, которое может быть представлено в обычном оборудовании с плавающей запятой и любое число в десятичном представлении с конечным числом цифр.

2 Кодовое представление этого типа, определенное в ГОСТ Р ИСО/МЭК 8825, допускает использование оснований 2, 8 и 16 для двоичного представления действительных чисел и основания 10 для десятичного в виде последовательности знаков. Выбор одного из вариантов осуществляется отправителем.

16.3 Тип «действительное число» имеет тег универсального класса 9.

16.4 Обозначения для определения значения типа «действительное число» должны иметь вид «Значение-ДействительноеЧисло»:

```
Значение-ДействительноеЧисло : : =
                    ЧисленноеДействительноеЗначение |
                    СпециальноеДействительноеЗначение
ЧисленноеДействительноеЗначение : : =
                    {Мантисса, Основание, Порядок} | 0
Мантисса . : = ЧислоСоЗнаком
Основание : : = 2 | 10
Порядок : : = ЧислоСоЗнаком
СпециальноеДействительноеЗначение : : =
                    PLUS-INFINITY | MINUS-INFINITY
```

Для нулевых значений должен использоваться вариант «0» конструкции ЧисленноеДействительноеЗначение, а альтернативный вариант этой конструкции использоваться не должен.

17 ОБОЗНАЧЕНИЯ ДЛЯ ТИПА «СТРОКА БИТОВ»

17.1 Для ссылки на тип «строка битов» (см. 3.19) должно использоваться обозначение «Тип-СтрокаБитов»:

```
Тип-СтрокаБит : : =
BIT STRING |
```

BIT STRING {СписокПоименованныхБит}
СписокПоименованныхБит : : =
ПоименованныйБит |
СписокПоименованныхБит, ПоименованныйБит
ПоименованныйБит : : =
Идентификатор (Число) |
Идентификатор (ОпределенноеЗначение)

17.2 «СписокПоименованныхБит» не является существенной составной частью определения типа. Эта последовательность используется только в обозначениях для записи значений, определенных в 17.8.

17.3 Первый бит в строке бит имеет нулевой номер. Замыкающим битом называется бит, который заканчивает строку бит.

Примечание. Термины первый бит и замыкающий бит используют при описании как обозначений для задания значений, так и правил кодирования.

17.4 «ОпределенноеЗначение» должно представлять собой ссылку на некоторое неотрицательное значение целочисленного типа, помеченного типа, производного от целочисленного типа, или подтипа целочисленного типа.

17.5 Значения последовательностей «Число» и «ОпределенноеЗначение», входящих в состав последовательности «СписокПоименованныхБит», должны быть попарно различными; каждое из этих значений используют в качестве номера выделенного бита при задании значения строки бит.

17.6 Все элементы «Идентификатор», входящие в «СписокПоименованныхБит», должны быть различными.

Примечание. Порядок следования последовательностей «ПоименованныйБит» в последовательности «СписокПоименованныхБит» не существенен.

17.7 Этот тип имеет тег универсального класса под номером 3.

17.8 Значение типа «строка бит» задается нижеследующим обозначением «Значение-СтрокаБит»:

Значение-СтрокаБит : : =
b-строка |
h-строка |
{СписокИдентификаторов} |
{ }
СписокИдентификаторов : : =
Идентификатор |
СписокИдентификаторов, Идентификатор

17.9 Каждый элемент «Идентификатор», входящий в последовательность «Значение-СтрокаБит», должен совпадать с некото-

рым элементом «Идентификатор» в составе последовательности «Тип-Строка-Бит», определяющей тип, к которому относится данное значение.

17.10 Пользователь нотации определяет (и может указать это в комментарии), является ли существенным наличие или отсутствие последних нулевых бит.

Примечание. Правилами кодирования обеспечивается передача строк бит произвольной длины и с произвольным чередованием бит в строке.

17.11 Обозначение вида «{СписокИдентификаторов}» и «{ }» для последовательности «Значение-СтрокаБит» могут использоваться только в том случае, когда присутствие или отсутствие замыкающих нулевых бит несущественно. Этот вариант обозначения задает строку бит с единицами в битовых позициях с номерами, задаваемыми последовательностями «Идентификатор», и со всеми остальными битами, равными нулю.

Примечание. Последовательность «{ }» используют для обозначения значения строки бит, не содержащей ни одного бита, равного единице.

17.12 При описании правил кодирования для строк бит для ссылок на соответствующие биты используют термины первый бит и замыкающий бит, определенные выше.

17.13 При использовании обозначения «b-строка» первый бит записывают слева, а замыкающий бит — справа.

17.14 При использовании обозначения вида «h-строка» старший бит каждой шестнадцатеричной цифры соответствует начальному (первому слева) биту в представляющей ее строке бит.

Примечание. Описанные обозначения никоим образом не ограничивают способа размещения строки бит по октетам для передачи, который реализуется правилами кодирования.

17.15 Обозначение «h-строка» может использоваться только в следующих случаях:

a) если значение строки бит состоит из количества бит, кратного четырем, или

b) присутствие или отсутствие последних нулевых бит несущественно.

Пример

A98A'H и

'1010100110001010'B

являются эквивалентными обозначениями одного и того же значения битовой строки.

18 ОБОЗНАЧЕНИЯ ДЛЯ ТИПА «СТРОКА ОКТЕТОВ»

18.1 Для ссылки на тип «строка октетов» (см. 3.20) должно использоваться обозначение «Тип-СтрокаОктетов»:

Тип-СтрокаОктетов : : = OCTET STRING

18.2 Этот тип имеет тег универсального класса 4.

18.3 Значение типа «строка октетов» должно определяться обозначением вида «Значение-СтрокаОктетов»:

Значение-СтрокаОктетов : : =
b-строка |
h-строка

18.4 При описании правил кодирования для строк октетов для ссылок на соответствующие октеты используют термины первый октет и замыкающий октет, а для ссылок на соответствующие биты внутри октета — термины «старший бит» и «младший бит».

18.5 При использовании обозначения вида «b-строка» самый левый бит является старшим битом первого октета. Если «b-строка» состоит из некоторого количества бит, не кратного восьми, она должна интерпретироваться таким образом, как если бы она содержала такое наименьшее количество дополнительных замыкающих нулевых бит, которое требуется для обеспечения кратности восьми.

18.6 При использовании обозначения вида «h-строка» крайняя левая шестнадцатеричная цифра является старшим полуоктетом первого октета. Если «h-строка» состоит из нечетного числа шестнадцатеричных цифр, то она должна интерпретироваться таким образом, как если бы она содержала еще одну дополнительную замыкающую нулевую шестнадцатеричную цифру.

19 ОБОЗНАЧЕНИЯ ДЛЯ ВЫРОЖДЕННОГО ТИПА

19.1 Для ссылки на вырожденный тип (см. 3.21) следует использовать обозначение «ВырожденныйТип»:

ВырожденныйТип : : = NULL

19.2 Этот тип имеет тег универсального класса 5.

19.3 Для ссылок на значение вырожденного типа должно использоваться обозначение «ВырожденноеЗначение»:

ВырожденноеЗначение : : = NULL.

20 ОБОЗНАЧЕНИЯ ДЛЯ ТИПОВ «ПОСЛЕДОВАТЕЛЬНОСТЬ»

20.1 Для построения типа «последовательность» (см. 3.22) из других типов используют определяемое ниже обозначение «Тип-Последовательность»:

```
Тип-Последовательность : : =
        SEQUENCE {СписокТипов-Компонентов} |
        SEQUENCE { }
СписокТипов-Компонентов : : =
        Тип-Компонент |
        СписокТипов-Компонентов, Тип-Компонент
Тип-Компонент : : =
        ПоименованныйТип                          |
        ПоименованныйТип OPTIONAL                 |
        ПоименованныйТип DEFAULT Значение         |
        COMPONENTS OF Тип
```

20.2 «Тип» в четвертом варианте продукции «Тип-компонент» должен быть типом «последовательность». Использование варианта «COMPONENTS OF Тип» определяет включение, в этом месте, в «СписокТипов-Компонентов» всех последовательностей «Тип-Компонент», имеющихся в типе, заданном обозначением «Тип».

Примечание. Это преобразование считают логически завершенным до удовлетворения требований, описанных в последующих подразделах.

20.3 Для каждой последовательности из одного или нескольких следующих друг за другом «Типов-Компонентов», отмеченных как «OPTIONAL» или «DEFAULT», теги этих «Типов-Компонентов» должны быть различными и отличаться от тега «Типа-Компонента», непосредственно следующего за этой последовательностью. (см. раздел 26).

20.4 Если используют ключевые слова «OPTIONAL» или «DEFAULT», то при формировании значения нового типа, а также при формировании, в соответствии с правилами кодирования, информации для передачи значение соответствующего типа-компонента может быть опущено.

Примечания

1 Если элементы «Идентификатор» присутствуют не в каждой последовательности «ПоименованнойТип», то обозначение значения для нового типа может оказаться неоднозначным.

2 Правилами кодирования гарантируется, что кодирование значения-последовательности, в котором был опущен элемент, заданный с использованием ключевых слов «DEFAULT» или «OPTIONAL», будет совпадать с кодированием такого значения последовательности, для которого соответствующий тип-компонент был опущен при описании типа. Это может оказаться полезным при определении подмножеств.

20.5 Пропуск значения типа, заданного как «DEFAULT», в точности эквивалентен включению значения, определенного последовательностью «Значение», которая должна быть допустимой последовательностью для обозначения значения типа, определенного

последовательностью «Тип» в последовательности «ПоименованныйТип».

20.6 Элементы «Идентификатор» (если они заданы) должны быть различными во всех последовательностях вида «ПоименованныйТип», входящих в последовательность «СписокТипов-Компонентов».

20.7 Все типы «последовательность» имеют тег универсального класса 16.

Примечание. Типы «последовательность из» имеют тот же самый тег (см. 21.3).

20.8 Обозначения для определения значения типа «последовательность» должны иметь вид «Значение-Последовательность»:

```
Значение-Последовательность : : =
                    {СписокЗначений-Компонентов}
                    { }
СписокЗначений-Компонентов : : =
    ПоименованноеЗначение |
    СписокЗначений-Компонентов, ПоименованноеЗначение
```

20.9 Обозначение «{ }» используют лишь в следующих случаях:

а) все последовательности вида «Тип-Компонент» в последовательности «Тип-Последовательность» отмечены как «DEFAULT» или «OPTIONAL», и все значения опущены; или

b) обозначение типа имело вид «SEQUENCE { }».

20.10 Для каждой последовательности «ПоименованныйТип», входящей в «Тип-Последовательность», которая не отмечена как «DEFAULT» или «OPTIONAL», в записи значения типа «последовательность» должно быть включено некоторое значение этого поименованного типа, в порядке следования последовательностей «ПоименованныйТип».

Примечание. Использование последовательностей «ПоименованныйТип», не содержащих идентификатор, не запрещается, но в случае использования DEFAULT» или «OPTIONAL» может привести к неоднозначным обозначениям.

21 ОБОЗНАЧЕНИЯ ДЛЯ ТИПОВ «ПОСЛЕДОВАТЕЛЬНОСТЬ ИЗ»

21.1 Обозначения для построения типа «последовательность из» (см. 3.23) из другого типа должны иметь вид «Тип-ПоследовательностьИз»:

```
Тип-ПоследовательностьИз : : = SEQUENCE OF Тип |
                               SEQUENCE
```

21.2 Обозначение «SEQUENCE» эквивалентно обозначению «SEQUENCE OF ANY» (см. раздел 27).

21.3 Все типы «последовательность из» имеют тег универсального класса 16.

Примечание. Типы «последовательность» имеют тот же самый тег (см. 20.7).

21.4 Для задания значения типа «последовательность из» используют определяемое ниже обозначение «Значение-ПоследовательностьИз»:

Значение-ПоследовательностьИз : : =
{СписокЗначений} | { }

СписокЗначений : : = Значение |
СписокЗначений, Значение

Обозначение «{ }» используют в тех случаях, когда в значении типа «последовательность из» нет ни одного значения-компонента.

21.5 Каждая последовательность «Значение», входящая в «СписокЗначений», должна быть обозначением некоторого значения того «Типа», который задан определении типа «последовательность из».

Примечание. Порядку следования этих значений может соответствовать некоторый семантический смысл.

22 ОБОЗНАЧЕНИЯ ДЛЯ ТИПА «МНОЖЕСТВО»

22.1 Обозначения для определения типа «множество» (см. 3.24) на основе других типов должны иметь вид «Тип-Множество»:

Тип-Множество : : = SET {СписокТипов-Компонентов} |
SET { }

«СписокТипов-Компонентов» определен в 20.1.

22.2 «Тип» в четвертом варианте продукции «Тип-Компонент» (см. 20.1) должен быть типом «множество». Использование варианта «COMPONENTS OF Тип» определяет включение в «СписокТипов-Компонентов» всех последовательностей «Тип-Компонент», имеющихся в типе «множество».

Примечание. Это преобразование считают логически завершенным до удовлетворения требований, описанных в последующих подразделах.

22.3 Все «Типы-компоненты» в типе «множество» должны иметь разные теги (см. раздел 26).

22.4 Для типов «множество» применимы формулировки 20.4—20.6.

22.5 Все типы «множество» имеют тег универсального класса 17.

Примечание. Типы «множество из» имеют тот же самый тег (см. 23.3).

22.6 Порядку следования значений-компонентов в типе «множество» не придают никакого семантического значения.

22.7 Обозначения для определения значения типа «множество» должны иметь вид «Значение-Множество»:

Значение-Множество : : = {СписокЗначений-Компонентов} | { }

Последовательность «СписокЗначений-Компонентов» определена в 20.8.

22.8 Вариант «{ }» в продукции «Значение-множество» может быть применен только в следующих случаях:

a) все последовательности «Тип-Компонент» в продукции «Тип-Множество» заданы как DEFAULT» или «OPTIONAL», и все значения-компоненты опущены, или

b) тип был задан последовательностью «SET{ }».

22 9 Для каждого «ПоименованногоТипа» в «Типе-Множестве», который не был задан как «ORTIONAL» или «DEFAULT», в записи значения должна присутствовать соответствующая последовательность «ПоименованноеЗначение».

Примечания

1 «ПоименованныеЗначения» могут следовать в произвольном порядке.

2 Использование последовательностей «ПоименованныйТип», в которых отсутствует идентификатор, не запрещается, но может привести к неоднозначности.

23 ОБОЗНАЧЕНИЯ ДЛЯ ТИПОВ «МНОЖЕСТВО ИЗ»

23.1 Обозначения для определения типа «множество из» (см. 3.25) на основе другого типа должны иметь вид «Тип-МножествоИз»:

Тип-МножествоИз : : = SET OF Тип ¦
SET

23.2 Обозначение «SET» эквивалентно обозначению «SET OF ANY» (см. 27).

23.3 Все типы «множество из» имеют тег универсального класса 17.

Примечание. Типы «множество» имеют тот же самый тег (см. 22.5).

23.4 Обозначения для определения значения типа «множество из» должны иметь вид «Значение-МножествоИз»:

Значение-МножествоИз : : = {СписокЗначений} ¦ { }

Последовательность «СписокЗначений» определена в 21.4

Обозначение «{ }» используют в том случае, когда значения-компоненты в значении-множестве отсутствуют.

23.5 Каждое «Значение» в «СпискеЗначений» должно быть обозначением некоторого значения «Типа», который используют в последовательности «Тип-МножествоИз».

Примечания

1 Порядок следования значений-компонентов не имеет никакого семантического значения.

2 К правилам кодирования не предъявляется требования сохранения порядка следования этих значений.

24 ОБОЗНАЧЕНИЯ ДЛЯ ВЫБОРОЧНЫХ ТИПОВ

24.1 Обозначения для определения выборочного типа (см. 3.27), исходя из имеющихся других типов, должны иметь вид «ВыборочныйТип»:

ВыборочныйТип : : = CHOICE{СписокТипов-Вариантов}
СписокТипов-Вариантов : : =
ПоименованныйТип
СписокТипов-Вариантов, ПоименованныйТип

Примечания

1 Правила кодирования обеспечивают идентичность способа кодирования выбранного значения выборочного типа способу кодирования такого же значения типа-варианта.

2 Кодирование значений «ВыборочногоТипа» с единственным типом-вариантом «ПоименованныйТип» в «СпискеТипов-Вариантов» невозможно отличить от кодирования соответствующих значений в случае прямого задания «Типа», входящего в последовательность «ПоименованныйТип».

24.2 Все типы, входящие в «СписокТипов-Вариантов», должны иметь различные теги (см. раздел 26).

24.3 Тег выборочного типа следует считать переменной. При выборе конкретного значения выборочного типа его тег становится равным тегу того типа-варианта, к которому относится значение.

24.4 Если выборочный тип используют в таких конструкциях, где требованием настоящего стандарта является использование типов с различными тегами (см 20.3, 22.3 и 24.2), то теги всех типов, определенных в «СпискеТипов-Вариантов», должны отличаться от тегов других типов (см. раздел 26). Приведенные ниже примеры иллюстрируют это требование. Примеры 1 и 2 являются примерами правильного использования нотации. Пример 3 является примером неправильного использования нотации, поскольку теги типов d, f, а также e, g совпадают.

