ГОСУДАРСТВЕННЫЙ СТАНДАРТ
СОЮЗА ССР

ИНФОРМАЦИОННАЯ ТЕХНОЛОГИЯ

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

ГОСТ 34.973—91
(ИСО 8824—87)

Издание официальное

48 р. 55 к. БЗ 7—91/794

КОМИТЕТ СТАНДАРТИЗАЦИИ И МЕТРОЛОГИИ СССР
Москва

gost.e-ecolog.ru

УДК 681.224 : 621.391 : 006.354 Группа П85

ГОСУДАРСТВЕННЫЙ СТАНДАРТ СОЮЗА ССР

Информационная технология

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

ГОСТ 34.973—91 (ИСО 8824—87)

Information technology.
Open systems interconnection.
Specification of abstract syntax notation one (ASN.1)

ОКСТУ 0034

Дата введения 01.07.92

Настоящий государственный стандарт распространяется на синтаксис передачи данных в прикладном уровне базовой эталонной модели взаимосвязи открытых систем и устанавливает спецификацию абстрактно-синтаксической нотации версии 1 (АСН.1), описывающую значения типов данных пользователя без указания конкретного способа представления этих значений при их передаче.

Требования настоящего стандарта являются обязательными.

Настоящий государственный стандарт эквивалентен стандарту Международной организации по стандартизации, за исключением:

а) упорядоченного использования аббревиатур;

б) в разд. 7.1 в примечание 2 введена новая табл. 3, содержащая знаки «кириллицы» с введением новой формулировки текста примечания 2;

в) введено дополнительно приложение 8, содержащее примеры использования идентификаторов объектов в виде графов компонентов идентификаторов объектов.

## 0. ВВЕДЕНИЕ

На нижних уровнях базовой эталонной модели (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. Стандарты по прикладному уровню

Издание официальное

© Издательство стандартов, 1991
Настоящий стандарт не может быть полностью или частично воспроизведен, тиражирован и распространен без разрешения Госстандарта СССР

требуют, чтобы данные пользователя услуг представления могли принимать значения весьма сложных типов, возможно, включающих строки знаков из различных знаковых наборов. Эти стандарты требуют наличия определенной нотации, которая определяла бы передаваемое значение, не задавая конкретный способ представления этого значения. 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 позволяет взаимодействующим объектам договариваться о том, какие синтаксисы передачи будут ими использоваться.

Цель задания значения состоит в том, чтобы отличить его от всех возможных других значений. Объединение некоторого значения с другими значениями, отличными от данного, называется типом, 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 используется термин «тип данных».

Для правильной интерпретации конкретного представления значения (например с помощью пометок на бумаге или битов, 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 типов и последующего образования различных комбинаций из этих простых типов. Некоторые из способов определения новых типов перечислены ниже:

а) имея (упорядоченный) список существующих типов, можно сформировать новое значение как последовательность, в которую входит по одному значению из каждого типа, входящего в список; совокупность всех возможных значений, получаемых 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);

б) имея список существующих типов можно сформировать новое значение как (неупорядоченное) множество значений, в которое входит по одному значению из каждого типа, входящего в список; совокупность всех значений, получаемых таким обра-

зом образует новый тип (этот способ можно также расширить, допуская пропуск некоторых значений);

в) имея один существующий тип, можно сформировать новое значение как (упорядоченную) последовательность или (неупорядоченное) множество, состоящее из пустого множества значений, одного или более значений исходного типа; (неограниченная) совокупность всех возможных значений, полученных таким образом, образует новый тип;

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

д) имея некоторый тип, можно сформировать новый тип как подмножество 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.

Настоящей нотацией определяется четыре класса тегов.

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

а) присвоен какому-то одному типу, либо

б) присвоен некоторому способу построения типов.

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

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

Четвертый класс тегов называется контекстно-зависимым классом. 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 обработку данных в машине 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 определить новый тип, присваивая новый тег ранее определенному типу

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

Настоящий стандарт определяет нотацию, которая является как средством определения сложных типов данных, так 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 (АСН.1). Абстрактно-синтаксическая нотация версии 1 является полуформальным средством, используемым при определении протоколов. Использование настоящей нотации само по себе не исключает неоднозначность соответствующих спецификаций. Обеспечение однозначности спецификаций должно осуществляться пользователем нотации.

Настоящий стандарт должен применяться вместе с другими стандартами, определяющими правила кодирования. Применение правил кодирования к значениям некоторго типа, установленным в АСН.1, полностью определяет представление этих значений при передаче, т. е. синтаксис передачи.

С технической точки зрения настоящий стандарт согласуется с соответствующими частями рекомендации МККТТ Х. 409 (1984).

Разд. 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 главе 1 также описываются обозначения, которые следует использовать при задании значений типов, определенных с помощью АСН.1.

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

В разд. 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.

Приложение 1 определяет нотацию, расширяющую базовую нотацию АСН.1. Эти расширения называются макросредствами.

Приложение 2 содержит определение логического дерева идентификаторов объекта для источников идентификации.

Приложение 3 содержит определение логического дерева идентификаторов объектов для источников идентификации.

Приложение 4 содержит определение логического дерева идентификаторов объектов для совместного использования ИСО и МККТТ.

Приложение 5 содержит примеры и рекомендации по использованию нотации АСН.1.

Приложение 6 содержит сводку синтаксиса нотации АСН.1, использующую правила описания синтаксиса, описанные в разд. 5.

В приложении 7 приводится алфавитный указатель ключевых слов, типов и основных терминов.

Приложение 8 содержит графы компонентов идентификаторов объектов (введено дополнительно для иллюстрации примеров использования идентификаторов объектов).

Приложение 9 содержит ссылочные нормативно-технические документы

Текст настоящего стандарта (в особенности это относится к приложениям 2—4) является результатом договоренности между ИСО и МККТТ.

## 1. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ

Настоящий стандарт описывает нотацию для определения абстрактного синтаксиса, называемую абстрактно-синтаксической нотацией версии 1 (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 (с помощью ссылок на другие стандарты).

Нотация АСН.1 может применяться во всех случаях, где требуется определить абстрактный синтаксис информации. Одной из сфер ее применения (но не единственной) являются стандарты по протоколам 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.

## 2. ССЫЛКИ

ИСО 8601 (ГОСТ 24520) «Цифровое обозначение дат») «Элементы данных и форматы обмена. Обмен информацией. Представление дат и времени».

ИСО 3166 «Коды для представления названий стран».

(ГОСТ 28906) «Системы обработки информации. Взаимосвязь открытых систем. Базовая эталонная модель» (см. также Рекомендацию МККТТ Х.200).

ИСО 8825 (ГОСТ 34.974) «Системы обработки информации. Взаимосвязь открытых систем. Описание базовых правил кодирования для абстрактно-синтаксической нотации версии 1 (АСН.1)».

МККТТ Х.409 «Системы обработки сообщений. Синтаксис и нотация представления информации при передаче».

## 3. ОПРЕДЕЛЕНИЯ

Термины, используемые в настоящем стандарте, и их определения — по ГОСТ 28906 со следующими дополнениями.

3.1. Значение — отличный от других элемент 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 тега (возможно, заданного по умолчанию) некоторого типа новым тегом.

3.8. Набор знаков АСН.1 — набор знаков, определенный в разд. 7, который используется в обозначениях нотации АСН.1.

3.9. Элементы — поименованные последовательности знаков из набора знаков АСН.1, определенные в разд. 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 присваиваются 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. Истина — одно из двух различных значений булевского типа.

3.15. Ложь — другое значение булевского типа.

3.16. Целочисленный тип — простой тип с различными значениями, являющимися положительными или отрицательными целыми числами, включая нуль (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 неощутимыми для любого из пользователей АСН.1. Тем не менее, конкретный пользователь АСН.1 может указать с помощью комментария, что используется более ограниченный диапазон чисел

3.17. Тип «строка знаков» — простой тип, различными значениями которого являются упорядоченные последовательности из пустого множества битов, одного или более битов.

Примечание. Правила кодирования не ограничивают число битов в строке битов. Тем не менее, конкретный пользователь может указать с помощью комментария, что используются строки, содержащие количество битов, находя-

щееся в диапазоне между некоторыми минимальным и максимальным значениями.

3.18. Тип «строка октетов» — простой тип, различные значения которого являются упорядоченными последовательностями из пустого множества октетов, одного или более октетов, где октет — это упорядоченная последовательность из восьми битов.

Примечание. Правила кодирования не ограничивают число октетов в строке октетов. Тем не менее, конкретный пользователь может указать с помощью комментария, что используются строки, содержащие количество октетов, находящееся в диапазоне между некоторыми минимальным и максимальным значениями.

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

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

3.20. 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.

3.21. 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». Тем не менее, конкретный пользователь АСН.1 может указать с помощью комментария минимальное и максимальное количество значений.

3.22. 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 необязательно содержит значение этого типа-компонента.

3.23. Тип «множество из» — структурированный тип, определяемый ссылкой на единственный существующий тип; каждое зна-

чение нового типа является неупорядоченным списком из нуля, одного или нескольких значений существующего типа.

Примечание. Правила кодирования не ограничивают число значений-компонентов в значении «множество из». Тем не менее, пользователь АСН 1 может указать с помощью комментария минимальное и максимальное количество значений, а также потребовать, чтобы все значения были различными

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

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

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

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

3.28. 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 необязательно) описаны с помощью АСН.1, и, соответственно, их кодовое представление может (но необязательно) соответствовать правилам кодирования для АСН.1.

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

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

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

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

3.32. Тип «описатель объекта» — тип, различные значения которого имеют вид текста, ориентированного на восприятие человеком и 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.

3.33. 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, чтобы используемые значения получаемых в результате типов имели конечное представление.

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

3.35. 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.

3.36. 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.

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

## 4. СОКРАЩЕНИЯ

АСН.1 — абстрактно-синтаксическая нотация версии 1.
UTC — всемирное согласованное время.
ICD — международный кодовый определитель.

## 5. СИСТЕМА ОБОЗНАЧЕНИЙ, ИСПОЛЬЗУЕМАЯ В НАСТОЯЩЕМ СТАНДАРТЕ

Нотация АСН.1 строится из последовательностей знаков знакового набора АСН.1, определенного в разд. 7.

Каждый конкретный случай использования нотации АСН.1 содержит знаки из знакового набора АСН.1, сгруппированные в эле-

менты. В разд. 8 определяются все последовательности знаков, образующие элементы нотации АСН.1, и каждому элементу присваивается имя.

