Statements

Basic

Variables

Syntax

```
var_stmt          ::= "var" (attributes identifier ((":" type_expr [var_stmt_asgn]) |  (var_stmt_asgn)) [","]
var_stmt_asgn  ::= ":=" expression
```

Description

Memory locations with specific types are explicitly named as **variables**. When values are assigned to variables, they are stored in the designated memory location for that variable. If a type cannot be converted and the variable is a class type that has an overloaded suitable operator, it will be converted using a custom operator defined by the programmer. Otherwise, an exception TR007 is thrown.

There are two types of variables:

- Variables declared outside of functions and within a module are referred to as **global variables**. They are accessible for the duration of the program's execution and are kept in the global context.
- **Local variables** are defined within a block of code, such as a function, class, or extension. Their stackframe stores their value.

When a variable is needed, it must be declared explicitly. In the absence of a declared variable, no memory is allocated. The interpreter will indicate an error TR005 if a variable identifier (such as a loop variable) is used without first being declared.

Example

```
var a: Integer := 5, b := 'text';
var c: UnicodeString;
```

Conditional

Loops

Functions

Types

Extensions