ПРИМЕРЫ

```
1  A : : =CHOICE
       {b B
       c NULL}
   B : : =CHOICE
       {d [0] NULL,
       e [1] NULL}
2  A : : =CHOICE
       {b B,
       c C}
   B : : =CHOICE
       {d [0] NULL
       e [1] NULL}
```

```
C ::=CHOICE
    {f [2] NULL,
    g [3] NULL}
```

3 (НЕПРАВИЛЬНОЕ ОПИСАНИЕ)

```
A ::=CHOICE
    {b B,
    c C}
B ::=CHOICE
    {d [0] NULL
    e [1] NULL}
C ::=CHOICE
    {f [0] NULL,
    g [1] NULL}
```

24.5 Все «Идентификаторы» (если они заданы) в последовательностях «ПоименованныйТип», входящих в «СписокТипов-Вариантов», должны быть различными.

24.6 Если выборочный тип используют в таких конструкциях, где требованием настоящего стандарта является использование «ПоименованныхТипов» с различными «Идентификаторами», то «Идентификаторы», если они заданы, всех «ПоименованныхТипов» в «СпискеТипов-Вариантов» должны отличаться от всех «Идентификаторов» других «ПоименованныхТипов» (входящих в конструкцию).

24.7 Обозначения, используемые для определения значения выборочного типа, определяются продукцией «ВыборочноеЗначение»:

ВыборочноеЗначение : : = ПоименованноеЗначение

24.8 Если «ПоименованноеЗначение» содержит «Идентификатор», то оно должно быть записью значения того типа в «СпискеЗначений-Вариантов», который имеет тот же «Идентификатор». Если «ПоименованноеЗначение» не содержит «Идентификатора», то оно должно быть записью значения одного из тех типов в «СпискеЗначений-Вариантов», которые не имеют «Идентификатора».

Примечание Пропуск идентификатора в «ПоименованномТипе» может привести к неоднозначности.

25 ОБОЗНАЧЕНИЯ ДЛЯ СЕЛЕКТИВНЫХ ТИПОВ

25.1 Поименованный тип в «СпискеТипов-Вариантов» «ВыборочногоТипа» может быть задан с помощью обозначения «СелективныйТип»:

СелективныйТип : : = Идентификатор<Тип

где «Тип» является обозначением, задающим «ВыборочныйТип», а «Идентификатор» является одним из «Идентификаторов» «ПоименованныхТипов» (входящих в «ВыборочныйТип»).

Примечание «СелективныйТип» может использоваться как в качестве «ПоименованногоТипа», и в этом случае «Идентификатор» используют при записи значения, так и в качестве «Типа» в «ПоименованномТипе», и в этом случае его «Идентификатор» при записи значения не используют.

25.2 Обозначения для записи значения селективного типа определяются продукцией «СелективноеЗначение»:

СелективноеЗначение : : = ПоименованноеЗначение

причем в «ПоименованноеЗначение» входит идентификатор, присутствующий в соответствующей последовательности «СелективныйТип», в том и только в том случае, когда «СелективныйТип» используют как «ПоименованныйТип».

26 ОБОЗНАЧЕНИЯ ДЛЯ ТЕГИРОВАННЫХ ТИПОВ

Тегированный тип (см. 3.26) является новым типом, изоморфным исходному типу, но имеющим другой тег. При всех способах кодирования обеспечивается отличие значения нового типа от значения исходного типа. Тегированный тип используют главным образом в тех случаях, когда требованием настоящего стандарта является использование типов с различными тегами (см. 20.3, 22.3, 24.2, 24.4 и 27.6).

Примечание. Различные теги могут понадобиться для правильного декодирования значения получателем в тех случаях, когда протокол допускает передачу значений различных типов данных в произвольный момент времени.

26.1 Обозначения для тегированного типа должны иметь вид «ТегированныйТип»:

```
ТегированныйТип : : = Тег Тип
                     Тег IMPLICIT Тип
                     Тег EXPLICIT Тип
Тег : : = [Класс НомерВКлассе]
НомерВКлассе : : = Число | ОпределенноеЗначение
Класс . . . . UNIVERSAL
              APPLICATION
              PRIVATE
              пусто
```

26.2 «ОпределенноеЗначение» - должно быть ссылкой на неотрицательное значение целочисленного типа или производного от целочисленного помеченного типа.

26.3 Новый тип изоморфен исходному типу, но имеет тег с классом, заданным последовательностью «Класс» и номер, заданный последовательностью «НомерВКлассе», если для последовательности «Класс» был выбран вариант, отличный от «пусто»; если же был выбран вариант «пусто», то тег относится к конте кст-

но-зависимому классу и имеет номер, заданный последовательностью «НомерВКлассе».

26.4 Для типов, определяемых вне рамок настоящего стандарта, «Класс» не должен быть универсальным классом («UNIVERSAL»).

Примечание. Использование тегов универсального класса периодически согласовывается между ИСО и МККТТ.

26.5 Если «Класс» тега является прикладным («APPLICATION»), то тот же самый «Тег» не может быть использован повторно в том же самом модуле.

26.6 Если «Класс» тега является пользовательским («PRIVATE»), то тег доступен для использования по правилам, устанавливаемых организациями-пользователями.

26.7 Конструкция тегирования задает явное тегирование, если выполняется одно из следующих условий:

a) используется вариант «Тег EXPLICIT Тип»;

b) используется вариант «Тег Тип» и значение конструкции «ТегиПоУмолчанию» для этого модуля — «EXPLICIT TAGS».

c) используется вариант «Тег Тип» и значение конструкции «ТегиПоУмолчанию» для этого модуля — «IMPLICIT TAGS», но тип, определенный конструкцией «Тип», — выборочный или неопределенный.

26.8 Если «Классом» тега является «пусто», то на использование «Тега» не накладывается никаких ограничений, кроме тех, которые вызваны требованиями различия тегов, сформулированными в 20.3, 22.3 и 24.2.

26.9 Вариант «IMPLICIT» является обозначением того, что для тех правил кодирования, которые обеспечивают эту возможность, явное указание тега, который имел исходный «Тип», не требуется при передаче данных.

Примечание. Сохранение старого тега может оказаться полезным, если он относится к универсальному классу, и поэтому однозначно идентифицирует исходный тип при отсутствии информации об определении, с помощью АСН.1, нового типа. Однако с помощью варианта «IMPLICIT» обеспечивается минимальное количество октетов при передаче. Пример кодирования для использования варианта «IMPLICIT» дан в ГОСТ 34.974.

26.10 Вариант «IMPLICIT» не должен использоваться, если тип, определенный последовательностью «Тип», является выборочным («CHOICE») или произвольным типом («ANY»).

26.11 Обозначения для определения значения тегированного типа должны иметь вид «ТегированноеЗначение»:

ТегированноеЗначение : : = Значение

где «Значение» является обозначением значения того типа, кото-

рый был задан последовательностью «Тип» в продукции «ТегированныйТип».

Примечание Отметим, что в этом обозначении отсутствует «Тег»

27 ОБОЗНАЧЕНИЯ ДЛЯ ПРОИЗВОЛЬНОГО ТИПА

27.1 Обозначение для произвольного типа (см. 3.29) определяется продукцией «ПроизвольныйТип»:

ПроизвольныйТип : : = ANY |
ANY DEFINED BY идентификатор

Примечание Использование «ANY» приводит к неполной спецификации, если не сопровождается дополнительной спецификацией. Конструкция «ANY DEFINED BY» предоставляет возможность задания типа, подставляемого вместо «ANY», во время обмена информацией, а также указывать на его семантику. Если выполняются описанные ниже правила использования этого типа, то он может обеспечить полную спецификацию Использование «ANY» без «DEFINED BY» не рекомендуется

27.2 Вариант «DEFINED BY» должен использоваться только в том случае, когда произвольный тип, или тип, полученный из него тегированием, является типом-компонентом типа «последовательность» или «множество» (содержащего произвольный тип).

27.3 «Идентификатор» в варианте «DEFINED BY» должен также присутствовать в конструкции «ПоименованныйТип», которая задает другой, обязательно присутствующий (то есть заданный не как «OPTIONAL») тип-компонент содержащего типа. «ПоименованныйТип» должен относиться либо к целочисленному типу, либо к идентификатору объекта.

27.4 Если «ПоименованныйТип» является целочисленным, перечислительным типом или типом «идентификатор объекта», либо типом, полученным из вышеупомянутых типов тегированием или образованием подтипа, то документ, использующий обозначение «DEFINED BY», должен содержать или явно ссылаться на единый список, в котором задано, какой тип АСН.1 должен быть подставлен вместо «ANY» для каждого разрешенного значения целочисленного типа. Этот список должен быть одним и тем же для всех случаев обмена информацией, затрагивающих содержащий тип.

27.5 Если «ПоименованныйТип» является типом «идентификатор объекта» или типом, полученным из него тегированием, то возникает необходимость в справочнике, «регистре», который для каждого выделенного значения «идентификатор объекта» связывал с ним единственный тип АСН.1 (возможно, выборочный тип), который следует подставить вместо типа «ANY».

Примечания

1 Возможно наличие произвольного количества регистров, связывающих значение-идентификатор объекта с типом АСН.1 для этих целей.

2 Регистрацию значений для идентификации типов при взаимосвязи открытых систем предполагается осуществлять в стандартах ИСО и рекомендациях МККТТ, в которых используют эти обозначения. В случаях, когда требуется специальная международная регистрационная служба для некоторого случая использования «ANY DEFINED BY», это должно быть явно указано в документе, использующем эти обозначения.

3 Основное различие между записью значений для идентификации типов в виде целого числа и идентификатора объекта заключается в том, что число задает один законченный список, полностью содержащийся в стандарте, использующей это обозначение, тогда как идентификатор объекта допускает использование открытого списка типов, определяемого регистрационной службой, выделяющей идентификаторы объектов.

27.6 Произвольный тип имеет неопределенный тег; он не должен использоваться в случаях, где требованием настоящего стандарта является использование типов с различными тегами (см 20.3, 22.3, 24.3 и 24.4).

27.7 Запись значения произвольного типа должна осуществляться по правилам АСН.1 и определяется следующей продукцией «ПроизвольноеЗначение».

ПроизвольноеЗначение : : = Тип Значение

где «Тип» — обозначение некоторого типа, а «Значение» обозначение для записи значения этого типа.

28 ОБОЗНАЧЕНИЯ ДЛЯ ТИПА «ИДЕНТИФИКАТОР ОБЪЕКТА»

28.1 Ссылка на тип «идентификатор объекта» (см. 3.33) должна иметь вид «Тип-ИдентификаторОбъекта»:

Тип-ИдентификаторОбъекта : : = OBJECT IDENTIFIER

28.2 Этот тип имеет тег универсального класса 6.

28.3 Обозначения для записи идентификатора объекта должны иметь вид «Значение-ИдентификаторОбъекта»:

```
Значение ИдентификаторОбъекта : : =
    {СписокКомпонентовИдОбъекта} |
    {ОпределенноеЗначение СписокКомпонентовИдОбъекта}
СписокКомпонентовИдОбъекта : : =
    КомпонентИдОбъекта |
    КомпонентИдОбъекта СписокКомпонентовИдОбъекта
КомпонентИдОбъекта : : = ИменнаяФорма |
                         ЧисловаяФорма |
                         СмешаннаяФорма
Именная Форма : : = Идентификатор
Числовая Форма : : = Число | ОпределенноеЗначение
Смешанная Форма : : = Идентификатор (ЧисловаяФорма)
```

28.4 Определенное значение в продукции «ЧисловаяФорма» должно быть ссылкой на значение целочисленного или помеченного типа, производного от целочисленного.

28.5 Определенное значение в продукции «Значение-ИдентификаторОбъекта» должно быть значением целочисленного или тегированного типа, производного от целочисленного.

28.6 «ИменнаяФорма» используется только для тех компонентов идентификатора объекта, численные значения которых определены в приложениях B—D к настоящему стандарту, и должна совпадать с одним из идентификаторов, определенных в этих приложениях.

28.7 «Число» в продукции «ЧисловаяФорма» должно быть числовым значением, присвоенным компоненту идентификатора объекта.

28.8 «Идентификатор» в продукции «СмешаннаяФорма» задается в случаях, когда данному компоненту идентификатора объекта присвоено целочисленное значение.

28.9 Семантическое значение идентификаторов задается деревом идентификаторов объектов. Дерево идентификаторов объектов является графом вида «дерево с корнем», причем корень дерева соответствует настоящему стандарту, вершины соответствуют службам — источникам идентификации, осуществляющим выделение ребер, исходящих из данной вершины. Каждому ребру дерева ставится в соответствие компонент идентификатора объекта, являющийся целочисленным значением. Каждому информационному объекту, подлежащему идентификации, соответствует в точности одна вершина (обычно — висящая вершина) и отсутствуют другие информационные объекты, которым назначена та же вершина. Таким образом, информационный объект однозначно и непротиворечиво идентифицируется последовательностью числовых значений (компонентов идентификаторов объектов), которые соответствуют ребрам в пути от корня графа до вершины, соответствующей данному информационному объекту.

Примечание. Значения вида «идентификатор объекта» состоят, по крайней мере, из двух компонентов, как определяется в приложениях B—D.

28.10 Значение вида «идентификатор объекта» представляет из себя упорядоченный список значений вида «компонент идентификатора объекта». Начиная с корня дерева идентификаторов объектов, каждое значение вида «компонент идентификатора объекта» соответствует ребру этого дерева. Последнее из этих значений идентифицирует ребро, ведущее к вершине, соответствующей информационному объекту. Это — тот самый информационный объект, который идентифицируется данным значением вида «иден-

тификатор объекта». Существенной частью значения вида — «компонент идентификатора объекта» является «именная форма» или «числовая форма», к которой сводится «именная форма». «Числовая форма» («NumberForm») представляет собой численное значение компонента идентификатора объекта.

Примечание. Как правило, информационный объект представляет собой класс информации (например, формат файла), а не конкретный представитель этого класса (например, конкретный файл). Таким образом, каждая вершина дерева задает некоторый информационный класс, (определяемый некоторым описанием, на которое дается ссылка), а не какой-то объем информации, относящейся к этому классу.

28.11 В случаях, когда «Значение-ИдентификаторОбъекта» включает «ОпределенноеЗначение», список компонентов идентификатора объекта, задаваемый этим «ОпределеннымЗначением», ставится непосредственно перед компонентами, явно присутствующими в значении.

Примеры

С учетом присвоения идентификаторов, описанного в приложении В, значения:

{iso standard 8571 pci(1)}

и

{ISO 8571 1}

идентифицируют один и тот же объект — протокольную управляющую информацию («pci»), определенную в ИСО 8571.

С учетом следующего дополнительного определения

ftam OBJECT IDENTIFIER : : =
{iso standard 8571}

значение

{ftam pci(1)}

также эквивалентно значениям, приведенным выше.

Примечание. Если в каком-либо международном стандарте, рекомендации МККТТ или другом документе содержится присвоение значений типа «идентификатор объекта» («OBJECT IDENTIFIER»), то рекомендуется снабжать этот документ приложением или дополнением, содержащим перечень всех содержащихся в нем присвоений. Желательно также, чтобы службы, присваивающие значения типа «идентификатор объекта» («OBJECT IDENTIFIER») информационным объектам, присваивали им также значения типа «описатель объекта».

29 ОБОЗНАЧЕНИЯ ДЛЯ ТИПОВ «СТРОКА ЗНАКОВ»

29.1 Обозначения для задания ссылки на тип «строка знаков» (см. 3.12 и раздел 31) должны иметь вид:

Тип-СтрокаЗнаков : : = СсылкаНаТип

где «СсылкаНаТип» — одно из имен типов «строка знаков», перечисленных в разделе 31.

29.2 Теги всех типов «строка знаков» определены в разделе 31.

29.3 Обозначения для записи значения «строка знаков» должны иметь вид:

Значение-СтрокаЗнаков : : = с-строка

Знаки, которые могут присутствовать в «строке знаков» («с-строке»), определятся типом, к которому относится строка.

30 ОБОЗНАЧЕНИЯ ДЛЯ ТИПОВ, ОПРЕДЕЛЕННЫХ В РАЗДЕЛАХ 32—35

30.1 Обозначения для задания ссылки на один из типов, определенных в разделах 32—35, должны быть следующими:

ОбщеупотребительныйТип : : = СсылкаНаТип

где «СсылкаНаТип» — одна из ссылок на тип, определенных в разделах 32—35 с помощью нотации АСН.1.

30.2 Теги для каждого из общеупотребительных типов определены в разделах 32—35.

30.3 Обозначения для записи значения общеупотребительного типа определяются в разделах 32—35.

31 ОПРЕДЕЛЕНИЕ ТИПОВ «СТРОКА ЗНАКОВ»

В этом разделе определяются типы, значения которых являются последовательностями, состоящими из пустого множества, одного или более знаков некоторого набора знаков.

31.1 Для определения типа («строка знаков») необходимо задать следующее:

а) тег, присвоенный типу;

b) имя, с помощью которого осуществляется ссылка на определение типа;

в) набор знаков, использующихся в данном типе; набор знаков задается либо с помощью ссылки на таблицу, содержащую изображения знаков этого набора, либо ссылкой на регистрационный номер в Международном регистре наборов кодируемых знаков, используемый в Esc-последовательностях (последовательностях смены набора знаков).