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

Для того, чтобы описать эту совокупность последовательностей, в настоящем стандарте используется формальная система обозначений, определяемая в последующих подразделах.

5.1. Продукция

Продукция — это правило образования новой (более сложной) совокупности последовательностей.

С помощью продукции, используя имена совокупностей последовательностей, определенных в настоящем стандарте, строят новую совокупность последовательностей двумя способами:

а) образованием новой совокупности из последовательностей, содержащихся в любой из исходных совокупностей, либо

б) образованием 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 включает части, образующие одну или несколько строк, в следующем порядке:

а) имя новой совокупности последовательностей;

б) знаки:

: : = ;

в) одна или несколько совокупностей-вариантов последовательностей, определенных по правилам, описанным в подразделе 5.2 и разделенных знаком:

I.

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 перечисления а).

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

Каждая из совокупностей-вариантов, упомянутых в перечислении в) п. 5.1, задается 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 в 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-СтрокаБитов : : = b-строка I
h-строка I
{Список-Идентификаторов}

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

а) произвольным элементом «b-строка»;

б) произвольным элементом «h-строка»;

в) 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-СтрокаБитов».

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.

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—15 | Зарезервированы для дополнений к настоящему стандарту |
| 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 дополнений к настоящему стандарту |

5.6. Ссылки на совокупность последовательностей

В настоящем стандарте ссылки на совокупность последовательностей (часть нотации АСН.1) осуществляются указанием первого имени в продукции (перед знаками «: : =»); если ссылка на имя не является частью продукции, то имя обрамляется знаками «кавычки», с тем 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, то оно обрамляется знаками «кавычки», с тем 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 (univesal);

прикладной (application);

пользовательский (private);

контекстно-зависимый (context-spesific).

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

Ограничения на теги, присваиваемые пользователем АСН.1, определены в разд. 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 табл. 1 перечисляются теги универсального класса.

Примечание Для дополнений к настоящему стандарту резервируется дополнительное множество тегов.

## 6. ИСПОЛЬЗОВАНИЕ НОТАЦИИ АСН.1

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

6.2. Обозначения нотации АСН.1 для определения значения типа определяются продукцией «Значение» (п. 12.6).

Примечание. Невозможно интерпретировать значение, не зная типа, к которому оно относится

6.3. Обозначения нотации АСН.1 для присвоения типа ссылке на тип определяется продукцией «Присвоение Типа» (п. 11.1).

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

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

## 7. НАБОР ЗНАКОВ НОТАЦИИ АСН.1

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

Таблица 2

**Набор знаков АСН.1**

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

Примечания.

1. Дополнительно к знакам, перечисленным в табл 2, в макронотации используются знаки «|» и «>»

2. Настоящим стандартом допускается присутствие знаков, перечисленных в табл 3, в следующих элементах (пять последних элементов определяются в приложении 1)·

СсылкаНаТип (п 8 2 1);
Идентификатор (п 8 3);
СсылкаНаЗначение (п. 8 4);
СсылкаНаМодуль (п 8 5);

Макроссылка (п. 1.2.1);
ССылкаНаПродукцию (п 1.2.2);
ЛокальнаяССылкаНаТип (п. 1.2.3);
Локальная ссылка на Значение (п. 1.2.4);
а-строка (п. 1.2.7).

Таблица 3

**Дополнительный набор знаков АСН.1**

| |
|---|
| От А до Я<br>От а до я |

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 «кавычки» (п. 8.11).

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

7.4. Не следует придавать ни 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 1 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. Всякий 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 для избежания неоднозначности, которая могла бы быть вызвана наличием комментария, следующего за элементом

8.2.2. Элемент «ССЫЛКАНАТИП» не должен совпадать с одной из зарезервированных знаковых последовательностей, перечисленных в табл. 4.

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 для избежания неоднозначности, которая могла бы быть вызвана наличием комментария, следующего за элементом

Таблица 4

**Зарезервированные последовательности знаков**

| |
|---|
| BOOLEAN |
| INTEGER |
| BIT |
| STRING |
| OCTET |
| NULL |
| SEQUENCE |
| OF |
| SET |
| IMPLICIT |
| CHOICE |
| ANY |
| EXTERNAL |
| OBJECT |
| IDENTIFIER |
| OPTIONAL |
| DEFAULT |
| COMPONENTS |
| TRUE |
| FALSE |
| BEGIN |
| END |

Примечание. В приложении 1 (п. 1.2.9) определены другие последовательности знаков, зарезервированные при использовании в тексте макроопределений.

8.4. Ссылки на значение

Имя элемента — ССылкаНаЗначение.

Элемент «ССылкаНаЗначение» должен состоять из последовательности знаков, удовлетворяющей определению элемента «Идентификатор» в п. 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 п. 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 используются 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 п. 7.1 (п. 7.3).

8.7. Пустой элемент

Имя элемента — Пусто.

Элемент «Пусто» не содержит никаких знаков. Он используется 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 — c-строка.

Элемент «c-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 «c-строка» парой знаков (″). Набор знаков, о котором идет речь, совсем необязательно ограничивается знаками, перечисленными в табл. 2, он определяется типом рассматриваемого элемента «c-строка» (типом, для которого «СтрокаЗнаков» является значением) (см. п. 7.2).

Пример. ″ ″

8.12. Элемент «присвоение»

Имя элемента — «: : = »

Этот элемент состоит из последовательности знаков: : : =

Примечание Эта последовательность не содержит пробелов (см. п. 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

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 п. 8.2.2), могут быть отличны от других элементов, содержащих такие же знаки, с помощью контекста, в котором они присутствуют.

## 9. ОПРЕДЕЛЕНИЕ МОДУЛЯ

9.1. «ОпределениеМодуля» задается следующими продукциями:

ОпределениеМодуля : : = ССылкаНаМодуль

```
                                DEFINITIONS
                                «: : =»
                                BEGIN
                                ТелоМодуля
                                 END
ТелоМодуля : : = СписокПрисвоений I
                 Пусто
СписокПрисвоений : : = Присвоение I
                       СписокПрисвоений Присвоение
Присвоение : : = ПрисвоениеТипа I
                 ПрисвоениеЗначения
```

Примечания

1 В приложении 1 определяются последовательности вида «Макроопределение», которые также могут присутствовать в последовательности вида «Список Присвоений» 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 «Тело-Модуля» может использоваться самостоятельно, а не как часть последовательности «Определение Модуля»

3 Продукции «ПрисвоениеТипа» и «ПрисвоениеЗначения» определяются в разд 11.

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

9.2. «ССылкаНаМодуль», присутствующая в продукции «ОпределениеМодуля», называется именем модуля. Имена модулей выбираются 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» считается полным и согласованным, если каждой содержащейся в ней последовательности вида «ССылкаНаТип» или «ССылкаНа Значение» соответствует в точности одна последовательность вида «ПрисвоениеТипа» или «ПрисвоениеЗначения» соответственно, которые устанавливают связь между именем и соответственно типом или значением.

9.3. Имена модулей должны использоваться только один раз в сфере применения определения модуля, за исключением случаев, определенных в п. 9 4.

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

ISOxxxx=yyyy,

где xxxx — номер стандарта, а yyyy — аббревиатура для названия этого стандарта (например JTM, FTAM или CCR). Аналогичное соглашение может применяться в других стандартах.

9.4. Для обращения к элементам «ССылкаНаТип» и «ССылкаНаЗначение» в модулях, отличных от тех, где вышеупомянутые элементы были определены, используются 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 используются последовательности, определяемые следующими продукциями:

ОпределенныйТип : : = ВнешняяССылкаНаТип | ССылкаНаТип

ОпределенноеЗначение : : = ВнешняяССылкаНаЗначение | ССылкаНаЗначение

10.2. Варианты «ССылкаНаТип» и «ССылкаНаЗначение» в приведенных выше продукциях должны использоваться только в тех же модулях, в которых данной ссылке на тип или значение соответственно был присвоен некоторый тип или некоторое значение (пп. 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 (пп. 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 разд. 2, а также не должен совпадать с каким-либо именем, используемым для ссылки на типы, определенные в разд. 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 (п. 12.6), 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
(п. 10.1)

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

Примечания:

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

2. В дополнениях к настоящему стандарту могут быть определены другие встроенные типы.

12.2. Обозначение «ВстроенныйТип» определяется в последующих разделах.

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

12.4. 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 разд. 23.

Примечание. Обозначение вида «СелективныйТип» содержит «Идентификатор». Если в качестве «ПоименованногоТипа» используется «СелективныйТип» (п. 23.1), то этот идентификатор может быть частью обозначения для записи значения.

12.5. «Идентификатор» не является составной частью типа и никак не влияет на тип. Тип, на который дается ссылка последовательностью «ПоименованныйТип», является тем же самым, что и тип, на который дается ссылка последовательностью «Тип», вложенной в последовательность «ПоименованныйТип».

12.6. 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 |
Значение-МножествоИз |
ВыборочноеЗначение |
СелективноеЗначение |
ТегированноеЗначение |
ПроизвольноеЗначение |
Значение-ИдентификаторОбъекта |
Значение-СтрокаЗнаков

Примечание. В качестве последовательности вида «Значение» также может использоваться обозначение значения, определенное с помощью макронотации (приложение 1).

12.7. 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-СтрокаБитов | Значение-СтрокаБитов |
| 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-Идентификатор-Объекта |
| Тип-СтрокаЗнаков | Значение-СтрокаЗнаков |

Примечание. В дополнениях к настоящему стандарту могут быть определены дополнительные обозначения для записи значений.

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

12.8. Обозначения для записи значений типов, определяемых обозначением «ОбщеупотребительныйТип», приведены в разд. 30, 31, 32, 33.

12.9. Обозначение «ВстроенноеЗначение» определяется в последующих пунктах.

12.10. 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 п. 23.2 указаны дополнительные ограничения на «ПоименованноеЗначение» для случая, когда «ПоименованныйТип» является «СелективнымТипом».

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

12.11. «Идентификатор» является составной частью последовательности «ПоименованноеЗначение» только в 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 БУЛЕВСКОГО ТИПА

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

БулевскийТип : : = BOOLEAN

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

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

БулевскоеЗначение : : = TRUE | FALSE

## 14. ОБОЗНАЧЕНИЯ ЦЕЛОЧИСЛЕННОГО ТИПА

14.1. Для ссылки на целочисленный тип (п. 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» не должен использоваться, если «Число» является нулем.

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

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

14.5. Значения каждой из последовательностей «ЧислоСоЗна-

ком» и «ОпределенноеЗначение» входящих в «СписокПоименованныхЧисел», должны быть попарно различными и выделенными значениями целочисленного типа.

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

14.7. Порядок следования последовательностей «ПоименованноеЧисло» в последовательности «СписокПоименованныхЧисел» несуществен.

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

14.9. Значение целочисленного типа должно определяться обозначением вида «ЦелочисленноеЗначение»:

ЦелочисленноеЗначение : : = ЧислоСоЗнааком |
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 «СТРОКА БИТОВ»

15.1. Для ссылки на тип «строка битов» должно использоваться обозначение «Тип-СтрокаБитов»:

Тип-СтрокаБитов : : =
BIT STRING |
BIT STRING{СписокПоименованныхБитов}

СписокПоименованныхБитов : : =
ПоименованныйБит |
СписокПоименованныхБитов, ПоименованныйБит

ПоименованныйБит : : = Идентификатор(Число) |
Идентификатор(ОпределенноеЗначение)

15.2. «СписокПоименованныхБитов» не является существенной составной частью определения типа. Эта последовательность используется только в обозначениях для записи значений, определенных в п. 15.8.

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

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

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

15.5. Значения последовательностей «Число» и «ОпределенноеЗначение», входящих в состав последовательности «СписокПоименованныхБитов», должны быть попарно различными; каждое из этих значений используется в качестве номера выделенного бита при задании значения строки битов.

15.6. Все элементы «Идентификатор», входящие в «СписокПоименованныхБитов», должны быть различными.

Примечание. Порядок следования последовательностей «ПоименованныйБит» в последовательности «СписокПоименованныхБитов» несуществен.

15.7. Этот тип имеет тег универсального класса с номером 3.

15.8. Значение типа «строка битов» задается нижеследующим обозначением «Значение-СтрокаБитов»:

```
Значение-СтрокаБитов : : = b-строка                    |
                           h-строка                    |
                           {СписокИдентификаторов}     |
                           { }

СписокИдентификаторов : : =
                    Идентификатор   |
                    СписокИдентификаторов, Идентификатор
```

15.9. Каждый элемент «Идентификатор», входящий в последовательность «Значение-СтрокаБитов», должен совпадать с некоторым элементом «Идентификатор» в составе последовательности «Тип-СтрокаБитов», определяющей тип, к которому относится данное значение.

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

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

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

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

15 12. При описании правил кодирования для строк битов при ссылках на соответствующие биты используются термины «первый бит» и «замыкающий бит», определенные выше.

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

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

Примечание. Описание обозначения не ограничивают способа размещения строки битов по октетам для передачи, который реализуется правилами кодирования.

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

а) если значение строки битов состоит из количества битов, кратного четырем, или

б) присутствие или отсутствие последних нулевых битов несущественно.

Пример

'A98A'H и '1010100110001010'B являются эквивалентными обозначениями одного и того же значения битовой строки.

## 16. ОБОЗНАЧЕНИЯ ТИПА «СТРОКА ОКТЕТОВ»

16.1. *Для ссылки на тип «строка октетов» должно использоваться* обозначение «Тип-СтрокаОктетов»:

Тип-СтрокаОктетов : : = OCTET STRING

16.2. Этот тип имеет тег универсального класса с номером 4.

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

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

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

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

16.6. 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.

## 17. ОБОЗНАЧЕНИЯ ВЫРОЖДЕННОГО ТИПА

17.1. Для ссылки на вырожденный тип (п. 3.19) следует использовать обозначение «ВырожденныйТип»:

ВырожденныйТип : : = NULL

17.2. Этот тип имеет тег универсального класса с номером 5.

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

ВырожденноеЗначение : : = NULL

## 18. ОБОЗНАЧЕНИЯ ТИПОВ «ПОСЛЕДОВАТЕЛЬНОСТЬ»

18.1. Для построения типа «последовательность» (п. 3.20) из других типов используется 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
```

18.2. «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 считается логически завершенным до удовлетворения требований, описанных в последующих пунктах.

18.3. Варианты с использованием ключевых слов «OPTIONAL» и «DEFAULT» допускаются только в тех случаях, когда все типы в последовательности «СписокТипов-Компонентов» имеют разные теги (разд. 24).

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.

## 17. ОБОЗНАЧЕНИЯ ВЫРОЖДЕННОГО ТИПА

17 1. Для ссылки на вырожденный тип (п. 3.19) следует использовать обозначение «ВырожденныйТип»:

ВырожденныйТип : : = NULL

17 2 Этот тип имеет тег универсального класса с номером 5.

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

ВырожденноеЗначение : : = NULL

## 18. ОБОЗНАЧЕНИЯ ТИПОВ «ПОСЛЕДОВАТЕЛЬНОСТЬ»

18.1. Для построения типа «последовательность» (п. 3.20) из других типов используется 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
```

18.2 «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 считается логически завершенным до удовлетворения требований, описанных в последующих пунктах.

18.3 Варианты с использованием ключевых слов «OPTIONAL» и «DEFAULT» допускаются только в тех случаях, когда все типы в последовательности «СписокТипов-Компонентов» имеют разные теги (разд. 24).

18.4. Если используются 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 «ПоименованныйТип», 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.

18.5. 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».

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

18.7. Все типы «последовательность» имеют тег универсального класса с номером 16.

Примечание. Типы «последовательность из» имеют тот же самый тег.

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

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

СписокЗначений-Компонентов : : =
ПоименованноеЗначение |
СписокЗначений-Компонентов, ПоименованноеЗначение

18.9. Обозначение «{ }» используется лишь в следующих случаях:

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

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

18.10. Для каждой последовательности «ПоименованныйТип», входящей в «Тип-Последовательность», которая не отмечена как «DEFAULT» или «OPTIONAL», в записи значения типа «последо-

вательность» 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.

## 19. ОБОЗНАЧЕНИЯ ТИПОВ «ПОСЛЕДОВАТЕЛЬНОСТЬ ИЗ»

19.1. Обозначения для построения типа «последовательность из» (п. 3.21) из другого типа должны иметь вид «Тип-ПоследовательностьИз»:

Тип-ПоследовательностьИз : : = SEQUENCE OF Тип |
SEQUENCE

19.2. Обозначение «SEQUENCE» эквивалентно обозначению «SEQUENCE OF ANY» (разд. 25).

19.3. Все типы «последовательность из» имеют тег универсального класса с номером 16.

Примечание. Типы «последовательность» имеют тот же самый тег (п. 18.7).

19.4. Для задания значения типа «последовательность из» используется определяемое ниже обозначение «Значение-ПоследовательностьИз»:

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

СписокЗначений : : = Значение |
СписокЗначений, Значение

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

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

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

## 20. ОБОЗНАЧЕНИЯ ТИПА «МНОЖЕСТВО»

20.1. Обозначения для определения типа «множество» на основе других типов должны иметь вид «Тип-Множество»:

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

«СписокТипов-Компонентов» определен в п 18 1

20.2 «Тип» в четвертом варианте продукции «Тип-Компонент» (п. 18.1) должен быть типом «множество». Использование варианта «COMPONENTS OF Тип» определяет включение в «Список-Типов-Компонентов» всех последовательностей «Тип Компонент», имеющихся в типе «множество»

Примечание Это преобразование считается логически завершенным до удовлетворения требований, описанных в последующих пунктах

20 3. Все «Типы-компоненты» в типе «множество» должны иметь разные теги (разд. 24).

20 4 Для типов «множество» применимы формулировки пп 18 4, 18 5 и 18 6.

20.5. Все типы «множество» имеют тег универсального класса с номером 17.

Примечание. Типы «множество из» имеют тот же самый тег (п 21 3).

20 6. Порядку следования значений-компонентов в типе «множество» не придается никакого семантического значения

20 7. *Обозначения для определения значения типа «множество»* должны иметь вид «Значение-Множество»

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

Последовательность «СписокЗначений-Компонентов» определен в п. 18.8.

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

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

б) тип был задан последовательностью «SET{ }»

20.9. Для каждого «ПоименованногоТипа» в «Типе-Множестве», который не был задан как «OPTIONAL» 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.

## 21. ОБОЗНАЧЕНИЯ ТИПОВ «МНОЖЕСТВО ИЗ»

21.1. Обозначения для определения типа «множество из» (см. п. 3 23) на основе другого типа должны иметь вид «Тип-МножествоИз»

ТипМножествоИз = SET OF Тип |
SET

21 2 Обозначение «SET» эквивалентно обозначению «SET OF ANY» (разд 25)

21 3 Все типы «множество из» имеют тег универсального класса с номером 17

Примечание Типы «множество» имеют тот же самый тег (см п 20 5).

21 4 Обозначения для определения значения типа «множество из» должны иметь вид «Значение-МножествоИз»:

Значение МножествоИз = {СписокЗначений} | { }

Последовательность «СписокЗначений» определена в п. 19 4

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

21 5 Каждое «Значение» в «СпискеЗначений» должно быть обозначением некоторого значения «Типа», который используется в последовательности «Тип МножествоИз»

Примечания

1 Порядок следования значений-компонентов не имеет никакого семантического значения

2 К правилам кодирования не предъявляются требования сохранения порядка следования этих значений

## 22. ОБОЗНАЧЕНИЯ ВЫБОРОЧНЫХ ТИПОВ

22 1 Обозначения для определения выборочного типа (п. 3.25), исходя из имеющихся других типов, должны иметь вид «ВыборочныйТип»

ВыборочныйТип = CHOICE{Список-Типов-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»

22 2 Все типы, входящие в «СписокТипов-Вариантов», должны иметь различные теги (разд 24).

22.3 Тег выборочного типа следует считать переменной При выборе конкретного значения выборочного типа его тег становит-

ся равным тегу того типа-варианта, к которому относится значение.

22.4. Если выборочный тип используется в таких конструкциях, где требованием настоящего стандарта является использование типов с различными тегами (пп. 18.3, 20.3 и 22.2), то теги всех типов, определенных в «СпискеТипов-Вариантов» должны отличаться от тегов других типов (разд. 24). 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 и f, а также e и g совпадают.

Примеры.

