

20 апреля 2024

Протоколы и интерфейсы БС LTE. Лекция 3.

Введение в ASN1C

ВЛАД РЫЖОВ

Fast Track в Телеком, 2024



Что сегодня будет происходить? Адженда

Введение в ASN1

15 минут

**Процесс
передачи
сообщения**

15 минут

Типы данных

15 минут

Пример

15 минут

Введение в ASN1

15 минут

**Процесс
передачи
сообщения**

15 минут

Типы данных

15 минут

Пример

15 минут

Компиляция кода – это...

Введение в ASN1

Abstract Syntax Notation number One (ASN.1) - формальное описание данных, передаваемых телекоммуникационным протоколом, которое не зависит от языка имплементации, представления данных или приложения, какими бы сложными они не были.

Базовые типы данных:

- integers (INTEGER)
- booleans (BOOLEAN)
- character strings (IA5String, UniversalString...)
- bit strings (BIT STRING)
- etc.

Составные типы данных:

- structures (SEQUENCE)
- lists (SEQUENCE OF)
- choice between types (CHOICE)
- etc.



ITU-T X.680	ISO/IEC 8824-1
ITU-T X.681	ISO/IEC 8824-2
ITU-T X.682	ISO/IEC 8824-3
ITU-T X.683	ISO/IEC 8824-4
ITU-T X.690	ISO/IEC 8825-1
ITU-T X.691	ISO/IEC 8825-2



**Какие языки описания данных вы
можете назвать?**

Пример

```
MyShopPurchaseOrders DEFINITIONS AUTOMATIC TAGS ::= BEGIN
```

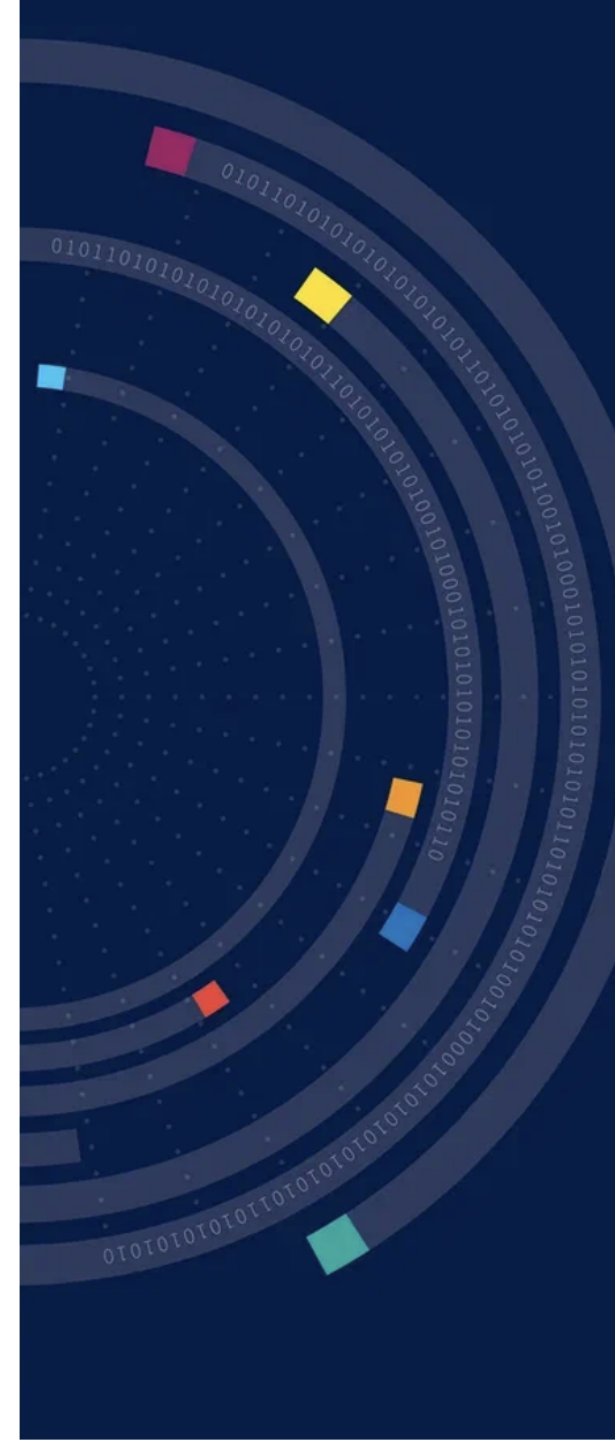
```
PurchaseOrder ::= SEQUENCE {  
  dateOfOrder DATE,  
  customer    CustomerInfo,  
  items       ListOfItems  
}
```

```
CustomerInfo ::= SEQUENCE {  
  companyName  VisibleString (SIZE (3..50)),  
  billingAddress Address,  
  contactPhone NumericString (SIZE (7..12))  
}
```

```
Address ::= SEQUENCE {  
  street  VisibleString (SIZE (5 .. 50)) OPTIONAL,  
  city    VisibleString (SIZE (2..30)),  
  state   VisibleString (SIZE(2) ^ FROM ("A".. "Z")),  
  zipCode NumericString (SIZE(5 | 9))  
}
```

```
ListOfItems ::= SEQUENCE (SIZE (1..100)) OF Item
```

```
Item ::= SEQUENCE {  
  itemCode    INTEGER (1..99999),  
  color        VisibleString ("Black" | "Blue" | "Brown"),  
  power        INTEGER (110 | 220),  
  deliveryTime INTEGER (8..12 | 14..19),  
  quantity     INTEGER (1..1000),  
  unitPrice    REAL (1.00 .. 9999.00),  
  isTaxable    BOOLEAN  
}  
END
```



Пример

```
MyShopPurchaseOrders DEFINITIONS AUTOMATIC TAGS ::= BEGIN
```

```
PurchaseOrder ::= SEQUENCE {  
  dateOfOrder DATE,  
  customer CustomerInfo,  
  items ListOfItems  
}
```

type

```
CustomerInfo ::= SEQUENCE {  
  companyName VisibleString (SIZE (3..50)),  
  billingAddress Address,  
  contactPhone NumericString (SIZE (7..12))  
}
```

