

M. C. E. S. C. H. E. R.

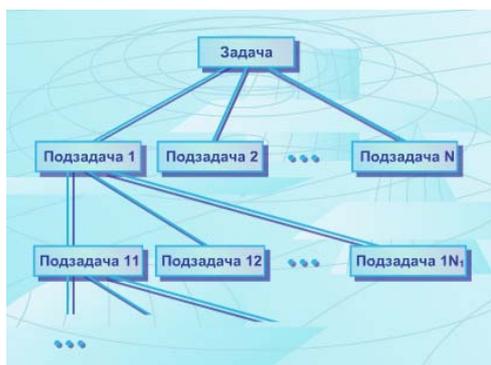
*M.C.E.S.C.H.E.R.*

## Подпрограммы

### Цель презентации

- Показать, как использовать структурный подход в программировании – разбиение сложной задачи на более простые.
- Узнать, как для этого можно использовать подпрограммы-процедуры.

При создании программы для решения более или менее сложной задачи выполняется разделение этой задачи на подзадачи, этих подзадач – на более мелкие подзадачи и так далее до тех пор, пока подзадачи не станут легко программируемыми.



Для такой организации используются **подпрограммы**.

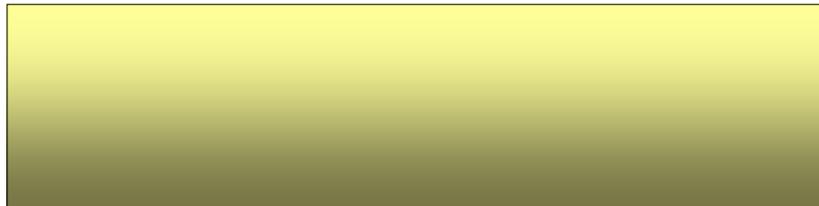
В программах часто приходится повторять некоторые аналогичные действия многократно.

Используя подпрограммы, можно единожды описать действия в подпрограмме, а затем лишь только вызывать ее.

Такой принцип дефрагментации программы называется **нисходящим программированием** и соответствует принципам структурного программирования, в основу которого и положено понятие подпрограммы.

**В Паскале используются подпрограммы двух видов:  
процедуры и функции.**

Описание функции имеет следующий вид:

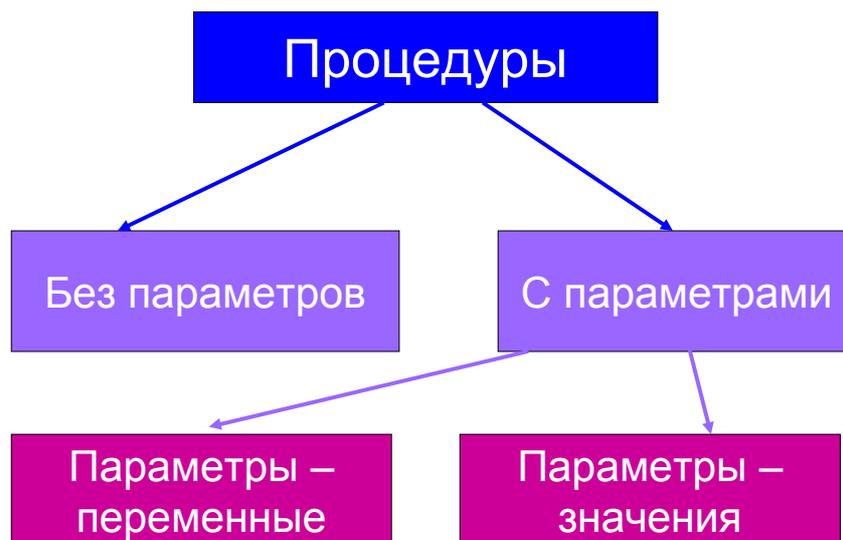


Описание процедуры имеет следующий вид:



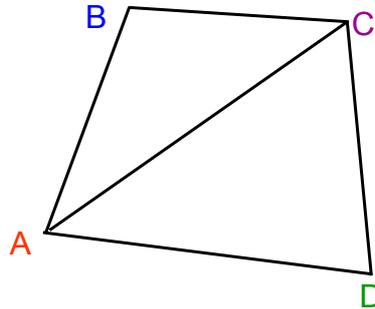
## Процедуры

(1)



### Задача

Составить программу для вычисления площади выпуклого 4-угольника, заданного длинами его сторон и диагональю.



Диагональ делит 4-угольник на два 3-угольника, к которым применима формула Герона:

$$s = \sqrt{p(p-a)(p-b)(p-c)},$$

$a, b, c$  – длины сторон  $\Delta$ ,

$$p = \frac{a + b + c}{2}$$

```
uses CRT;
var AB, BC, CD, DA, AC, S1, S2, S, a, b, c, p: real;
Procedure Ploshad1;
begin
  p:=(a+b+c)/2;
  s:=sqrt(p*(p-a)*(p-b)*(p-c));
end;
begin
  Clrscr;
  Writeln('Задайте стороны 4-х угольника ABCD и его диагональ AC');
  readln (AB, BC, CD, DA, AC);
  a:=AB; b:=BC; c:=AC;
  Ploshad1;
  S1:=s;
  a:=DA; b:=AC; c:=CD;
  Ploshad1;
  S2:=s;
  Writeln ('Площадь ABCD= ', S1+S2:8:2);
  readln;
end.
```

Связь процедуры Ploshad1 с остальными операторами происходит с помощью переменных a, b, c и s. Переменная p используется только внутри процедуры. Ее можно и описать в этой процедуре.