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

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

22.6. Если выборочный тип используется в таких конструкциях, где требованием настоящего стандарта является использование «ПоименованныхТипов» с различными «Идентификаторами», то «Идентификаторы» если они заданы, всех «Поименованных-

Типов» в «СпискеТипов-Вариантов» должны отличаться от всех «Идентификаторов» других «ПоименованныхТипов» (входящих в конструкцию).

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

ВыборочноеЗначение : : = ПоименованноеЗначение

22.8. 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.

## 23. ОБОЗНАЧЕНИЯ СЕЛЕКТИВНЫХ ТИПОВ

23.1. 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» используется при записи значения, так и в качестве «Типа» в «ПоименованномТипе», и в этом случае его «Идентификатор» при записи значения не используется.

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

СелективноеЗначение : : = ПоименованноеЗначение

При этом в «ПоименованноеЗначение» входит идентификатор, присутствующий в соответствующей последовательности «СелективныйТип», только в том случае, когда «СелективныйТип» используется как «ПоименованныйТип».

## 24. ОБОЗНАЧЕНИЯ ТЕГИРОВАННЫХ ТИПОВ

Тегированный тип (п. 3.24) 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 используется главным образом в тех случаях, когда требованием настоящего стандарта

является использование типов с различными тегами (пп. 18.3, 20.3, 22.2, 22.4 и 25.2).

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.

24.1. Обозначения для тегированного типа должны иметь вид «ТегированныйТип»:

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

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

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

24.4. 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.

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

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

24.7. Если «Классом» тега является «пусто», то на использование «Тега» не накладывается никаких ограничений, кроме тех, которые вызваны требованиями различия тегов, сформулированными в пп. 18.3, 20.3 и 22.2.

24.8. 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 ИСО 8825 (ГОСТ )

24 9 Вариант «IMPLICIT» не должен использоваться, если тип, определенный последовательностью «Тип», является выборочным («CHOICE») или произвольным типом («ANY»)

24 10 Обозначения для определения значения тегированного типа должны иметь вид «ТегированноеЗначение»:

ТегированноеЗначение = Значение,

где «Значение» является обозначением значения того типа, который был задан последовательностью «Тип» в продукции «ТегированныйТип»

Примечание Отметим, что в этом обозначении отсутствует «Тег».

## 25 ОБОЗНАЧЕНИЯ ПРОИЗВОЛЬНОГО ТИПА

25 1 Обозначение для произвольного типа определяется продукцией «ПроизвольныйТип»

ПроизвольныйТип = ANY

Примечание «ANY» используется в тех случаях, когда требуется дополнительная стандартизация При использовании «ANY» до того, как станет возможным обмен информацией, требуется дополнительное описание типов данных и их семантики, соответствующее описание типа должно быть подставлено вместо «ANY» Использование произвольного типа («ANY») в стандартах ИСО или Рекомендациях МККТТ сопровождается, как правило, комментарием, указывающим источник дополнительных описаний

25 2 Произвольный тип имеет неопределенный тег, он не должен использоваться в тех случаях, где требованием настоящего стандарта является использование типов с различными тегами (пп 18 3, 20 3, 22.2 и 22 4).

25 3 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.

## 26. ОБОЗНАЧЕНИЯ ТИПА «ИДЕНТИФИКАТОР ОБЪЕКТА»

26 1. Ссылка на тип «идентификатор объекта» (п. 3.31) должна иметь вид «Тип-ИдентификаторОбъекта»:

Тип-ИдентификаторОбъекта : : = OBJECT IDENTIFIER

26.2. Этот тип имеет тег универсального класса с номером 6.

26.3. 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)

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

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

26.6. «ИменнаяФорма» используется только для тех компонентов идентификатора объекта, численные значения которых определены в приложениях 2—4 настоящего стандарта, и должна совпадать с одним из идентификаторов, определенных в этих приложениях.

26.7. «Число» в продукции «ЧисловаяФорма» должна быть числовым значением, присвоенным компоненту идентификатора объекта.

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

26.9. 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 однознач-

но 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 2—4.

26.10. 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 «идентификатор 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.

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

Примечание. Если в каком-либо 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».

## 27. ОБОЗНАЧЕНИЯ ТИПОВ «СТРОКА ЗНАКОВ»

27.1. Обозначения для задания ссылки на тип «строка знаков» (п. 3.12 и глав. 2) должны иметь вид:

Тип-СтрокаЗнаков : : = ССылкаНаТип,

где «ССылкаНаТип» — одно из имен типов «строка знаков», перечисленных в разд. 29.

27.2. Теги всех типов «строка знаков» определены в разд. 29.

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

Значение-СтрокаЗнаков : = с-строка

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

## 28. ОБОЗНАЧЕНИЯ ТИПОВ ОБЩАЯ ФОРМА ЗАПИСИ ВРЕМЕНИ, ВСЕМИРНОЕ ВРЕМЯ, ВНЕШНИЙ ТИП, ОПИСАТЕЛЬ ОБЪЕКТА

28.1. Обозначения для задания ссылки на один из типов, приведенных в разд. 30, 31, 32, 33, должны быть следующими

ОбщеупотребительныйТип : . = ССылкаНаТип,

где «ССылкаНаТип» — одна из ссылок на тип, определенных в главе 3 с помощью нотации АСН 1.

28.2. Теги для каждого из общеупотребительных типов определены в разд. 30, 31, 32, 33

28.3. Обозначения для записи значения общеупотребительного типа приведены в разд. 30, 31, 32, 33

## 29. ОПРЕДЕЛЕНИЕ ТИПОВ «СТРОКА ЗНАКОВ»

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

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

а) тег, присвоенный типу;

б) имя, с помощью которого осуществляется ссылка на определение типа,

в) набор знаков, используюшихся 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 перечисления б) может использоваться при «ССылка НаТип» при задании типа с помощью нотации АСН 1 (см. разд. 27).

29.2. В табл. 6 приведены имена, с помощью которых могут осуществляться ссылки на определения типов, номера типов универсального класса, присвоенных типам, и регистрационные номера или номера приведенных ниже таблиц (определяющие набор знаков) В случае необходимости строка таблицы сопровождается

примечанием. В тех 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 тех 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.

29.3. В табл. 5 перечислены знаки, которые могут присутствовать в значении типа, имеющего имя NumericString (числовая строка).

29.4. В табл. 6 перечислены знаки, которые могут присутствовать в значении типа, имеющего имя PrintableString (печатная строка).

29.5. Обозначения для всех значений типов «строка знаков» должны иметь вид «с-строка». Список типов «строка знаков» приведен в табл. 7.

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

29.6. Во всех случаях набор допустимых знаков может ограничиваться с помощью комментария, но не может быть расширен.

Таблица 5

**Тип NumericString (числовая строка)**

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

Таблица 6

**Тип PrintableString (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 | . |

*Продолжение табл. 6*

| Наименование | Изображение |
|---|---|
| Косая черта | / |
| Двоеточие | : |
| Знак равенства | = |
| Вопросительный знак | ? |

Таблица 7

**Список типов «строка знаков»**

| Имя для ссылки на тип | Номер тега универсального класса | Регистрационные номера или номер таблицы, определяющей набор знаков | Примечания |
|---|---|---|---|
| NumericString | 18 | Табл. 5 | (1) |
| PrintebleString | 19 | Табл. 6 | (1) |
| TeletexString (T61String) | 20 | 87, 102, 103, 106, 107 +ПРОБЕЛ+ЗАБОЙ | (3) |
| 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.

## 30. ОБЩАЯ ФОРМА ЗАПИСИ ВРЕМЕНИ

Примечание. Предполагается, что эта часть настоящего стандарта будет впоследствии расширена с тем, чтобы упорядочить использование других широко распространенных типов данных, таких как относящиеся к диагностике, идентификации отправителя, а также учетная информация, параметры защиты информации и т. д.

Обозначения для записи значений каждого из этих типов и их семантика могут быть выведены из соответствующих определений типа, приведенных в нотации АСН.1. Допускаются ссылки на эти определения в стандартах, определяющих правила кодирования, с тем, чтобы определить кодовые представления значений этих типов.

30.1. Для ссылок на этот тип должно использоваться имя GeneralizedTime.

30.2. Этот тип состоит из значений, содержащих:

а) календарную дату в форме, определенной ИСО 8601 (ГОСТ 24520);

б) время суток с любой точностью, предусмотренной в ИСО 8601 (ГОСТ 24520);

в) сдвиг местного времени относительно всемирного времени в виде, описанном в ИСО 8601 (ГОСТ 24520).

30.3. 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:

а) «отображаемая строка» («VisibleString») является строкой, представляющей календарную дату в соответствии с правилами, определенными в ИСО 8601 (ГОСТ 24520), с четырехзначной записью года, двузначной записью месяца и двузначной записью числа без разделителей, за которой следует строка, представляющая время суток, как предписано ИСО 8601 (ГОСТ 24520), не содержащая других разделителей, кроме десятичной точки или десятичной запятой, без оконечного знака Z (как предусмотрено в ИСО 8601 (ГОСТ 24520);

б) «отображаемая строка» является строкой знаков, определенной в перечислении а), за которой следует прописная буква Z;

в) «отображаемая строка» является строкой знаков, определенной в перечислении а), за которой следует строка, определяющая сдвиг местного времени относительно всемирного времени, соответствующая требованиям стандарта ИСО 8601 (ГОСТ 24520) без разделителей.

В случае перечисления а) запись времени обозначает местное время. В случае перечисления б) запись времени представляет всемирное время (UTC). В случае перечисления в) часть строки, описанная в перечислении а) представляет местное время (t1), а сдвиг местного времени относительно всемирного времени (t2)

позволяет вычислить всемирное время (UTC) по следующей формуле

время UTC равно t1—t2

Примеры:

Случай а). Значение

19851106210627.3

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

Случай б). Значение

19851106210627.3Z

задает всемирное время (UTC), в остальном заданное как в перечислении а).

Случай в). Значение

19851106210627.3—0500 задает местное время как в перечислении а) с указанием сдвига местного времени относительно всемирного времени (местное время отстает на 5 ч).

30.4. Тег этого типа должен соответствовать определению типа в п. 30.3.

30.5. Обозначения для записи значения должны быть обозначениями для записи значения типа «отображаемая строка», определенного в п. 30.3.

## 31. ВСЕМИРНОЕ ВРЕМЯ

31.1. Для ссылки на этот тип должно использоваться имя UTCTime.