identifier

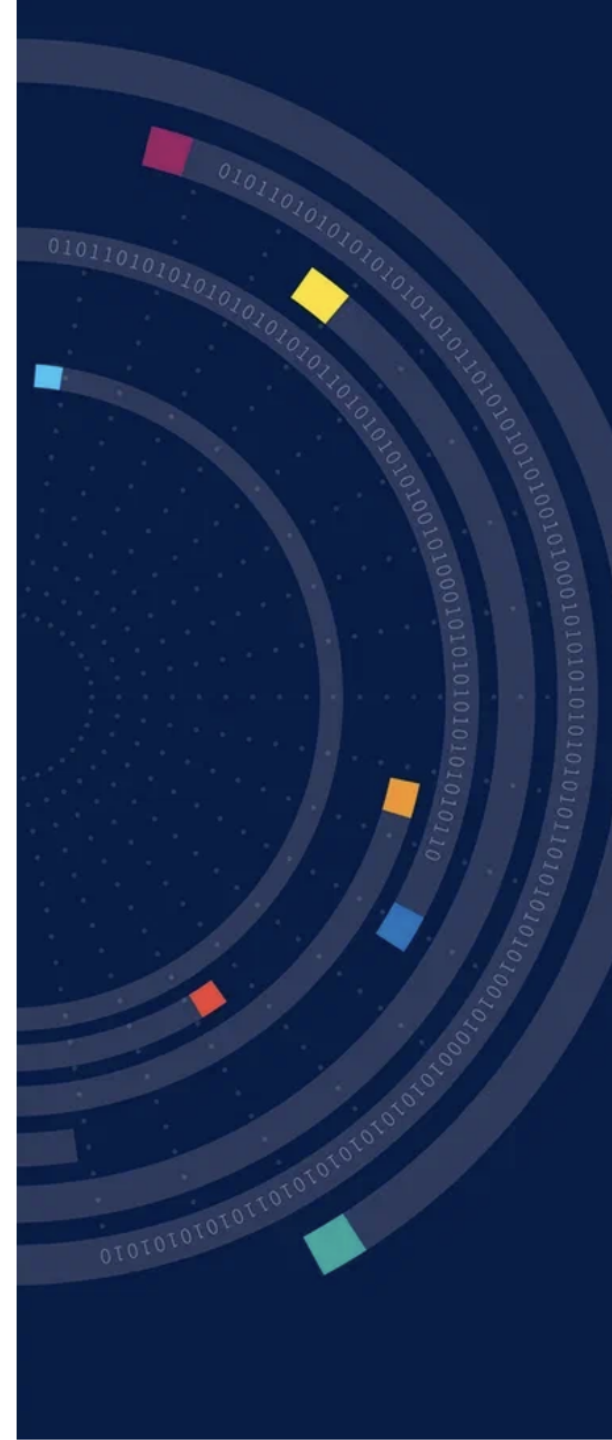
```
Address ::= SEQUENCE {  
  street VisibleString (SIZE (5 .. 50)) OPTIONAL,  
  city VisibleString (SIZE (2..30)),  
  state VisibleString (SIZE(2) ^ FROM ("A".. "Z")),  
  zipCode NumericString (SIZE(5 | 9))  
}
```

constraint

```
ListOfItems ::= SEQUENCE (SIZE (1..100)) OF Item
```

```
Item ::= SEQUENCE {  
  itemCode INTEGER (1..99999),  
  color VisibleString ("Black" | "Blue" | "Brown"),  
  power INTEGER (110 | 220),  
  deliveryTime INTEGER (8..12 | 14..19),  
  quantity INTEGER (1..1000),  
  unitPrice REAL (1.00 .. 9999.00),  
  isTaxable BOOLEAN  
}  
END
```

basic ASN1 type



Введение в ASN1

15 минут

**Процесс
передачи
сообщения**

15 минут

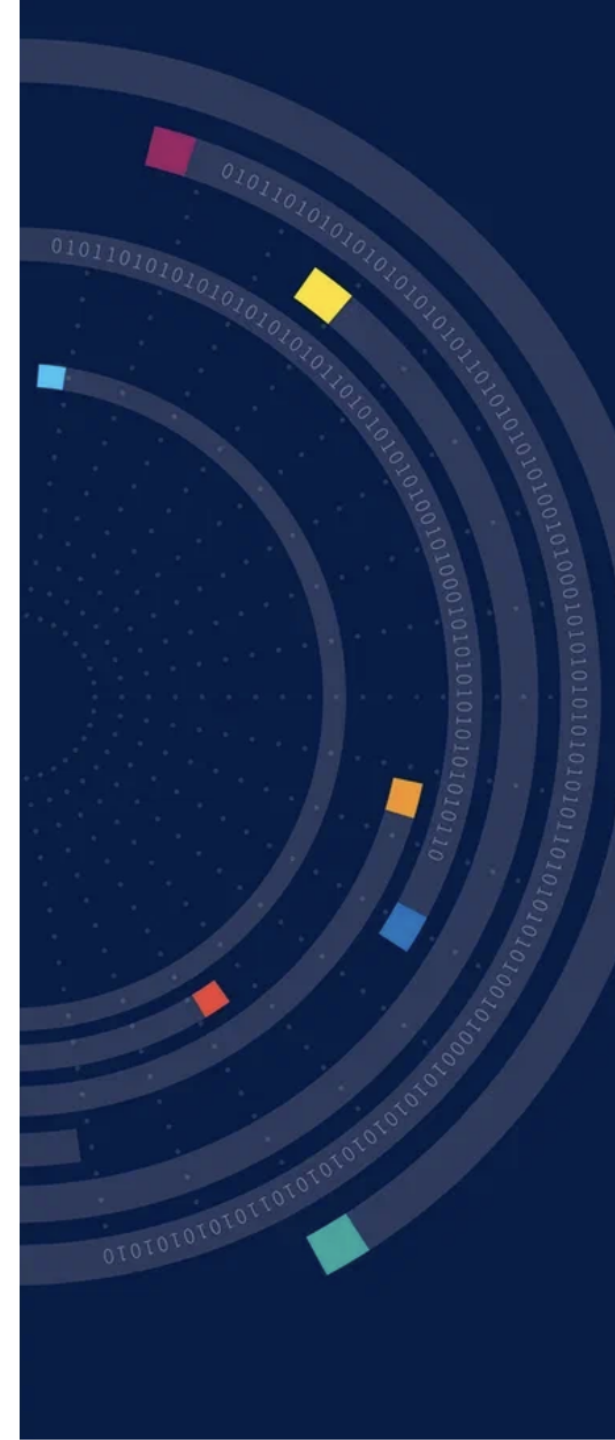
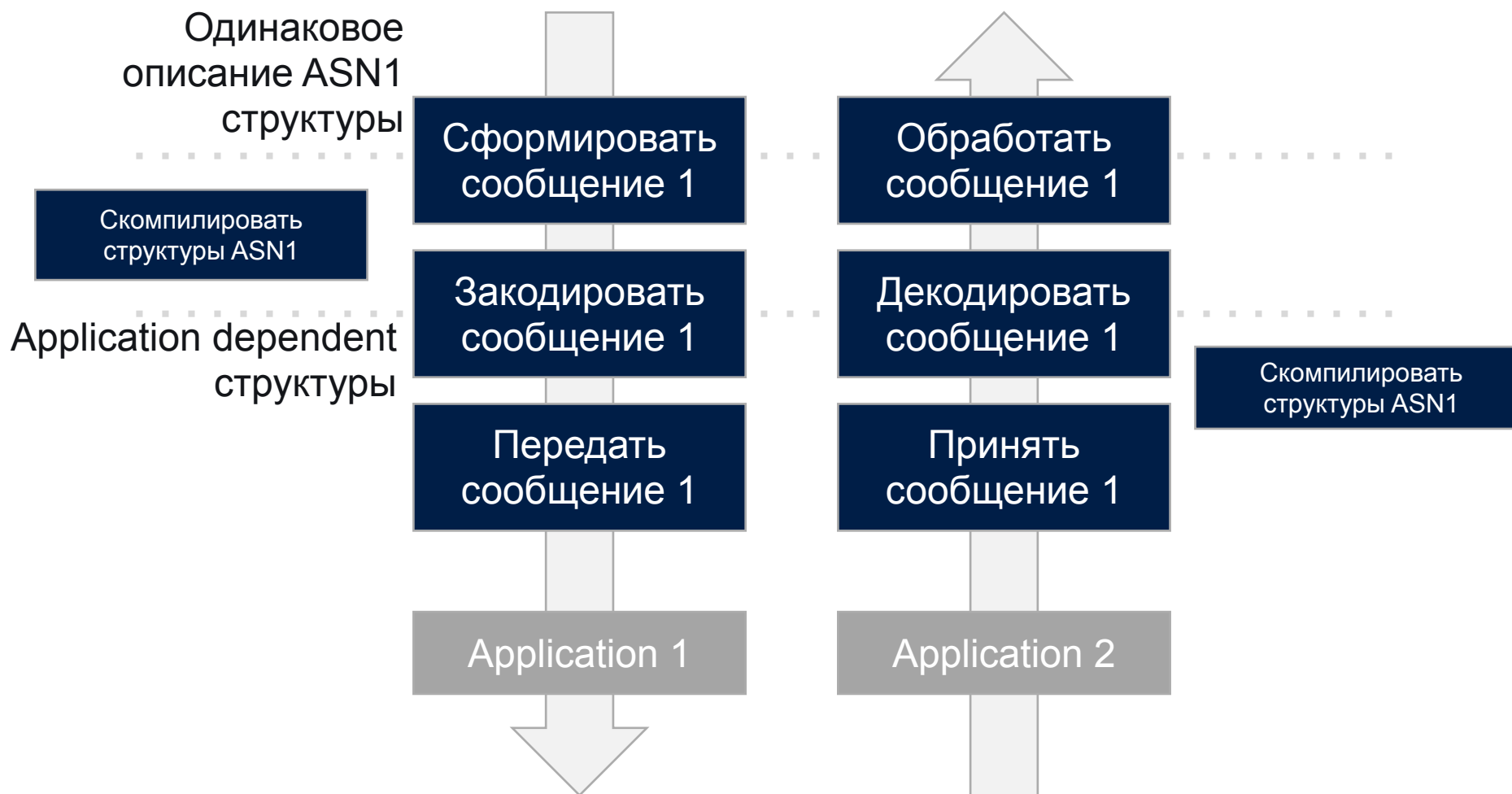
Типы данных

