**function**

*Задание пользовательской функции.*

**Синтаксис:**

**function**  <Имя функции> (<имя входной переменной 1> {:<тип входной переменной 1>} {=<инициирующее выражение 1>} {,**out** <имя выходной переменной> {:<тип выходной переменной>} {=<инициирующее выражение>}}){:<тип выходного значения>}

<операции выполняемые при вызове функции>;

**end**;

**Описание:**

Задание новой пользовательской функции. Значение функции может быть не присвоено. Выходные переменные декларируются словом **out** – в такую переменную функция записывает значение. Если тип значения функции не указан, то он определяется автоматически по типу выражения, где присваивается выходное значение функции. Вместо **function** можно использовать **procedure**. Если типы параметров функции не указаны, то при создании кода функции типы принимаются как у операндов при соответствующем вызове.

**Пример:**

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

В примере, если при вызове операнды A и B будут действительными числами, то результат тоже будет действительным числом. Если A и B будут векторами, то результат будет вектором.

В тексте программы одна и та же функция может быть по-разному определена несколько раз, а также переопределена под другим именем в соответствии со следующим форматом:

**function** <имя функции>=<имя ранее определённой функции>

**Пример:**

|  |  |
| --- | --- |
|  | **function** SumQ(A,**out** R:integer):complex  *//Присваивание значений функции*  SumQ = sqr(A);  *//Присваивание выходной переменной*  R = round(real(SumQ));  **end**;  *//Переопределение SumQ как FSum*  **function** Fsum=SumQ; |

Язык программирования позволяет создавать также библиотеки функций в виде текстовых файлов. Это может быть сделано при помощи ключевого слова **include**. Оно позволяет загрузить фрагмент программы из текстового файла.