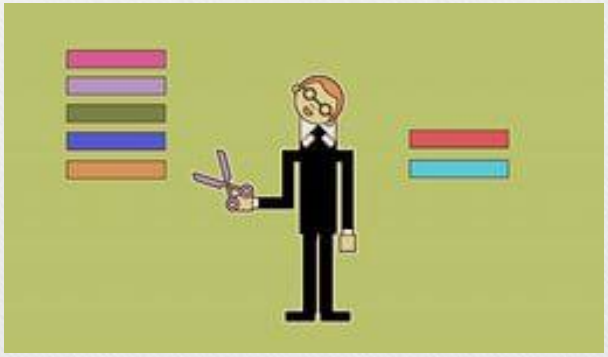
Процедуры и функции в Паскале. Рекурсия

- Подпрограмма - автономная часть программы, выполняющая определенный алгоритм и допускающая обращение к ней из различных частей общей программы. Использование подпрограмм позволяет реализовать один из самых современных методов программирования - структурное программирование.

Подпрограммы

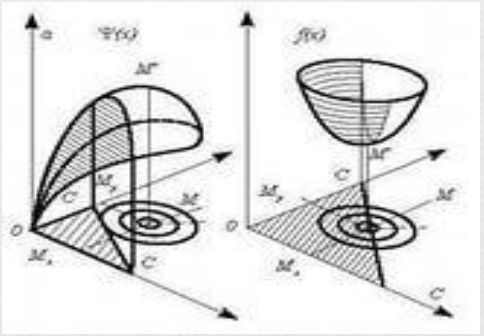
процедура

PROCEDURE



функция

FUNCTION



Процедуры и функции в Паскале объявляются в разделе описания за разделом переменных.

У функций и процедур существуют **параметры** (переменные, которые передают какое - либо значение). Они бывают двух видов:

1) **Формальные** - те, которые находятся в описании подпрограммы

2) **Фактические** - те, которые передаются из основной программы в функцию или процедуру.

Фактические параметры должны соответствовать формальным по количеству, порядку следования и типу.

Также у подпрограммы существую **переменные**. с которыми она в дальнейшем работает. Они делятся опять же на два типа:

1) **Глобальные переменные**, то есть действующие во всей программе

2**) Локальные** - те, которые действуют только в процедуре или функции

Процедуры

Используются в случаях, когда в подпрограмме необходимо получить несколько результатов. Существует два вида процедур:

- с параметром ;
- без параметра.

- Структура процедуры аналогична структуре программы и состоит из заголовка и блока (тела процедуры).

```
procedure ИмяПроцедуры;
var …
begin
…//Тело процедуры
end;
begin
//тело главной программы
end.
```

Процедуры без параметров

```
procedure  pr;
  var      i :  integer ;
  begin
for i :=1 to 60 do write (‘ * ');    writeln;
  end.
```

```
begin
  pr;
end.
```

Данная программа выводит строку из 60 звездочек.

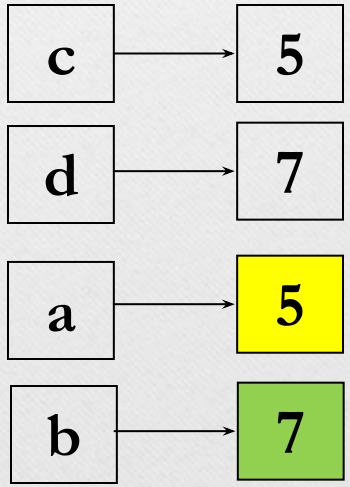
Процедура с параметром.

Составить программу обмена местами двух чисел c=5 и d=7

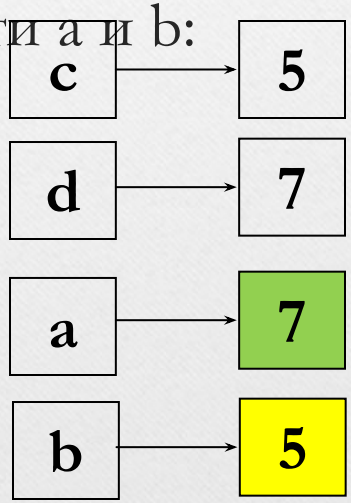
```
program obmenDan;
var c,d:integer;
procedure obmen ( a,b:integer);
var m:integer;
begin
    m:=a; a:=b; b:=m;
writeln(a,b);
end;
begin
        writeln ('Введите 2 числа: ');
         readln(c,d);
         obmen(c,d);
         writeln(c,' ',d);  End.
```

Разбор задачи

1) при вызове процедуры obmen с двумя параметрами 5 и 7, в переменные a и b помещаются тоже числа 5 и 7 соответственно:

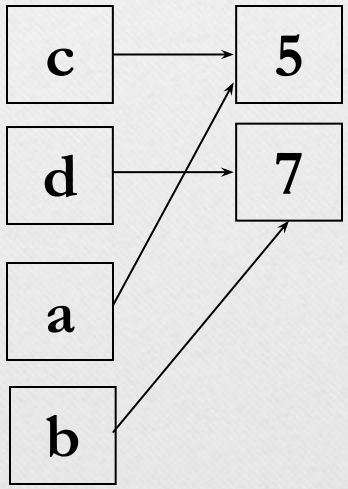


2) далее в процедуре осуществляется перестановка значений ячеек памяти a и b:



- Для того чтобы переменные **c** и **d**, **a** и **b** ссылались на одни и те же ячейки памяти (если изменятся значения **a** и **b**, то изменятся значения и **c, d**) необходимо при описании формальных параметров, перед нужными переменными добавить слово **VAR**:

procedure obmen (**var a,b:integer**);



Функции

- 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 операнд.

Описание функции:

- **function**<*имя*> (<список формальных параметров>): тип;
- {раздел описания **локальных** имён}
- **Begin**
- {раздел выполняемых операторов}
- *<Имя функции>:=<значение>;* {обязательный параметр}
- **End;**

- ***Вызов функции:***
- ***< оператор>:= <имя функции>***
- **(<список фактических переменных>);**

1. В правой части оператора присваивания.
2. В выражении, стоящем в условии оператора разветвления.
3. В процедуре вывода, как результат работы функции.

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 зацикливания программы. При каждом рекурсивном вызове создается новое множество локальных переменных. То есть переменные, расположенные вне вызываемой функции, не изменяются.

Составить рекурсивную функцию, вычисляющую факториал числа **n** следующим образом: **n! = 1 , если n= 1**

n!= (n -1)! · n , если n > 1

```
function   f  ( n : integer): integer;
 begin
      if   n = 1 then  f := 1  else f := n * f  ( n -1 );
      {функция f  вызывает саму себя}
 end;
```

Домашнее задание

1. Дано n целых чисел. Найти среди них число, у которого сумма цифр имеет максимальное значение.