Имя из b) может использоваться как «СсылкаНаТип» при задании типа с помощью нотации АСН.1 (см. раздел 29).

31.2 В таблице 6 приведены имена, с помощью которых могут осуществляться ссылки на определения типов, номера тегов универсального класса, присвоенных типам, и регистрационные номера или номера приведенных ниже таблиц (определяющие набор зна-

ков). В случае необходимости строка таблицы сопровождается примечанием. В случаях, когда для типа определено имя-синоним, оно приводится в скобках.

Примечание. Тег, назначенный типу «строка знаков», однозначно идентифицирует этот тип. Отметим, однако, что в случаях, когда с помощью АСН.1 определяются новые типы, исходя из имеющегося типа «строка знаков» (в особенности, с использованием ключевого слова «IMPLICIT»), может оказаться невозможным распознавание этих типов без знания их определений.

31.3 В таблице 4 перечислены знаки, которые могут присутствовать в значении типа, имеющего имя NumericString (числовая строка).

Таблица 4 — **Тип NumericString (числовая строка)**

Наименование	Изображение
Цифры	0, 1, . . . 9
Пробел	(пробел)

31.4 В таблице 5 перечислены знаки, которые могут присутствовать в значении типа, имеющего имя PrintableString (печатная строка).

Таблица 5 — **Тип PrintebleString (печатная строка)**

Наименование	Изображение
Прописные буквы	A, B . . Z
Строчные буквы	a, b, . . z
Цифры	0, 1, . . 9
Пробел	(пробел)
Апостроф	'
Левая скобка	(
Правая скобка	)
Знак плюс	+
Запятая	,
Дефис	—
Точка	.
Косая черта	/
Двоеточие	:
Знак равенства	=
Вопросительный знак	?

31.5 Обозначения для всех значений типов «строка знаков» должны иметь вид «с-строка».

Примечание. Это обозначение может быть использовано только в том случае, когда используют средства отображения, допускающие изображение знаков, составляющих данное значение. Обозначения для записи значения в других случаях не определены.

31.6 Во всех случаях набор допустимых знаков может ограничиваться комментарием, но не может быть расширен.

Таблица 6 — **Список типов «строка знаков»**

Имя для ссылки на тип	Номер тега универсального класса	Регистрационные номера (см. ИСО 2375) или номер таблицы, определяющей набор знаков	Примечания
NumericString	18	Таблица 4	(1)
PrintebleString	19	Таблица 5	(1)
TeletexString (T61String)	20	87, 102, 103, 106, 107 +ПРОБЕЛ+ЗАБОЙ	(2)
VideotexString	21	1, 72, 73, 102, 108, 128, 129 +ПРОБЕЛ +ЗАБОЙ	(3)
VisibleString (ISO646String)	26	2+ПРОБЕЛ	
IA5String	22	1, 2+ПРОБЕЛ+ЗАБОЙ	
GraphicString	25	Все наборы G +ПРОБЕЛ	
GeneralString	27	Все наборы G и C +ПРОБЕЛ+ЗАБОЙ	

Примечания

1 Для данного типа такие характеристики, как шрифт, размер, цвет, яркость и т. п. несущественны.

2 Для случаев, соответствующих этим регистрационным номерам, правила использования определяются рекомендацией МККТТ Т.61.

3 Для случаев, соответствующих этим регистрационным номерам, обеспечиваются функциональные возможности, определяемые рекомендациями МККТТ Т.100 и Т.101.

32 ОБЩАЯ ФОРМА ЗАПИСИ ВРЕМЕНИ

32.1 Для ссылок на этот тип должно использоваться имя GeneralizedTime.

32.2 Этот тип состоит из значений, содержащих:

a) календарную дату в форме, определенной ИСО 2014 («Записи календарной даты в числовом представлении»);

b) время суток с любой точностью, предусмотренной в разделе 2 ИСО 3307 («Формы представления времени суток»);

c) сдвиг местного времени относительно всемирного времени в виде, описанном в ИСО 4031 («Представление разности местного времени по отношению ко всемирному времени»)

32.3 Описываемый тип может быть определен с помощью нотации АСН.1 следующим образом:

GeneralizedTime : : =

[UNIVERSAL 24] IMPLICIT VisibleString

где значения «отображаемой строки» («VisibleString») ограничиваются строками знаков, удовлетворяющих одному из следующих условий:

a) «отображаемая строка» («VisibleString») является строкой, представляющей календарную дату в соответствии с правилами, определенными в ИСО 2014 с четырехзначной записью года, двузначной записью месяца и двузначной записью числа без разделителей, за которой следует строка, представляющая время суток, как предписано ИСО 3307, не содержащая других разделителей, кроме десятичной точки или десятичной запятой(как предусматривается в 2.3—2.5 ИСО 3307) без оконечного знака Z (как предусмотрено в разделе 3 ИСО 3307) или

b) «отображаемая строка» является строкой знаков, определенной в a), за которой следует прописная буква Z, или

c) «отображаемая строка» является строкой знаков, определенной в a), за которой следует строка, определяющая сдвиг местного времени относительно всемирного времени, соответствующая требованиям ИСО 4031, без разделителей.

В случае a) запись времени обозначает местное время. В случае b) запись времени представляет всемирное время (UTC). В случае c) часть строки, описанная в a), представляет местное время (t_1), а сдвиг местного времени относительно всемирного времени (t_2) позволяет вычислить всемирное время (UTC) по следующей формуле:

время UTC равно $t_1 - t_2$

Примеры

Случай a)

Значение 19851106210627.3

задает местное время 9 ч вечера 6 мин 27,3 с 6 ноября 1985 г.

Случай b)

Значение 19851106210627.3Z

задает всемирное время (UTC), в остальном заданное как в п. a).

Случай c)

Значение 19851106210627.3—0500

задает местное время как в а) с указанием сдвига местного времени относительно всемирного времени (местное время отстает на 5 ч)

32.4 Тег этого типа должен соответствовать определению типа в 32.3.

32.5 Обозначения для записи значения должны быть обозначениями для записи значения типа, «отображаемая строка», определенного в 32.3.

33 ВСЕМИРНОЕ ВРЕМЯ

33.1 Для ссылки на этот тип должно использоваться имя UTCTime

33.2 Этот тип состоит из значений, содержащих:

a) календарную дату;

b) время с точностью до 1 мин или 1 с.;

и

c) (в качестве необязательной возможности) сдвиг местного времени относительно всемирного времени.

33.3 Этот тип может быть определен с помощью нотации АСН.1 следующим образом:

UTCTime : : =

[UNIVERSAL 23] IMPLICIT VisibleString

где значения «отображаемой строки» (VisibleString») ограничены строками знаков, состоящими из следующих частей, следующих друг за другом:

a) шести цифр YYMMDD, где YY — две последние цифры номера года по грегорианскому календарю, MM — номер месяца (январь — 01) и DD — число данного месяца (от 01 до 31);

b)

1) четырех цифр hhmm, где hh — обозначение часа (от 01 до 23) и mm — обозначение минут (от 00 до 59),

2) шести цифр hhmmss, где hh и mm — как в 1), а ss — обозначение секунд;

c)

1) знака Z

2) одного из знаков + или —, за которым следует последовательность hhmm, где hh — часы, а mm — минуты.

Разные варианты в b) обеспечивают разную точность при задании времени.

В варианте c) 1) задается всемирное время. В варианте c)2) задается местное время (t_1) (часть обозначения, определяемая a), b) и сдвиг местного времени относительно всемирного времени (t_2) (часть обозначения, определяемая в c)2), при этом всемирное время (UTC) может быть вычислено следующим образом:

время UTC равно $t_1 - t_2$

Пример — Пусть местное время — 6 ч после полудня 2 января 1982 г., а всемирное время — 12 ч (полдень) 2 января, тогда время может быть записано в виде:

UTCTime «8201021200Z» или

UTCTime «8201020700—0500»

33.4 Тег этого типа должен соответствовать определению типа в 33.3.

33.5 Обозначения для записи значения этого типа должны быть отображениями для записи типа «отображаемая строка», определенного в 33.3.

34 ВНЕШНИЙ ТИП

34.1 Обозначение внешнего типа (см. 3.30) должно иметь вид «ВнешнийТип»:

ВнешнийТип : : = EXTERNAL

34.2 Этот тип состоит из значений, включающих следующие составные части:

a) кодовое представление какого-то одного значения данных, которое может быть (но не обязательно должно быть) значением какого-то одного определенного в АСН.1 типа данных,

b) идентифицирующую информацию, устанавливающую семантику значения и правила кодирования и

c) (в качестве необязательной информации) описатель объекта, содержащий описание данного информационного объекта.

Необязательный описатель объекта может задаваться только в том случае, если это явно разрешается в комментарии, описывающем использование обозначения EXTERNAL.

34.3 Тип EXTERNAL допускает включение в него любых значений данных из некоторого* идентифицируемого множества значений.

Примечания

1 Определение этого множества значений данных, их семантики, присвоение идентификаторов объектов и описателей объектов, а также доведение этой информации до всех обменивающихся информацией сторон называется регистрацией абстрактного синтаксиса. Эта совокупность действий может быть выполнена любой службой, которая имеет право присваивать значения идентификаторов объектов (OBJECT IDENTIFIER), как описано в приложениях B—D.

2 Множество значений данных, зарегистрированных в качестве абстрактного синтаксиса (вместе с соответствующими правилами кодирования) считают правильно сформированным только в том случае, если кодовое представление каждого значения данных обладает свойством самоидентифицируемости в рамках множества кодовых представлений значений данных. Если для определения абстрактного синтаксиса используют нотацию АСН.1, то для обеспечения самоидентифицируемости кодового представления используют присвоение тегов. В случаях, когда абстрактный синтаксис не является правильно сформированным, использование коммуникационного канала либо контекстно-зависимо, либо ведет к неоднозначности

34.4 Внешний вид (EXTERNAL) может быть определен с помощью нотации АСН.1 следующим образом:

```
EXTERNAL ::= [UNIVERSAL 8] IMPLICIT SEQUENCE
    {direct-reference OBJECT IDENTIFIER OPTIONAL,
    indirect-reference INTEGER OPTIONAL,
    data-value-descriptor ObjectDescriptor OPTIONAL,
    encoding CHOICE
    {single-ASN1-type [0] ANY,
    octet-aligned [1] IMPLICIT OCTET STRING,
    arbitrary [2] IMPLICIT BIT STRING}}
```

34.5 Когда для данного значения внешнего типа («EXTERNAL») не используют диалог по установлению правил кодирования на уровне представления, а используют предварительную договоренность относительно синтаксиса передачи, то в значении должен присутствовать идентификатор объекта для прямой ссылки («direct-reference OBJECT IDENTIFIER»). В этом случае идентификация множества значений осуществляется посредством идентификатора объекта (являющегося прямой ссылкой на абстрактный синтаксис) и подставляемого в поле «идентификатор объекта для прямой ссылки» («direct reference OBJECT IDENTIFIER»). В этом случае правила кодирования также определяются с использованием осуществленной ранее регистрации абстрактного синтаксиса и целочисленная косвенная ссылка («indirect-reference INTEGER») не используется.

34.6 Когда используют диалог по установлению правил кодирования на уровне представления, для данного значения внешнего типа («EXTERNAL»), то должна быть задана «целочисленная косвенная ссылка» («indirect-reference INTEGER»). В этом случае идентификатором множества значений данных является целое число, являющееся ссылкой на конкретное использование абстрактного синтаксиса. Это число называется идентификатором контекста представления и подставляется в поле «целочисленная косвенная ссылка» («indirect-reference INTEGER»). Если диалог по установлению правил кодирования на уровне представления завершен, то идентификатор контекста представления также идентифицирует

правила кодирования (синтаксис передачи), применяемые к данному значению данных, при этом «идентификатор объекта для прямой ссылки» («direct reference OBJECT IDENTIFIER») не задается. Если диалог по установлению правил кодирования на уровне представления не завершен, то значение идентификатора объекта также необходимо для идентификации используемых в данный момент правил кодирования (синтаксиса передачи). При использовании диалога по установлению правил кодирования допустимость (или необходимость) элемента «идентификатор объекта для прямой ссылки» («direct reference OBJECT IDENTIFIER») должна быть указана в комментарии, описывающем использование обозначения «EXTERNAL»), если же соответствующее указание отсутствует, то этот идентификатор не должен использоваться.

Примечания

1 Следствием из 34.5 и 34.6 является требование обязательного присутствия по крайней мере одного из вариантов «прямая ссылка» или «косвенная ссылка».

2 Если диалог по установлению параметров на уровне представления используется, но не завершен, то используются обе ссылки.

34.7 Если рассматриваемое значение данных является значением какого-либо одного типа данных, определенного с помощью АСН.1, и если правила кодирования для этого значения данных те же, что и для всего внешнего типа («EXTERNAL»), то в конкретной реализации на стороне отправителя может быть использован любой из вариантов элемента «кодовое представление» («encoding»):

single-ASN1-type
octet-aligned
arbitrary

34.8 Если кодовое представление данных, использующее правила кодирования, установленные в результате предварительной договоренности или в результате диалога по установлению параметров, имеет целое число октетов, то в конкретной реализации на стороне отправителя может быть использован любой из следующих вариантов элемента «кодовое представление» («encoding»):

octet-aligned
arbitrary

Примечание. Если значение данных является последовательностью значений, относящихся к различным типам АСН.1, и правила кодирования предписывают простую постановку друг за другом строк октетов, полученных применением базовых правил кодирования к каждому из значений разных типов, то такое значение относится к категории, описываемой в данном пункте; а не к категории, описанной в 34.7.

34.9 Если кодовое представление значения данных, использующее правила кодирования, установленные в результате предварительной договоренности или в результате диалога по установлению параметров, содержит не целое число октетов, то должен использоваться следующий вариант элемента «кодовое представление» («encoding»):

arbitrary

34.10 Если для элемента «кодовое представление» («encoding») выбран вариант «единственный тип АСН.1» («single-ASN1-type»), то вместо «ANY» должен быть подставлен тип значения, подлежащего кодированию.

Примечание. Множество значений, которые могут встретиться в этом случае, определяются осуществленной ранее регистрацией значения вида «идентификатор объекта», задаваемого «прямой ссылкой» («direct reference») и/или целочисленного значения, задаваемого «косвенной ссылкой» («indirect reference»).

34.11 Если для элемента «кодовое представление» («encoding») выбран вариант «октетно-выравненное» («octet-aligned»), то значение данных должно быть закодировано в соответствии с синтаксисом передачи, установленным в результате предварительной договоренности или диалога по установлению параметров, и полученные в результате октеты должны составлять значение — строку октетов.

34.12 Если для элемента «кодовое представление» («encoding») был выбран вариант «произвольное» («arbitrary»), то значение данных должно быть закодировано в соответствии с синтаксисом передачи, установленным в результате предварительной договоренности или диалога по установлению правил кодирования, а результат должен быть значением — строкой бит.

34.13 Тег этого типа должен соответствовать определению типа в 34.4.

34.14 Обозначения для записи значения должны быть обозначениями для записи значения типа, определенного в 34.4.

35 ТИП «ОПИСАТЕЛЬ ОБЪЕКТА»

35.1 Ссылку на этот тип осуществляют с помощью имени ObjectDescriptor.

35.2 Этот тип составляют тексты, ориентированные на восприятие человеком и предназначенные для описания информационных объектов. Текст, являющийся значением данного типа, не обязательно должен идентифицировать информационный объект, но предполагается, что использование одного и того же текста для описания разных информационных объектов маловероятно.

Примечание. Желательно, чтобы значения типа «описатель объекта» («ObjectDescriptor») присваивались информационным объектам теми же службами, которые присваивают информационным объектам значения типа «идентификатор объекта» («OBJECT IDENTIFIER»).

35.3 Этот тип может быть определен с помощью нотации АСН.1 следующим образом:

ObjectDescriptor : : =
    [UNIVERSAL 7] IMPLICIT GraphicString

«Графическая строка» («GraphicString») содержит текст, описывающий информационный объект.

35.4 Тег этого типа должен соответствовать определению типа в 35.3.

35.5 Обозначения для записи значения должны быть обозначениями для записи значения типа, определенного в п. 37.3.

36 ОБОЗНАЧЕНИЯ ДЛЯ ПОДТИПОВ

36.1 Подтип определяется с помощью обозначения порождающего типа, за которым следует спецификация требуемого подтипа. Обозначения для спецификации подтипа определяются через обозначения подмножеств значений подтипа. Множество значений подтипа определяется в соответствии с 36.7 взятием теоретико-множественного объединения подмножеств значений подтипа.

36.2 Не следует использовать обозначения подтипа таким образом, при котором порождается подтип, не содержащий ни одного значения.

36.3 Обозначения для подтипа должны иметь вид:

Подтип : : =
    ПорождающийТип СпецификацияПодтипа |
    SET ОграничениеРазмера OF Тип |
    SEQUENCE ОграничениеРазмера OF Тип

ПорождающийТип : : = Тип

36.4 В случаях, когда обозначение «СпецификацияПодтипа» следует за обозначением «СелективныйТип», порождающим типом является «СелективныйТип», а не «Тип» в обозначении «СелективныйТип».

36.5 В случаях, когда обозначение «СпецификацияПодтипа» следует за обозначением типа «множество из» или «последовательность из», оно относится к «Типу» в обозначении «множество из» или «последовательность из», а не к самому типу «множество из» или «последовательность из».