15 минут

Пример

15 минут

Процесс передачи сообщения.



ASN1C compiler.

ASN1C компилятор конвертирует нотацию ASN1 в нотацию, которая используется в языке C/C++.

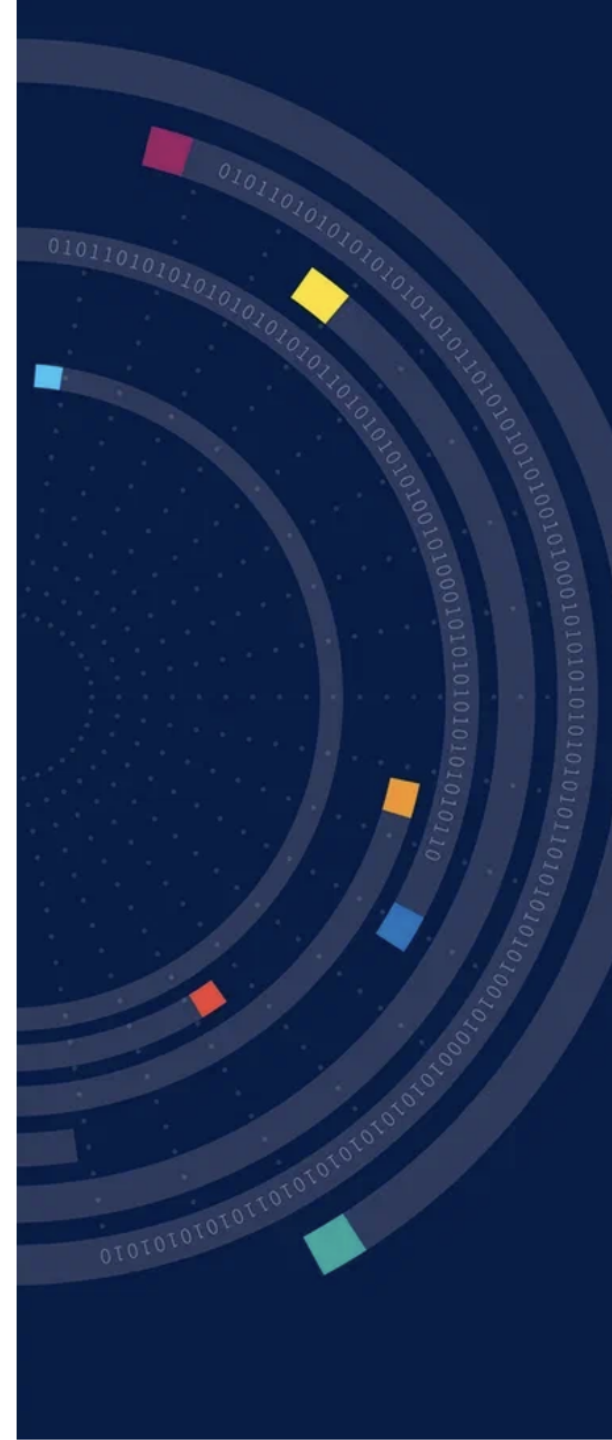
```
RectangleTest DEFINITIONS ::= BEGIN

Rectangle ::= SEQUENCE {
    height  INTEGER,      -- Height of the rectangle
    width   INTEGER      -- Width of the rectangle
}

END
```

ASN1C compiler

```
typedef struct Rectangle_s {
    long height;
    long width;
} Rectangle_t;
```



Введение в ASN1

15 минут

**Процесс
передачи
сообщения**

10 минут

Типы данных

20 минут

Пример

15 минут

Базовые типы данных

The BOOLEAN type

Тип BOOLEAN представляет собой простой выбор из двух TRUE/FALSE, YES/NO, ON/OFF.

The INTEGER type

Тип INTEGER – знаковое натуральное число без ограничения на его размер.

