3.6.3 String Types

Static Semantics

(1)

A one-dimensional array type whose component type is a character type is called a string type.

(2)

There are two predefined string types, String and Wide_String, each indexed by values of the predefined subtype Positive; these are declared in the visible part of package Standard:

(3)

```
subtype Positive is Integer range 1 .. Integer'Last;
```

(4)

```
type String is array(Positive range <>) of Character;
type Wide_String is array(Positive range <>) of Wide_Character;
```

NOTES

(5)

(49) String literals (see 2.6 and 4.2) are defined for all string types. The concatenation operator & is predefined for string types, as for all nonlimited one-dimensional array types. The ordering operators <, <=, >, and >= are predefined for string types, as for all one-dimensional discrete array types; these ordering operators correspond to lexicographic order (see 4.5.2).

Examples

(6)

Examples of string objects:

(7)

```
Stars      : String(1 .. 120) := (1 .. 120 => '*' );
Question   : constant String  := "How many characters?";
                                                    --  Question'First =
1, Question'Last = 20
                                                    --  Question'Length =
20 (the number of characters)
```

(8)

```
Ask_Twice  : String  := Question & Question;    --  constrained to
(1..40)
Ninety_Six : constant Roman   := "XCVI";       --  see 3.5.2 and 3.6
```

2.6 String Literals

(1)

A string_literal is formed by a sequence of graphic characters (possibly none) enclosed between two quotation marks used as string brackets. They are used to represent operator_symbols (see 6.1), values of a string type (see 4.2), and array subaggregates (see 4.3.3).

Syntax

(2)

```
string_literal ::= "{string_element}"
```

(3)

```
string_element ::= "" | non_quotation_mark_graphic_character
```

(4)

- A string_element is either a pair of quotation marks (""), or a single graphic_character other than a quotation mark.

Static Semantics

(5)

The sequence of characters of a string_literal is formed from the sequence of string_elements between the bracketing quotation marks, in the given order, with a string_element that is "" becoming a single quotation mark in the sequence of characters, and any other string_element being reproduced in the sequence.

(6)

A null string literal is a string_literal with no string_elements between the quotation marks.

NOTES

(7)

(5) An end of line cannot appear in a string_literal.

Examples

(8)

Examples of string literals:

(9)

```
"Message of the day:"

""                          --  a null string literal
" "   "A"   """"            --  three string literals of length 1

"Characters such as $, %, and } are allowed in string literals"
```