ПРОЦЕДУРЫ И ФУНКЦИИ

Pascal

C

Подпрограмма – группа операторов реализующая законченный алгоритм и оформленная как самостоятельная синтаксическая единица вызываемая по имени.

Модульное программирование – разбиение больших сложных алгоритмов на отдельные подзадачи и реализация подзадач (в том числе, иерархических) в виде подпрограмм.

Подпрограммы могут быть реализованы в виде **Процедур** и в виде **Функций**.

Процедура и **Функция** – независимые части программы, имеющие имя и реализующие определенный алгоритм.
Отличие Процедуры от Функции состоит в том, что Процедура может вычислять и возвращать в основную программу векторный результат (несколько переменных), а Функция только скалярный, при этом вычисленное значение присваивается имени Функции.

В языке С Процедур нет – есть только Функции.

ПРОЦЕДУРЫ И ФУНКЦИИ

Pascal

C

Процедуры

Формат описания Процедуры:

Procedure <имя_процедуры>
[(<имя_формального_параметра1> [:Тип1]; …,
<имя_формального_параметраN> [:ТипN])];
< Раздел описаний>
Begin
  Оператор1; …, ОператорN;
End;

Формат вызова Процедуры:

<Имя_процедуры>
[(фактич_параметр1, …,
фактич_параметрN)];

Пример вызова Процедуры:
Calc (a1, a2, a3);

В языке C \ C++
Процедур нет.

ПРОЦЕДУРЫ И ФУНКЦИИ

Pascal

Процедуры. *Замечания*

~~C~~

- Количество и тип фактических параметров, передаваемых в Процедуру при вызове должны точно соответствовать объявленному количеству и типам формальных параметров,
- Если в объявлении Процедуры перед именем формального параметра стоит var, то фактическим параметром в вызове может быть только переменная, иначе – и переменная и константа,
- Имя процедуры не может использоваться в качестве операнда в выражениях,
- Перед аргументами Процедуры (параметрами) возвращающими результаты в вызывающую программу нужно ставить var,
- При возврате из Процедуры в вызывающую программу управление передается оператору следующему за оператором вызова процедуры,
- Параметры одного типа можно указывать списком в объявлении Процедуры, так же и var,
- Все переменные объявленные внутри Процедуры являются локальными.

ПРОЦЕДУРЫ И ФУНКЦИИ

Pascal

Процедуры

C

Пример процедуры

```
Program Star;

 Procedure StarLine (len: integer);
    (* Выводит строку звездочек.
      len – количество звездочек*)
  var
    z: integer;
  begin
    for z := 1  to len do
       write (‘*’);
  end;

Begin
   StarLine (50);
     writeln (‘Пример вывода строки звездочек’);
   StarLine (50);
End.
```

ПРОЦЕДУРЫ И ФУНКЦИИ

Функции

Pascal

Формат описания Функции:

```
Function <имя_функции>
[(<имя_формального_параметра1
        [:тип1]>; …,
<имя_формального_параметраN
        [:типN])]> :
<тип_результата>;
< Раздел описаний>
Begin
  Оператор1; …, ОператорN;
<имя_функции> :=
<выражение>
End;
```

Формат вызова Функции:

```
<Имя_функции>
[(фактич_параметр1, …,
фактич_параметрN  )];
```

C / C++

Формат описания Функции:

```
[класс] <возвр_тип>
<имя_функции> [(<тип1>
<имя_формального_парамет
- ра1>, …, <типN>
<имя_формаль-
ного_параметраN>)] [throw
(исключения)]
{
<тело_функции >
  return <возвращаемое_значение>;
}
```

где - класс – extern или static – явно задает область видимости функции: глобальная (умолчание) или в пределах модуля;
- исключения – обрабатываемые функцией исключения.

Пример вызова Функции: Pascal - y := cube (a);
C - y = cube (a);

ПРОЦЕДУРЫ И ФУНКЦИИ

Pascal

Функции

Замечания

C / C++

- Количество и тип фактических параметров, передаваемых в Функцию при вызове должны точно соответствовать объявленному количеству и типам формальных параметров,
- Имя Функции обычно используется в качестве операнда в выражениях,
- При возврате из Функции в вызывающую программу управление передается оператору следующему за оператором вызова процедуры,
- Все переменные объявленные внутри Функции являются локальными.