SimpleInteger ::= INTEGER

-- An integer with a very limited range

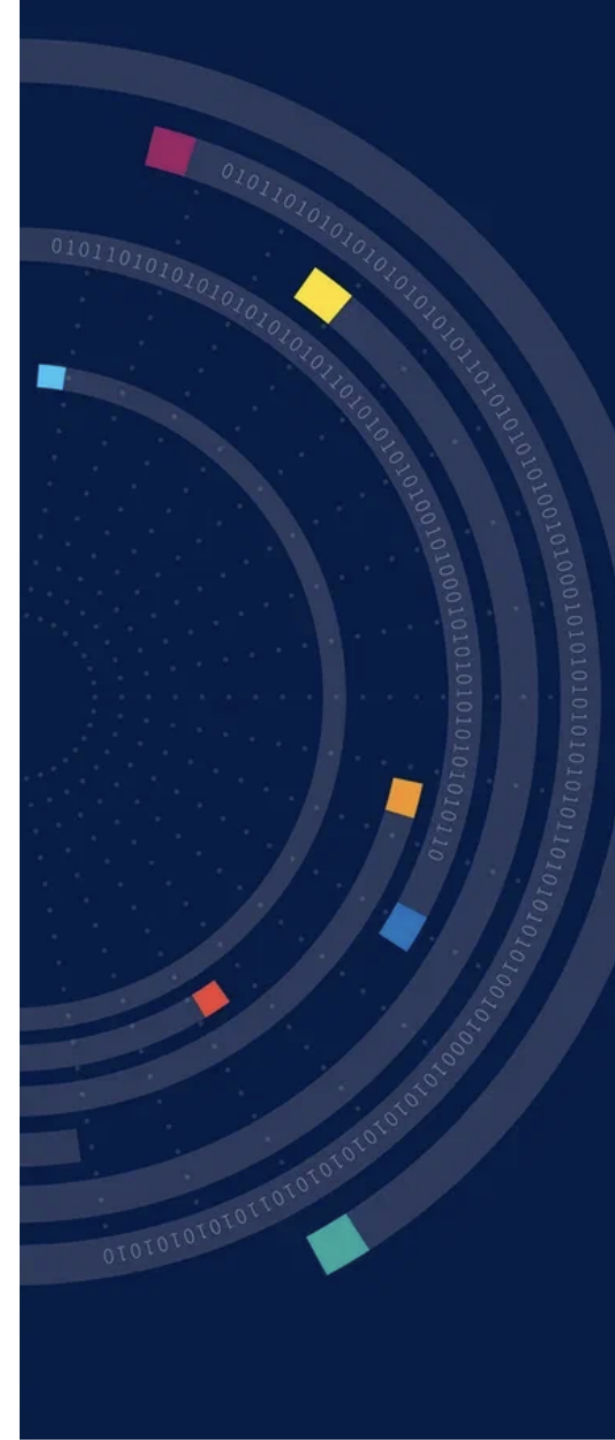
SmallPositiveInt ::= INTEGER (0..127)

-- Integer, negative

NegativeInt ::= INTEGER (MIN..0)

The OCTET STRING type

Тип OCTET STRING представляет собой последовательность 8-битных байт. Часто используется для передачи некоторых “непрозрачных” данных или данных, сериализованных другими типами кодеров.



Базовые типы данных

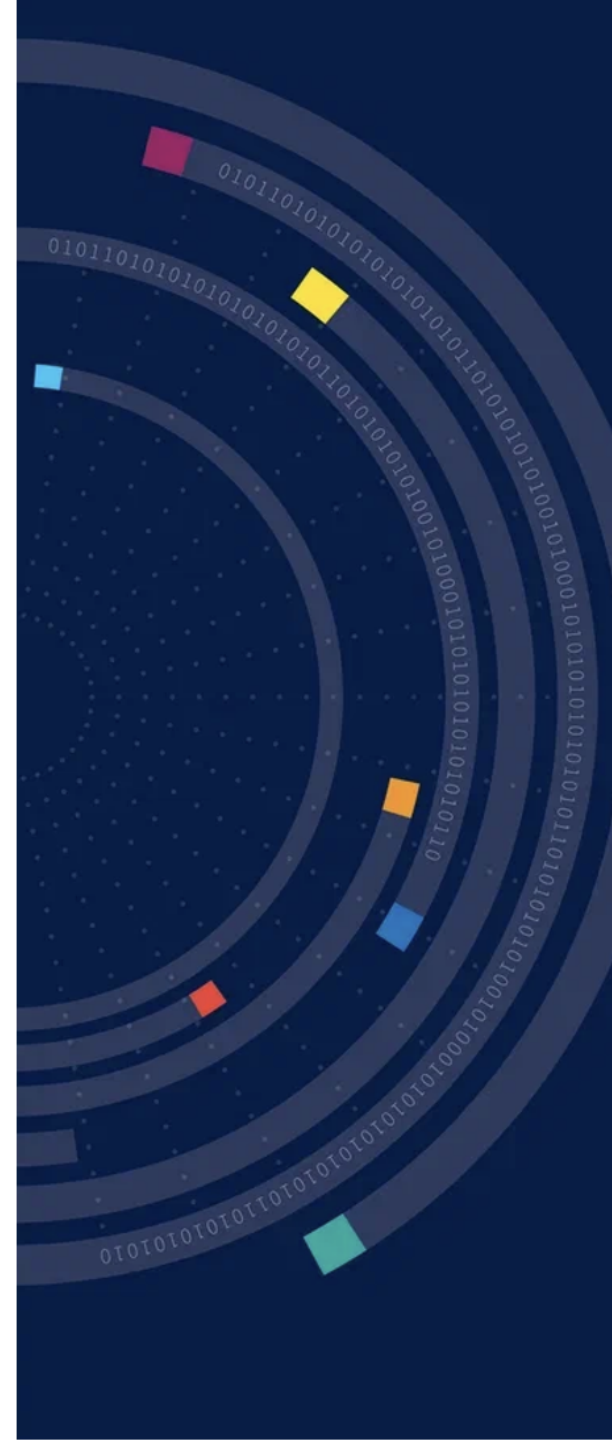
The ENUMERATED type

Тип ENUMERATED семантически эквивалентен типу INTEGER за исключением того, что некоторые из его значений явно названы.

```
FruitId ::= ENUMERATED { apple(1), orange(2) }
```

- The numbers in braces are optional,
- the enumeration can be performed
- automatically by the compiler

```
ComputerOSType ::= ENUMERATED {  
    FreeBSD, -- acquires value 0  
    Windows, -- acquires value 1  
    Solaris(5), -- remains 5  
    Linux, -- becomes 6  
    MacOS -- becomes 7  
}
```



Базовые типы данных

The OBJECT IDENTIFIER type

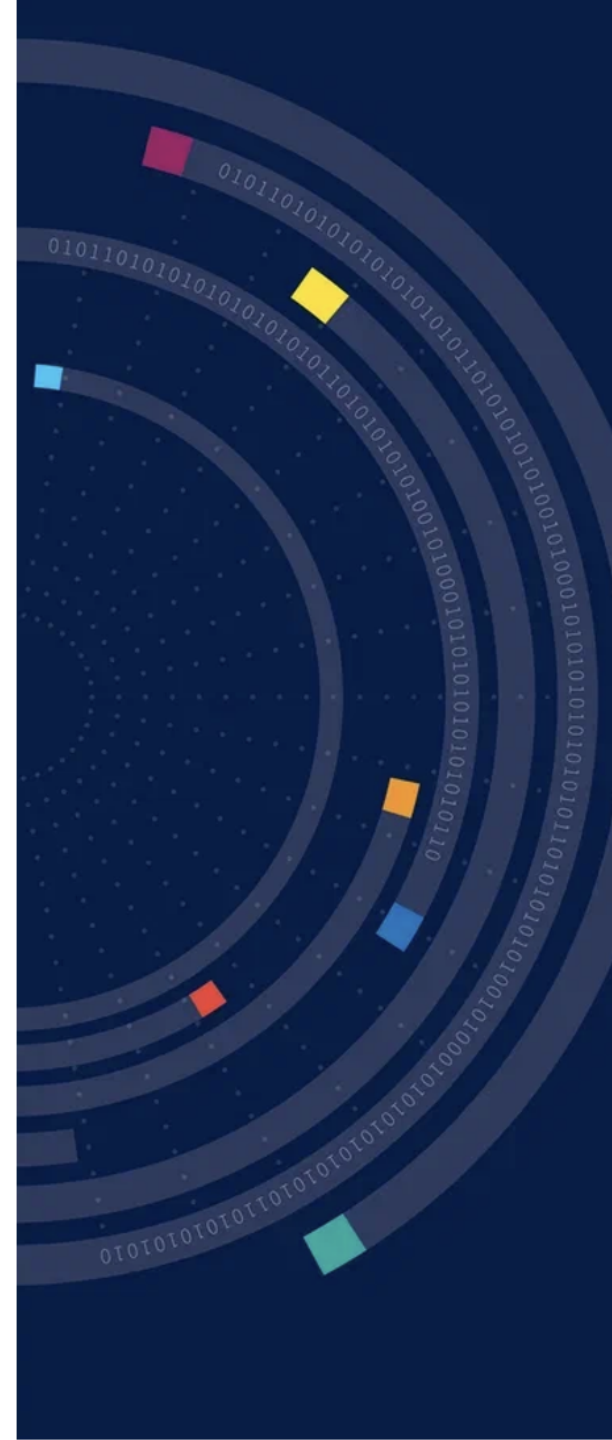
Тип OBJECT IDENTIFIER используется для представления уникального идентификатора любого объекта, начиная с самого корня дерева регистрации.

