**function**

*Setting of user function.*

**Syntax:**

**function** <Function name> (<input variable name 1> {:<input variable type 1>} {=<triggering expression 1>} {,**out** <output variable name> {:<output variable type>} {=<triggering expression>}}){:<output value type>}

<operations to be executed on function activation>;

**end**;

**Description:**

Setting of a new user function. The function value can be unassigned. Output variables are declared by **out** word – the function writes down the value in such variable. If any function value type is not specified, then it is defined automatically according to the type of expression where the output function value is assigned. **procedure** can be used in place of **function**. If types of function parameters are not specified, then in the process of creation of the function code the types are accepted as those of operands during a correspondent call.

**Example:**

|  |  |
| --- | --- |
|  | **function** PolyF(A,B)  PolyF = A\*A + B\*B;  **end**; |

In the example, if while calling A and B operands are real numbers, then the result will be a real number too. If A and B are vectors, then the result will be vector too.

The same function in the program text can be defined several times in different ways and also redefined under another name according to the next format:

**function** <function mane>=<predefined function name>

**Example:**

|  |  |
| --- | --- |
|  | **function** SumQ(A,**out** R:integer):complex  *//Function value assignment*  SumQ = sqr(A);  *//Output variable assignment*  R = round(real(SumQ));  **end**;  *//Redefining of SumQ as FSum*  **function** Fsum=SumQ; |

The programming language also allows libraries of functions to be created as text files. This can be done using **include** key word. It allows the program fragment to be loaded from the text file.