31.2. Этот тип состоит из значений, содержащих:

а) календарную дату;

б) время с точностью до одной минуты или одной секунды;

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

31.3. 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 частей, следующих друг за другом:

а) 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);

б) одна из двух возможностей:

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 перечислении 1), а ss — обозначение секунд;

в) одна из двух возможностей:

1) знака Z,

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

Разные варианты в перечислении б) обеспечивают разную точность при задании времени.

В варианте в)1) задается всемирное время. В варианте в)2) задается местное время (t1) (часть обозначения, определяемая в перечислении а) и б) и сдвиг местного времени относительно всемирного времени (t2) (часть обозначения, определяемая в перечислении в)2), при этом всемирное время (UTC) может быть вычислено следующим образом:

время UTC равно t1—t2.

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

UTCTime «8201021200Z» или

UTCTime «8201020700—0500».

31.4. Тег этого типа должен соответствовать определению типа в п. 31.3.

31.5. Обозначения для записи значения этого типа должны быть отображениями для записи типа «отображаемая строка», определенного в п. 31.3.

## 32. ВНЕШНИЙ ТИП

32.1. Обозначение внешнего типа (см. п. 3.28) должно иметь вид «ВнешнийТип»:

ВнешнийТип : : = EXTERNAL

32.2. Этот тип состоит из значений, включающих следующие составные части:

а) кодовое представление какого-то одного значения данных, которое может быть (но необязательно должно быть) значением какого-то одного определенного в АСН 1 типа данных;

б) идентифицирующую информацию, устанавливающую семантику значения и правила кодирования;

в) (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.

32.3. 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 2—4.

2. Множество значений данных, зарегистрированных в качестве абстрактного синтаксиса (вместе с соответствующими правилами кодирования), считается 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 используется нотация АСН. 1, то для обеспечения самоидентифицируемости кодового представления используется присвоение тегов. В тех случаях, когда абстрактный синтаксис не является правильно сформированным, использование коммуникационного канала либо контекстно зависимо, либо ведет к неоднозначности.

32.4. Внешний тип (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-ASNI-type [0] ANY,
octet-aligned [1] IMPLICIT OCTET STRING,
arbitrary [2] IMPLICIT BIT STRING}}
```

32.5. Когда для данного значения внешнего типа («EXTERNAL») не используется диалог по установлению правил кодирования на уровне представления, а используется предварительная 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 «идентификатора 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.

32.6. Когда используется 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 абстракт-

ного 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 пп 32 5 и 32 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

32 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

32 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 п. 32.7.

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

arbitrary

32.10. 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»)

32.11. 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 в результате диалога по установлению параметров, и полученные в результате октеты должны составлять значение — строку актетов.

32.12. 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 в результате диалога по установлению правил кодирования, а результат должен быть значением — строкой битов.

32.13. Тег этого типа должен соответствовать определению типа в п. 32.4.

32.14. Обозначения для записи значения должны быть обозначениями для записи значения типа, определенного в п. 32.4.

## 33. ТИП «ОПИСАТЕЛЬ ОБЪЕКТА»

33.1. Ссылка на этот тип осуществляется с помощью имени ObjectDescriptor.

33.2. Этот тип составляют тексты, ориентированные на восприя-

тие человеком и предназначенные для описания информационных объектов. Текст, являющийся значением данного типа, необязательно 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»)

33.3. 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.

33.4. Тег этого типа должен соответствовать определению типа в п. 33.3.

33.5. Обозначения для записи значения должны быть обозначениями для записи значения типа, определенного в п. 33.3.

*ПРИЛОЖЕНИЕ 1*
*Обязательное*

## МАКРОНОТАЦИЯ

1 1 Введение

В АСН 1 предусмотрен способ, с помощью которого пользователь АСН 1 может определить новую систему обозначений, с помощью которой он может строить те же типы данных, что и в АСН 1, и 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 (п 6 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

а) 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),

б) 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),

в) 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 присвоены на время выполнения синтаксического анализа локальным ссылкам на тип или локальным ссылкам на значение с помощью соответствующих конструкций в макроопределении Возможно также включение 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 считается 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 необязательно 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, отмеченное в макроопределении ключевым словом VALUE

12 Дополнение к набору знаков и множеству элементов АСН I

В макронотации используются знаки « » и «>»

Также используются элементы, описанные в последующих пунктах

121 *Макроссылка*

Имя элемента — макроссылка

«Макроссылка» должна состоять из последовательности знаков, удовлетворяющей требованиям, установленным для «Ссылки — НаТип» в п 82, с тем отличием, что все буквы должны быть прописными Внутри одного модуля одна и та же последовательность знаков не должна использоваться одновременно, как ссылка на тип и как макроссылка

122 *Ссылка на продукцию*

Имя элемента — ССЫЛКАНАПРОДУКЦИЮ

«ССЫЛКАНАПРОДУКЦИЮ» должна состоять из последовательности знаков, удовлетворяющих требованиям, установленным для «ССЫЛКИНАТИП» в п 82

123 *Локальная ссылка на тип*

Имя элемента — ЛОКАЛЬНАЯССЫЛКАНАТИП

«ЛОКАЛЬНАЯССЫЛКАНАТИП» должна представлять собой последовательность знаков вида «ССЫЛКАНАТИП», определенного в п 82 «ЛОКАЛЬНАЯССЫЛКАНАТИП» используется в качестве идентификатора для типов, распознаваемых во время синтаксического анализа макрообозначения типа или значения

124 *Локальная ссылка на значение*

Имя элемента — ЛОКАЛЬНАЯССЫЛКАНАЗНАЧЕНИЕ

«ЛОКАЛЬНАЯССЫЛКАНАЗНАЧЕНИЕ» должна представлять собой последовательность знаков вида «ССЫЛКАНАТИП», определенного в п 82 «ЛОКАЛЬНАЯССЫЛКА НАТИП» используется в качестве идентификатора для значений, распознаваемых во время синтаксического анализа макрообозначения типа или значения

Примечание «ЛОКАЛЬНАЯССЫЛКАНАТИП» должна начинаться с прописной буквы

125 *Элемент разграничения вариантов*

Имя элемента — I

Этот элемент состоит из единственного знака «I»

126 *Элемент, обозначающий конец определения*

Имя элемента — >

Этот элемент состоит из единственного знака «>»

Примечание Элемент «<», обозначающий начало определений, приведен в п 813

127 *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 I (см разд 7), выделенных 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 (»)

1.2.8. 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 табл. 8 посредством ссылки на раздел настоящего стандарта.

Примечание. В макронотации не поддерживается различие между идентификаторами и ссылками, отличающимися только тем, является ли их первая буква строчной или заглавной. Это вызвано требованием совместимости с рекомендацией МККТТ X.409.

Таблица 8

**Классы последовательностей, задаваемые элементами**

| Имя элемента | Раздел, содержащий определение |
|---|---|
| «string» | Произвольная последовательность знаков |
| «identifier» | 8.3 — Идентификаторы |
| «number» | 8.8 — Число |
| «empty» | 8.7 — Пустой элемент |

1.2.9 *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 пп. 1.2.2—1.2.4 приложения 1 включительно, не должны совпадать с какой-либо последовательностью знаков из п. 1.2.9 приложения 1, 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)

1.3 Обозначения для записи макроопределений

1.3.1 Макросредство должно быть определено с использованием обозначений вида «Макроопределение»

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

1.3.2 Каждая «ССылкаНаПродукцию», присутствующая в «ОпределенииСимвола» (п.1.3.6 приложения 1), должна присутствовать один раз в качестве первого элемента некоторой «Продукции»

1.3.3 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.

1.3.4 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.

1.3.5. 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».

1.3.6 Обозначения для записи последовательности «ВариантыМакро» в макроопределении определяются следующим образом

ВариантМакро = СписокСимволов
СписокСимволов =
Символьный Элемент I
СписокСимволов Символьный Элемент
Символьный Элемент =
ОпределениеСимвола I
ВложенноеОпределение
МакроТип = ЛокальнаяСсылкаНаТип I
Тип
ОпределениеСимвола =
—а строка» I
ССылкаНаПродукцию I
«string» I
«identifier» I
«number» I
«empty» I
type I
type (ЛокальнаяССылкаНаТип) I
value (МакроТип) I
value (ЛокальнаяССылкаНаЗначение МакроТип) I
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 «ВариантМакро», 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

1 3 7 Элемент («а строка») описывает последовательности знаков, составляющие «а строку», без обрамляющей пары кавычек (»)

1 3 8 Элемент «ССылкаНаПродукцию» описывает произвольную последовательность знаков, задаваемых «Продукцией», идентифицируемой данной ссылкой

1 3 9 Последовательности знаков, описываемые следующими четырьмя вариантами продукции «ОпределениеСимвола», задаются табл 8

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

1 3 10 Вариант «type» описывает последовательности знаков, образующие обозначение вида «Тип», соответствующее определению в п 12 1

Примечание Выражение «ОпределенныйТип» в определении из п 12 1 может в данном случае содержать «ЛокальнуюССылкуНаТип», служащую для обозначения типа, определенного с помощью макроопределения

1 3 11 Вариант «type (ЛокальнаяССылкаНаТип)» описывает последовательности знаков, образующие выражение «Тип», соответствующее определению из п 12 1, и, кроме того, присваивает этот тип «ЛокальнойССылкеНаТип»

1 3 12 Вариант «value (МакроТип)» описывает последовательности знаков,

образующие выражение «Значение», соответствующее определению из п 12 6, *того типа, который задан выражением «Макротип»*

1 3 13 Выражение «value (ЛокальнаяССылкаНаЗначение МакроТип)» описывает последовательности знаков, образующих выражение «Значение», соответствующее определению из п 12 6, того типа, который задан выражением «МакроТип», и, кроме того, присваивает этот тип «ЛокальнойССылкеНаТип» Впоследствии той же «ЛокальнойССылкеНаТип» может быть присвоен другой тип

*1 3 14 Вариант* «value (VALUE МакроТип)» описывает последовательности знаков, образующих выражение «Значение» (соответствующее определению из п 12 6), 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»

1 3 15 При синтаксическом анализе любого правильного использования макрообозначений должно осуществиться в точности одно присвоение значения выражению «VALUE» (описанному в пп 1 3 14 или 1 3 16 приложения 1).

1 3 16 Выражение «ВложенныеОпределения» определяется следующими продукциями

ВложенноеОпределение =
<СписокВложенныхОпределений>

*СписокВложенныхОпределений* =
ВложенноеОпределение
СписокВложенныхОпределений I
*ВложенноеОпределение*

ВложенноеОпределение =
ЛокальноеПрисвоениеТипа I
ЛокальноеПрисвоениеЗначения

ЛокальноеПрисвоениеТипа =
ЛокальнаяССылкаНаТип
« =»
МакроТип

ЛокальноеПрисвоениеЗначения =
ЛокальнаяССылкаНаЗначение
МакроТип
« =»
МакроЗначение

МакроЗначение =
Значение I
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 «ВложенныминОпределениями», 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 п. 1.3.14 приложения 1.

1.4. Использование новой нотации

Использование обозначения типа или значения, определенного с помощью макроопределения в том же модуле, допускается во всех случаях, где требуется обозначение вида «Тип» или «Значение».

Примечаний:

1. Описываемое в этом приложении расширение нотации АСН 1 не используется 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 2*
*Обязательное*

## ЗНАЧЕНИЯ КОМПОНЕНТОВ OBJECT IDENTIFIER, ОПРЕДЕЛЯЕМЫЕ ИСО

2.1 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 1 |

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

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

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

| Значение | Идентификатор | Служба, отвечающая за присвоение следующих компонентов |
|---|---|---|
| 0 | standard | см. п. 11.2.4 |
| 1 | registration authority | см. п. 11.2.5 |
| 2 | member-body | см. п. 11.2.6 |
| 3 | identified-organization | см. п. 11.2.7 |

Эти идентификаторы могут использоваться в качестве выражения «ИменнаяФорма».

2.4 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 этом 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 производиться так, как если бы стандарт состоял из одной части.

2.5 Ребра, выходящие из вершины «регистрационная служба» («registration authority»), резервируются для последующего определения в дополнении к настоящему стандарту, которое будет разрабатываться параллельно с определением структур для идентификации различных регистрационных служб взаимодействующих открытых систем.

2.6 Ребрам, выходящим из вершины, в которую ведет ребро «комитет — член» («member-body»), должны быть присвоены значения трехзначного кода страны, 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.

Примечания:

1. Наличие кода страны необязательно означает, что действительно существует комитет — член ИСО, представляющий эту страну, или что в сферу деятельности этого комитета входит схема выделения компонентов идентификаторов объектов.

2. СССР имеет трехзначный код 810, Белорусская ССР — 112 и Украинская ССР — 804. Организация, ведающая присвоением компонентов идентификаторов объектов в рамках СССР, пока еще не сформирована.

2.7. 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, ICD идентифицирует организацию — источник кодов, зарегистрированную вышеупомянутой регистрационной службой в качестве источника компонентов идентификаторов объектов (примечания 1 и 2). Ребрам, непосредственно следующим за ICD, должны быть присвоены значения вида «код организации», которые выделяются организацией — источником кодов. 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.

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 п. 2.7 приложения 2 является то, что любая организация может получить код от соответствующей организации, присваивающей коды, и после этого присваивать значения вида «идентификатор объекта» (OBJECT IDENTIFIER) 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 3*
*Обязательное*

## ЗНАЧЕНИЯ КОМПОНЕНТОВ OBJECT IDENTIFIER, ОПРЕДЕЛЯЕМОЕ МККТТ

3 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 4* |

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

3 2 Каждый из идентификаторов «ccitt» «iso» и «joint iso ccitt», определенных выше может использоваться в качестве выражения «ИменнаяФорма»

3 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 п 3 4 |
| 1 | question | см п 3 5 |
| 2 | administration | см п 3 6 |
| 3 | network-operator | см п 3 7 |

Эти идентификаторы могут использоваться в качестве выражения «ИменнаяФорма»

3 4 Ребра следующие за ребром «рекомендация» («recommendation»), имеют значения от 1 до 26, которым присвоены идентификаторы от a до z После дующие 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»

3 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, соответствующие 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.

3.6. Ребра, следующие за ребром «администрация» («administration»), имеют значения цифровых кодов стран (DCC), определенных рекомендацией МККТТ X.121. Последующие ребра определяются в соответствии с потребностями управления телекоммуникациями страны, определенной кодом DCC.

3.7. 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 управления телекоммуникациями или зарегистрированного частного агенства (RPOA), идентифицируемого кодом DNIC.

*ПРИЛОЖЕНИЕ 4*
*Обязательное*

## ЗНАЧЕНИЯ КОМПОНЕНТОВ OBJECT IDENTIFIER, ОПРЕДЕЛЯЕМЫЕ СОВМЕСТНО

4 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 пп 4 2, 4 3 |

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

4 2 Каждый из идентификаторов ccitt», «iso» и «joint-iso ccitt», определенных выше, может использоваться в качестве выражения СимвольнаяФорма»

4 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».

Примечание. Регистрационной службой, ответственной за присвоение значений компонентам идентификаторов объектов для совместного использования ИСО и МККТТ, является Американский национальный институт стандартов (ANSI).

4 4 Ребра, которые следуют за упомянутыми в п 4 3, приложения 4 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

4 5 Поскольку в первых стандартах ИСО и рекомендациях МККТТ, относящихся к совместным областям стандартизации, потребовалось выделение идентификаторов объектов (OBJECT IDENTIFIER) до установления процедур, упомянутых в п 4 3, приложения 4, то они выделяются так, как описано в приложениях 2 или 3 Идентификаторы информационных объектов, определенные таким способом в стандарте ИСО или рекомендации МККТТ, не будут меняться после установления процедур, упомянутых в п. 4 3 приложения 4.

*ПРИЛОЖЕНИЕ 5*
*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.

5.1. Пример записи учета кадров

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

5.1.1. *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 |

5.1.2. *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 эле-

ментам «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»)

5 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»
```