ExampleOID ::= OBJECT IDENTIFIER

rectangleModule1-oid ExampleOID ::= { 1 3 6 1 4 1 9363 1 5 2 1 1 }

-- An identifier of the Internet.

internet-id OBJECT IDENTIFIER ::= { iso(1) identified-organization(3) dod(6) internet(1) }



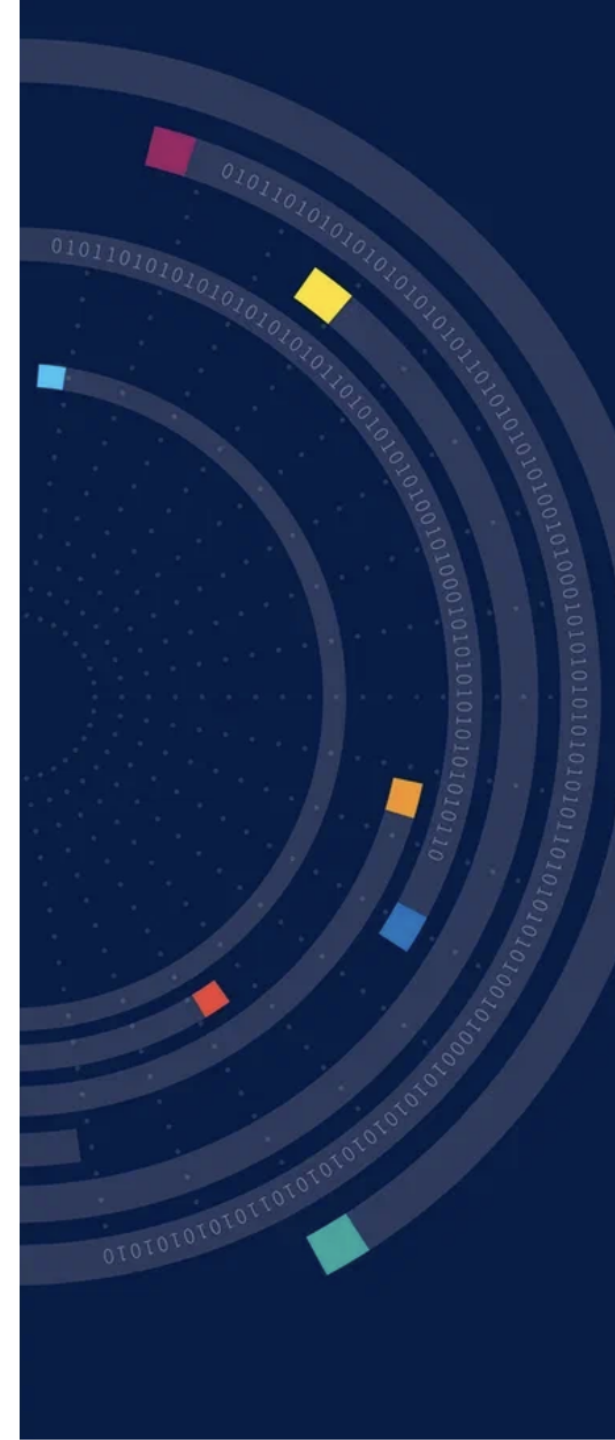
Базовые типы данных

The RELATIVE-OID type

Тип RELATIVE-OID имеет семантику поддерева OBJECT IDENTIFIER. Иногда нет необходимости повторять полную последовательность чисел от корня дерева регистрации, если только некоторая подпоследовательность дерева представляет собой интерес.

this-document RELATIVE-OID ::= { docs(2) usage(1) }

this-example RELATIVE-OID ::= { this-document assorted-examples(0) this-example(1) }



Строковые типы данных

The IA5String type

По сути, это формат ASCII, содержащий 128 доступных кодов символов (7 младших битов 8-битного байта).

The UTF8String type

Это символьная строка, которая кодирует полный Unicode диапазон в 4 байта с использованием многобайтовых последовательностей символов.

The NumericString type

Тип NumericString представляет собой символьную строку с алфавитом, состоящим из цифр (от “0” до “9”) и пробела.