Примечание. Обозначения для частных случаев, имеющие вид «SET ОграничениеРазмера OF Тип» и «SEQUENCE ОграничениеРазмера OF Тип» обеспечивают альтернативные (более легко воспринимаемые) конструкции для

простых случаев. Более сложные случаи требуют использования конструкции общего вида.

36.6 Обозначение для спецификации подтипа должно иметь вид «СпецификацияПодтипа»:

СпецификацияПодтипа : : =
(ПодмножествоЗначенийПодтипа СписокПодмножествЗначенийПодтипа)

СписокПодмножествЗначенийПодтипа : : =
« ¦ »
ПодмножествоЗначенийПодтипа
СписокПодмножествЗначенийПодтипа ¦
пусто

36.7 Каждое «ПодмножествоЗначенийПодтипа» задает какое-то количество (возможно, равное нулю) значений порождающего типа, которые включаются в подтип. Значение порождающего типа является значением подтипа тогда и только тогда, когда оно входит в одно или несколько подмножеств значений подтипа. Подтип, таким образом, образуется как объединение подмножеств значений подтипа.

36.8 Предусмотрено несколько различных форм обозначения «ПодмножествоЗначенийПодтипа». Они перечислены ниже, а их синтаксис и семантика определены в разделе 37. Как указано в разделе 37 и проиллюстрировано в таблице 7, некоторые виды обозначений применимы лишь к некоторым порождающим типам.

ПодмножествоЗначенийПодтипа : : =
ОдноЗначение ¦
ВложенныйПодтип ¦
ДиапазонЗначений ¦
РазрешенныйАлфавит ОграничениеРазмера ¦
ОграниченияВнутреннихТипов

37 ПОДМНОЖЕСТВА ЗНАЧЕНИЙ ПОДТИПА

37.1 **Одно значение**

37.1.1 Обозначение «одно значение» должно иметь вид

ОдноЗначение : : = Значение

где «Значение» — обозначение значения порождающего типа.

37.1.2 Подмножество вида «ОдноЗначение» состоит из одного значения порождающего типа, заданного «Значением». Это обозначение может быть применено ко всем порождающим типам.

37.2 **Вложенный подтип**

37.2.1 Обозначения «ВложенныйПодтип» должны иметь вид:

ВложенныйПодтип : : = INCLUDES Тип

37.2.2 Значения, задаваемые обозначением «ВложенныйПод-

тип», — это все значения, входящие в «Тип», который сам должен быть подтипом порождающего типа. Это обозначение может быть применено ко всем порождающим типам.

37.3 **Диапазон значений**

37.3.1 Обозначение «ДиапазонЗначений» должно иметь вид:

ДиапазонЗначений : : =
НижняяГраница . . ВерхняяГраница

Таблица 7— Применимость способов образования подмножеств значений подтипов

Тип или полученный из него тегированием	Одно значение	Вложенный подтип	Диапазон значений	Ограничение размера	Ограничение алфавита	Внутренние типы
Булевский	да	да	нет	нет	нет	нет
Целочисленный	да	да	да	нет	нет	нет
Перечислительный	да	да	нет	нет	нет	нет
Действительное число	да	да	да	нет	нет	нет
Идентификатор объекта	да	да	нет	нет	нет	нет
Строка бит	да	да	нет	да	нет	нет
Строка октетов	да	да	нет	да	нет	нет
Типы «строка знаков»	да	да	нет	да	да	нет
Последовательность	да	да	нет	нет	нет	да
Последовательность из	да	да	нет	да	нет	да
Множество	да	да	нет	нет	нет	да
Множество из	да	да	нет	да	нет	да
Произвольный	да	да	нет	нет	нет	нет
Выборочный	да	да	нет	нет	нет	да

37.3.2 Множество значений, задаваемое «ДиапазономЗначений», состоит из всех значений, заключенных в диапазоне, заданном числовыми значениями границ диапазона. Это обозначение может применяться только для целочисленных типов, типов «действительное число», а также типов, полученных из них тегированием или взятием подтипа.

Примечание При образовании подтипов «PLUS-INFINITY» («плюс бесконечность») превышает все значения «ДействительноеЧисло», а «MINUS-

INFINITY» («минус бесконечность») меньше всех значений «ДействительноеЧисло».

37.3.3 Граница диапазона может быть замкнутой (в этом случае граница диапазона входит в множество значений) или открытой (в этом случае граница диапазона не входит в множество значений). Если граница диапазона открытая, то обозначение нижней или верхней границы включает в себя знак «меньше» («<»):

НижняяГраница : : =
НижнееГраничноеЗначение | НижнееГраничноеЗначение <
ВерхняяГраница : : =
ВерхнееГраничноеЗначение | <ВерхнееГраничноеЗначение

37.3.4 Граница диапазона может также быть не задана, при этом диапазон продолжается в данном направлении до границы порождающего типа:

НижнееГраничноеЗначение : : = Значение | MIN
ВерхнееГраничноеЗначение : : = Значение | MAX

37.4 **Ограничение размера**

37.4.1 Обозначение «ограничение размера» должно иметь вид:

ОграничениеРазмера : : = SIZE СпецификацияПодтипа

37.4.2 Обозначение «ОграничениеРазмера» применимо только к типам «строка бит», «строка октетов», «строка знаков», «множество из» и «последовательность из», а также типам, полученным из них тегированием или взятием подтипа.

37.4.3 «СпецификацияПодтипа» задает разрешенные целочисленные значения длины элементов множества значений и имеет вид любой спецификации подтипа, которая может быть применена к следующему порождающему подтипу:

INTEGER (0..MAX)

37.4.4 Единица измерения зависит от порождающего типа следующим образом:

Тип	Единица измерения
строка бит	бит
строка октетов	октет
строка знаков	знак
множество-из	значение-компонент
последовательность-из	значение-компонент

37.5 **Разрешенный алфавит**

37.5.1 Обозначение «РазрешенныйАлфавит» должно иметь вид:

РазрешенныйАлфавит : : = FROM СпецификацияПодтипа

37.5.2 Значения варианта «РазрешенныйАлфавит» — это все значения, которые могут быть построены с использованием подалфавита порождающей строки. Это обозначение может быть приме-

нено только к типам «строка знаков» или к типам, полученным из них тегированием или образованием подтипа.

37.5.3 Обозначение «СпецификацияПодтипа» задает знаки, которые могут присутствовать в строке знаков, оно может быть любой спецификацией подтипа, которая применима к подтипу, полученному из порождающего типа применением спецификации подтипа «SIZE(1)».

37.6 **Образование внутренних подтипов**

37.6.1 Обозначение «ОграниченияВнутреннихТипов» должно иметь вид:

```
ОграниченияВнутреннихТипов : : =
    WITH COMPONENT ОграничениеОдногоТипа |
    WITH COMPONENTS ОграниченияНескольких Типов
```

37.6.2 «ОграниченияВнутреннихТипов» задают включение в подмножество только тех значений, которые удовлетворяют совокупности ограничений на присутствие и/или значения компонентов порождающего типа. Значение порождающего типа включается в подтип только в том случае, когда удовлетворяются все ограничения, как явные, так и неявные (см. 37.6.6). Это обозначение может быть применено к типам «множество-из», «последовательность-из», «множество», «последовательность» и выборочному типу, а также к типам, полученным из них тегированием или образованием подтипа.

37.6.3 Для типов, определенных в терминах некоторого другого (внутреннего) типа (типы «множество-из», «последовательность-из» и полученные из них тегированием или образованием подтипа), предусмотрено ограничение, имеющее вид спецификации значения подтипа. Обозначение для этого случая должно иметь вид:

```
ОграничениеОдногоТипа : : = СпецификацияПодтипа
```

«СпецификацияПодтипа» определяет подтип единственного другого (внутреннего) подтипа Значение порождающего является элементом подмножества значений подтипа в том и только в том случае, когда каждое внутреннее значение принадлежит подтипу, образованному применением «СпецификацииПодтипа» к внутреннему типу.

37.6.4 Для типов, определенных в терминах нескольких других (внутренних) типов (типы «множество», «последовательность», выборочный и полученные из них тегированием или образованием подтипа), предусмотрено несколько возможностей накладывать ограничения на эти внутренние типы. Обозначение для этого случая должно иметь вид:

```
ОграниченияНескольких Типов : : =
    ПолнаяСпецификация | ЧастичнаяСпецификация
ПолнаяСпецификация : : = {ОграниченияТипа}
```

ЧастичнаяСпецификация : : = {..., ОграниченияТипа}
ОграниченияТипа : : =
ПоименованноеОграничение |
ПоименованноеОграничение ОграниченияТипа

ПоименованноеОграничение : : =

Идентификатор Ограничение | Ограничение

37.6.5 «ОграниченияТипа» содержат список ограничений на типы-компоненты порождающего типа. Для типа «последовательность» ограничения должны следовать в требуемом порядке. Внутренний тип, к которому относится ограничение, определяется с помощью идентификатора, если он имеется, или местом в перечне ограничений, как в случае типов «последовательность».

Примечание. Если внутренний тип не имеет идентификатора, то обозначение может оказаться неоднозначным.

37.6.6 Обозначение «ОграниченияНескольких Типов» охватывает два варианта обозначений: «ПолнаяСпецификация» и «ЧастичнаяСпецификация». Если используется «ПолнаяСпецификация», то неявно подразумевается ограничение присутствия вида «ABSENT» («отсутствует») для всех внутренних типов, не присутствующих явно в перечне ограничений (см. 37.6.9), а каждый внутренний тип, не отмеченный ключевым словом «OPTIONAL» («необязательный») или «DEFAULT» («по умолчанию») в описании порождающего типа, должен явно присутствовать в перечне ограничений. Если используется «ЧастичнаяСпецификация», то никаких ограничений присутствия не подразумевается, и любой внутренний тип может быть опущен из перечня ограничений.

37.6.7 На конкретный внутренний тип могут накладываться ограничения, как в терминах его присутствия (в значениях порождающего типа), так и в терминах допускаемых для него значений, либо и того и другого. Соответствующее обозначение должно иметь вид «Ограничение»:

Ограничение : : =

ОграничениеЗначения ОграничениеПрисутствия

37.6.8 Ограничение значения внутреннего типа выражается обозначением «ОграничениеЗначения»:

ОграничениеЗначения : : = СпецификацияПодтипа | пусто

Значение порождающего типа удовлетворяет ограничению в том и только том случае, когда значение внутреннего типа относится к подтипу, заданному обозначением «СпецификацияПодтипа» в применении к внутреннему типу.

37.6.9 Ограничение присутствия внутреннего типа выражается обозначением «ОграничениеПрисутствия»:

ОграничениеПрисутствия : : =
PRESENT | ABSENT | OPTIONAL |
пусто

Смысл этих вариантов и ситуации, в которых они разрешены, определяются в 37.6.9.1—37.6.9.3.

37.6.9.1 Если порождающий тип — «последовательность» или «множество», присутствие типа-компонента, отмеченного ключевым словом «OPTIONAL», может быть ограничено ключевым словом «PRESENT» (в этом случае ограничение выполняется в том и только в том случае, когда присутствует соответствующее значение-компонент), ключевым словом «ABSENT» (в этом случае ограничение выполняется только в том случае, когда отсутствует соответствующее значение-компонент), или ключевым словом «OPTIONAL» (в этом случае ограничение накладывается на присутствие соответствующего значения-компонента).

37.6.9.2 Если порождающий тип — выборочный тип, то присутствие типа-компонента может быть ограничено ключевым словом «ABSENT» (в этом случае ограничение выполняется в том и только в том случае, когда соответствующий тип-компонент отсутствует в значении.

37.6.9.3 Смысл варианта «пусто» в обозначении «ОграничениеПрисутствия» зависит от того, используется вариант «ПолнаяСпецификация» или «ЧастичнаяСпецификация»:

a) в «ПолнойСпецификации» этот вариант эквивалентен ограничению типа «PRESENT»;

b) в «ЧастичнойСпецификации» никаких ограничений не налагается.

ПРИЛОЖЕНИЕ А
Обязательное

МАКРОНОТАЦИЯ

А.1 Введение

В АСН.1 предусмотрен способ, с помощью которого пользователь АСН.1 может определить новую систему обозначений, с использованием которой он может строить те же типы данных, что и в АСН.1, ссылаться на них, а также задавать значения этих типов. Эта новая система обозначений определяется с помощью обозначения нотации АСН.1 «Макроопределение». «Макроопределение» одновременно определяет как обозначения для построения типов и осуществления ссылок на них, так и обозначения для записи значений. (См. Е.3, содержащий пример использования макронотации.)

С помощью «Макроопределения» пользователь АСН.1 определяет новые обозначения с помощью некоторого множества продукций, аналогично тому, как это делается в настоящем стандарте. Составитель макроопределения должен:

a) полностью определить синтаксис, который предполагается использовать для определения всех типов, обеспечиваемых данным макроопределением; (это описание синтаксиса вызывается для синтаксического анализа каждый раз при появлении в обозначении типа АСН.1 имени макроопределения);

b) полностью определить синтаксис, который предполагается использовать для определения значений этих типов (это описание синтаксиса вызывается для синтаксического анализа каждый раз, когда ожидается значение типа, описанного с помощью макроопределения), и

c) определить в виде значений стандартных типов АСН.1 (произвольного уровня сложности) получающиеся в результате типы и значения для всех случаев использования макрообозначений для записи значений.

Конкретное использование синтаксиса, определенного с помощью макроопределения, может содержать обозначения типов и значений, заданных в обычной нотации АСН.1. Эти типы или значения (присутствующие в конкретном использовании макрообозначений) могут быть логически связаны в период выполнения синтаксического анализа с локальными ссылками на тип или на значение с помощью соответствующих конструкций в макроопределении. Можно также встроить в макроопределение присвоение ссылкам обычных определений типов АСН.1. Описанные выше присвоения активизируются, когда соответствующая синтаксическая категория макроопределения сопоставляется с элементом (или элементами), входящими в использование новой нотации. Время их действия ограничено временем выполнения синтаксического анализа.

При синтаксическом анализе значения, заданного с помощью макронотации, присвоения, выполненные при синтаксическом анализе типа, продолжают быть доступными. Синтаксический анализ обозначения типа считают логически предшествующим анализу соответствующих значений.

Каждое «Макроопределение» определяет систему обозначений (синтаксис) для определения типа и систему обозначений (синтаксис) для определения значений. Тип данных АСН.1, определенный с помощью макрообозначения типа, может зависеть (но не обязательно зависит) от макрообозначения конкретного значения, к которому относится макрообозначение типа. В этом отношении использование макронотации типа сходно с использованием выборочного типа («CHOICE») — тег типа в данном случае не определен. Поэтому в случаях зависимости типа от значения макронотация не может быть использована там, где

требуется наличие известного тега; в этих случаях не может также применяться неявное задание тега.

Тип и значение, полученные в результате использования макронотации, определяются тем, какое значение (и каков его тип) присваивается (в результате последовательной обработки макроопределения типа, а затем макроопределения значения) локальной ссылке на значение, отмеченной в макроопределении ключевым словом VALUE.

А.2 Дополнение к набору знаков и множеству элементов АСН.1

В макронотации используют знаки : и >.

Также используют элементы, описанные в последующих пунктах.

А.2.1 Макроссылка

Имя элемента — макроссылка

«Макроссылка» должна состоять из последовательности знаков, удовлетворяющей требованиям, установленным для «СсылкиНаТип» в 8.2, с тем отличием, что все буквы должны быть прописными. Внутри одного модуля одна и та же последовательность знаков не должна использоваться одновременно и как ссылка на тип, и как макроссылка.

А.2.2 Ссылка на продукцию

Имя элемента — ССылкаНаПродукцию

«ССылкаНаПродукцию» должна состоять из последовательности знаков, удовлетворяющих требованиям, установленным для «СсылкиНаТип» в 8.2.

А.2.3 Локальная ссылка на тип

Имя элемента — ЛокальнаяССылкаНаТип

«ЛокальнаяССылкаНаТип» должна представлять собой последовательность знаков вида «СсылкаНаТип», определенного в 8.2. «ЛокальнаяССылкаНаТип» используется в качестве идентификатора для типов, распознаваемых во время синтаксического анализа макрообозначения типа или значения.

А.2.4 Локальная ссылка на значение

Имя элемента — ЛокальнаяССылкаНаЗначение

«ЛокальнаяССылкаНаЗначение» должна представлять собой последовательность знаков вида «СсылкаНаЗначение», определенного в 8.2. «ЛокальнаяССылкаНаТип» используется в качестве идентификатора для значений, распознаваемых во время синтаксического анализа макрообозначения типа или значения.

Примечание. «ЛокальнаяССылкаНаТип» должна начинаться с прописной буквы.

А.2.5 Элемент разграничения вариантов

Имя элемента — « . »

Этот элемент состоит из единственного знака : .

А.2.6. Элемент, обозначающий конец определения

Имя элемента — >

Этот элемент состоит из единственного знака >.

Примечание. Элемент <, *обозначающий начало определений*, определен в 8.13.

А.2.7 Синтаксический терминальный элемент

Имя элемента — а-строка

Элемент «а-строка» («строка в кавычках») должен состоять из произвольного (возможно, равного нулю) количества знаков из знакового набора АСН.1 (см. раздел 7), обрамленных кавычками ". Знак " представляется в «строке в кавычках» в виде пары знаков "".