5 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 разд 3, здесь не рассматриваются.

5 2 1. *Булев тип*

5 2 1 1. Булев тип следует использовать для представления значений логических переменных (имеющих два разных состояния), например ответа на вопрос «да» или «нет».

Пример

Employed : : = BOOLEAN

5 2 1 2 При назначении ссылочного имени булеву типу следует использовать слово, подходящее для обозначения состояния истина»

Пример

Married : : =BOOLEAN

а не

MaritalStatus : : =BOOLEAN

См также п 5 2.2.4.

5 2 2 *Целочисленный тип*

5 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
```

5 2.2 2 Следует определять допустимые максимальное и минимальное значения целочисленного типа в качестве выделенных значений.

Пример

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

5.2.2.3. Целочисленный 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 : : = INTEGER {sunday(0), monday (1),
                   tuesday(2), wednesday(3), thursday(4),
                   friday(5), saturday(6)}
```

5.2.2.4. Целочисленный тип с выделенными значениями 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 : : = INTEGER {single(0), married (1)},
```

если предполагается его последующая замена на следующее:

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

5.2.3. *Тип «строка битов»*

5.2.3.1. 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.
```

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

Пример.

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

5.2.3.3. 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 был солнечный в том и только
- - в том случае, если i-й бит равен единице
```

5.2.3.4. 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)}

5.2.4. *Тип «строка октетов»*

5.2.4.1. 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.

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

Пример.

Surname : : = PrintableString

5.2.4.3. 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

5.2.5. *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».

5.2.6. *Типы «последовательность» и «последовательность из»*

5.2.6.1. Тип «последовательность из» следует использовать для

представления совокупности переменных одного и того же типа, количество которых велико или непредсказуемо, а порядок следования существенен.

Пример.

```
NamesOfMemberNations : : = SEQUENCE OF VisibleString
- - перечисление в порядке вступления
```

5.2.6.2. 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}
```

5.2.6.3. 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}
```

5.2.6.4. 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 пп. 5.2.3.3, 5.2.3.4 и 5.2.5 Приложения 5.

5.2.7. *Тип «множество»*

5.2.7.1. 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,
    counteryName        [2] IMPLICIT VisibleString}
```

5.2.7.2. 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 контекстнозависимого 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 - -}
```

5.2.7.3. 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 контекстнозависимого 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 - -                          }
```

5.2.7.4. 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,
    . . .                                              }
```

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

Пример.

```
Keywords :  : = SET OF VisibleStringh
- - в произвольном порядке
```

См. также пп. 5.2.3.4 и 5.2.8.3 приложения 5.

5.2.8. *Тегированный тип*

5.2.8.1. В рамках настоящего стандарта тегированный тип универсального класса используется 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
```

5.2.8.2. 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
    {directorName                VisibleString,
    DirectoryRelativeFileName    VisibleString}
```

5.2.8.3. Контекстнозависимые тегированные типы используются 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,
                             16
   mailingAddress     [1] IMPLICIT VisibleString,
    accountNumber     [2] IMPLICIT  VisibleString,
    balanceDue        [3] IMPLICIT VisibleString
                       - - в центах - -}
```

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

Пример.

```
ProductRecord : : = SET
    {                  UniformCode,
    description        [0] IMPLICIT INTEGER,
```

```
    inventoryNo          [1] IMPLICIT  INTEGER,
    inventoryLevel       [2]  IMPLICIT INTEGER}
  UniformCode : : = [APPLICATION 13] IMPLICIT INTEGER
```

5.2.8.5. 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
```

5.2.8.6. В тех 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 тех 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
```

5.2.8.7. В тех 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 тех случаев, когда ставится цель обеспечить выбор из нескольких (более одного) разных универсальных типов.

Пример.

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

5.2.8.8. 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
```

5.2.8.9. В приводящихся здесь рекомендациях во всех примерах, где это допустимо, используется «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 пп 5 2.7.1, 5 2 7 2, 5 2 9 1 и 5 2 9 2 приложения 5.

5.2.9. *Выборочный тип*

5 2.9 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 - - }
```

5.2.9.2 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 - -                }
```

5.2.9.3. 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- - }

5.2.9.4. В тех случаях, когда в некотором конкретном применении настоящего стандарта нормой является неявное задание тегов (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,
                   Voise          }
```

5.2.10. *Селективный тип*

5.2.10.1. Селективный тип следует использовать для представления переменной, тип которой является типом некоторого варианта в определенном ранее выборочном типе.

