

Пример 2. Процедура с параметрами. Даны 3 различных массива целых чисел (размер каждого не превышает 15). В каждом массиве найти сумму элементов и среднеарифметическое значение.

```
program proc;
var i, n, sum: integer;
    sr: real;
procedure work (r:integer; var s:integer; var s1:real); {процедура work}
var mas: array [1..15] of integer; {объявление массива mas}
    j: integer;
begin
    s:=0;
    for j:=1 to r do begin {ввод элементов массива mas}
        write(' Введите элемент ', j, ': ');
        read (mas[j]);
        s:=s+mas[j];
    end;
    s1:=s/r;
end;
begin {главная программа}
    for i:=1 to 3 do begin
        write ('Введите размер массива: ', i);
        readln(n);
        work (n, sum, sr); {вызов процедуры work}
        writeln ('Сумма элементов ', sum);
        writeln ('Среднее арифметическое = ', sr:4:1);
    end;
    readln;
end.
```

Результат работы программы: В программе трижды вызывается процедура work, в которой формальные переменные r, s, s1 заменяются фактическими n, sum, sr. Процедура выполняет ввод элементов массива, вычисляет сумму и среднее значение. Переменные s и s1 возвращаются в главную программу, поэтому перед их описанием ставится служебное слово var. Локальные параметры mas, j действуют только в процедуре. Глобальные - i, n, sum, sr доступны во всей программе.

Функции в Паскале

Набор встроенных функций в языке Паскаль достаточно широк (ABS, SQR, TRUNC и т.д.). Если в программу включается новая, нестандартная функция, то ее необходимо описать в тексте программы, после чего можно обращаться к ней из программы. Обращение к функции осуществляется в правой части оператора присваивания, с указанием имени функции и фактических параметров. Функция может иметь собственные локальные константы, типы, переменные, процедуры и функции. Описание функций в Паскале аналогично описанию процедур. Отличительные особенности функций: - результат выполнения - одно значение, которое присваивается имени функции и передается в основную программу;

- имя функции может входить в выражение как операнд.

Пример 3. Написать подпрограмму-функцию степени a^x , где a, x – любые числа. Воспользуемся формулой:

```

program stepen;
var f, b,d : real; { глобальные параметры}
    function stp (a, x : real) : real;
        var y : real; { локальные параметры}
        begin
            y := exp (x * ln ( a)) ;
            stp:= y; {присвоение имени функции результата вычислений подпр-мы}
        end; { описание функции закончено }
begin {начало основной программы }
    writeln(stp(3,5));    {можем использовать конкретные значения}

    read(f,b);
    d:=stp(f,b);          {или переменные}
    writeln (d);

end.

```

Тоже самое через процедуру

```

•Program1.pas
program stepen;
var f, b, d : real; { глобальные параметры}
    procedure stp (a,x:real; var y: real);

        begin
            y := exp (x * ln ( a))
        end;
begin {начало основной программы }
    read(f,b);
    stp(f,b,d);
    writeln (d);

end.

```

Рекурсия

Процедуры и функции в Паскале могут вызывать сами себя, т.е. обладать свойством рекурсивности. Рекурсивная функция обязательно должна содержать в себе условие окончания рекурсивности, чтобы не вызвать заикливания программы. При каждом рекурсивном вызове создается новое множество локальных переменных. То есть переменные, расположенные вне вызываемой функции, не изменяются.

Пример 4. Составить рекурсивную функцию, вычисляющую факториал числа n следующим образом: $n! = 1$, если $n = 1$ и $n! = (n - 1)! \cdot n$, если $n > 1$

```
var
p,j:integer;
function f ( n : integer): integer;
begin
if n = 1 then
f := 1
else
f := n * f ( n -1 ); { функция f вызывает саму себя}
end;
begin
read(p);
    j:=f(p);
    write(j);
end.
|
```

Задания:

<http://informatics.mccme.ru/mod/statements/view.php?id=11146>