Примечание. Использование «строки в кавычках» («а-строки») в макрообозначении задает появление в соответствующем месте анализируемого синтаксиса знаков, заключенных между кавычками ("").

А.2.8 Элементы — ключевые слова синтаксических категорий

Имена элементов — «string»
«identifier»
«number»
«empty»

Элементы с приведенными выше именами должны состоять (в макроопределении) из последовательностей знаков, совпадающих с их именами, исключая кавычки (“). Эти элементы используются в макроопределении для задания вхождения в использовании новой нотации последовательностей знаков определенного вида. Вид последовательностей знаков, соответствующих каждому из элементов, определяется в таблице А.1 посредством ссылки на раздел настоящего стандарта.

Примечание. В макронотации не поддерживается различие между идентификаторами и ссылками, отличающихся только типом первой буквы: строчная или прописная. Это обусловлено историческими причинами.

Таблица А.1 — **Классы последовательностей, задаваемые элементами**

Имя элемента	Раздел, содержащий определение
«string»	произвольная последовательность знаков
«identifier»	8.3 — Идентификаторы
«number»	8.8 Числа
«empty»	8.7 — Пустой элемент

А.2.9 Дополнительные элементы — ключевые слова

Имена элементов — MACRO
TYPE
NOTATION
VALUE
value
type

Элементы с приведенными выше именами должны состоять из последовательностей знаков, совпадающих с их именами.

Элементы, определяемые в А.2.2—А.2.4, не должны совпадать с какой-либо последовательностью знаков из А.2.9, за исключением случаев, оговоренных ниже.

Ключевое слово «MACRO» должно использоваться для того, чтобы обозначить начало макроопределения. Ключевое слово «TYPE NOTATION» должно использоваться в качестве имени продукции, определяющей новые обозначения типа. Ключевое слово «VALUE NOTATION» должно использоваться в качестве имени продукции, определяющей новые обозначения значения. Ключевое слово «VALUE» должно использоваться в качестве «ЛокальнойСсылкиНаЗначение», которой присваивается получаемое в результате значение. Ключевое слово «value» должно использоваться для указания того, что в каждом применении новых обозначений в этом месте должно содержаться значение (записанное в обозначениях АСН.1) некоторого типа (задаваемого макроопределением). Ключевое слово «type» должно использоваться для указания того, что в каждом применении новых обозначений в этом месте должно содержаться обозначение вида «Тип» (в обозначениях АСН.1).

А.3 Обозначения для записи макроопределений

А.3.1 Макросредство должно быть определено с использованием обозначений вида «Макроопределение»:

```
Макроопределение : : =
                    макроссылка
                    MACRO
                    « : : = »
                    СодержаниеМакро
СодержаниеМакро : : =
                    BEGIN ТелоМакро END |
                    макроссылка         |
                    ВнешняяМакроссылка
ТелоМакро : : =
                    ПродукцияТипа
                    ПродукцияЗначения
                    ВспомогательныеПродукции
ПродукцияТипа : : =
                    TYPE NOTATION
                    « : : = »
                    СписокВариантовМакро
ПродукцияЗначения : : =
                    VALUE NOTATION
                    « : : =»
                    СписокВариантовМакро
ВспомогательныеПродукции : : =
                    СписокПродукции |
                    empty
СписокПродукций : : =
                    Продукция                |
                    СписокПродукций Продукция
Продукция : : =
                    ССылкаНаПродукцию
                    « : : =»
                    СписокВариантовМакро
ВнешняяМакроссылка : : =
                    ССылкаНаМодуль.макроссылка
```

Примечание. Предполагается дать возможность в макроопределении ссылаться на другие макроопределения. Вопрос о том, согласуется ли это с правилами настоящей нотации, подлежит дальнейшему изучению.

А.3.2 Если для конструкции «СодержаниеМакро» был выбран вариант «макроссылка», то модуль, в котором содержится данное макроопределение, должен либо

a) содержать другое макроопределение, либо

b) содержать это макроопределение среди своих «ВходныхСимволов».

А.3.3 Если для конструкции «СодержаниеМакро» был выбран вариант «ВнешняяМакроссылка», то модуль, заданный «ССылкойНаМодуль», должен содержать макроопределение, определяющее эту «макроссылку». Это макроопределение и является определением данной макроссылки.

А.3.4 Цепочка определений, могущая возникнуть в результате многократного применения правил А.3.2 и А.3.3, должна заканчиваться «Макроопределением, в котором используется вариант «BEGIN ТелоМакро END»; «ТелоМакро» этого макроопределения определяет обозначения типа и значения данного макросредства.

А.3.5 Каждая «ССылкаНаПродукцию», присутствующая в «Определении-Символа» (см. А.3.9), должна присутствовать в точности один раз в качестве первого элемента некоторой «Продукции».

А.3.6 Каждый случай использования макрообозначения типа должен начинаться с последовательности знаков «Макроссылка», за которой следует одна из последовательностей знаков, удовлетворяющих продукции «TYPE NOTATION» после применения всех продукций, заданных в макроопределении.

А.3.7 Каждый случай использования макронотации для записи значения должен быть некоторой последовательностью знаков из множества, описанного продукцией «VALUE NOTATION» после применения всех продукций, заданных в макроопределении.

А.3.8 Множество последовательностей знаков, описываемое продукцией «TYPE NOTATION» или VALUE NOTATION», определяется соответствующим «СпискомВариантовМакро», имеющим вид:

```
СписокВариантовМакро : : =
    ВариантМакро |
    СписокВариантовМакро « | » ВариантМакро
```

Множество последовательностей знаков, описываемое продукцией «СписокВариантовМакро», состоит из множеств последовательностей знаков, описываемых продукциями «ВариантМакро», входящими в «СписокВариантовМакро».

А.3.9 Обозначения для записи последовательности «ВариантыМакро» в макроопределении определяются следующим образом:

```
ВариантМакро . : = СписокСимволов
СписокСимволов : : =
    Символьный Элемент |
    СписокСимволов Символьный Элемент .
Символьный Элемент : : =
    ОпределениеСимвола ,
    ВложенноеОпределение
МакроТип : :  ЛокальнаяССылкаНаТип :
    Тип
ОпределениеСимвола : : =
    «a-строка»
    ССылкаНаПродукцию
    «string»
    «identifier»
    «number»
    «empty»
    type
    type (ЛокальнаяССылкаНаТип)
    value (МакроТип)
    value (ЛокальнаяССылкаНаЗначение МакроТип)
    value (VALUE МакроТип)
```

Примечание. В макроопределении выражение «МакроТип», определенное в этом же макроопределении, может присутствовать в любом месте, где в соответствии с правилами нотации АСН.1 требуется выражение «Тип».

Множество последовательностей знаков, описываемое продукцией «Вариант-Макро», образуется последовательным соединением любой из строк знаков, описываемой первой последовательностью «ОпределениеСимвола» с любой из строк знаков, описываемой второй последовательностью «ОпределениеСимвола» и так далее, вплоть до последнего «ОпределенияСимвола» из «СпискаСимволов».

Примечание. «ВложенныеОпределения», если они присутствуют, не играют явной роли при определении множества допустимых строк.

А.3.10 Элемент («а-строка») описывает последовательности знаков, составляющие «а строку», без обрамляющей пары кавычек (").

А.3.11 Элемент «ССылкаНаПродукцию» описывает произвольную последовательность знаков, задаваемых «Продукцией», идентифицируемой данной ссылкой.

А.3.12 Последовательности знаков, описываемые следующими четырьмя вариантами продукции «ОпределениеСимвола», задаются таблицей А.1.

Примечание. Последовательности знаков, описываемые вариантом «string», должны заканчиваться в использовании макрообозначения появлением последовательности, описываемой очередным выражением «ОпределениеСимвола» в «СпискеСимволов».

А.3.13 Вариант «type» описывает последовательности знаков, образующие обозначение вида «Тип», соответствующее определению в 12.1.

Примечание. Выражение «ОпределенныйТип» в определении из 12.1 может в данном случае содержать «ЛокальнуюССылкуНаТип», служащую для обозначения типа, определенного с помощью макроопределения.

А.3.14 Вариант «type (ЛокальнаяССылкаНаТип)» описывает последовательности знаков, образующие выражение «Тип», соответствующее определению из 12.1, и кроме того присваивает этот тип «ЛокальнойССылкеНаТип».

А.3.15 Вариант «value (МакроТип)» описывает последовательности знаков, образующие выражение «Значение», соответствующее определению из 12.7, того типа, который задан выражением «Макротип».

А.3.16 Выражение «value (ЛокальнаяССылкаНаЗначение МакроТип)» описывает последовательности знаков, образующих выражение «Значение», соответствующее определению из 12.7, того типа, который задан выражением «МакроТип», и кроме того присваивает этот тип «ЛокальнойССылкеНаТип». Впоследствии той же «ЛокальнойССылкеНаТип» может быть присвоен другой тип.

А.3.17 Вариант «value (VALUE МакроТип)» описывает последовательности знаков, образующих выражение «Значение», (соответствующее определению из 12.7) того типа, который задан выражением «МакроТип», но кроме того возвращает данное значение в качестве значения, заданного данным использованием макрообозначений. Тип возвращаемого значения задается выражением «МакроТип»

А.3.18 При синтаксическом анализе любого правильного использования макрообозначений должно осуществляться в точности одно присвоение значения выражению «VALUE» (описанному в А.3.17 или А.3.19).

А.3.19 Выражение «ВложенныеОпределения» определяется следующими продукциями:

```
ВложенноеОпределение : : =
    <СписокВложенныхОпределений>
СписокВложенныхОпределений : : =
    ВложенноеОпределение |
    СписокВложенныхОпределений
    ВложенноеОпределение
ВложенноеОпределение : : =
    ЛокальноеПрисвоениеТипа |
    ЛокальноеПрисвоениеЗначения
ЛокальноеПрисвоениеТипа : : =
    ЛокальнаяССылкаНаТип
    « : : = »
```

МакроТип
ЛокальноеПрисвоениеЗначения : : =
ЛокальнаяСсылкаНаЗначение
МакроТип
« : : = »
МакроЗначение
Макрозначение : : =
Значение |
ЛокальнаяСсылкаНаЗначение

Присвоение «МакроТипа» «ЛокальнойСсылкеНаТип», а также «МакроЗначения» «ЛокальнойСсылкеНаЗначение» во ВложенныхОпределениях» осуществляется при синтаксическом анализе использования макрообозначения при обработке «ВложенныхОпределений» и сохраняется до переопределения «ЛокальнойСсылкиНаТип» или «ЛокальнойСсылкиНаЗначение»

Примечания

1 Использование связанной «ЛокальнойСсылкиНаТип» или «ЛокальнойСсылкиНаЗначение» в другом месте того же варианта возможно только при определенных предположениях относительно алгоритма синтаксического разбора. Эти предположения должны быть описаны в комментарии. Например, использование «ЛокальнойСсылкиНаТип», текстуально следующей за «ВложеннымиОпределениями», возможно только при синтаксическом разборе слева направо.

2 «ЛокальнойСсылкеНаТип», заданной как «VALUE», может быть присвоено значение как с помощью конструкции «value (VALUE МакроТип)», так и с помощью вложенного определения. В обоих случаях присвоенное значение является возвращаемым значением, в соответствии с А.3.17.

А.4 Использование новой нотации

Обозначение типа (или значения), определенное с помощью макросредства, может применяться во всех случаях, когда в соответствии с правилами настоящего стандарта требуется использование обозначения «Тип» (или «Значение»), при условии, что выполнены следующие требования:

a) макросредство определено в том же модуле; либо

b) макросредство вводится в модуль посредством включения «макроссылки» в конструкцию «ВходныеСимволы» этого модуля.

Чтобы обеспечить возможность b), «макроссылка» может быть подставлена в качестве «Символа» в обозначениях 9.1.

Примечания.

1 Описываемое в этом приложении расширение нотации АСН.1 не используют в основном тексте настоящего стандарта.

2 Возможно построение модулей, содержащих последовательности присвоения типа и макроопределения, приводящих к сколь угодно сложному синтаксическому разбору значений, заданных с использованием ключевого слова DEFAULT.

ПРИЛОЖЕНИЕ В
Обязательное

ОПРЕДЕЛЯЕМЫЕ ИСО ЗНАЧЕНИЯ КОМПОНЕНТОВ OBJECT IDENTIFIER

В.1 Из корневой вершины дерева идентификаторов объектов выходят три ребра. Присвоенные им значения и идентификаторы, а также службы, ответственные за присвоение значений следующих компонентов, таковы:

Значение	Идентификатор	Служба, отвечающая за присвоение следующих компонентов
0	ccitt	МККТТ
1	iso	ИСО
2	joint-iso-ccitt	См. Приложение D

Примечание. В остальной части приложения рассматривают присвоение значений, осуществляемое ИСО.

В.2 Каждый из идентификаторов «ccitt», «iso» и «joint-iso-ccitt», определенных выше, может использоваться в качестве выражения «ИменнаяФорма».

В.3 Из вершины, проидентифицированной «iso», выходят четыре ребра. Присвоенные им значения и идентификаторы таковы:

Значение	Идентификатор	Служба, отвечающая за присвоение следующих компонентов
0	standard	см. В.4
1	registration-authority	см. В.5
2	member-body	см. В.6
3	identified-organization	см. В.7

Эти идентификаторы могут использоваться в качестве выражения «ИменнаяФорма».

В.4 Каждому из ребер, выходящих из вершины «стандарт» («standard»), должно быть присвоено значение, равное номеру некоторого стандарта. Если стандарт состоит из нескольких частей, то из соответствующей ему вершины должны исходить ребра, соответствующие номерам частей, если в тексте стандарта явно не указано иначе. Последующим ребрам должны быть присвоены значения, определяемые в настоящем стандарте.

Примечание. Если в некотором стандарте, состоящем из одной части, были присвоены значения некоторым идентификаторам объектов, после чего стандарт был преобразован и стал состоять из нескольких частей, то после преобразования присвоение значений идентификаторов объектов должно продолжаться так, как если бы стандарт состоял из одной части.

В.5 Ребра, выходящие из вершины «регистрационная служба» («registration authority»), резервируются для последующего определения в дополнении к настоящему стандарту, которые будут разрабатываться параллельно с определением процедур для идентификации различных регистрационных служб ВОС.

В.6 Ребрам, выходящим из вершины, в которую ведет ребро «комитет-член ИСО» («member-body»), должны быть присвоены значения трехзначного цифро-

вого кода страны в соответствии с ИСО 3166; этот код идентифицирует национальный комитет-член ИСО (см примечание) «ИменнаяФорма» этого компонента идентификатора объекта не допускается. Ребра, следующие за «кодом страны», не определяются в настоящем стандарте.

Примечание. Наличие кода страны в ИСО 3166 не обязательно означает, что действительно существует комитет-член ИСО, представляющий эту страну, или что в сферу деятельности этого комитета входит схема выделения компонентов идентификаторов объектов.

В.7 Ребрам, выходящим из вершины, в которую ведет ребро «зарегистрированная организация» («identified-organization»), должны быть присвоены значения международного кодового определителя (International Code Designator — ICD), которые назначаются регистрационной службой, связанной с ИСО 6523; ICD идентифицирует организацию — источник кодов, зарегистрированную вышеупомянутой регистрационной службой в качестве источника компонентов идентификаторов объектов (см. примечания 1 и 2). Ребрам, непосредственно следующим за ICD, должны быть присвоены значения вида «код организации», которые выделяются организацией — источником кодов в соответствии с ИСО 6523. Ребра, следующие за «кодом организации», в настоящем стандарте не определяются

Примечания

1 Гарантия того, что коды организаций, используемые как цифровые значения идентификаторов объектов и назначенные организациями, присваивающими коды, будут соответствовать требованиям настоящего стандарта, обеспечивается требованием регистрации этих организаций регистрационной службой, связанной с ИСО 6523

2 Упоминание того, что коды, назначенные организациям, используются в качестве компонентов идентификаторов объекта, не исключает возможности их использования также для других целей.

3 Предполагается, что организации, идентифицируемые с помощью «кода организации», будут определять следующие ребра таким образом, чтобы гарантировать неповторяемость соответствующих значений.

4 Следствием В.7 является то, что любая организация может получить код от соответствующей организации, присваивающей коды, и после этого присваивать значения вида «идентификатор объекта» (OBJECT IDNTIFIER) для своих собственных целей, причем гарантируется, что эти значения не будут повторять значения, назначенные другими организациями. Таким образом некоторая фирма может присвоить значение типа «идентификатор объекта (OBJECT IDENTIFIER)» своим собственным информационным форматам.

ПРИЛОЖЕНИЕ С
Обязательное

ОПРЕДЕЛЯЕМЫЕ МККТТ ЗНАЧЕНИЯ КОМПОНЕНТОВ OBJECT IDENTIFIER

С.1 Из корневой вершины дерева идентификаторов объектов выходят три ребра. Присвоенные им значения и идентификаторы, а также службы, ответственные за присвоение значений следующих компонентов, таковы:

Значение	Идентификатор	Служба, отвечающая за присвоение следующих компонентов
0	ccitt	МККТТ
1	iso	ИСО
2	joint-iso-ccitt	См. приложение D

Примечание В остальной части этого приложения рассматривается присвоение значений, осуществляемое МККТТ.