5.2.10.2. 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» } .
```

5.2.11. *Произвольный тип*

5.2.11.1. 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 ACH.1 за рамками настоящего стандарта

5.2.12. *Внешний тип*

5.2.12.1. 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

5.3. 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 ACH.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)
«=»
        type (Local-type-2)
```

```
- - Здесь ожидается описание второго типа
- - АСН.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 АСН.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.

5.4. Использование нотации для определения абстрактных синтаксисов

5.4.1. Для использования услуг уровня представления требуется описание значений, называемых значениями данных уровня представления, и группирование этих значений в множества, назы-

ваемые абстрактными синтаксисами. Каждому из этих множеств присваивается имя абстрактного синтаксиса, относящееся к определенному в АСН.1 типу «идентификатор объекта».

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

5.4.3. 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.

5.4.4. 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 значение:

« . . . . . »

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

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

и

«Basic Encoding of a single ASN.1 type»

(присваиваемые информационному объекту в (ГОСТ 34.974).

Конец примера

5.4.5. Стандарт может также потребовать для этого абстрактного синтаксиса обязательной поддержки синтаксиса передачи, получаемого применением правил кодирования.

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

*ПРИЛОЖЕНИЕ 6*
*Справочное*

## СВОДКА СИНТАКСИСА НОТАЦИИ АСН.1

Перечисленные ниже элементы определяются в разд 8

| | | |
|---|---|---|
| СсылкаНаТип | ( | ANY |
| Идентификатор | ) | EXTERNAL |
| СсылкаНаЗначение | [ | OBJECT |
| СсылкаНаМодуль | ] | IDENTIFIER |
| Комментарий | - | OPTIONAL |
| Пусто | BOOLEAN | DEFAULT |
| Число | INTEGER | COMPONENTS |
| b-строка | BIT | UNIVERSAL |
| h-строка | STRING | APPLICATION |
| c-строка | OCTET | PRIVATE |
| «: :=» | NULL | TRUE |
| { | SEQUENCE | FALSE |
| } | OF | BEGIN |
| < | SET | END |
| , | IMPLICIT | DEFINITIONS |
| . | CHOICE | |

Ниже приведены продукции, используемые в настоящем стандарте, терминальными символами продукций являются приведенные выше элементы:

ОпределениеМодуля : . = СсылкаНаМодуль
DEFINITIONS
« =»
BEGIN
ТелоМодуля
END

ТелоМодуля : : =СписокПрисвоений

СписокПрисвоений = Присвоение |
СписокПрисвоений Присвоение

Присвоение = ПрисвоениеТипа | ПрисвоениеЗначения

ВнешняяСсылкаНаТип · . = СсылкаНаМодуль
СсылкаНаТип

ВнешняяСсылкаНаЗначение · = СсылкаНаМодуль
СсылкаНаЗначение

ОпределенныйТип · = ВнешняяСсылкаНаТип I
СсылкаНаТип

*ОпределенноеЗначение* = ВнешняяСсылкаНаЗначение |
СсылкаНаЗначение

ПрисвоениеТипа = СсылкаНаТип
‹ =»
Тип

ПрисвоениеЗначения =СсылкаНаЗначение
Тип
«· =»
Значение

Тип = ВстроеныйТип | ОпределенныйТип

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

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

ПоименованноеЗначение ::= Идентификатор Значение | Значение

БулевскийТип ::= BOOLEAN

БулевскоеЗначение ::= TRUE : FALSE

ЦелочисленныйТип ::= INTEGER | INTEGER{СписокПоименованных Чисел}

СписокПоименованныхЧисел ::= ПоименованноеЧисло | СписокПоименованныхЧисел, ПоименованноеЧисло

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

ЧислоСоЗнаком ::= Число | -Число

ЦелочисленноеЗначение ::= ЧислоСоЗнаком | Идентификатор

Тип-СтрокаБитов ::= BIT STRING | BIT STRING{СписокПоименованных Битов}

СписокПоименованныхБитов ::= ПоименованныйБит | СписокПоименованныхБитов, ПоименованныйБит

ПоименованныйБит ::= Идентификатор (Число) | Идентификатор (ОпределенноеЗначение)

Значение СтрокаБитов ::= '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 ::= ПоименованноеЗначение | 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 ::= [Класс НомерВКлассе]

НомерВКлассе ::= Число | ОпределенноеЗначение

Класс ::= UNIVERSAL | APPLICATION | PRIVATE | Пусто

ТегированноеЗначение ::= Значение

ПроизвольныйТип ::= ANY

ПроизвольноеЗначение : : =Тип Значение
Тип-ИдентификаторОбъекта : : =OBJECT IDENTIFIER
Значение-ИдОбъекта : : ={СписокКомпонентовИдОбъекта} I
{ОпределенноеЗначение СписокКомпонентовИдОбъекта}
СписокКомпонентовИдОбъекта : : =КомпонентИдОбъекта I
КомпонентИдОбъектаСписокКомпонентовИдОбъекта
КомпонентИдОбъекта : : =Именнаяформа I
Числоваяформа I
Смешаннаяформа
Именнаяформа : : =Идентификатор
Числоваяформа : :=Число I ОпределенноеЗначение
Смешаннаяформа : : =Идентификатор (Числовая
форма)
Тип-СтрокаЗнаков : : =ССылкаНаТип
Значение-СтрокаЗнаков : : =с-строка
ОбщеупотребительныйТип : : =ССылкаНаТип

Перечисленные ниже типы «строка знаков» определены в разд. 29

| | |
|---|---|
| **NumericString** | **VisibleString** |
| PrintableString | **ISO646String** |
| TeletexString | IA5String |
| T61String | GraphicString |
| VideotexString | GeneralString |

Общеупотребительные типы имена которых перечислены ниже, определены в разд. 30, 31, 32, 33.

| | |
|---|---|
| GeneralizedTime | EXTERNAL |
| UTCTime | ObjectDescriptor |

Перечисленные ниже дополнительные элементы определены в п. 1.2 примечания 1 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 П1; терминальными символами этих продукций являются приведенные выше элементы, а также элементы, перечисленные в начале настоящего приложения

Макроопределение : : =макроссылка
MACRO
«: : =»
BEGIN
ТелоМакро
END
ТелоМакро : : = ПродукцияТипа
ПродукцияЗначения
ВспомогательныеПродукции
ПродукцияТипа : : =
TYPE NOTATION
«: : =»
СписокВариантовМакро
ПродукцияЗначения : : =
VALUE NOTATION

«: : =»
СписокВариантовМакро
ВспомогательныеПродукции : : =
СписокПродукций I
empty
СписокПродукций : : =
Продукция I
СписокПродукций Продукция
Продукция : : = ССылкаНаПродукцию
«: : =»
СписокВариантовМакро
СписокВариантовМакро : : =
ВариантМакро I
СписокВариантовМакро
«I»
ВариантМакро
ВариантМакро : : = СписокСимволов
СписокСимволов : : =
СимвольныйЭлемент I
СписокСимволов СимвольныйЭлемент
Символьный Элемент : : =
ОпределениеСимвола I
ВложенноеОпределение
МакроТип : : = ЛокальнаяССылкаНаТип I
Тип
ОпределениеСимвола : : =
«а-строка» I
ССылкаНаПродукцию I
«string» |
«identifier» |
«number» |
«empty» |
type |
type (ЛокальнаяССылкаНаТип) I
value (МакроТип) I
value (ЛокальнаяССылкаНаЗначение МакроТип) I
value (VALUE МакроТип)
ВложенноеОпределение : : =
<СписокВложенныхОпределений>
СписокВложенныхОпределений : : =
ВложенноеОпределение I
СписокВложенныхОпределений
ВложенноеОпределение
ВложенноеОпределение : : = ЛокальноеПрисвоениеТипа I
ЛокальноеПрисвоениеЗначения
ЛокальноеПрисвоениеТипа : : = ЛокальнаяССылкаНаТип
«: : =»
МакроТип
ЛокальноеПрисвоениеЗначения : : =
ЛокальнаяССылкаНаЗначение
МакроТип
«: : =»
МакроЗначение
МакроЗначение : : = Значение I
ЛокальнаяССылкаНаЗначение

*ПРИЛОЖЕНИЕ 7*
*Справочное*

## АЛФАВИТНЫЙ УКАЗАТЕЛЬ


| Термин | Раздел, пункт |
|---|---|
| булев тип (BOOLEAN) | 3 14, 8 2, 8 14, 13, 5 2 1 |
| внешняя ссылка на тип | 9.4, 10 |
| время | 30, 31 |
| время UTC (UTCTime) | 31 |
| всемирное время | 31 |