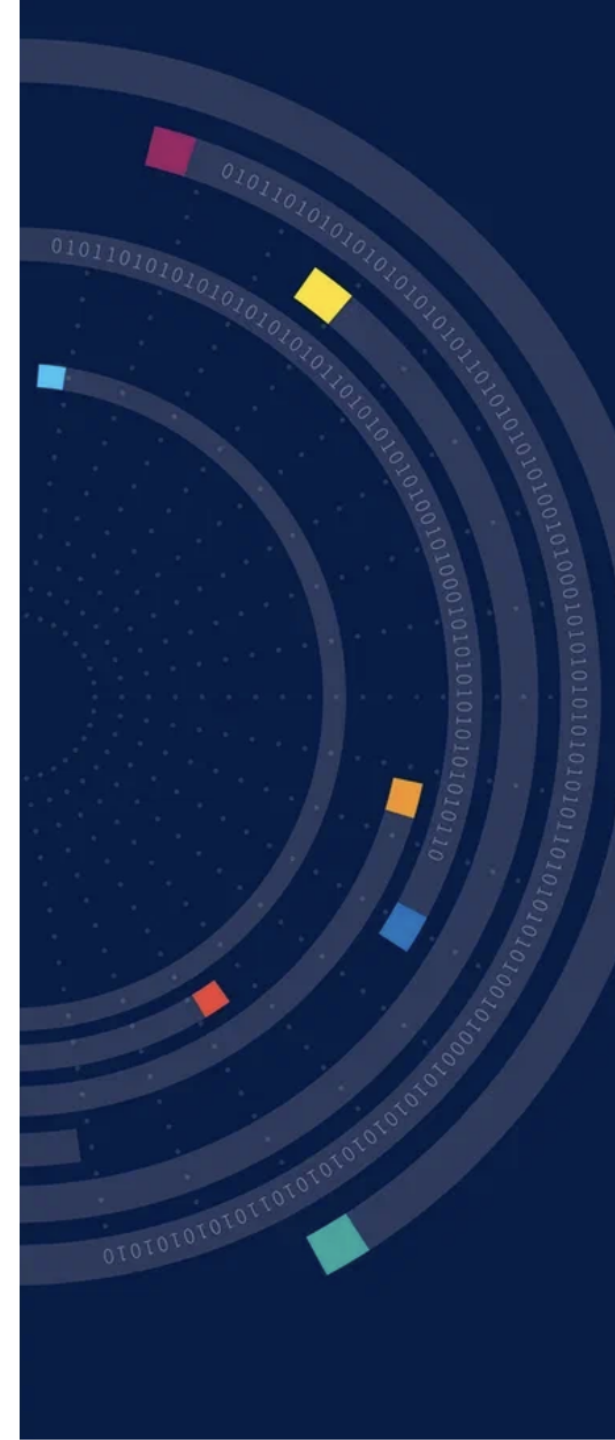
The PrintableString type

Строка символов со следующим алфавитом: пробел, “” (одинарная кавычка), “(”, “)”, “+”, “,” (запятая), “-”, “.”, “/”, цифры (от “0” до “9”), “:”, “=”, “?”, заглавные и строчные буквы (от “A” до “Z” и от “a” до “z”).

Строковые типы данных

The VisibleString type

Строка символов с алфавитом, который является большим или меньшим подмножеством ASCII символов между пробелом и символом “~” (тильда). Иначе алфавит можно описать как алфавит с печатной строкой, представленный выше вместе со следующими символами: “!”, ““”, “#”, “\$”, “%”, “&”, “*”, “.”, “:”, “[”, “\”, “]”, “^”, “_”, “”” (одиночная левая кавычка), “{”, “|”, “}”, “~”.

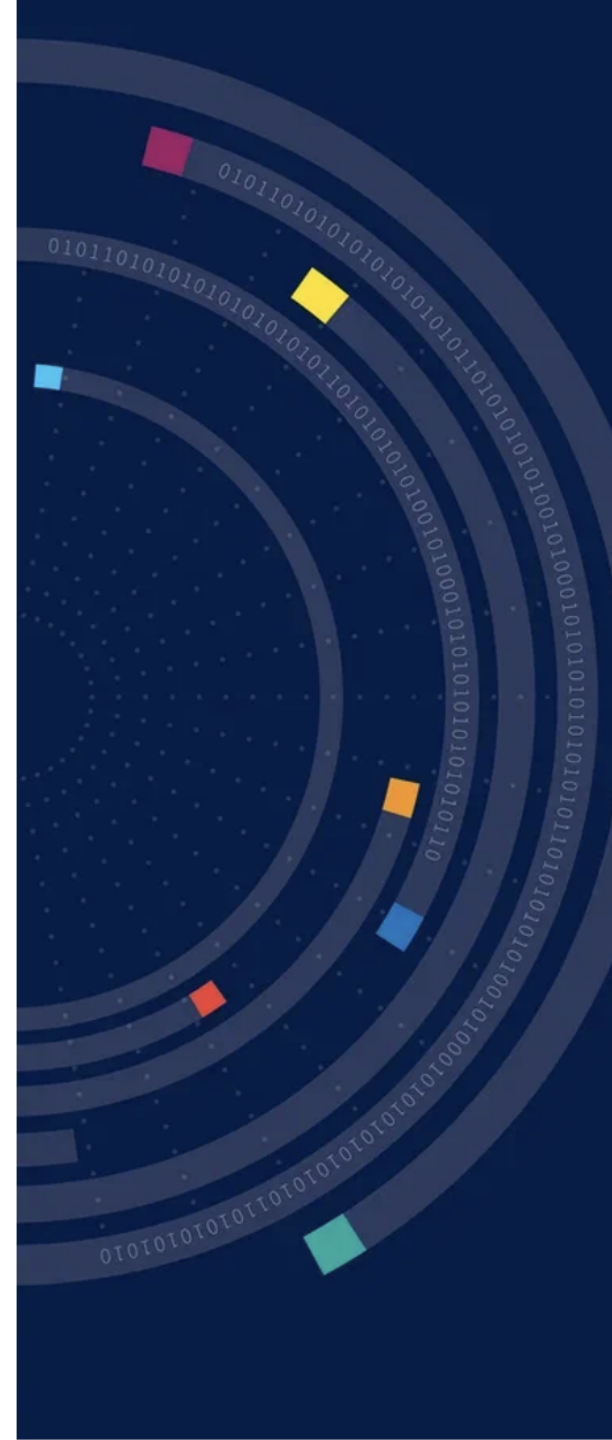


Составные типы данных

The SEQUENCE type

Это упорядоченная коллекция других простых или составных типов. Сконструированный тип SEQUENCE похож на struct в языке Си.

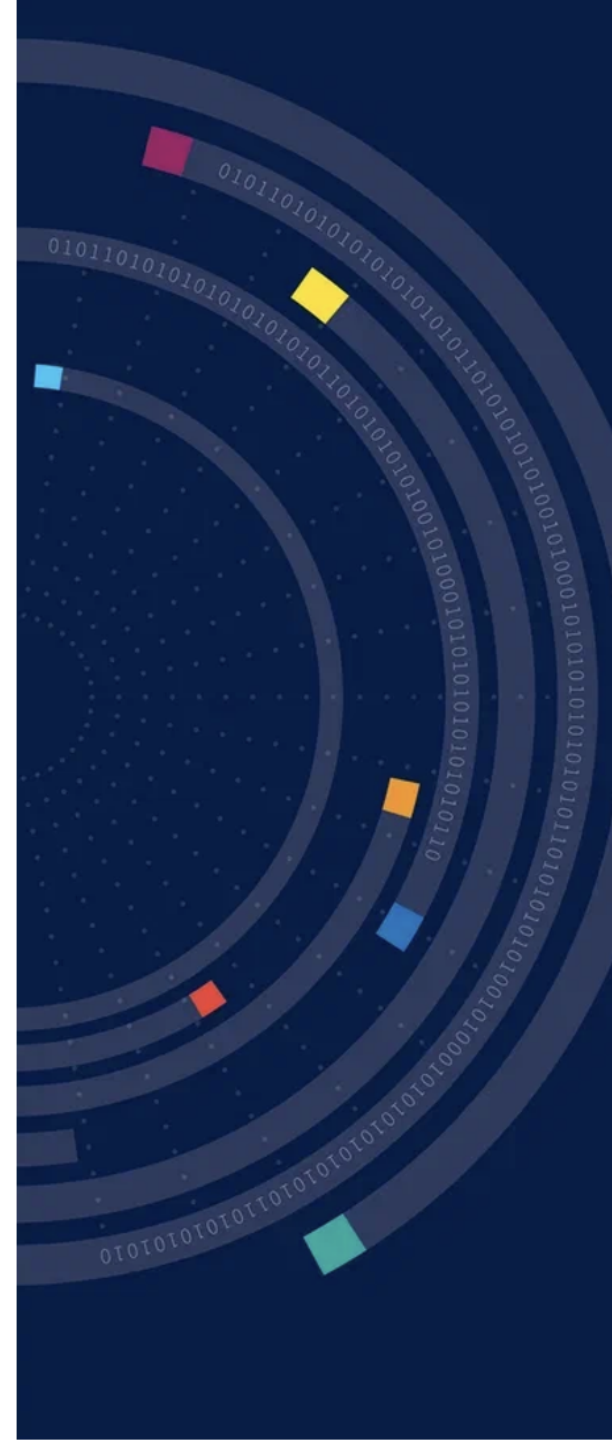
```
Address ::= SEQUENCE {  
    -- The apartment number may be omitted  
    apartmentNumber NumericString OPTIONAL,  
    streetName PrintableString,  
    cityName PrintableString,  
    stateName PrintableString,  
    -- This one may be omitted too  
    zipNo NumericString OPTIONAL  
}
```



Составные типы данных

The SET type

Это набор других простых или составных типов. Порядок не имеет значения. Данные могут поступать в порядке, отличном от порядка спецификации. Данные кодируются в порядке, который не обязательно соответствует порядку спецификации.