С.2 Каждый из идентификаторов «ccitt», «iso» и «joint-iso-ccitt», определенных выше, может использоваться в качестве выражения «ИменнаяФорма».

С.3 Из вершины, проидентифицированной «ccitt», выходят четыре ребра. Присвоенные им значения и идентификаторы таковы:

Значение	Идентификатор	Служба, отвечающая за присвоение следующих компонентов
0	recommendation	см. С.4
1	question	см. С.5
2	administration	см. С.6
3	network-operator	см. С.7

Эти идентификаторы могут использоваться в качестве выражения «ИменнаяФорма».

С.4 Ребра, следующие за ребром «Рекомендация» («recommendation»), имеют значения от 1 до 26, которым присвоены идентификаторы от a до z. Последующие ребра имеют значения номеров рекомендаций МККТТ той серии, которая обозначается предыдущей буквой. Ребра, следующие за этими ребрами, определяются соответствующими рекомендациями МККТТ. Идентификаторы от a до z могут использоваться в качестве «ИменнойФормы».

С.5 Ребра, следующие за ребром «вопрос» («question»), имеют значения, определяемые номером исследовательской группы МККТТ и периодом времени. Эти значения вычисляются по следующей формуле:

номер исследовательской группы + (период * 32)

где «период» имеет значение 0 для 1984—1988 гг., 1 — для 1988—1992 гг., и т. д., а множитель 32 задан в десятичной системе счисления.

Ребра, следующие за каждым ребром «исследовательская группа», имеют значения, отвечающие вопросам, входящим в сферу деятельности соответствующей исследовательской группы. Последующие ребра определяются по мере необходимости соответствующей группой (например, рабочим совещанием или специальной группой экспертов), сформированной для изучения вопроса.

С.6 Ребра, следующие за ребром «администрация» («administration»), имеют значения цифровых кодов стран (DCC), определенных рекомендацией

МККТТ X.121. Последующие ребра определяются в соответствии с потребностями управлением телекоммуникаций страны, определенной кодом DCC.

C.7 Ребра, следующие за ребром «оператор сети» («network-operator»), имеют значения цифровых кодов сетей (DNIC), определенных в рекомендации МККТТ X.121. Последующие ребра определяются в соответствии с потребностями управлением телекоммуникаций или зарегистрированным частным эксплуатационным агентством, RPOA, идентифицируемым кодом DNIC.

ПРИЛОЖЕНИЕ D
Обязательное

СОВМЕСТНО ОПРЕДЕЛЯЕМЫЕ ЗНАЧЕНИЯ КОМПОНЕНТОВ OBJECT IDENTIFIER

D.1 Из корневой вершины дерева идентификаторов объектов выходят три ребра. Присвоенные им значения и идентификаторы, а также службы, ответственные за присвоение значений следующих компонентов, таковы:

Значение	Идентификатор	Служба, отвечающая за присвоение следующих компонентов
0	ccitt	МККТТ
1	iso	ИСО
2	joint-iso-ccitt	См. ниже

Примечание. В остальной части этого приложения рассматривается присвоение значений, осуществляемое совместно ИСО и МККТТ.

D.2 Каждый из идентификаторов «ccitt», «iso» и «joint-iso-ccitt», определенных выше, может использоваться в качестве выражения «СимвольнаяФорма».

D.3 Ребрам, следующим за ребром «совместное определение ИСО и МККТТ» («joint-iso-ccitt»), присвоены значения, периодически назначаемые по согласованию между ИСО и МККТТ для идентификации их совместных областей деятельности по стандартизации в соответствии с «Процедурами присвоения значений компонентам идентификаторов объектов для совместного использования ИСО и МККТТ»[1].

D.4 Ребра, которые следуют за упомянутыми в D.3, выделяются в соответствии с процедурами, действующими в момент выделения.

Примечание. Предполагается, что в данном случае для присвоения значений будет осуществляться передача полномочий смешанной группе экспертов ИСО и МККТТ, ответственных за область совместной работы.

D.5 Поскольку в первых международных стандартах и рекомендациях МККТТ, относящимся к совместным областям стандартизации, потребовалось выделение идентификаторов объектов (OBJECT IDENTIFIER) до установления процедур, упомянутых в D.3, то они выделяются так, как описано в приложениях B или C. Идентификаторы информационных объектов, определенные таким способом в международном стандарте или рекомендации МККТТ, не будут меняться после установления процедур, упомянутых в D.3.

[1] Регистрационной службой, ответственной за присвоение значений компонентам идентификаторов объектов для совместного использования ИСО и МККТТ, является Американский национальный институт стандартов (ANSI), 1430 Broadway, New York, NY

ПРИЛОЖЕНИЕ Е
Справочное

ПРИМЕРЫ И РЕКОМЕНДАЦИИ

В этом приложении приводятся примеры использования нотации АСН.1 для описания некоторых взятых произвольно структур данных. В нем также содержатся пояснения и рекомендации по использованию различных возможностей АСН 1.

Е.1 Пример записи учета кадров

В этом подразделе использование АСН.1 иллюстрируется на примере произвольно взятой простой записи учета кадров.

Е.1.1 Неформальное описание записи учета кадров

Ниже приводится структура записи учета кадров и ее значение для конкретного служащего.

Имя:	John P Smith
Должность:	Директор
Учетный номер:	51
Дата найма:	17 сентября 1971
Имя супруги:	Mary T Smith
Число детей:	2
Информация о ребенке	
Имя:	Ralph T Smith
Дата рождения:	11 ноября 1957
Информация о ребенке	
Имя:	Susan B. Tones
Дата рождения:	17 июня 1959

Е.1.2 Описание структуры записи с помощью АСН.1

Структура каждой записи учета кадров формально описана ниже с помощью стандартных обозначений для типов данных.

```
PersonnelRecord : : = [APPLICATION 0] IMPLICIT SET
{                                       Name,
title                                   [0] VisibleString,
number                                  EmployeeNumber,
dateOfHire                              [1] Date,
nameOfSpouse                            [2] Name,
children                                [3] IMPLICIT SEQUENCE OF
                                            ChildInformation DEFAULT { } }

ChildInformation : : = SET

{                                       Name
dateOfBirth                             [0] Date}

Name : = [APPLICATION 1] IMPLICIT SEQUENCE

{givenName                              VisibleString,
initial                                 VisibleString,
familyName                              VisibleString}

EmployeeNumber : = [APPLICATION 2] IMPLICIT INTEGER
Date : : = [APPLICATION 3] IMPLICIT VisibleString —— YYYYMMDD
```

В этом примере иллюстрируется один из аспектов синтаксического разбора выражений АСН.1 Синтаксическая конструкция «по умолчанию» («DEFAULT») может применяться только к элементам выражения «последовательность» («SEQUENCE») или «множество» («SET»), но не может быть применена к элементам «последовательность из» («SEQUENCE OF»). Поэтому выражение «DEFAULT { }» в «записи учета кадров» («PersonnelRecord») относится к выражению «дети» («children»), а не к выражению информация о ребенке» («ChildInformation»).

Е.1.3 Описание значения записи с помощью АСН 1

Значение записи учета кадров для описанного в начале подраздела, служащего в стандартных обозначениях для значений данных, будет иметь вид:

```
{       {givenName «John», initial «P», familyName «Smith»},
  title    «Director»
number   51
  dateOfHire «19710917»
  nameOfSpouse {givenName «Mary», initial «T», familyName «Smith»},
  children
  {     { {givenName «Ralph», initial «T», familyName «Smith»},
        dateOfBirth «19571111»},
{ {givenName «Susan», initial «B», familyName «Jones»},
                dateOfBirth «19590717»}} }
```

Е.2 Рекомендации по использованию нотации

Типы данных и формальная нотация, определенные в настоящем стандарте, обладают большой гибкостью, предоставляя возможность их использования в широком спектре протоколов. Эта гибкость, однако, может привести к недоразумениям, в особенности при недостаточном опыте ее использования. В этом приложении делается попытка свести к минимуму возможность недоразумений с помощью рекомендаций по использованию нотации и примеров ее использования. Для каждого из встроенных типов данных предлагается одна или несколько рекомендаций по использованию. Типы «строка знаков» (например, «отображаемая строка» («VisibleString») и типы, определенные в разделах 32—35, здесь не рассматривают.

Е.2.1 Булевский тип

Е.2.1.1 Булевский тип следует использовать для представления значений логических переменных (то есть имеющих два разных состояния), например, ответа на вопрос «да» или «нет»

Пример

```
Employed : : = BOOLEAN
```

Е.2.1.2 При назначении ссылочного имени булевскому типу следует использовать слово, подходящее для обозначения состояния «истина».

Пример

```
Married : : = BOOLEAN
```

а не

```
MaritalStatus : : = BOOLEAN
```

См. также Е.2.3.2.

Е.2.2 Целочисленный тип

Е.2.2.1 Целочисленный тип следует использовать для представления значений количественных числительных и целочисленных переменных для всех практических целей, без ограничения величины.

Пример

```
CheckingAccountBalance : : = INTEGER
    -- значение выражено в центах; отрицательное
    -- значение означает перерасход
```

Е.2.2.2 Следует определять допустимые максимальное и минимальное значения целочисленного типа в качестве выделенных значений.

Пример

DayOfTheMonth : : = INTEGER {first(1), last(31)}

Е.2.3 Перечислительный тип

Е.2.3.1 Перечислительный тип с выделенными значениями следует использовать для представления значений переменной с тремя или более состояниями. Если единственным требованием к значениям переменной является их отличие друг от друга, то значения следует присваивать, начиная с нуля.

Пример

DayOfTheWeek : : = ENUMERATED {sunday(0), monday(1), tuesday(2), wednesday(3), thursday(4), friday(5), saturday(6)}

Е.2.3.2 Перечислительный тип следует использовать для представления значений переменной, которая имеет только два состояния, но у которой могут появиться дополнительные состояния в следующей версии протокола.

Пример

Следует использовать описание

MaritalStatus : : = ENUMERATED {single(0), married(1)}

«если предполагается его последующая замена на следующее:

MaritalStatus : : = ENUMERATED {single(0), married(1), widowed(2)}

Е.2.4 Тип «действительное число»

Е.2.4.1 Тип «действительное число» следует использовать для представления приближенных значений.

Пример

AngleInRadians : : = REAL

pi REAL : : = {3141592653589793238462643383279, 10, —30}

Е.2.5 Тип «строка битов»

Е.2.5.1 Тип «строка битов» следует использовать для представления двоичных данных, формат и длина которых не заданы или заданы в другом месте, при этом длина данных в битах не обязательно кратна восьми.

Пример

G3FacsimilePage : : = BIT STRING

— — последовательность битов, удовлетворяющая

— — рекомендации МККТТ Т.4.

Е.2.5.2 Следует определять первый и последний имеющие смысл биты строки фиксированной длины в качестве выделенных битов.

Пример

Nibble : : = BIT STRING {first(0), last(3)}

Е.2.5.3 Тип «строка битов» следует использовать также для представления значений битового отображения, то есть упорядоченной совокупности логических переменных, показывающих, выполняется или нет некоторое определенное условие для каждого объекта из некоторой упорядоченной совокупности.

Пример

SunnyDaysOfTheMonth : : = BIT STRING {first(1), last(31)}

— — i-й день — солнечный в том и только

— — в том случае, если i-й бит равен единице

Е.2.5.4 Тип «строка битов» с поименованными значениями следует использовать для представления значений совокупности взаимосвязанных логических переменных.

Пример

PersonalStatus : : = BIT STRING

{married(0), employed(1), veteran(2), collegeGraduate(3)}

Е.2.6 Тип «строка октетов»

Е.2.6.1 Тип «строка октетов» следует использовать для представления значений двоичных данных, формат и длина которых не заданы или заданы в другом месте, причем длина данных в битах кратна восьми.

Пример

G4FacsimileImage : : = OCTET STRING

— — последовательность октетов,

— — удовлетворяющая рекомендациям

— — МККТТ Т.5 и Т.6

Е.2.6.2 В случаях, когда доступен для использования требуемый тип «строка знаков», его использование следует предпочесть использованию типа «строка октетов».

Пример

Surname : : = PrintableString

Е.2.6.3 Тип «строка октетов» следует использовать для представления любой строки информации, которая не может быть представлена с помощью одного из типов «строка знаков». Обязательно следует определить используемые знаки и способ их представления в виде октетов.

Пример

PackedBCDString : : = OCTET STRING

— — цифры от 0 до 9, две цифры на октет,

— — каждая цифра кодируется как 0000—1001,

— — комбинация 1111 используется для

— — дополнения до полного октета

Е.2.7 Вырожденный тип

Вырожденный тип («NULL») следует использовать в том случае, когда требуется задать отсутствие элемента некоторой последовательности.

Пример

PatientIdentifier : : SEQUENCE

{name VisibleString,

roomNumber CHOICE

{INTEGER,

NULL — — в случае выбывающего пациента — — }}

Примечание. Тот же результат может быть достигнут с помощью параметра «OPTIONAL».

Е.2.8 Типы «последовательность» и «последовательность из»

Е.2.8.1 Тип «последовательность из» следует использовать для представления совокупности переменных одного и того же типа, количество которых велико или непредсказуемо, и порядок следования существенен.

Пример

NamesOfMemberNations : : = SEQUENCE OF VisibleString

— — перечисление в порядке вступления

Е.2.8.2 Тип «последовательность» следует использовать для представления совокупности переменных одного и того же типа, количество которых невелико и известно, порядок следования существенен, при условии, что состав совокупности не должен меняться при переходе к следующей версии протокола.

Пример

NamesOfOffisers : : = SEQUENCE

{president VisibleString,

vicePresident VisibleString,

secretary VisibleString}

Е.2.8.3 Тип «последовательность» следует также использовать для представления совокупности переменных различных типов, количество которых невелико и известно, при условии, что состав совокупности не должен меняться при переходе к следующей версии протокола

Пример

```
Credentials : : = SEQUENCE
{userName VisibleString,
password VisibleString,
accountNumber INTEGER}
```

Е.2.8.4 Если в типе «последовательность» имеется фиксированное число элементов различных типов, то каждому элементу, чье назначение не очевидно из его типа, следует присвоить ссылочное имя.

Пример

```
File . : = SEQUENCE
{  ContentType,
other FileAttributes,
content ANY   }
```

См. также Е.2.3.3, Е.2.3.4 и Е.2.5

Е.2.9 Тип «множество»

Е.2.9.1 Тип «множество» следует использовать для представления совокупности переменных, количество которых невелико и известно, а порядок следования несущественен. Каждой переменной следует присвоить метку контекстно-зависимого класса.

Пример

```
UserName : . = SET
personalName                [0] IMPLICIT VisibleString
organisationName            [1] IMPLICIT VisibleString,
countryName                 [2] IMPLICIT VisibleString}
```

Е.2.9.2 Тип «множество» с включением параметра «OPTIONAL» следует использовать для представления совокупности переменных, которая является собственным или несобственным подмножеством другой совокупности переменных, причем количество переменных достаточно мало, а их порядок несущественен. Каждой переменной следует присвоить метку контекстно-зависимого класса.

Пример

```
UserName : . = SET
{personalName               [0] IMPLICIT VesibleString,
organisationName            [1] IMPLICIT VisibleString
                                OPTIONAL
— — по умолчанию — название данной организации — — ,
countryName                 [2] IMPLICIT VisibleString
                                OPTIONAL
— — по умолчанию — название данной страны — —       }
```

Е.2.9.3 Тип «множество» следует также использовать для представления совокупности переменных, состав которой с большой вероятностью будет меняться при переходе к следующей версии протокола. Каждой переменной следует присвоить метку контекстно-зависимого класса.

Пример

```
UserName : : = SET
{personalName               [0] IMPLICIT VisibleString,
organisationName            [1] IMPLICIT VisibleString
                                OPTIONAL
```

```
        по умолчанию — название данной организации — ,
    countryName [2] IMPLICIT VisibleString OPTIONAL
    -- — по умолчанию — название данной страны
    -- другие необязательные атрибуты оставлены для
    ---- дальнейшей проработки -- — }
```

Е.2.9.4 Если количество элементов, входящих в тип «множество», фиксировано, то каждому элементу, чье назначение не очевидно из его типа, следует присвоить ссылочное имя.

```
Пример
FileAttributes : : = SET
{owner                          [0] IMPLICIT UserName   ,
sizeOfContentInOctets           [1] IMPLICIT INTEGER    ,
                                [2] IMPLICIT AccessControls,
                                                        }
```

Е.2.9.5 Тип «множество» следует также использовать для представления совокупности переменных разных типов, порядок которых несущественен.

```
Пример
Keywords : : = SET OF VisibleString
-- — в произвольном порядке
```

См. также Е.2.5.4 и Е.2.10.3

Е.2.10 Тегированный тип

Е.2.10.1 В рамках настоящего стандарта тегированный тип универсального класса используют для определения широко используемых, не зависящих от конкретного применения типов данных, представление которых должно отличаться от представления всех других типов данных.

```
Пример
EncriptionKey  . = [UNIVERSAL 30] IMPLICIT OCTET STRING
-- семь октетов
```

Е.2.10.2 Тегированный тип прикладного класса следует использовать при определении типа данных, которые находят широкое, не локализованное использование в некотором контексте представления, причем представление этого типа должно отличаться от представления всех других типов данных в том же контексте представления.

```
Пример
FileName : : = [APPLICATION 8] IMPLICIT SEQUENCE
{directoryName                  VisibleString,
DirectoryRelativeFileName       VisibleString}
```

Е.2.10.3 Контекстно-зависимые тегированные типы используют для различения элементов множества. Если единственным требованием, предъявляемым к числовым тегам, является отличие их друг от друга, то их следует присваивать, начиная с нуля.

```
Пример
CustomerRecord . = SET
{name                           [0] IMPLICIT VisibleString,
mailingAddress                  [1] IMPLICIT VisibleString,
accountNumber                   [2] IMPLICIT VisibleString,
balanceDue                      [3] IMPLICIT VisibleString
                                    — в центах — — }
```

Е.2.10.4 В случаях, когда некоторый элемент множества имеет тег прикладного класса (APPLICATION), то не требуется присвоения нового контекстно-зависимого тега, за исключением случаев, когда требуется (или может потребоваться в будущем) обеспечить его отличие от других элементов множества. В случаях, когда элемент множества имеет тег универсального класса (UNIVERSAL), ему следует дополнительно присвоить контекстно-зависимый тег.