| встроенный тип | 12 1 |
| встроенное значение | 12 6 |
| выборочный тип (CHOICE) | 8 14, 22, П5 2 9 |
| *вырожденный тип (NULL)* | 8 14, 17, П5 2.5 |
| дерево идентификаторов объектов | 29.6, П2, П3 |
| знаковый набор ACH 1 | 3 8, 7 |
| значение | 3 1 |
| идентификатор | 8 3 |
| идентификатор объекта | 8 2, 8 14, 26 |
| истина (TRUE) | 3 14, 8 14 |
| комментарий | 7 3, 8 6 |
| компоненты типа (COMPONENTS OF) | 8 2, 8 14, 18 |
| ложь (FALSE) | 3 15, 8 2, 8 14 |
| макро | П1 1—П1 4, П5 3 |
| множество (SET) | 8,14, 20, П5 2 7 |
| множество из (SET OF) | 8 14, 21 |
| модуль | 9 |
| *необязательная возможность* (OPTIONAL) | 8 14, 18, 20 |
| неявное задание тега | 24 |
| общая форма записи времени (GeneralizedTime) | 30 |
| общеупотребительные типы | 28, 30, 31, 32, 33 |
| описатель объекта (OBJECT DISCRIPTOR) | 33 |
| отображаемая строка (VisibleString) | 29 |
| печатная строка (PrintableString) | 29, 29 4 |
| пользовательский (PRIVATE) | 5 8, 24 |
| последовательность (SEQUENCE) | 8 2, 8 14, 18, 5 2 6 |
| последовательность из (SEQUENCE OF) ... | 8 2, 8 14, 19, 5 2 6 |
| прикладной класс (APPLICATION) | 5 8, 24 |
| примеры | 5 1—5 3 |
| присвоение значения | 8 12 |
| присвоение типа | 11 1 |
| продукции | 5 1 |
| произвольный тип (ANY) | 8 2, 8 14, 19 2, 25, 5 2 11 |
| простой тип | 3 3 |
| резервируемая последовательность знаков | 8 2 2 |
| сводка синтаксиса | 6 |
| селективный тип | 26, 12 4, 23 |
| ссылка на значение | 8 4 |
| ссылка на модуль | 8 5 |
| ссылка на тип | 8 2 |
| строка битов (BIT STRING) | 8 2, 8 14, 15, 5 2 3 |
| строка знаков | 3 17, 27, 29 |

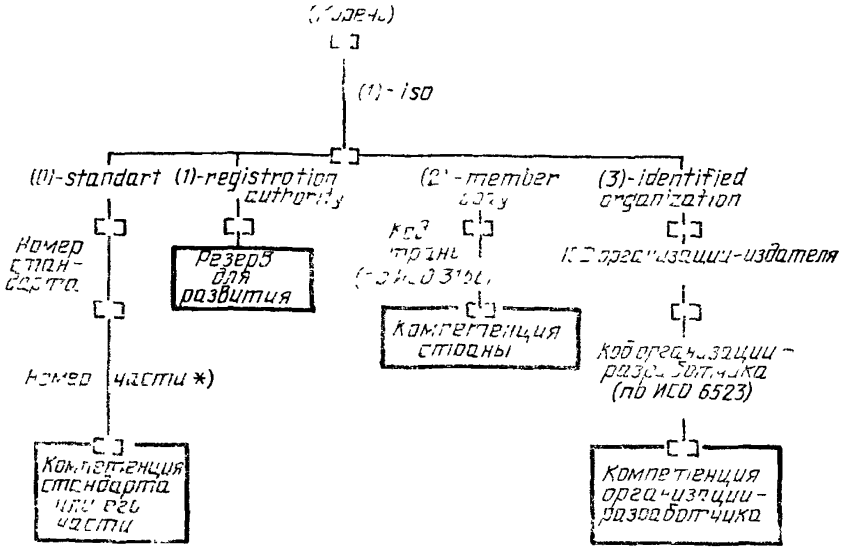
строка октетов (OCTET STRING) 3.18, 8.14, 16, 5.2.4
структурированный тип 3.4
теги 3.6, 5.8, 5.2.8
тегированный тип 24, 5.2.8
тегирование 3.7
тип 3.2
тип T61String 29
тип TeletexString 29
тип VideotexString 29
тип-компонент 3.5
универсальный (UNIVERSAL) 5.8, 24
целочисленный тип (INTEGER) 3.16, 8.2, 8.14, 14, 5.2.2
числовая строка (NumericString) 29, 29.3
элемент «двоичная строка» (b-строка) 8.9
элемент «присвоение». 8.12
элемент «пусто» 8.7
элемент «строка знаков» (c-строка) 8.11
элемент «число» («number») 8.8
элемент «шестнадцатеричная строка» (h-строка) 8.10
элементы 3.9, 8
элементы — ключевые слова 8.14

*ПРИЛОЖЕНИЕ 8*
*Справочное*

## ГРАФЫ КОМПОНЕНТОВ ИДЕНТИФИКАТОРОВ ОБЪЕКТОВ

8.1. Поддерево компонентов идентификаторов объектов, определяемое ИСО

На черт. 8.1 изображена схема дерева, составляемая по пп. 2.1—2.7 приложения 2.

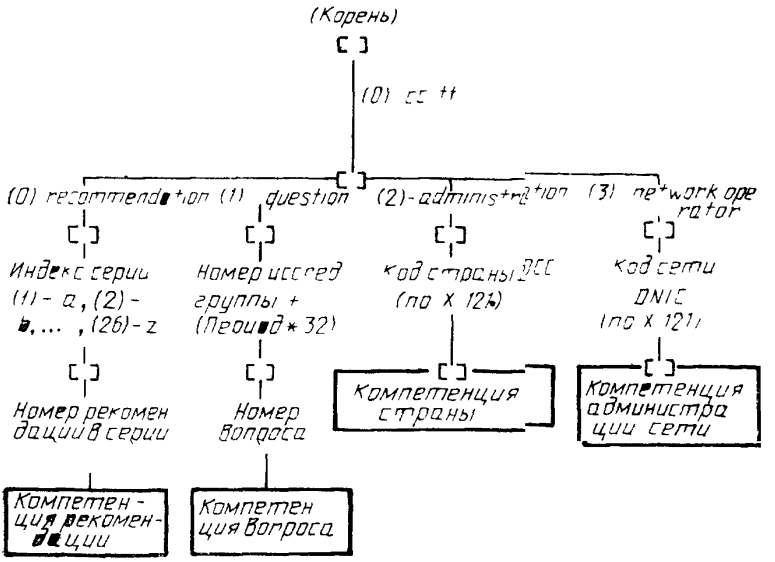



* Присутствует только тогда, когда стандарт разделен на части (см. п. 2.4 приложения 2).

Черт. 8.1. Поддерево компонентов идентификаторов объектов по линии ИСО

8.2. Поддерево компонентов идентификаторов объектов, определяемое МККТТ

На черт. 8.2 изображена схема дерева, составленная по пп. 3.1—3.7 приложения 3.




Черт. 8.2. Поддерево компонентов идентификаторов объектов по линии МККТТ

## ИНФОРМАЦИОННЫЕ ДАННЫЕ

1. **ПОДГОТОВЛЕН И ВНЕСЕН НПО «Персей».**

2. **УТВЕРЖДЕН И ВВЕДЕН В ДЕЙСТВИЕ Постановлением Комитета стандартизации и метрологии СССР от 30.07.91 № 1307 настоящий стандарт подготовлен методом прямого применения международного стандарта ИСО 8824—87 «Системы обработки информации. Взаимосвязь открытых систем. Спецификация абстрактно-синтаксической нотации версии 1 (АСН.1) и полностью ему соответствует**

3. **Срок проверки — 1997 г., периодичность проверки — 5 лет**

4. **ССЫЛОЧНЫЕ НОРМАТИВНО-ТЕХНИЧЕСКИЕ ДОКУМЕНТЫ**

| Раздел, пункт, в котором приведена ссылка | Обозначение соответствующего международного стандарта | Обозначение отечественного нормативного документа, на который дана ссылка |
|---|---|---|
| 0, 2, 3 | | ГОСТ 28906—91 |
| 0, 24.8, Приложение 5 | ИСО 8825—87 | ГОСТ 34.974—91 |
| 29.5 (табл. 7) | МККТТ Т.61 | |
| | МККТТ Т.100 | — |
| | МККТТ Т.101 | — |
| 2, 30.2, 30.3 | ИСО 8601—88 | ГОСТ 24520—80 |
| 0, 2, Приложение 1 | МККТТ Х.409—84 | — |
| Приложение 3 | МККТТ Х.121 | — |
| Приложение 5 | МККТТ Т.5 и Т.6 | — |

СОДЕРЖАНИЕ

0. Введение . . . . . 1
1. Назначение и область применения . . . . . 5
2. Ссылки . . . . . 6
3. Определения . . . . . 6
4. Сокращения . . . . . 10
5. Система обозначений, используемая в настоящем стандарте . . 10

5.1. Продукция . . . . . 11
5.2. Совокупности-варианты . . . . . 11
5.3. Пример продукции . . . . . 12
5.4. Расположение текста . . . . . 12
5.5. Рекурсия . . . . . 12
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 . . . . . 14
8. Элементы АСН.1 . . . . . 15

8.1. Общие правила . . . . . 15
8.2. Ссылки на тип . . . . . 16
8.3. Идентификаторы . . . . . 17
8.4. Ссылки на значение . . . . . 17
8.5. Ссылка на модуль . . . . . 17
8.6. Комментарий . . . . . 17
8.7. Пустой элемент . . . . . 17
8.8. Элемент «число» . . . . . 17
8.9. Элемент «двоичная строка» . . . . . 18
8.10. Элемент «шестнадцатеричная строка» . . . . . 18
8.11. Элемент «строка знаков» . . . . . 18
8.12. Элемент «присвоение» . . . . . 18
8.13. Элементы, состоящие из одного знака . . . . . 18
8.14. Элементы — ключевые слова . . . . . 19

9. Определение модуля . . . . . 19
10. Ссылки на определения типов и значений . . . . . 21
11. Присвоение типов и значений . . . . . 21
12. Определение типов и значений . . . . . 22
13. Обозначения булевского типа . . . . . 25
14. Обозначения целочисленного типа . . . . . 25
15. Обозначения для «строка битов» . . . . . 26
16. Обозначения типа «строка октетов» . . . . . 28
17. Обозначения вырожденного типа . . . . . 29
18. Обозначения типов «последовательность» . . . . . 29
19. Обозначения типов «Последовательность из» . . . . . 31
20. Обозначения типа «Множество» . . . . . 31
21. Обозначения типов «Множество из» . . . . . 32
22. Обозначения выборочных типов . . . . . 33
23. Обозначения селективных типов . . . . . 35
24. Обозначения тегированных типов . . . . . 35
25. Обозначения произвольного типа . . . . . 37
26. Обозначения типа «Идентификатор объекта» . . . . . 37
27. Обозначения типов «Строка знаков» . . . . . 39
28. Обозначения типов: общая форма записи времени, всемирное время, внешний тип. Описатель объекта . . . . . 40
29. Определение типов «строка знаков» . . . . . 40

30. Общая форма записи времени . . . . . 42
31. Всемирное время . . . . . 44
32. Внешний тип . . . . . 45
33. Тип «Описатель объекта» . . . . . 48

Приложения:

1. Макронотация . . . . . 50
2. Значения компонентов OBJECT IDENTIFIER, определяемые ИСО . . 57
3. Значения компонентов OBJECT IDENTIFIER, определяемые МККТТ . 59
4. Значения компонентов OBJECT IDENTIFIER, определяемые совместно . 61
5. Примеры и рекомендации . . . . . 62
6. Сводка синтаксиса нотации АСН.1 . . . . . 76
7. Алфавитный указатель . . . . . 81
8. Графы компонентов идентификаторов объектов . . . . . 83

Информационные данные . . . . . 85

Редактор *Р. Г. Говердовская*
Технический редактор *В. Н. Малькова*
Корректор *А. И. Зюбан*

Сдано в наб 06.09.91 Подп. к печ. 28.11.91 5,5 усл. п. л. 5,63 усл. кр.-отт. 6,10 уч.-изд. л. Тираж 440 экз. Цена 48 р. 55 к.

Ордена «Знак Почета» Издательство стандартов, 123557, Москва, ГСП Новопресненский пер., 3.
Калужская типография стандартов, ул. Московская, 256. Зак. 1720

gost.e-ecolog.ru