

# *Подпрограммы в Turbo Pascal*

- **Подпрограмма** – это именованная логически законченная группа команд, которую можно вызывать для выполнения любое количество раз из разных мест программы.

# Причины использования подпрограмм

- упростить разработку больших программ методом декомпозиции (разделения) задачи на несколько подзадач;
- большая наглядность программы;
- экономия памяти.

# Виды подпрограмм

```
graph TD; A[Виды подпрограмм] --> B[процедуры]; A --> C[функции]
```

процедуры

функции

- **Процедура** – это независимая именованная часть программы, предназначенная для выполнения конкретных действий.

# Процедуры без параметров

- *Формат записи:*

```
procedure <имя процедуры>;  
    begin  
        <тело процедуры>;  
    end;
```

- Все переменные, которые используются в процедурах без параметров, описываются в основной программе (в модуле Var).

```
Program cylinder;  
  Const PI=3.14;  
  Var R, H, V: real;
```

```
    Procedure Input;      { процедура ввода данных }
```

```
    Begin
```

```
      Writeln ('введите значение радиуса');
```

```
      Readln (R);
```

```
      writeln ('введите значение высоты');
```

```
      readln (H);
```

```
    end;
```

```
    Procedure formula;   { процедура вычисления объёма }
```

```
    begin
```

```
      V: =PI*sqr(R)*H;
```

```
    end;
```

```
    Procedure Output;   { процедура вывода результатов }
```

```
    begin
```

```
      writeln ('V=',V);
```

```
    end;
```

```
BEGIN
```

```
  Input;
```

```
  formula;
```

```
  Output;
```

```
END.
```

# Пример.

Составить  
программу  
нахождения  
объёма  
цилиндра.

# Процедуры с параметрами

- В процедурах можно описывать константы, переменные, другие процедуры.
- Раздел описания в процедурах имеет такую же структуру, как и в основной программе.
- **Локальные переменные** – это переменные, описанные внутри процедуры.
- Локальные переменные недоступны за пределами процедуры.
- Изменения, происходящие с локальными переменными внутри процедуры, не влияют на значения переменных с такими же именами, но описанными вне данной процедуры.

# Процедуры с параметрами

- **Глобальные переменные** – переменные, которые используются в процедуре, но описаны вне её.
- Изменения, происходящие с глобальными переменными внутри процедуры, изменяют значения этих переменных вне данной процедуры.



# Пример.

```
Program zadacha;  
    Var a, b: integer;  
Procedure lokal;  
    Var a, x: char;  
begin  
    a:=!';  
    x:=':';  
    b:=b+1;  
end;  
BEGIN  
    a:=0;  
    b:=100;  
    lokal;  
    writeln ('a =', a);  
    writeln ('b=', b);  
END.
```

## *Для процедуры lokal:*

*переменная x* – локальная переменная  
(программа не может изменить её значение)

*переменная b* – глобальная переменная  
(все изменения значения этой переменной в процедуре  
сохраняются и после выхода из процедуры)

*переменная a* в основной программе – целого типа,  
а в процедуре – символьного типа. Переменная a  
целого типа недоступна в процедуре lokal.

Результат выполнения программы: a=0; b=101.

# Передача параметров в Turbo Pascal

## 1. Передача параметров по значению

- **Параметры-значения** – переменные, описанные после имени процедуры в круглых скобках. Перед ними отсутствует служебное слово `Var`.
- **Формат записи:**

```
procedure <имя процедуры> (<имя переменной>:<тип  
                             переменной>);  
  
    begin  
        <тело процедуры>;  
  
    end;
```

- Переменные, описанные в заголовке процедуры, наз. **формальными параметрами**.
- Переменные или константы, которые описаны в команде процедуры при её вызове, наз. **фактическими параметрами**.
- При обращении к процедуре с параметрами все формальные параметры заменяются фактическими в порядке их перечисления.

# Пример.

```
Program parametr;
```

```
  Var m, n: integer;
```

```
  Procedure summa (a, b: integer);
```

```
    Var S: integer;
```

```
    Begin
```

```
      S:=0;
```

```
      S:=a+b;
```

```
      writeln ('S=',S);
```

```
    end;
```

```
BEGIN
```

```
  m:=100;
```

```
  n:=10;
```

```
  summa (m,n);  или  summa (100,10);
```

```
END.
```

*Переменные a и b являются формальными параметрами, а переменные m и n - фактическими. Значения фактических параметров m=100 и n=10 передаются формальным параметрам a и b.*

*Изменения фактических параметров происходит только внутри процедуры и не влияют на них вне данной процедуры.*

# Передача параметров в Turbo Pascal

## 2. Передача параметров по имени

- **Параметры-переменные** – переменные, описанные после имени процедуры в круглых скобках и перед которыми записано служебное слово *Var*.

- **Формат записи:**

```
procedure <имя> (<имя переменной>:<тип  
                переменной>; Var <имя  
                переменной>:<тип переменной>);
```

```
begin
```

```
    <тело процедуры>;
```

```
end;
```

- При выполнении процедуры формальные параметры-переменные заменяются фактическими параметрами.
- Любые изменения формального параметра-переменной приводят к изменению значения фактического параметра-переменной.
- С помощью параметров-переменных в основную программу передаются результаты действий команд над данными.

## Пример.

Даны две  
тройки чисел:  
a1,b1,c1 и  
a2,b2,c2.

Найти  
значение  
суммы:  $S = \min$   
(a1, b1, c1) + min  
(a2, b2, c2)

Program primer;

Var a1,b1,c1,a2,b2,c2, min1, min2, S: real;

Procedure minimum (a,b,c: real; Var min: real);

Begin

min:=a;

if min > b then min:= b;

if min > c then min:= c;

end;

BEGIN

writeln ('введите три числа');

readln (a1, b1, c1);

writeln ('введите три числа');

readln (a2,b2,c2);

minimum (a1,b1,c1, min1);

minimum (a2,b2,c2, min2);

S:= min1 + min 2;

writeln ('S=', S);

END.

**Функция** – это подпрограмма, результатом работы которой является некоторое значение.

- **Формат записи:**

```
function <имя функции> (<имена параметров>:  
    <тип параметров>): <тип результата>;  
    begin  
        <тело функции>;  
    end;
```

- В теле функции имени функции необходимо присвоить результат её выполнения.
- При вызове функции её имя с указанием списка фактических параметров должно быть включено в выражение как операнд.



Program vyrazenie;

Var x, y: real;

function modul (a: real) : real;

Begin

if a < 0 then a:= -a;

modul:=a;

end;

BEGIN

writeln ('введите значение переменной');

readln (x);

y:= modul (x-3) + modul (x+6);

writeln ('y=', y);

END.

**Пример.**

Вычислить

значение

выражения:

$y = |x-3| +$

$|x+6|$