Пример

```
ProductRecord : : = SET
{                                   UniformCode
description                         [0] IMPLICIT INTEGER,
inventoryNo                         [1] IMPLICIT INTEGER,
inventoryLevel                      [2] IMPLICIT INTEGER}
UniformCode : : = [APPLICATION 13] IMPLICIT INTEGER
```

Е.2.10.5 Контекстно-зависимые тегированные типы следует использовать также при определении вариантов выборочного типа (CHOICE). Если единственным требованием к тегам является их отличие друг от друга, то следует начинать с нулевого значения.

Пример

```
CustomerAttribute : : = SET
{name                               [0] IMPLICIT VisibleString,
mailingAddress                      [1] IMPLICIT VisibleString,
accountNumber                       [2] IMPLICIT INTEGER,
balanceDue                          [3] IMPLICIT INTEGER
                                        -- в центах --  }
```

Е.2.10.6 В случаях, когда некоторый вариант, входящий в определение выборочного типа (CHOICE), представляет собой тип, имеющий тег прикладного класса (APPLICATION), то не требуется использовать контекстно-зависимый тег, за исключением случаев, когда нужно (или может понадобиться в будущем) различать между собой варианты выборочного типа.

Пример

```
ProductDesignator : : = CHOICE
{                                   UniformCode
description                         [0] IMPLICIT VisibleString,
inventoryNo                         [1] IMPLICIT INTEGER, }
UniformCode : : = [APPLICATION 13] IMPLICIT INTEGER
```

Е.2.10.7 В случаях, когда некоторый вариант, входящий в определение выборочного типа (CHOICE), представляет собой тип, имеющий тег универсального класса (UNIVERSAL), необходимо использовать контекстно-зависимый тег, за исключением случаев, когда ставится цель обеспечить выбор из нескольких (более одного) разных универсальных типов.

Пример

```
CustomerIdentifier : : = CHOICE
{name  VisibleString,
number INTEGER  }
```

Е.2.10.8 Тегированный тип пользовательского класса следует использовать для определения данных, используемых в рамках некоторой организации или страны, причем требуется, чтобы была возможность отличить эти данные (по их представлению) от всех остальных данных, используемых этой организацией или страной.

Пример

```
AcmeBadgeNumber : : = [PRIVATE 2] IMPLICIT INTEGER
```

Е.2.10.9 В приводящихся здесь рекомендациях во всех примерах, где это допустимо, используют «неявное» задание тегов, то есть с указанием «IMPLICIT». При использовании соответствующих правил кодирования это приводит к компактному представлению данных, что весьма желательно в некоторых приложениях. В других приложениях компактность может быть менее важна, чем, например, возможность выполнения детальной проверки типов данных. В последнем случае может использоваться явное задание тегов.

См. также Е.2.9.1, Е.2.9.2, Е.2.11.1 и Е.2.11.2.

Е.2.11 Выборочный тип

Е.2.11.1 Выборочный тип (CHOICE) следует использовать для представления переменной, выбираемой из совокупности переменных, количество которых известно и невелико. Каждая переменная из упомянутой совокупности идентифицируется с помощью контекстно-зависимого тега.

Пример

```
FileIdentifier : : = CHOICE
{relativeName [0] IMPLICIT VisibleString,
   — имя файла (например, «MarchProgressReport»)
absoluteName [1] IMPLICIT VisibleString,
— — имя файла и содержащий это имя каталог
— — (например, «<Williams>MarchProgressReport»)
serialNumber [2] IMPLICIT INTEGER
— — идентификатор, присвоенный файлу системой — — }
```

Е.2.11.2 Выборочный тип (CHOICE) следует использовать также для представления переменной, выбираемой из совокупности переменных, состав которых с большой вероятностью будет меняться при переходе от одной версии протокола к следующей версии. Каждую переменную из упомянутой совокупности следует идентифицировать с помощью контекстно-зависимого тега.

Пример

```
FileIdentifier : : = CHOICE
{relativeName [0] IMPLICIT VisibleString,
— — имя файла (например, «MarchProgressReport»)
absoluteName [1] IMPLICIT VisibleString,
— — имя файла и содержащий это имя каталог
— — (например, «<Williams>MarchProgressReport»)
— — другие формы идентификаторов файлов оставлены
— — для дальнейшего исследования — — }
```

Е.2.11.3 Каждому элементу, чье назначение не очевидно из его типа, следует присвоить ссылочное имя.

Пример

```
FileIdentifier : : = CHOICE
{relativeName [0] IMPLICIT VisibleString,
— — имя файла (например, «MarchProgressReport»)
absoluteName [1] IMPLICIT VisibleString,
— — имя файла и содержащий это имя каталог
    например, «<Williams>MarchProgressReport»)
serialNumber [2] IMPLICIT INTEGER
— — идентификатор, присвоенный файлу системой — — }
```

Е.2.11.4 В случаях, когда в некотором конкретном применении настоящего стандарта нормой является неявное задание тегов (то есть с указанием «IMPLICIT») и планируется переход от использования одного типа к использованию нескольких типов, рекомендуется использование выборочного типа (CHOICE), состоящего из одного варианта. Этим исключается возможность того, что будет иметь место неявное задание тегов, и таким образом облегчается переход к использованию нескольких типов.

Пример

```
Greeting : : = [APPLICATION 12] CHOICE
                {VisibleString}
```

в будущем предполагается переход к

```
Greeting : : = [APPLICATION 12] CHOICE
                {VisibleString,
                 Voice          }
```

Е.2.12 Селективный тип

Е.2.12.1 Селективный тип следует использовать для представления перемен-

ной, тип которой является типом некоторого варианта в определенном ранее выборочном типе.

E.2.12.2 Пусть дано определение

FileAttribute : : = CHOICE
{date-last-used INTEGER,
file-name VisibleString}

тогда возможно следующее определение:

CurrentAttributes : : = SEQUENCE
{date-last-used < FileAttribute,
file-name < FileAttribute}

с возможной записью значения

{date-last-used 27,
file name «PROGRAM»}

Возможно также следующее определение:

AttributeList : : = SEQUENCE
{first-attribute date-last-used < FileAttribute,
second-attribute file-name < FileAttribute}

с возможной записью значения

{first-attribute 27,
second-attribute «PROGRAM»}

E.2.13 Произвольный тип

E.2.13.1 Произвольный (ANY) тип следует использовать для представления переменной, тип которой не задан или задан в другом месте с помощью нотации АСН 1

Пример

MessageContents : : = ANY

— — элемент данных, тип которого определен с помощью
. . . нотации АСН.1 за рамками настоящего стандарта

E.2.14 Внешний тип

E.2.14.1 Внешний (EXTERNAL) тип следует использовать для представления переменной, тип которой не задан или задан в другом месте без ограничений на используемую для этого нотацию.

Пример

FileContents : : = EXTERNAL
DocumentList : : = SEQUENCE OF EXTERNAL

E.3 Пример использования макронотации

Предположим, что желательно иметь обозначения для определения типа следующего вида:

PAIR TYPEX = . . . TYPEY = . . .

допускающие запись соответствующего значения в виде

(X = — — — —, Y = — — — —)

где . . . и — — — — означают, соответственно, любой тип АСН 1 и соответствующее значение

Такие макрообозначения типа могли бы быть использованы для определения типов и значений, как показано ниже:

T1 : : = PAIR TYPEX = INTEGER
TYPEY = BOOLEAN
T2 : : = PAIR TYPEX = VisibleString
TYPEY = T1

Тогда значение типа T1 может иметь вид:

(X = 3, Y = TRUE)

а значение типа T2 может иметь вид:

(X = «Name», Y = (X = 4, Y = FALSE))

Для обеспечения этих обозначений, которые могут рассматриваться как расширение базовой нотации ACH.1, может быть использовано следующее макроопределение:

```
PAIR
MACRO ::=BEGIN
TYPE NOTATION ::=
  «TYPEX»
  «,»
  type (Local-type-1)
       -- Здесь ожидается описание любого типа
       -- ACH.1, которое будет присвоено
       -- переменной Local-type-1;
  «TYPEY»
  «,»
  type (Local-type-2)
            Здесь ожидается описание второго типа
          -- ACH.1, которое будет присвоено
          -- переменной Local-type-2;
VALUE NOTATION ::=
  «(»
  «X»
  «,»
  value (Local-value-1 Local-type-1)
            Здесь ожидается описание значения типа,
          -- определяемого переменной Local-type-1,
          -- которое будет присвоено переменной
             Local-value-1;
  «,»
  «Y»
  «,»
  value (Local-value-2 Local-type-2)
          -- Здесь ожидается описание значения типа,
          -- определяемого переменной Local-type-2,
          -- которое будет присвоено переменной
          -- Local value-2;
  <VALUE SEQUENCE (Local-type 1,
                   Local-type-2)
     ::= {Local-value-1, Local-value-2}>
            Это «вложенное определение» возвращает
            окончательное значение, являющееся
            значением последовательности,
       -- состоящей из двух типов
  «)»
END
```

В этом примере тип возвращаемого значения (в смысле базовой нотации ACH.1) не зависит от подставляемого значения, но зависит от подставляемого типа. В других случаях тип может либо полностью определяться макроопределением, либо может зависеть от подставляемого значения. Отметим, тем не менее, что во всех случаях для определения типа возвращаемого значения следует рассматривать продукцию «VALUE NOTATION». Продукция «TYPE NOTATION» просто определяет синтаксис для определения типа и устанавливает начальные значения локальных переменных, используемых при анализе подставляемой нотации значения.

Е.4 Использование нотации для определения абстрактных синтаксисов

Е.4.1 Для использования услуг уровня представления (ГОСТ 34.971) требуется описание значений, называемых значениями данных уровня представления, и группирование этих значений в множества, называемые абстрактными синтаксисами Каждому из этих множеств присваивается имя абстрактного синтаксиса, относящееся к определенному в АСН.1 типу «идентификатор объекта».

Е.4.2 Нотация АСН.1 может быть использована в качестве единого средства при описании значений данных уровня представления и их группировании в поименованные абстрактные синтаксисы.

Е.4.3 Для простейших случаев такого использования в нотации АСН.1 может найтись единственный тип, такой, что все значения данных представления, составляющие поименованный абстрактный синтаксис, будут значениями этого типа. Как правило, этим типом является выборочный тип, и все значения данных представления будут относиться к различным типам-вариантам из этого выборочного типа. В этом случае рекомендуется оформить определяемые данные в виде модуля, в котором первым определяемым типом будет вышеупомянутый выборочный тип, а далее будут приведены определения тех (не универсальных) типов, на которые имеются прямые или косвенные ссылки в определении выборочного типа.

Примечание. Это не означает, что исключаются ссылки на типы, определенные в других модулях.

Е.4.4 Ниже приведен пример текста, который мог бы быть частью некоторого стандарта, относящегося к прикладному уровню. Для исключения неоднозначности конец примера явно выделен строкой «Конец примера».

Пример

```
ISOxxxx—yyyy DEFINITIONS ::=
BEGIN
PDU ::= CHOICE
          {connect-pdu . . . . ,
          data pdu CHOICE
              { .          .
                          },
      . . . . . . }
. . . . .
END
```

Настоящий стандарт присваивает следующее значение АСН.1 — идентификатор объекта

{iso standard xxxx abstract-syntax (1)}

в качестве имени абстрактного синтаксиса для множества значений данных представления, каждое из которых является значением типа данных АСН.1 «ISOxxxx—yyyy.PDU». Соответствующим описателем объекта АСН.1 устанавливается

« . . . . . »

В качестве имени синтаксиса передачи, используемым вместе с определенным выше именем абстрактного синтаксиса, могут использоваться значения АСН.1 для идентификатора объекта и описателя объекта.

{joint-iso-ccitt asn1 (1) basic-encoding (1)}

и

«Basic Encoding of a single ASN.1 type»

(присваиваемые информационному объекту в ГОСТ 34.974.

Конец примера

Е.4.5 Стандарт может также потребовать для этого абстрактного синтаксиса

обязательной поддержки синтаксиса передачи, получаемого применением правил кодирования

{joint-iso-ccitt asn1 (1) basic-encoding (1)}

Е.5 **Подтипы**

Е.5.1 Подтипы следует использовать для ограничения значений существующего типа множеством значений, допустимых к конкретной ситуации.

Примеры

```
AtomicNumber ::= INTEGER (1..104)
TouchToneString ::= IA5String (FROM («0» | «1» | «2» | «3» | «4» | «5» |
                                «6» | «7» | «8» | «9» | «*» | «#»)) «SIZE(1..63))
ParameterList ::= SET SIZE(0..63) OF Parameter
SmallPrime ::= INTEGER (2 | 3 | 5 | 7 | 11 | 13 | 17 | 19 | 23 | 29)
```

Е.5.2 В случаях, когда два или более типа имеют много общего, следует рассмотреть возможность явного определения в качестве исходного типа их общего порождающего типа и использовать механизм образования подтипов для определения требуемых типов. Этот подход подчеркивает взаимосвязь и общность типов и является привлекательным (хотя и необязательным) подходом к развитию типов. Такой подход облегчает, таким образом, реализацию обработки значений этих типов.

Пример

```
Envelope ::= SET {typeA TypeA,
                  typeB TypeB OPTIONAL,
                  typeC TypeC OPTIONAL}
                  -- — общий порождающий тип
ABEnvelope ::= Envelope (WITH COMPONENTS
                  {...,
                  typeB PRESENT, typeC ABSENT})
                  -- — где тип typeB должен всегда
                  -- — присутствовать, а тип typeC
                  -- — должен отсутствовать
ACEnvelope ::= Envelope (WITH COMPONENTS
                  {...,
                  typeB ABSENT, typeC PRESENT})
                  -- — где тип typeC должен всегда
                  -- — присутствовать, а тип typeB
                  -- — должен отсутствовать
```

Два последних определения могут быть также записаны в следующем виде:

```
ABEnvelope ::= Envelope (WITH COMPONENTS{typeA, typeB})
ACEnvelope ::= Envelope (WITH COMPONENTS{typeA, typeC})
```

Выбор между альтернативными вариантами описания должен основываться на таких факторах, как число компонентов в порождающем типе, число необязательных компонентов, значительность различий между отдельными типами, а также вероятность их совместной дальнейшей эволюции.

Е.5.3 Механизм подтипов следует использовать для частичного определения значения, например, протокольного блока данных, с которым будет производиться сравнение в тесте конформности, при условии, что этот тест конформности анализирует только некоторые компоненты ПБД.

Пример

Пусть задан протокольный блок данных

```
PDU ::= SET
            {alpha [0] INTEGER,
            beta [1] IA5String OPTIONAL,
            gamma [2] SEQUENCE OF Parameter,
            delta [3] BOOLEAN}
```

и необходимо составить тест, требующий, чтобы булевское значение было «ложно», а целочисленное значение было отрицательным. Тогда может быть использовано описание

```
TestPDU ::= PDU (WITH COMPONENTS
                 {...,
                 delta(FALSE),
                 alpha(MIN ... <0)})
```

Если к тому же требуется, чтобы значение типа-компонента IA5String, beta, обязательно присутствовало и имело длину либо 5, либо 12 знаков, может быть использовано описание:

```
FutherTestPdu ::= TestPDU (WITH COMPONENTS
                           {...,
                           beta (SIZE(5 | 12)) PRESENT})
```

Е.5.4 Если тип данных общего назначения был определен как SEQUENCE OF, следует использовать механизм образования подтипов для определения подтипа, являющегося частым случаем общего типа.

Пример

```
Text-block ::= SEQUENCE OF VisibleString
Addres ::= Text-block (SIZE(1..6) : WITH COMPONENT(SIZE(1..32)))
```

Е.5.5 Для образования новых подтипов из существующих подтипов следует использовать вложенные подтипы.

