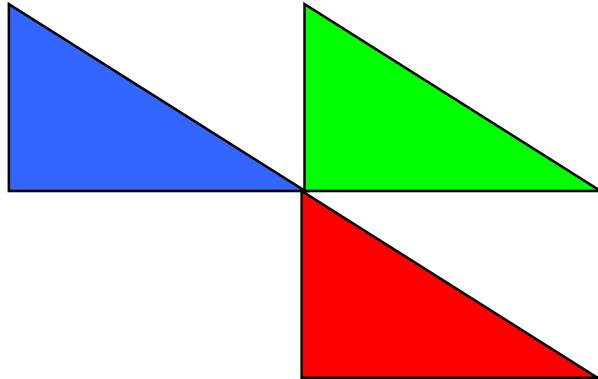


Программирование на языке Паскаль

**Тема. Подпрограммы.
Процедуры**

Процедуры

Задача: Построить фигуру:



Можно ли решить известными методами?

Особенность: Три похожие фигуры.

общее: размеры, угол поворота

отличия: координаты, цвет



Сколько координат надо задать?

Подпрограмма

Подпрограмма – это обособленная, оформленная в виде самостоятельной программной конструкции и снабжённая именем часть программы.

Особенности:

- Записывается **однократно** и может **вызываться из соответствующих мест программы по имени**.
- Использование подпрограмм позволяет **сократить объем** программы, улучшить её **наглядность**, облегчает процесс **отладки**.

В Паскале механизм подпрограмм реализуется в виде **процедур** и **функций**, которые вводятся в программе с помощью своего описания, но их структура такая же, как и структура программы. Процедуры и функции различаются назначением и способом их использования.

Процедуры

Процедура – это независимая именованная часть программы, которую можно вызвать по имени для выполнения некоторой последовательности действий.

Любая процедура начинается с **заголовка**, который является её обязательной частью (в отличие от заголовка программы).

Процедуры

Он состоит из служебного слова **Procedure**, за которым следует имя процедуры, а в круглых скобках – список формальных параметров, в конце ставим точку с запятой. После заголовка могут идти те же разделы, что и в программе.

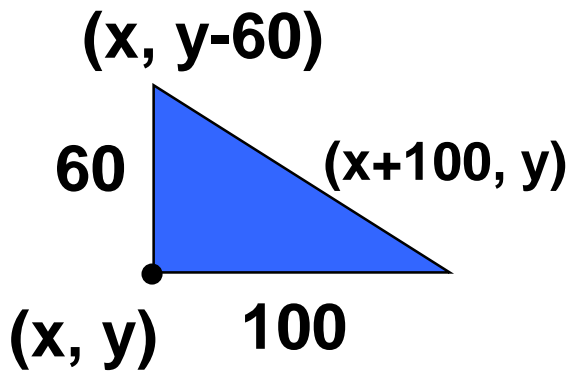
Общий вид процедуры:

```
Procedure    <ИМЯ>    (<список    формальных  
параметров>);  
<раздел описаний>;  
Begin  
<тело процедуры>;  
End;
```

Процедуры

Порядок разработки:

- выделить одинаковые или похожие действия (три фигуры)
- найти в них **общее** (размеры, форма, угол поворота) и **отличия** (координаты, цвет)
- отличия записать в виде неизвестных переменных, они будут параметрами процедуры



заголовок

параметры

```
procedure Tr ( х,у:integer; с: color );
begin
```

цвет

```
  MoveTo (х, у);
  LineTo (х, у-60);
  LineTo (х+100, у);
  LineTo (х, у);
  FloodFill (х+20, у-20, с);
```

координаты

тело
процедуры

```
end;
```

Процедуры

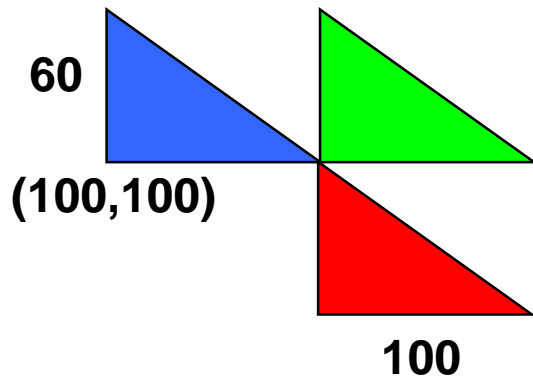
Процедура не может выполняться сама, её необходимо вызвать по имени и указать **фактические параметры** того же типа, что и **формальные**.

Количество и тип формальных параметров равны количеству и типу фактических параметров.

При вызове процедуры её формальные параметры заменяются фактическими в порядке их следования.

Результат выполнения процедуры – это одно или несколько значений. Они передаются в основную программу как значения её параметров.

Программа



ВЫЗОВЫ
процедуры

формальные параметры

```
program qq;  
  procedure Tr( x, y: integer; c: color)  
  begin  
    ...  
  end;  
begin  
  Tr (100, 100, clBlue) ;  
  Tr (200, 100, clgreen) ;  
  Tr (200, 160, clred) ;  
end.
```

процедура

фактические параметры

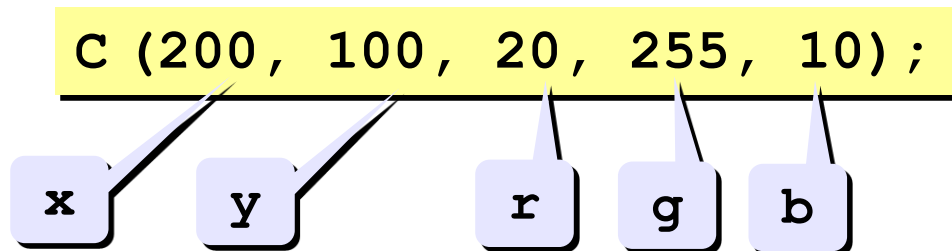
Процедуры

Особенности:

- все процедуры расположены **выше** основной программы
- в заголовке процедуры перечисляются **формальные** параметры, они обозначаются именами, поскольку могут меняться

```
procedure C( x, y, r, g, b: integer);
```

- при вызове процедуры в скобках указывают **фактические** параметры (числа или арифметические выражения) **в том же порядке**



Процедуры

Особенности:

- для каждого формального параметра после двоеточия указывают его тип

```
procedure A (x: real; y: integer; z: real);
```

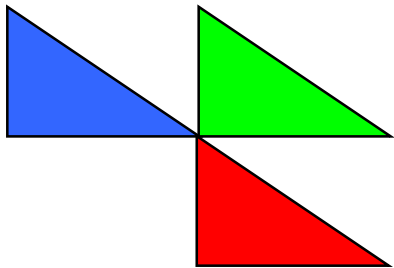
- если однотипные параметры стоят рядом, их перечисляют через запятую

```
procedure A (x, z: real; y, k, l: integer);
```

- внутри процедуры параметры используются так же, как и переменные

Задание

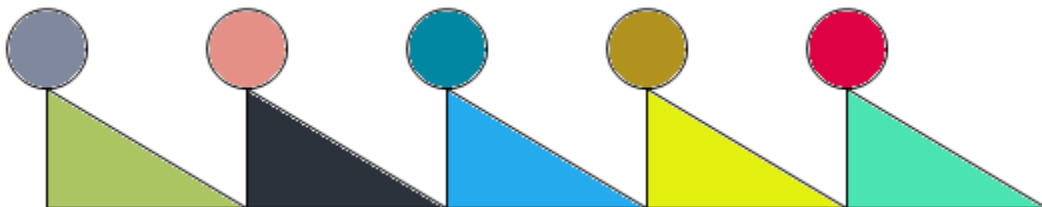
Задача «А»: Построить фигуру:



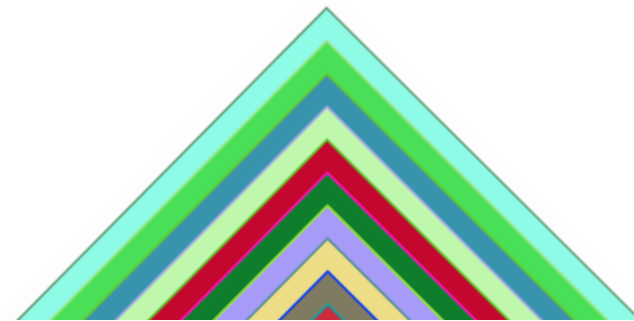
Задача «В»: Построить фигуру, используя процедуру из задачи «А»:



Задача «С»: Построить фигуру, используя 2 процедуры:



Задача «D»:



Задача «Е»:

