

Углубленный курс информатики

Подпрограммы в PascalABC.NET



Чузов Вячеслав Алексеевич

к.т.н., доцент ОХИ ИШПР

ПОДПРОГРАММЫ

- **Подпрограмма** — это фрагмент кода, который имеет свое имя и создается в случае необходимости выполнять этот код несколько раз.

Причины использования подпрограмм:

1. Улучшение структуры программы, что облегчает ее понимание.
2. Отсутствие необходимости повторять в программе многие фрагменты.
3. Избавление от многих ошибок, которые можно допустить при повторном написании кода программы.

Виды подпрограмм:

1. **Функции** – при выполнении всегда возвращают результат (число, массив, логическую переменную), который может быть использован в выражении;
2. **Процедуры** – не возвращают результат при вызове.

ПОДПРОГРАММЫ

```
begin
  var (x, y) := (ReadInteger, y);

  Оператор1;
  Оператор2;
  Оператор3;

  print(y);

  x := ReadInteger;

  Оператор1;
  Оператор2;
  Оператор3;

  print(y);

  x := ReadInteger;

  Оператор1;
  Оператор2;
  Оператор3;

  print(y);
end.
```



```
Описание подпрограммы
  Оператор1;
  Оператор2;
  Оператор3;

begin
  var (x, y) := (ReadInteger, y);
  Вызов подпрограммы;
  print(y);

  x := ReadInteger;
  Вызов подпрограммы;
  print(y);

  x := ReadInteger;
  Вызов подпрограммы;
  print(y);
end.
```

ОПИСАНИЕ ФУНКЦИЙ

Описание функции имеет следующий вид:

```
function имя (список формальных параметров) : тип возвращаемого значения;  
раздел описаний  
begin  
    операторы  
end;
```

- Операторы подпрограммы, выделенные операторными скобками **begin** / **end**, называются телом этой подпрограммы.
- Список формальных параметров вместе с окружающими скобками может отсутствовать. Он состоит из одной или нескольких секций, разделенных символом ";". Каждая секция состоит из списка переменных, перечисляемых через запятую, после которого следуют двоеточие и тип.
- Раздел описаний функции устроен так же, как и раздел описаний основной программы. Здесь описываются так называемые **локальные** переменные и константы, а также вложенные процедуры и функции. *Все такие локальные объекты доступны лишь внутри данной подпрограммы и не видны извне.*
- Внутри тела любой функции определена специальная переменная с именем **Result**, которая хранит результат вычисления функции. Ее тип совпадает с типом возвращаемого значения функции.

ОПИСАНИЕ ФУНКЦИЙ

```
function Sum(a, b: real): real;
begin
    Result := a + b;
end;

function MinElement(a: array of integer): integer;
begin
    Result := a[0];
    foreach var item in a do
        if item < Result then
            Result := item;
    end;

begin
    var a := Arr(1, 5, 3);
    var a1 := Arr(10, 20, 30);
    var a2 := Arr(2, 15, 45);

    print(MinElement(a) + Sum(2, 3));
    print(MinElement(a1));
    print(Sum(a2[0], a2[2]))
end.
```

ОПИСАНИЕ ПРОЦЕДУР

Описание процедур имеет следующий вид:

```
procedure имя (список формальных параметров);  
раздел описаний  
begin  
    операторы  
end;
```

```
procedure PrintSum(a: array of real);  
begin  
    var sum := a.Sum;  
    println('$'Сумма элементов массива: {sum}');  
end;  
  
begin  
    var arr := ArrRandomReal(10, 0, 10);  
    PrintSum(arr);  
    PrintSum(ArrRandomReal(100, 1, 100))  
end.
```

Сумма элементов массива: 57.6656841755219

Сумма элементов массива: 5340.62982357555

ГЛОБАЛЬНЫЕ И ЛОКАЛЬНЫЕ ПЕРЕМЕННЫЕ

```
function Sum(a, b: real): real;
begin
    var x := a + b; // x - локальная переменная функции Sum
    Result := x;
    x.Println
end;

begin
    var x := 1; // x - Глобальная переменная основной программы
    x.Println;

    var s := Sum(10, 5);
    s.Println;
    x.Println;
end.
```

```
1
15
15
1
```

- В данном примере выполняется вывод значения x внутри функции исключительно для демонстрации смысла локальных параметров подпрограмм.
- Подпрограммы всегда должны выполнять одно действие, иначе их поведение становится неявным и может осложнить работу программы.

