

L#6

Основы алгоритмизации и программирования.

Введение

Педагогическое образование, 3 семестр

Mayer Svetlana Fyodorovna

Процедуры: определение

- Процедура – это фрагмент кода для выполнения какой-то задачи.
- Процедура – это подпрограмма основной программы.
- Процедура состоит из **заголовка** (имя и параметры) и **тела**.
- Процедура в коде объявляется один раз, но, при этом, может быть вызвана много раз.

Процедура без параметров. Пример

- **Задача.** Печатать звездочку (*) 60 раз, каждую с новой строки. Использовать процедуру без параметров.
- **Правило.** Процедура должна быть объявлена и определена до ключевого слова **begin** основной программы

```
procedure printStars; {заголовок процедуры}
begin {тело процедуры}
  loop 60 do
  begin
    println('*');
  end;
end; {конец процедуры}
begin
  printStars; {вызов процедуры}
end.
```

Процедура с параметрами. Пример

- **Задача.** Давайте решим почти такое же задание, но теперь необходимо распечатывать 60 раз любой символ (введенный с клавиатуры), каждый с новой строки.

```
1  procedure pr(a: char);
2  begin
3      loop 60 do
4          begin
5              println(a);
6          end;
7  end;
8
9  begin
10     writeln('enter character:');
11     var s := readchar;
12     pr(s);
13 end.
```

a - формальный параметр

s - фактический параметр или аргумент

Входные и выходные параметры (параметры-значения и параметры-переменные). Пример

- **Задача.** Написать процедуру для вычисления среднего арифметического значения двух вводимых целых чисел.

```
1  procedure CalcAMean(a, b: integer; var Mean: real);
2  begin
3      Mean := (a + b) / 2;
4  end;
5
6  begin
7      var (x, y) := (3, 5);
8      var Mean: real;
9      CalcAMean(x, y, Mean);
10     Print(Mean);
11     CalcAMean(2 * 2, 8, Mean);
12     Print(Mean);
13 end.
```

a,b - входные параметры
(параметры-значения)

Mean- выходной параметр
(с ключевым словом **var**)
или параметр-переменная

первый вызов процедуры

второй вызов процедуры

Формальные и фактические параметры. Пример

- **Задача.** Написать процедуру для вычисления среднего арифметического значения двух вводимых целых чисел.
- Фактический параметр может быть константой или выражением.

```
1  procedure CalcAMean(a, b: integer; var Mean: real);
2  begin
3      Mean := (a + b) / 2;
4  end;
5
6  begin
7      var (x, y) := (3, 5);
8      var Mean: real;
9      CalcAMean(x, y, Mean);
10     Print (Mean);
11     CalcAMean(2 * 2, 8, Mean);
12     Print (Mean);
13 end.
```

a,b - формальные параметры

Фактические параметры

Передача аргументов по значению и по ссылке

- В PascalABC.NET есть возможность передавать аргумент процедуре по значению или по ссылке.
- **Передача по значению:** Значение фактического параметра копируется в соответствующий формальный параметр. Изменение формального параметра не ведет к изменению фактического параметра

```
procedure p(a: integer);  
begin  
    a := 666; // изменяется только формальный параметр!  
end;  
  
begin  
    var x := 555;  
    p(x);  
    Print(x); // 555  
end.
```

Передача аргументов по значению и по ссылке

- В PascalABC.NET есть возможность передавать аргумент процедуре по значению или по ссылке.
- **Передача по ссылке:** Оба параметра и формальный и фактический ссылаются на одно и то же расположение в памяти, поэтому все изменения в теле процедуры отражаются и на фактическом параметре.
- Изменение формального параметра изменяет и фактический параметр тоже:

```
procedure p(var a: integer);
begin
    a := 666;
end;

begin
    var x := 555;
    p(x);
    Print(x); // 666 – фактический параметр изменился тоже!
end.
```

Передача аргументов по значению и по ссылке

- **Задача.** Создать процедуру обмена значений двух переменных:

```
a = 10  
b = 12  
Result: a=12, b=10
```

Решение 1:

```
1 procedure Swap(var a, b: integer);  
2 begin  
3     var t := a;  
4     a := b;  
5     b := t;  
6 end;  
7  
8 begin  
9     var (x, y) := (3, 5);  
10    Swap(x, y);  
11    println(x, y);  
12 end.
```