2 обращения к процедуре

Команды присваивания, задающие значения a, b, c перед каждым вызовом процедуры

Команды присваивания для сохранения результатов

Переменные, которые описаны и используются внутри процедуры, называются – **локальными**.

Переменные, которые описаны в основной программе, называются – **глобальными**.

Локальные переменные действуют только внутри данной процедуры.

Глобальные переменные действуют во всей программе и в каждой ее процедуре.

```
{глобальные переменные}
var AB, BC, CD, DA, AC, S1, S2, S, a, b, c: real;
Procedure Ploshad1;
  var p:real;          {локальная переменная}
  begin
    p:=(a+b+c)/2;
    s:=sqrt(p*(p-a)*(p-b)*(p-c));
  end;
begin
  Writeln('Задайте стороны 4-х угольника ABCD и его
диагональ AC');
  readln (AB, BC, CD, DA, AC);
  a:=AB; b:=BC; c:=AC;
  Ploshad1;
  S1:=s;
  a:=DA; b:=AC; c:=CD;
  Ploshad1;
  S2:=s;
  Writeln ('Площадь ABCD= ', S1+S2:8:2);
  readln;
end.
```

**Использование процедур без параметров не удобно** из-за большого количества операторов присваивания, которые до обращения к процедуре определяют значения переменных a, d с.

**Процедуры с параметрами очень удобны.** Они позволяют при вызове процедуры напрямую указывать переменные или константы для определения значений переменных a, b, c (тех, которые используются в тексте процедуры).

## **Процедуры с параметрами-переменными**

**Procedure <ИМЯ> (Список параметров);**

**Procedure Ploshad2 (var a, b, c, s: real);**

В основной программе процедура вызывается тоже с указанием параметров:

**<ИМЯ> (Список параметров);**

**Ploshad2 (AB, BC, AC, S);**

**Список параметров**  
предназначен для обмена информацией между  
вызывающей программой и вызываемой  
подпрограммой.

**Параметры:**  
• формальные  
• фактические

**Формальные параметры** – это параметры, которые указаны в скобках после имени процедуры в разделе описания процедуры и служат для описания порядка действий в процедуре.

**Фактические параметры** – это параметры, которые указываются в скобках при вызове процедуры. Они заменяют формальные параметры при выполнении программы.

```
uses CRT;  
var AB, BC, CD, DA, AC, S1, S2: real;  
Procedure Ploshad2(var a, b, c, s: real);  
  var p: real;  
  begin  
    p:=(a+b+c)/2;  
    s:=sqrt(p*(p-a)*(p-b)*(p-c));  
  end;  
begin  
Clrscr;  
Writeln('Задайте стороны 4-угольника ABCD и его диагональ AC');  
readln (AB, BC, CD, DA, AC);  
Ploshad2(AB, BC, AC, S1);  
Ploshad2(DA, AC, CD, S2);  
Writeln ('Площадь ABCD= ', S1+S2:8:2);  
readln;  
end.
```

Формальные параметры

Фактические параметры

## Процедуры с параметрами-значениями

Procedure <ИМЯ> (Список параметров);  
Procedure Ploshad3 (a, b, c: real; var s:real);

Параметры-значения

Параметры-переменные

В основной программе процедура вызывается  
аналогично:

**Ploshad3 (AB, BC, AC, S);**

**Параметры-значения** используются для задания начальных значений в процедуре,

**Параметры-переменные** для вывода результата.

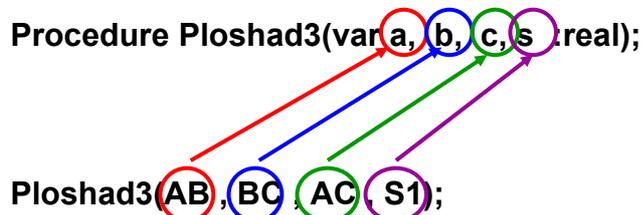
```
uses CRT;  
var AB, BC, CD, DA, AC, S1, S2: real;  
Procedure Ploshad3( a, b, c: real; var s:real);  
  var p:real;  
  begin  
    p:=(a+b+c)/2;  
    s:=sqrt(p*(p-a)*(p-b)*(p-c));  
  end;  
begin  
Clrscr;  
  Writeln('Задайте стороны 4-х угольника ABCD и его  
диагональ AC');  
  readln (AB, BC, CD, DA, AC);  
  Ploshad3(AB,BC,AC,S1);  
  Ploshad3(DA,AC,CD,S2);  
  Writeln ('Площадь ABCD= ', S1+S2:8:2);  
  readln; end.
```

## Правила вызова процедур с параметрами

1. Соответствие между параметрами команды вызова процедуры и параметрами описания процедуры устанавливается не по именам параметров, а по порядку следования:

Первый параметр в команде вызова соответствует первому параметру в описании процедуры, второй – второму и т.д. → число фактических параметров должно соответствовать числу формальных параметров.

```
Procedure Ploshad3(var a, b, c, s :real);  
  
Ploshad3(AB, BC, AC, S1);
```



## Правила вызова процедур с параметрами

2. Типы фактических параметров должны соответствовать типам формальных параметров.

3. Фактические параметры, соответствующие результатам вычислений в процедуре, должны быть только именами переменных.

```
Procedure Ploshad3 (a, b, c: real; var s :real);
```

```
Ploshad3 (AB, 3, 6-2.7, S1);
```

4. Фактические параметры, соответствующие начальным значениям для процедуры, могут быть именами переменных, константами и выражениями соответствующего типа.