Процедуры

Существуют очень нужные и интересные фрагменты программы! С их помощью тело программы становится небольшим и компактным! Их можно написать ОДИН раз, а вызывать для выполнения очень-очень много раз!

Подпрограммы *– многократно повторяющаяся часть программы, оформленная отдельно от основной программы. Место расположения в программах - Раздел описаний, за описанием переменных.*

Преимущества подпрограмм *- уменьшение объема программы, сокращение времени на отладку.*

Подпрограммы – процедуры и функции

Процедуры и функции.

Процедуры и функции обеспечивают возможность практической реализации принципов структурного программирования.

С их помощью Вы можете разделить свою большую программу на меньшие фрагменты - процедуры или функции.

Они отлаживаются отдельно и независимо, а затем могут использоваться в основном блоке программы или в других процедурах или функциях.

При создании процедур и функций пытаются достичь следующих целей:

1
разделение необъятной общей задачи на несколько меньших по объему и сложности (обозримых) частных задач;

2
уменьшение объема программы за счет многократного применения типовых программных действий, оформленных один раз в виде процедуры или функции;

3
усовершенствование программных модулей - по возможности полученное решение частной задачи обобщается, чтобы его можно было использовать при решении других задач.

Процедуры и функции позволяют включать в основной программный блок дополнительные блоки. Каждое описание процедуры или функции содержит **заголовок**, за которым следует **программный блок**.

Имея один и тот же смысл и аналогичную структуру, процедуры и функции различаются назначением и способом их использования.

Процедура – это самостоятельное какое-то действие!
Что бы вызвать процедуру достаточно воспользоваться оператором вызова процедуры.

Функция нам необходима КАК РЕЗУЛЬТАТ вычисления!
Ее мы используем в выражениях или при выполнении другого оператора, когда ее результат вычисления подставляется в это выражение или оператор.

Процедура

Процедуру можно представить в виде:

```
Procedure ИМЯ ( список формальных
                параметров);
   label; const
   type              Описание локальных меток, типов,
   var                констант и переменных;
   procedure         Описание внутренних процедур и
   function           функций;
begin
        Операторы;
end;
```

Структура такая же как и у программы!

Процедуры состоят из
операторов,
локальных данных
и внутренних процедур и функций.

Процедура

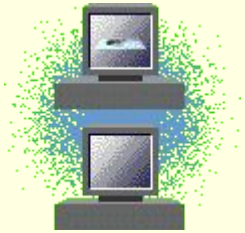
При описании процедур в общем случае необходимо задать три основные компоненты:

1
интерфейс,
то есть информацию, необходимую для ее вызова;

2
локальный контекст:
совокупность описаний (рабочих) объектов, с которыми осуществляются действия;

3
операторы,
составляющие смысл процедуры
(тело)

В заголовке процедуры указывается имя процедуры и описывается список формальных параметров (если он присутствует).



Описание локальных объектов и операторы процедуры составляют внутреннюю ее часть и, как правило, имеют синтаксис блока.

Можно сказать, что заголовок содержит информацию о том, ЧТО делает процедура , а тело процедуры (блок) описывает, КАК она это делает.

Запуск процедуры осуществляется с помощью оператора вызова процедуры, в котором содержатся ИМЯ процедуры и необходимые параметры.

Если в теле процедуры так же есть оператор вызовы процедуры, то процедура будет выполняться рекурсивно.

Принцип работы процедуры:

Выполнение программы начинается с выполнения основной части программы.

Как только появляется необходимость в выполнении процедуры, она вызывается по имени с передачей входных данных.

После выполнения процедура передает в основную часть программы выходных данных (результатов), в то место, откуда была вызвана процедура.

Затем продолжается выполнение основной части программы.

Приведем пример программы, которая выводит на экран строку, состоящую из заданного числа звездочек с использованием процедуры.

```
Uses crt;

Var n: integer;

    Procedure sterline (var len: integer);
       Var i:integer;
    Begin
        For i:=1 to len do Write (‘*’);
    End;

Begin

Write(‘Введите количество звездочек’); Readln(n);
Sterline (n);
Readln;

End.
```