Пример 1

Составить подпрограмму-функцию для вычисления данного выражения:

$$k = \lg \left(\frac{a \cdot x^{2.5}}{3 + x} \right) + 2 \cdot 10^3 \cdot (1 + 2 \cdot x)^{(1 - \sqrt{a})}$$

Значения переменных x и a считайте с клавиатуры.

```
function get_k(a, x: real): real;  
begin  
    Result := log10(a * x ** 2.5 / (3 + x)) + 2e3 * (1 + 2 * x) ** (1 - a ** 0.5)  
end;  
  
begin  
    var (x, a) := ReadReal2();  
    get_k(a, x).Println;  
  
    (x, a) := ReadReal2();  
    get_k(a, x).Println  
end.
```

```
1.5 2  
1126.37377625492  
3 2.5  
646.334088554873
```


Пример 2

Описать вычисление переменной **y** по формуле:

$$y = \begin{cases} x^2 + 1 & \text{при } x \leq 3 \\ x^2 + \ln(x) & \text{при } x > 3 \end{cases}$$

Реализовать вычисление переменной **y** в виде подпрограммы функции.
x изменяется в интервале от 1.5 до 3.5 с шагом 0.5.

Пример 2

```
function get_y(x: real): real;
begin
  if x <= 3 then
    Result := sqr(x) + 1
  else
    Result := sqr(x) + ln(x)
  end;

begin
  var (x, h, xk) := (1.5, 0.5, 3.5);
  var y: real;

  for var i := 1 to Trunc((xk - x) / h) + 1 do
    begin
      y := get_y(x);
      Writeln(x:4:1, y:7:2);
      x += h
    end;
  end..
```

```
1.5    3.25
2.0    5.00
2.5    7.25
3.0   10.00
3.5   13.50
```

Задание 1

Составить подпрограмму-функцию для вычисления данного выражения:

$$y = \left(x^6 \cdot (x - 5)^3 \right) / (2 \cdot x + 1)^5$$

$$x = 10$$

Задание 2

Описать вычисление $f(x)$ по формуле, используя подпрограмму-функцию:

$$f(x) = \begin{cases} x^2 & \text{при } -2 \leq x < 2 \\ x^2 + 4x + 5 & \text{при } x \geq 2 \\ 1 & \text{при } x < -2 \end{cases}$$

x изменяется в интервале от -3 до 3 с шагом 1

Задание 3

Составить подпрограмму-функцию для вычисления значения переменной y . Вычисление z и a также выполнить в виде функций.

$$y = \frac{\operatorname{ctg}(x) - \operatorname{ctg}(z)}{\operatorname{ctg}(a) + \operatorname{ctg}(x)} - \operatorname{ctg}(z)$$

$$z = 2 \cdot x^{3/2} + \operatorname{ctg}(x + 2)$$

$$a = 3 \cdot x^{7/9} + \operatorname{ctg}(x + 3)$$

$$x = 2$$

Задание 4

Составить подпрограмму-функцию для вычисления Z . Вычисление a и $f(a)$ выполнить в виде функций.

$$Z = \frac{[f(a)]^2}{\sin(a)} + \frac{1}{a}$$

$$f(a) = \begin{cases} \frac{1}{a}, & \text{при } a < 5 \\ 1 + 2 \cdot a^{-3}, & \text{при } 5 < a \leq 35 \\ 25^{1/\sqrt{a}}, & \text{при } a > 35 \end{cases}$$


$$a = i^2 + \cos\left(-i + \sqrt[3]{(2 \cdot i + 3)}\right)$$


i изменяется от 1 до 10 с шагом 1


КОНТАКТНАЯ ИНФОРМАЦИЯ

ЧУЗЛОВ ВЯЧЕСЛАВ АЛЕКСЕЕВИЧ

к.т.н., доцент ОХИ ИШПР

 Учебный корпус №2, ауд. 136

 +7-962-782-66-15

 chuva@tpu.ru