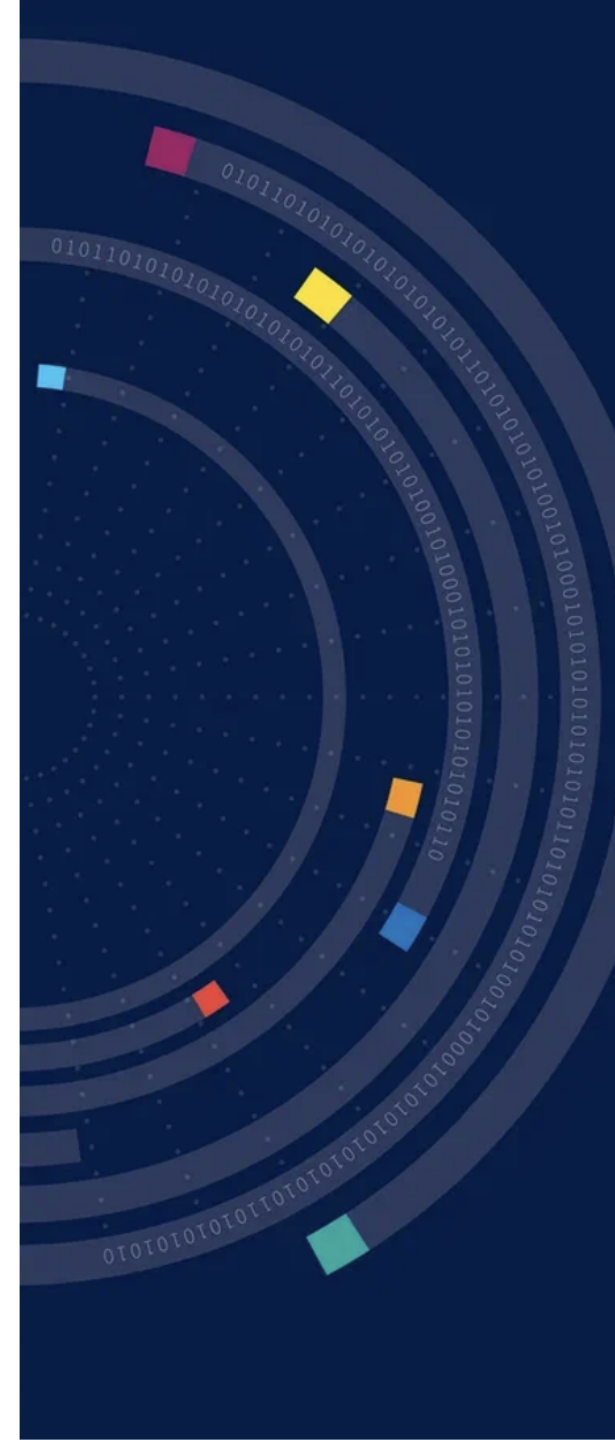
Составные типы данных

The CHOICE type

Этот тип представляет собой просто выбор между указанными в нем подтипами. Тип CHOICE содержит не более одного из указанных подтипов, и всегда неявно известно, какой вариант декодируется или кодируется. Этот вариант напоминает оператор C “union”.

Пример ниже определяет код ответа, который может быть либо целым кодом, либо логическим кодом “истина”/“ложь”.

```
ResponseCode ::= CHOICE{  
    intCode INTEGER,  
    boolCode BOOLEAN  
}
```



Составные типы данных

The SEQUENCE OF type

Это список (массив) простых или составных типов:

-- Example 1

ManyIntegers ::= SEQUENCE OF INTEGER

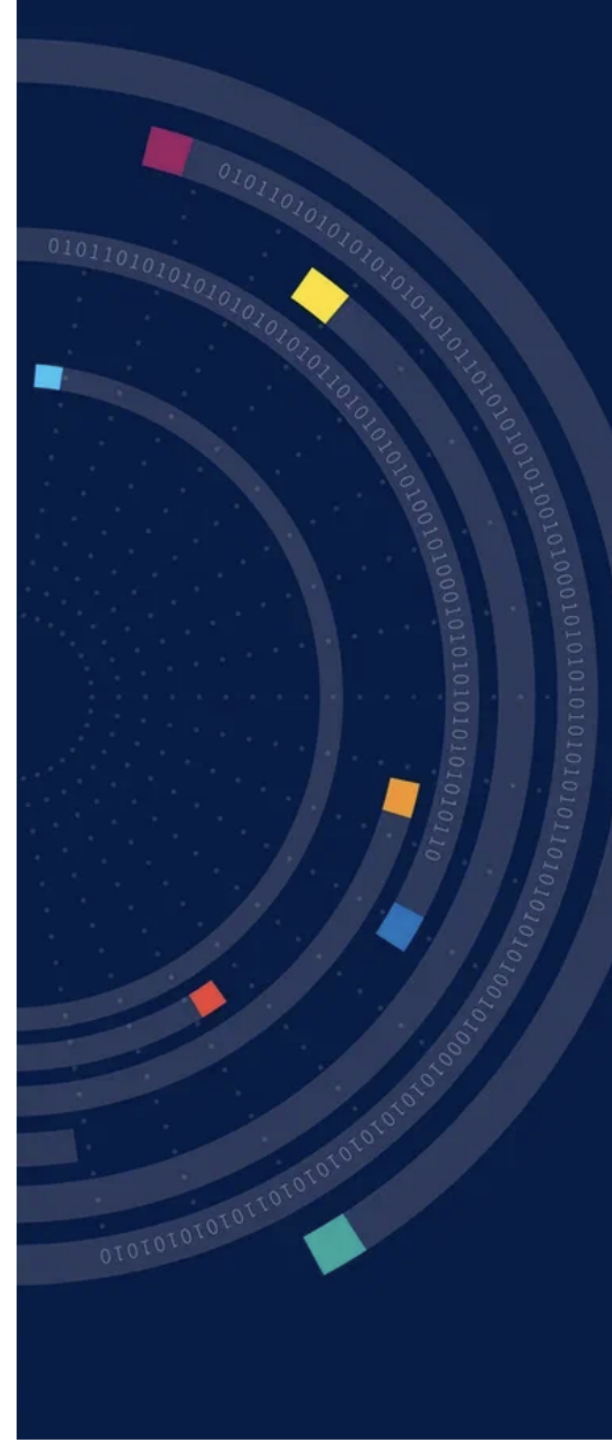
-- Example 2

ManyRectangles ::= SEQUENCE OF Rectangle

-- More complex example:

-- an array of structures defined in place.