Пример

```
Months ::= ENUMERATED {
                       january(1),
                       february(2),
                       march(3),
                       april(4),
                       may(5),
                       june(6),
                       july(7),
                       august(8),
                       september(9),
                       october(10),
                       november(11),
                       december(12)}
First-quarter ::= Months {
                          january |
                          february |
                          march}
Second-quarter ::= Months {
                           april |
                           may |
                           june}
Third-quarter ::= Months {
                          july |
                          august |
                          september}
Fourth-quarter ::= Months {
                           october |
                           november |
                           december}
First-half ::= Months {
                       INCLUDES First-quarter |
                       INCLUDES Second-quarter}.
```

```
Second-half ::= Months {
                    INCLUDES Third-quarter ,
                    INCLUDES Fourth-quarter}
```

ПРИЛОЖЕНИЕ F
Справочное

СВОДКА СИНТАКСИСА НОТАЦИИ АСН.1

Перечисленные ниже элементы определяются в разделе 8

ССылкаНаТип	INTEGER	BEGIN
Идентификатор	BIT	END
ССылкаНаЗначение	STRING	DEFINITIONS
ССылкаНаМодуль	OCTET	EXPLICIT
Комментарий	NULL	ENUMERATED
Пусто	SEQUENCE	EXPORTS
Число	OF	IMPORTS
b-строка	SET	
h-строка	IMPLICIT	REAL
c-строка	CHOICE	INCLUDES
« : : = »	ANY	MIN
{	EXTERNAL	MAX
}	OBJECT	SIZE
<	IDENTIFIER	FROM
,	OPTIONAL	WITH
	DEFAULT	COMPONENT
(	COMPONENTS	PRESENT
)	UNIVERSAL	ABSENT
[	APPLICATION	DEFINED
]	PRIVATE	BY
—	TRUE	PLUS-INFINITY
BOOLEAN	FALSE	MINUS-INFINITY

Ниже приведены продукции, используемые в настоящем стандарте; терминальными символами продукций являются приведенные выше элементы.

```
ОпределениеМодуля : : =
    ИдентификаторМодуля
    DEFINITIONS
    ТегиПоУмолчанию
    « : : = »
    BEGIN
    ТелоМодуля
    END
ТегиПоУмолчанию : : =
    EXPLICIT TAGS |
    IMPLICIT TAGS |
    пусто
ИдентификаторМодуля : : =
    ССылкаНаМодуль
    ПрисвоенныйИдентификатор
```

```
ПрисвоенныйИдентификатор : : =
    Значение-ИдентификаторОбъекта |
    пусто
ТелоМодуля : : =
    ВыходныеССылки ВходныеССылки СписокПрисвоений |
    пусто
ВыходныеССылки : : =
    EXPORTS ВыходныеСимволы |
    пусто
ВыходныеСимволы : : =
    СписокСимволов |
    пусто
ВходныеССылки : : =
    IMPORTS ВходныеСимволы |
    пусто
ВходныеСимволы : : =
    СписокСимволовИзМодуля |
    пусто
СписокСимволовИзМодуля : : =
    СимволыИзМодуля СписокСимволовИзМодуля |
    СимволыИзМодуля
СимволыИзМодуля : : =
    СписокСимволовFROM ИдентификаторМодуля
    СписокСимволов : : =
    Символ, СписокСимволов | Символ
Символ : : = ССылкаНаТип-| ССылкаНаЗначение
СписокПрисвоений : : = Присвоение
                       СписокПрисвоений Присвоение
Присвоение : : = ПрисвоениеТипа | ПрисвоениеЗначения
ВнешняяССылкаНаТип : : =
                              ССылкаНаМодуль
                              .
                              ССылкаНаТип
ВнешняяССылкаНаЗначение : : =
                                    ССылкаНаМодуль
                                    .
                                    ССылкаНаЗначение
ОпределенныйТип      = ВнешняяССылкаНаТип |
                       ССылкаНаТип
ОпределениеЗначение : : = ВнешняяССылкаНаЗначение |
                          ССылкаНаЗначение
ПрисвоениеТипа :  = ССылкаНаТип
                    « : : = »
                    Тип
ПрисвоениеЗначения : : = ССылкаНаЗначение
                         Тип
                         « : : = »
                         Значение
Тип : : = ВстроенныйТип | ОпределенныйТип | Подтип
ВстроенныйТип : : =  БулевскийТип
                     ЦелочисленныйТип
                     Тип-СтрокаБитов
                     Тип-СтрокаОктетов
                     ВырожденныйТип
```

Тип-Последовательность
Тип-ПоследовательностьИз
Тип-Множество
Тип-МножествоИз
ВыборочныйТип
СелективныйТип
ТегированныйТип
ПроизвольныйТип
Тип-ИдентификаторОбъекта
Тип-СтрокаЗнаков
ОбщеупотребительныйТип
ПеречисленныйТип
Тип-ДействительноеЧисло

ПоименованныйТип ::= Идентификатор Тип | Тип | СелективныйТип

Значение ::= ВстроенноеЗначение | ОпределенноеЗначение

ВстроенноеЗначение ::= БулевскоеЗначение
ЦелочисленноеЗначение
Значение-СтрокаБитов
Значение-СтрокаОктетов
ВырожденноеЗначение
Значение-Последовательность
Значение-ПоследовательностьИз
Значение-Множество
Значение-МножествоИз
ВыборочноеЗначение
СелективноеЗначение
ТегированноеЗначение
ПроизвольноеЗначение
Значение-ИдентификаторОбъекта
Значение-СтрокаЗнаков
ПеречислительноеЗначение
Значение-ДействительноеЧисло

ПоименованноеЗначение ::= Идентификатор Значение | Значение

БулевскийТип ::= BOOLEAN

БулевскоеЗначение ::= TRUE | FALSE

ЦелочисленныйТип ::= INTEGER | INTEGER {СписокПоименованныхЧисел}

СписокПоименованныхЧисел ::= ПоименованноеЧисло | СписокПоименованныхЧисел, ПоименованноеЧисло

ПоименованноеЧисло ::= Идентификатор (ЧислоСоЗнаком) | Идентификатор (ОпределенноеЗначение)

```
ЧислоСоЗнаком                    ::= Число | - Число
ЦелочисленноеЗначение            ::= ЧислоСоЗнаком | Идентификатор
ПеречислительныйТип              ::= ENUMERATED {Перечисление}
Перечисление                     ::= ПоименованноеЧисло
                                   | ПоименованноеЧисло, Перечисление
ПеречислительноеЗначение         ::= Идентификатор
Тип-ДействительноеЧисло          ::= REAL
Значение-ДействительноеЧисло     ::=
                                   ЧисленноеДействительноеЗначение
                                   | СпециальноеДействительноеЗначение
ЧисленноеДействительноеЗначение  ::=
                                   {Мантисса, Основание, Порядок} | 0
Мантисса                         ::= ЧислоСоЗнаком
Основание                        ::= 2 | 10
Порядок                          ::= ЧислоСоЗнаком
СпециальноеДействительноеЗначение ::=
                                   PLUS-INFINITY | MINUS-INFINITY
Тип-СтрокаБит                    ::= BIT STRING
                                   | BIT STRING {СписокПоименованныхБитов}
СписокПоименованныхБит           ::= ПоименованныйБит
                                   | СписокПоименованныхБит, ПоименованныйБит
ПоименованныйБит                 ::= Идентификатор(Число)
                                   | Идентификатор(ОпределениеЗначение)
Значение-СтрокаБит               ::= b-строка | h-строка
                                   | {СписокИдентификаторов} | { }
СписокИдентификаторов            ::= Идентификатор
                                   | СписокИдентификаторов, Идентификатор
Тип-СтрокаОктетов                ::= OCTET STRING
Значение-СтрокаОктетов           ::= b-строка | h-строка
ВырожденныйТип                   ::= NULL
ВырожденноеЗначение              ::= NULL
Тип-Последовательность           ::=
                                   SEQUENCE{СписокТипов-Компонентов}
                                   | SEQUENCE { }
СписокТипов-Компонентов          ::=
                                   СписокТипов-Компонентов, Тип-Компонент
Тип-Компонент                    ::= ПоименованныйТип
                                   | ПоименованныйТип OPTIONAL
                                   | ПоименованныйТип DEFAULT Значение
                                   | COMPONENTS OF Тип
Значение-Последовательность      ::=
                                   СписокЗначений-Компонентов | { }
СписокЗначений-Компонентов       ::= Поименованное Значение
                                   | СписокЗначений-Компонентов, ПоименованноеЗначение
Тип-ПоследовательностьИз         ::= SEQUENCE OF Тип | SEQUENCE
Значение-ПоследовательностьИз    ::= {СписокЗначений} | { }
СписокЗначений                   ::= Значение | СписокЗначений, Значение
Тип-Множество                    ::= SET{СписокТипов-Компонентов}
                                   | SET { }
```

```
Значение-Множество                  : : = {СписокЗначений-Компонентов} | {}
Тип-МножествоИз                     : : = SET OF Тип | SET
Значение-МножествоИз                : : = {СписокЗначений} | { }
ВыборочныйТип                       : : = CHOICE{СписокТипов-Вариантов}
СписокТипов-Вариантов               : : = ПоименованныйТип |
                                          СписокТипов-Вариантов, ПоименованныйТип
ВыборочноеЗначение                  : : = ПоименованноеЗначение
СелективныйТип                      : : = Идентификатор < Тип
СелективноеЗначение                 : : = ПоименованноеЗначение
ТегированныйТип                     : : = Тег Тип |
                                          Тег IMPLICIT Тип |
                                          Тег EXPLICIT Тип
Тег                                 : : = [Класс НомерВКлассе]
НомерВКлассе                        : : = Число | ОпределенноеЗначение
Класс                               : : = UNIVERSAL | APPLICATION |
                                          PRIVATE | Пусто
ТегированноеЗначение                : : = Значение
ПроизвольныйТип                     : : = ANY | ANY DEFINED BY
                                          идентификатор
ПроизвольноеЗначение                : : = Тип Значение
Тип-ИдентификаторОбъекта            : : = OBJECT IDENTIFIER
Значение-ИдОбъекта                  : : = {СписокКомпонентовИдОбъекта} |
                                          {ОпределенноеЗначение СписокКомпонентовИдОбъекта}
СписокКомпонентовИдОбъекта          = КомпонентИдОбъекта |
                                          КомпонентИдОбъекта СписокКомпонентовИдОбъекта
КомпонентИдОбъекта                  : : = ИменнаяФорма | ЧисловаяФорма |
                                          СмешаннаяФорма
ИменнаяФорма                        : : = Идентификатор
ЧисловаяФорма                       : : = Число | ОпределенноеЗначение
СмешаннаяФорма                      : : = Идентификатор(ЧисловаяФорма)
Тип-СтрокаЗнаков                    : : = ССылкаНаТип
Значение-СтрокаЗнаков               : : = с-строка
ОбщеупотребительныйТип              : : = ССылкаНаТип
```

Перечисленные ниже типы «строка знаков» определены в разделе 31

NumericString	VisibleString
PrintableString	ISO646String
TeletexString	IA5String
T61String	GraphicString
VideotexString	GeneralString

Общеупотребительные типы, имена которых перечислены ниже, определены в разделах 32—35:

GeneralizedTime	EXTERNAL
UTCTime	ObjectDescriptor

Приведенные ниже продукции используют в разделах 36—37

```
Подтип : : =
    ПорождающийТип СпецификацияПодтипа |
    SET ОграничениеРазмера OF Тип |
    SEQUENCE ОграничениеРазмера OF Тип
ПорождающийТип : : = Тип
СпецификацияПодтипа : : =
    (ПодмножествоЗначенийПодтипа
    СписокПодмножествЗначенийПодтипа)
СписокПодмножестваЗначенийПодтипа : : =
                    « | »
                ПодмножествоЗначенийПодтипа
```

СписокПодмножествЗначенийПодтипа |
пусто

ПодмножествоЗначенийПодтипа : : =
ОдноЗначение |
ВложенныйПодтип |
ДиапазонЗначений |
РазрешенныйАлфавит ОграничениеРазмера |
ОграниченияВнутреннихТипов

ОдноЗначение : : = Значение
ВложенныйПодтип : : = INCLUDES Тип
ДиапазонЗначений : : = НижняяГраница . . ВерхняяГраница
НижняяГраница : : = НижнееГраничноеЗначение |
НижнееГраничноеЗначение <
ВерхняяГраница : : = ВерхнееГраничноеЗначение |
< ВерхнееГраничноеЗначение
НижнееГраничноеЗначение : : = Значение | MIN
ВерхнееГраничноеЗначение : : = Значение | MAX
ОграничениеРазмера : : = SIZE СпецификацияПодтипа
РазрешенныйАлфавит : : = FROM СпецификацияПодтипа
ОграниченияВнутреннихТипов : : =
WITH COMPONENT ОграничениеОдногоТипа |
WITH COMPONENTS ОграниченияНескольких Типов
ОграничениеОдногоТипа : : = СпецификацияПодтипа
ОграниченияНескольких Типов : : =
ПолнаяСпецификация | ЧастичнаяСпецификация
ПолнаяСпецификация : : = {ОграниченияТипа}
ЧастичнаяСпецификация : : = {. . . , ОграниченияТипа}
ОграничениеТипа : : = ПоименованноеОграничение |
ПоименованноеОграничение ОграниченияТипа
ПоименованноеОграничение : : = Идентификатор Ограничение |
Ограничение
Ограничение : : = ОграничениеЗначения ОграничениеПрисутствия
ОграничениеЗначения : : = СпецификацияПодтипа | пусто
ОграничениеПрисутствия : : = PRESENT | ABSENT | пусто |
OPTIONAL

Перечисленные ниже дополнительные элементы определены в разделе А.2 и используются в макронотации.

макроссылка	«number»
ССылкаНаПродукцию	«empty»
ЛокальнаяССылкаНаТип	MACRO
ЛокальнаяССылкаНаЗначение	TYPE
«	»	NOTATION
>	VALUE
а-строка	value
«string»	type
«identifier»	

Ниже приводятся продукции, используемые в приложении А; терминальными символами этих продукций являются приведенные выше элементы, а также элементы, перечисленные в начале настоящего приложения.

Макроопределение : : = макроссылка
MACRO
« : : = »
СодержаниеМакро

```
    СодержаниеМакро : : =
                        BEGIN ТелоМакро END |
                        макроссылка |
                        ВнешняяМакроссылка
    ТелоМакро : : = ПродукцияТипа
                    ПродукцияЗначения
                    ВспомогательныеПродукции
    ПродукцияТипа : : =
                      TYPE NOTATION
                      «: : = »
                      СписокВариантовМакро
    ПродукцияЗначения : : =
                      VALUE NOTATION
                      «: : = »
                      СписокВариантовМакро
    ВспомогательныеПродукции : : =
                             СписокПродукций |
                             empty
    СписокПродукций : : =
                      Продукция |
                      СписокПродукций Продукция
    Продукция : : = ССылкаНаПродукцию
                    «: : = »
                    СписокВариантовМакро
ВнешняяМакроссылка : : =
                       ССылкаНаМодуль.макроссылка
    СписокВариантовМакро : : =
                       ВариантМакро |
                       СписокВариантовМакро
                       « | »
                       ВариантМакро
    ВариантМакро : : = СписокСимволов
    СписокСимволов : : =
      СимвольныйЭлемент |
      СписокСимволов СимвольныйЭлемент,
    Символьный Элемент : : =
      ОпределениеСимвола |
      ВложенноеОпределение
    МакроТип : : = ЛокальнаяССылкаНаТип |
              Тип
ОпределениеСимвола : : =
  «a-строка»                                    |
  ССылкаНаПродукцию                             |
  «string»                                      |
  «identifier»                                  |
  «number»                                      |
  «empty»                                       |
  type                                          |
  type (ЛокальнаяССылкаНаТип)                   |
  value (МакроТип)                              |
  value (ЛокальнаяССылкаНаЗначение МакроТип)    |
  value (VALUE МакроТип)
ВложенноеОпределение : : = <СписокВложенныхОпределений>
```

```
СписокВложенныхОпределений  ::= ВложенноеОпределение |
                                СписокВложенныхОпределений
                                ВложенноеОпределение
ВложенноеОпределение ::= ЛокальноеПрисвоениеТипа |
                         ЛокальноеПрисвоениеЗначения
ЛокальноеПрисвоениеТипа ::= ЛокальнаяСсылкаНаТип
                            «::=»
                            МакроТип
ЛокальноеПрисвоениеЗначения ::=
                                ЛокальнаяСсылкаНаЗначение
                                МакроТип
                                «::=»
                                МакроЗначение
МакроЗначение ::= Значение |
                  ЛокальнаяСсылкаНаЗначение
```

УДК 681.324:006.354 П85

Ключевые слова: информационная технология, взаимосвязь открытых систем, спецификация абстрактно-синтаксической нотации, простой тип, тег, значения типов, методы построения новых типов, структурированные типы, нотация АСН.1, абстрактный синтаксис информации, протоколы прикладного уровня, правила кодирования

ОКСТУ 4002

Редактор *В. П. Огурцов*
Технический редактор *В. Н. Прусакова*
Корректор *А. С. Черноусова*

Сдано в набор 05.01.94. Подп. в печ. 04.03.94. Усл. печ. л. 5,81. Усл. кр.-отт. 5,96. Уч.-изд. л. 6,20. Тир. 325 экз. С 1067

gost.e-ecolog.ru

Ордена «Знак Почета» Издательство стандартов, 107076, Москва, Колодезный пер., 14.
Калужская типография стандартов, ул. Московская, 256. Зак. 223