Решение 2:

```
1 procedure Swap(var a, b: integer);  
2 begin  
3     (a, b) := (b, a);  
4 end;  
5  
6 begin  
7     var (x, y) := (3, 5);  
8     Swap(x, y);  
9     println(x, y);  
10 end.
```

Короткий способ определения процедуры

- Если тело процедуры состоит только из одного выражения, то можно использовать короткую запись:

```
1  procedure p := Print(1);  
2  
3  begin  
4    p; p; p;  
5  end.
```



Функции

Функции

- Функция – это своего рода процедура, которая возвращает значение для использования в выражении.
- Определение функции отличается от определения процедуры в двух моментах:
 1. Мы должны написать **тип значения**, которое возвращает функция.
 2. Переменная **Result** хранит результат функции.

```
function f(x: real): real;  
begin  
  Result := x * x + 1;  
end;
```

Тип возвращаемого значения

Переменная **Result**

Функции

```
function f(x: real): real;  
begin  
    Result := x * x + 1;  
end;  
begin  
    var x := 5;  
    var r := Sqrt(x) + f(x);  
end.
```



Вызов функции стоит в
выражении

Способы определения функции

- объявление **формальных** параметров (тех, значения которых передаются из основной программы в функцию):

```
function ff( a, b: integer; x: real ): real;
```

- **выходные** параметры, значения которых становятся доступными в основной программе (**возвращаются** в программу):

```
function Max ( var a, b: integer): integer;
```

- **Тип возвращаемого значения** функции добавляется в конце заголовка функции после двоеточия :

```
function Max (a, b: integer): integer;
```

Нахождение максимального из двух чисел с использованием функции

- Для возврата результата в основную программу можно использовать переменную **Result** или **имя функции** :

Решение 1:

```
function Max (a, b: integer): integer;
begin
  if a > b then Max := a
  else         Max := b;
end;

begin
  writeln('Введите два числа');
  read(a, b);
  c := Max ( a, b );
  writeln('Наибольшее число ', c );
end.
```

формальные параметры

результат функции

практические параметры

вызов функции

Нахождение максимального из двух чисел с использованием функции

Решение 2:

```
function max (a,b: integer):integer;  
begin  
  if a > b then Result:=a  
  else Result:=b;  
end;  
begin  
var x:=readinteger;  
var y:=readinteger;  
println( 'maximum =', max(x,y) )  
end.
```

Тип функции

Переменная Result для хранения результата функции

Пример: факториал числа

- В теле функции могут находиться циклы:

```
function Fact(n: integer): integer;  
begin  
    Result := 1;  
    for var i:=1 to n do  
        Result *= i  
end;
```

```
begin  
    println ('5! = ', Fact(5)) // 5! = 120  
end.
```

Короткое определение функции

- Если функция состоит из единственного выражения, мы можем использовать короткое определение: функции

```
function Sq(x: real) := x * x + 1;
```

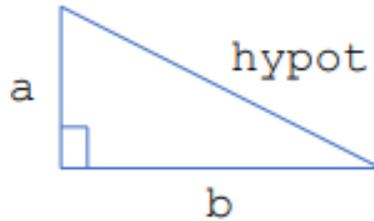
Without Result variable
Without begin-end in a body
Without a type of a result value

```
begin  
  var x := 10;  
  var r := Sq(x) + Sq(2)  
end.
```

Примеры

- Функция для расчета гипотенузы.

```
function Hypot(a,b: real) := Sqrt(a*a + b*b);
```



```
begin  
  loop 5 do  
  begin  
    var (a,b) := ReadlnReal('Enter a,b:');  
    Println(Hypot(a,b))  
  end;  
end.
```

Примеры

- Вызов функции может находиться внутри другой функции:

```
function Hypot (a,b: real) := Sqrt (a*a + b*b);
```



```
function Hypot1 (x1,y1,x2,y2: real) := Hypot (x2-x1,y2-y1);
```

```
begin  
  loop 5 do  
    begin  
      var (x1,y1,x2,y2) := ReadlnReal4;  
      Println (Hypot1 (x1,y1,x2,y2));  
    end;  
end.
```



Q & A