ManyCircles ::= SEQUENCE OF SEQUENCE { radius INTEGER }



Составные типы данных

The SET OF type

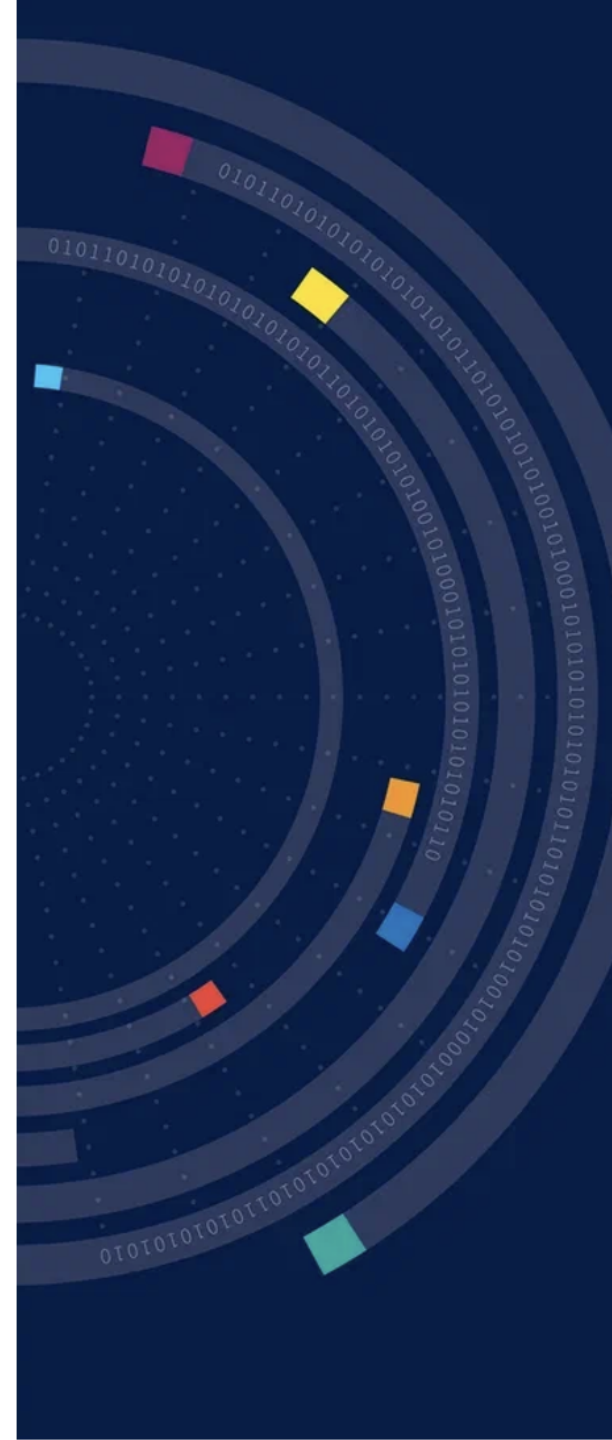
Тип SET OF моделирует набор структур. Он похож на тип SEQUENCE OF, но для него не важен порядок: элементы могут поступать в порядке, который не обязательно совпадает с порядком в памяти на удаленных компьютерах.

-- A set of structures defined elsewhere

SetOfApples :: SET OF Apple

-- Set of integers encoding the kind of a fruit

FruitBag ::= SET OF ENUMERATED { apple, orange }



Ограничения

Single value constraint

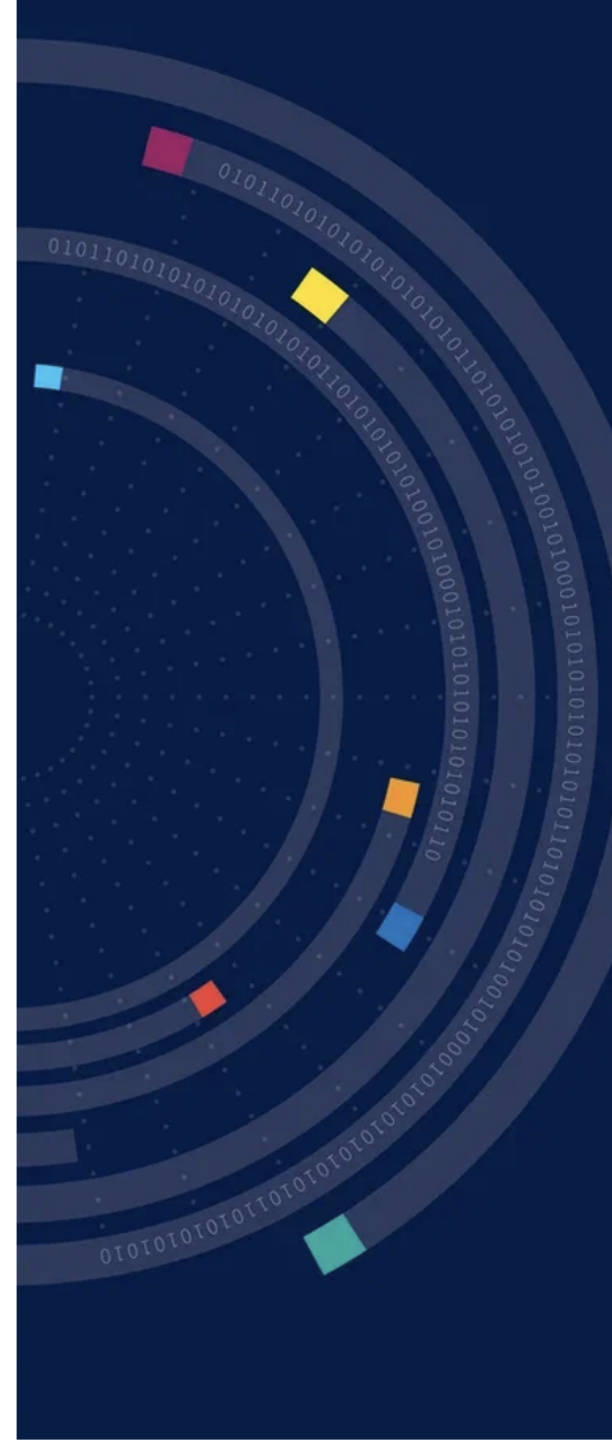
- color `VisibleString` ("Black" | "Blue" | "Brown")
- power `INTEGER` (110 | 220)

Value and alphabet range constraint

- quantity `INTEGER` (1..1000)
- deliveryTime `INTEGER` (8..12 | 14..19)
- unitPrice `REAL` (1.00..9999.00)
- State ::= `VisibleString` `SIZE`(2)

Size constraint

- contactPhone `NumericString` (`SIZE` (7..12))
- ListOfItems ::= `SEQUENCE` (`SIZE` (1..100)) `OF` Item
- zipCode `NumericString` (`SIZE` (5 | 9))



Введение в ASN1

15 минут

**Процесс
передачи
сообщения**

15 минут

Типы данных

15 минут

Пример

15 минут

Разбор примера + ДЗ



СПАСИБО