- Параметры одного типа можно указывать списком в объявлении Функции,
- Тип возвращаемого Функцией значения может быть: порядковым, вещественным, указателем,
- В теле Функции её имени хотя бы раз должно быть присвоено значение,
- Если в объявлении Функции перед именем формального параметра стоит var, то фактическим параметром в вызове может быть только переменная, иначе – и переменная и константа.

- Для каждого параметра, передаваемого в функцию указывается его тип и имя (в описании Функции имена можно опускать,
- Тип возвращаемого Функцией значения может быть любым, кроме массива и функции (но может быть указателем на массив или функцию,
- Если Функция не должна возвращать значения указывается тип void, но тогда она не может входить в выражения.

ПРОЦЕДУРЫ И ФУНКЦИИ

Pascal

C / C++

Функции

Примеры функции

```
Program max2;
   var a, b, m :  integer;
function max(a, b : integer) : integer;
(* Функция возвращает максимальное
из двух чисел *)
  begin
    if a > b
      then max := a;
      else max := b;
  end;
Begin
  writeln ('Введите два целые числа ->');
  readln (a,b);
  m := max (a,b);
    writeln ('Максимальное значение
            = ', m);
End.
```

```
#include <STDIO.H>
int max (int a, int b)
{  /* Функция возвращает максимальное из
                  двух чисел */
   if (a > b)
     return(a);
    else
     return (b);
}
main ()
{
 int a, b;
 printf ("Введите  два целые
          числа -> ");
 scanf("%d  %d", &a, &b);
 printf ("Максимальное
   значение -> %d\n", max(a, b));
return 0;
}
```

ПРОЦЕДУРЫ И ФУНКЦИИ

C

C

Практическое занятие

Написать программу, использующую три последовательно выполняемые функции:

- **F1 вычисляет произведение 3-х чисел,**
- **F2 – вычисляет корень квадратный из F1,**
- **F3 – выводит на печать результат F2.**

```
#include <conio.h>
#include <stdio.h>
#include <math.h>
float F1 (float a, float b, float c)                    // int F1(int a, int b, int c)
//  считает произведение трех вещественных чисел
{  float k;      k=a*b*c;   return (k);      }          // { return (a*b*c); }
float  F2 (float  d)
//  вычисляет корень  квадратный числа
{  float j;      j = pow(d,0.5);   return (j);   }      //  {   return (pow(d,0.5));   }
void F3 (float e)
// выводит на экран вещественное число
{ printf ("\nчисло -> %6.3f\n", e); }
main ()
 {
   float a,b,c;   clrscr ();
   printf ("Введите через пробел 3-и вещественных числа и нажмите
                                                Enter\n");
   scanf ("%d %d %d", &a, &b, &c);
 F3(F2(F1(a,b,c)));
  getch ();  return 0;
}
```

ПРОЦЕДУРЫ И ФУНКЦИИ

Pascal

Pascal

Практическое занятие

Написать программу, использующую три последовательно выполняемые функции / процедуры:

- 1-я вычисляет произведение 3-х чисел,
- 2-я – вычисляет корень квадратный из 1-й,
- 3-я – выводит на печать результат 2-й.

```
Program proc_func;
Procedure P1 (x,y,z : real; var product : real);
(* Считает произведение трех вещественных чисел *)
  Begin    product := x*y*z;    End;
Procedure P2 (undersqr : real; var rootsqr : real);
(* Вычисляет квадратный корень *)
  Begin    rootsqr := sqrt (undersqr);    End;
Procedure P3 (Print : real);
(* Выводит на экран вещественное число *)
  Begin WriteLn (' Число -> ', Print:6:3);   End;

(* Головная программа *)
 var
   a,b,c : real;
   Prod,rsqr : real;
Begin
 write ('Введите 3 числа через пробел и
                        нажмите Enter -> ');
     read (a,b,c);
     P1(a,b,c,Prod);
     P2(Prod, rsqr);
     P3(rsqr);
End.
```

```
Program func_3;
 var      m1,m2,m3 : integer;
   function F1(a, b, c : integer) :  integer;
      (* считает произведение трех целых чисел*)
     begin     F1 := a*b*c;   end;
   function F2 (d : real) : real;
       (*  вычисляет корень  квадратный числа   *)
     begin      F2 := sqrt(d);  end;
   function F3 (e : real) :  real;
      (* выводит на экран вещественное число *)
    begin   writeln ('Число -> ', e);    end;

Begin
 writeln ('Введите через пробел 3-и целые числа
                              и нажмите Enter');
  readln (m1, m2, m3);
 F3(F2(F1(m1,m2,m3)));
  readln;
End.
```