1.1.34. Examples of the Use of Size

```
MonthNumber ::= NumericString (SIZE (1 ..2))
MonthNumber ::= NumericString (SIZE (1 |2))
Base ::= BIT STRING (SIZE ( 0 | 2 .. 7 | 10 ))
Couple ::= SET SIZE(2) OFHuman
BridgeDeal ::= SET SIZE (13) OFPlayingCard
BridgeHand ::= SET SIZE (0..13) OFPlayingCard
lineLength INTEGER 80
Line ::= VisibleString (SIZE (0 .. lineLength)
```

Exercise 15

The X.400 standard specifies that a name can consist of several subfields. One of the subfields is called OrganizationName and can have as value between 1 and 64 characters from the character se PrintableString. Suggest a definition of this in ASN.1.

1.1.35. Character String Types

ASN.1 has several Character String types for different charactersets.

NumericString“	“0” .. “9” and “ ”
PrintableString	“a”..“z”, “A”..“Z”, “0”..“9” ' ( ) + , - . / : = ?
TeletexString T61String	The T.61 or ISO 6937 character set, a set which uses one or two octets to specify more than 255 different characters, for example, the character É is specified by the two characters “ ' E”.
VisibleString ISO646String	Printable characters, including space, from ISO 646 (”ASCII”), but no format control characters like Carriage Return or Line Feed.
IA5String	IA5 (ISO 646, ”ASCII”).
GraphicString	Can contain characters from several different character sets, using ISO 2022 codes to switch from one character set to another character set within the string. Can only contain printable characters and space, not format control characters.
GeneralString	Same as GraphicString, but can also contain formatting characters.
UniversalString	ISO 10646.
CharacterString	Can contain characters from multiple character sets, using ISO 2022 codes to switch between the sets.

Character Strings have a special kind of subtype only available for Character Strings. It is called Permitted Alphabet, and uses a list of characters allowed in a new type. Example: **PrintableString (FROM("0" | "1" | "2" | "3" | "4" | "5" | "6" | "7"))**

1.9. Structured types

Structured types specify new types by combining several components of one or more already defined types. This table lists the basic constructed types in ASN.1.

SET	A list of component fields, like a record in a data base. the components can be included in any order, and the order of the components when transmitted does not convey any information.	**Chairman ::= SET {** **democratic chairman [0] General String, republican chairman [1] General String }**
SEQUENCE	Similar to **SET**, but the fields must be sent in a certain order.	**Ingredients ::= SEQUENCE {** **peas REAL,** **eggs INTEGER }**
SET OF	Zero, one or more components, all of the same type. The order of the components conveys no information.	**Ingredients ::= SET OF Ingredient** **Couple ::= SET SIZE (2) OF Person**
SEQUENCE OF	Like **SET OF**, but order has significance.	**Children ::= SET OF Person**
CHOICE	Has as value one of a listed number of alternative types.	**Vehicle ::= CHOICE {** **Bus, Car, Bicycle }**

For the **SET OF** and **SEQUENCE OF** types, it is possible to indicate that one or more of the components need not be included. Example:

```
KnownParents ::= SEQUENCE OF {
father Male OPTIONAL,
mother Female OPTIONAL }
```

The two declarations of **Month** above define the same value set. **MAX** and **MIN** means that there is no limit. This is not the same thing as +∞ and -∞), an **INTEGER** cannot have infinity as a value, but it can be of arbitrary size.

Exercise 7

Change the definition of **Measurement** in Exercise 2 so that feet can only have the values 0, 1 or 2 (since 3 feet will be a yeard), and so that inches is specified as an integer between 0 and 1199 giving the value in hundreds of an inch (since 1200 or 12 inces will be a foot).

1.1.26. Boolean Type

The Boolean type has only two values, **TRUE** and **FALSE**. Example:

```
ShopOpen ::= BOOLEAN
```

It is *not* permitted to write:

```
Gender ::= BOOLEAN {male (TRUE), female (FALSE) }
```

but instead, you can write

```
Gender ::= BOOLEAN
male Gender ::= TRUE
female Gender ::= FALSE
```

Exercise 8

In an opinion poll, made at the exit door from the election rooms, every voter is asked to indicate which party they voted for. Allowed values are Labour, Liberals, Conservatives or “other”. The age of each voter is also registered as a positive integer above the voting age of 18 years, and the gender is registered. Define a data type to transfer this information from the poll station to a server.

Exercise 9

In the local election in Hometown, there are also two local parties, the Hometown party and the Drivers party. Extend solution 1 to exercise 8 to a new datatype **HometownVoter** where also these two additional parties are allowed.

1.1.27. Enumerated

The **ENUMERATED** type can only have the values which are enumerated in its declaration. The syntax is similar to the **INTEGER** type. Example:

```
DayOfTheWeek ::= ENUMERATED {monday (1), tuesday (2), wednesday (3), thursday (4),
friday (5), saturday (6), sunday (7) }
```

A difference between **ENUMERATED** and **INTEGER** is that the values of the **ENUMERATED** type are not ordered. The following construct:

```
WeekDayNumber ::= INTEGER {monday (1), tuesday (2), wednesday (3), thursday (4),
friday (5), saturday (6), sunday (7) }

WorkingDayNumber ::= WeekDayNumber ( 1 .. 5 )
```

is thus not permitted, with **ENUMERATED**, you have to define this subtype as:

```
WorkingDay ::= DayOfTheWeek ( monday | tuesday | wednesday | thursday | friday |
saturday | sunday )
```

Compare the following three definitions of DayOfTheWeek:

① `DayOfTheWeek ::= INTEGER { monday(1), tuesday(2), wednesday(3), thursday(4), friday(5), saturday(6), sunday(7) }`

② `DayOfTheWeek ::= INTEGER { monday(1), tuesday(2), wednesday(3), thursday(4), friday(5), saturday(6), sunday(7) } (1..7)`

③ `DayOfTheWeek ::= ENUMERATED { monday(1), tuesday(2), wednesday(3), thursday(4), friday(5), saturday(6), sunday(7) }`

Case ① allows all possible integers as values, case ② and ③ only allows the seven values 1 to 7. Case ② has a defined order, case ③ has no defined order of the values.

1.1.28. Real Type

The **REAL** type includes the following allowed values: +∞, --∞ and values of the form
$M * B^E$, where M and E can be any ASN.1 **INTEGER** and B can only have the value 2 or 10. Examples: