

Подпрограммы

Подпрограммы

Любая программа может состоять из *основной программы* и произвольного числа *подпрограмм*, включенных в основную программу.

Подпрограмма – это самостоятельная программная единица (блок), оформленная по определенным правилам.

В виде подпрограмм могут оформляться повторяющиеся фрагменты программы, либо не повторяющиеся, но представляющие собой логически целостные вычислительные блоки.

Обращение к подпрограмме

В тексте основной программы вместо фрагмента, помещенного в подпрограмму, оформляется обращение к этой подпрограмме.

После вызова подпрограммы осуществляется ее выполнение, а затем происходит возврат в основную программу и продолжается выполнение основной программы с оператора, следующего за командой обращения к подпрограмме.

Обращение к подпрограмме осуществляется по ее **имени** с указанием параметров, связывающих основную программу с

Структура подпрограммы. Подпрограмма состоит из *заголовка* со списком параметров и *тела* подпрограммы, в котором реализуется алгоритм выполнения подпрограммы. Внутри подпрограммы могут определяться *локальные* переменные, которые используются только в теле подпрограммы.

Переменные, объявленные в основной программе, являются *глобальными* и могут использоваться в любой точке программы, в том числе и в подпрограмме.

Процедуры и функции. Подпрограммы бывают двух видов – *процедуры* и *функции*. Отличаются они тем, что процедура просто выполняет группу операторов, а функция *вдобавок* возвращает (передает) вычисленное значение в основную программу. Поэтому функция имеет определенный

тип

Параметры подпрограммы

Чтобы подпрограмма могла выполняться, ей надо получить данные из основной программы. Данные передаются через списки параметров или аргументов.

Различают *формальные* и *фактические* параметры.

Формальные параметры указываются в списке заголовка подпрограммы. Они нужны только для описания тела подпрограммы.

Фактические параметры (конкретные значения) указываются при обращении к подпрограмме.

Формальные параметры должны быть согласованы с фактическими по *количеству, типу и порядку следования*.

При выполнении подпрограммы каждый формальный параметр заменяется на соответствующий ему фактический параметр.

Примеры использования подпрограмм-функций

Задание 1. Вычислить периметр P треугольника $\triangle ABC$, заданного на плоскости координатами его вершин: $A(x_A, y_A)$, $B(x_B, y_B)$, $C(x_C, y_C)$.

Решение

Периметр треугольника вычисляется по формуле:

$$P = AB + BC + CA, \text{ где}$$

AB, BC, CA – длины сторон треугольника.

Вычисление длины одной стороны треугольника оформим подпрограммой-функцией, используя формулу вычисления расстояния между двумя точками.

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

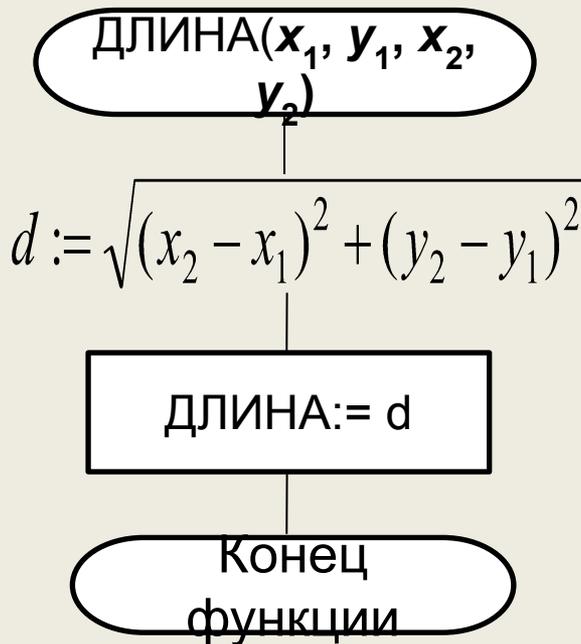
Подпрограмме-функции зададим имя **ДЛИНА**.

В список формальных параметров включим имена

переменных x_1, x_2, y_1, y_2 .

Оформление подпрограммы-функции

Блок-схема



Псевдокод

Функция ДЛИНА (x_1, y_1, x_2, y_2)

$$d := \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

ДЛИНА := d

Конец функции

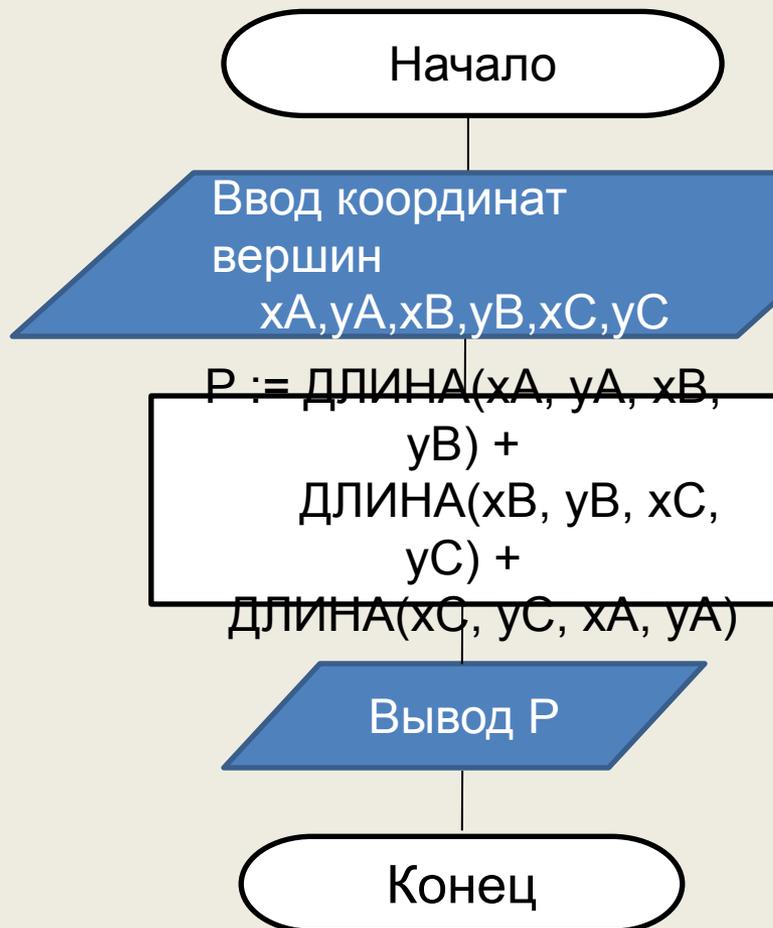
Турбо-Паскаль

```
Function Dlina (x1, y1, x2, y2: real):real;  
var  
  d: real;  
begin  
  d:=sqrt(sqr(x2 - x1) + sqr(y2 - y1));  
  Dlina:= d;  
end;
```

Обращение к подпрограмме-функции

Обращение к функции осуществляется из основной программы и включается в состав выражения в качестве операнда.

Блок-схема основной программы



Переменные $x_A, y_A, x_B, y_B, x_C, y_C$ являются **фактическими параметрами** - аргументами функции.

Для вычисления длин сторон треугольника АВ, ВС и СА надо **три раза** обратиться к подпрограмме-функции ДЛИНА, передав ей значения соответствующих фактических аргументов.

Программирование основной программы с использованием функции

Псевдокод

Начало программы

Функция ДЛИНА (x_1, y_1, x_2, y_2)

$$d := \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

ДЛИНА := d

Конец функции

Ввод координат вершин $x_A, y_A,$
 x_B, y_B, x_C, y_C

$P :=$ ДЛИНА(x_A, y_A, x_B, y_B) +
ДЛИНА(x_B, y_B, x_C, y_C) +
ДЛИНА(x_C, y_C, x_A, y_A)

Вывод P

Конец программы

Программирование основной программы с использованием функции

Турбо-Паскаль

```
Program PP_F;  
  Var  
    xA, yA, xB, yB, xC, yC, P: real;  
  
  Function Dlina (x1, y1, x2, y2: real):real;  
    var  
      d: real;  
    begin  
      d:=sqrt(sqr(x2 - x1) + sqr(y2 - y1));  
      Dlina:= d;  
    end;  
  
  Begin  
    writeln (' Введите координаты вершин');  
    read (xA, yA, xB, yB, xC, yC);  
    P := Dlina(xA,yA,xB,yB)+  
          Dlina (xB,yB,xC,yC)+  
          Dlina (xC,yC,xA,yA);  
    writeln ('Периметр P=', P:6:2);  
  End.
```

Задание 2. Начинаящим программистом была разработана следующая подпрограмма-функция **Graphic**.

```
Pascal  
function Graphic(X,N:integer):real;  
begin  
  if N=0 then Graphic:=0;  
  if N=1 then  
    begin Graphic:=sin(x) - 1;  
  end else  
    begin Graphic:=x*x + 1;  
  end;  
  if N=2 then  
    begin Graphic:=x*x*x - 1;  
  end else  
    begin Graphic:=cos(x) + 1;  
  end;  
end;
```

Переменная **Analiz** в основной программе после выполнения оператора присваивания:

**Analiz :=
Graphic(x,0) –
Graphic(x,1) +
Graphic(x,2);**

при **x=0** примет значение.... **?**

Analiz = -1