

# Язык программирования Паскаль (Pascal)



## КОМПЛЕКС ОПОРНЫХ КОНСПЕКТОВ И ПРАКТИЧЕСКИХ ЗАНЯТИЙ

г. Новокузнецк,  
МБНОУ «Гимназия № 59»  
Дымович Е.Е.

# Содержание



- Структура программы (опорный конспект: типы данных, процедуры ввода-вывода)
- Процедуры ввода-вывода (пример учебной программы)
- Ветвление (опорный конспект)
- Циклы с условием (опорный конспект)
- Циклы с пред- и постусловием (пример учебной программы)

# Опорный конспект

## Структура программы

```
Program имя_программы ;  
Uses Crt ;  
Const ... ; {раздел констант}  
Var ... ; {раздел переменных}  
BEGIN  
  Clrscr;  
  ... {Операторы –  
        команды основной программы}  
  Readkey;  
END.
```

## Процедура ввода (с подсказкой)

```
Write('Zadaite a: ');  
Readln(a);
```

## / ВЫВОДА

```
Writeln('Perimetr= ', p);
```

## Типы данных

Целый: **Integer**  
Вещественный: **Real**

## Объявление переменных

```
Var x, y: Integer ;  
    k, z: Real ;
```

## Оператор присваивания

имя\_переменной := <выражение>;  
Пример: **p:=2\*(a+b);**

## Работа с экраном

**ClrScr** – очистка текущего окна  
**GotoXY(20, 4)**– перемещение курсора в позицию: 20<sup>ый</sup> столбец [1÷80] и 4<sup>ая</sup> строка [1÷25] экрана.

# Язык программирования Паскаль (Pascal)



**ПРИМЕР УЧЕБНОЙ ПРОГРАММЫ  
«ПРОЦЕДУРЫ ВВОДА-ВЫВОДА.  
РАБОТА С ЭКРАНОМ»**

Составить программу вычисления среднего значения.  
Цифры для текста подсказок задать const, значения заданий задать с клавиатуры. Вывод оформить в виде:

Лабораторная работа № 1

Ученика(цы) 9 класса ...

**Фамилия**

**Имя**

Задание 1 (баллов): \_

Задание 2 (баллов): \_

Задание 3 (баллов): \_

Средний балл: ...

# Математическое описание



## {Описание входных переменных}

**r** – номер работы (const)

**k1** – номер класса (const)

**b1** – балл 1 задания

**b2** – балл 2 задания

**b3** – балл 3 задания

## {Формулы}

$$sr = (b1+b2+b3) / 3$$

## {Описание выходных переменных}

**sr** – среднее арифметическое баллов

# Пример программы (без оформления)

```
Program Lr1;  
  Uses Crt;  
  Const  r = 1; kl = 9;  
  Var   b1, b2, b3: Integer;  
        sr: Real;  
Begin  
  Clrscr;  
  Writeln ('Лабораторная работа № ', r);  
  Writeln ('Ученика ', kl, ' класса ...');  
  Writeln ('Фамилия');  
  Writeln ('Имя');  
  Write ('Задание 1 (баллов): '); Readln(b1);  
  Write ('Задание 2 (баллов): '); Readln(b2);  
  Write ('Задание 3 (баллов): '); Readln(b3);  
  sr := (b1+b2+b3) / 3;  
  Writeln ('Средний балл: ', sr:0:2);  
Readkey;  
End.
```

**Переписать текст программы в тетрадь**

# Пример программы (с оформлением)

```
Program Lr1;  
  Uses Crt;  
  Const  r = 1; kl = 9;  
  Var   b1, b2, b3: Integer;  
        sr: Real;  
  Begin  
    Clrscr;  
    GotoXY(30,2); Writeln ('Лабораторная работа № ', r);  
    GotoXY(30,4); Writeln ('Ученика ', kl, ' класса ...');  
    GotoXY(35,6); Writeln ('Фамилия');  
    GotoXY(45,8); Writeln ('Имя');  
    GotoXY(15, 11); Write ('Задание 1 (баллов): '); Readln(b1);  
    GotoXY(15, 13); Write ('Задание 2 (баллов): '); Readln(b2);  
    GotoXY(15, 15); Write ('Задание 3 (баллов): '); Readln(b3);  
    sr := (b1+b2+b3) / 3;  
    GotoXY(20, 22); Writeln ('Средний балл: ', sr:0:2);  
  Readkey;  
  End.
```

**Набрать программу в среде Pascal**



**Задача:** Напишите программу, запрашивающую высоту дома  $h$  (в метрах), и **вычисляющую время падения кирпича  $t$**  (в секундах) с крыши этого дома по формуле:

$$t = \sqrt{\frac{2 \cdot h}{g}}$$

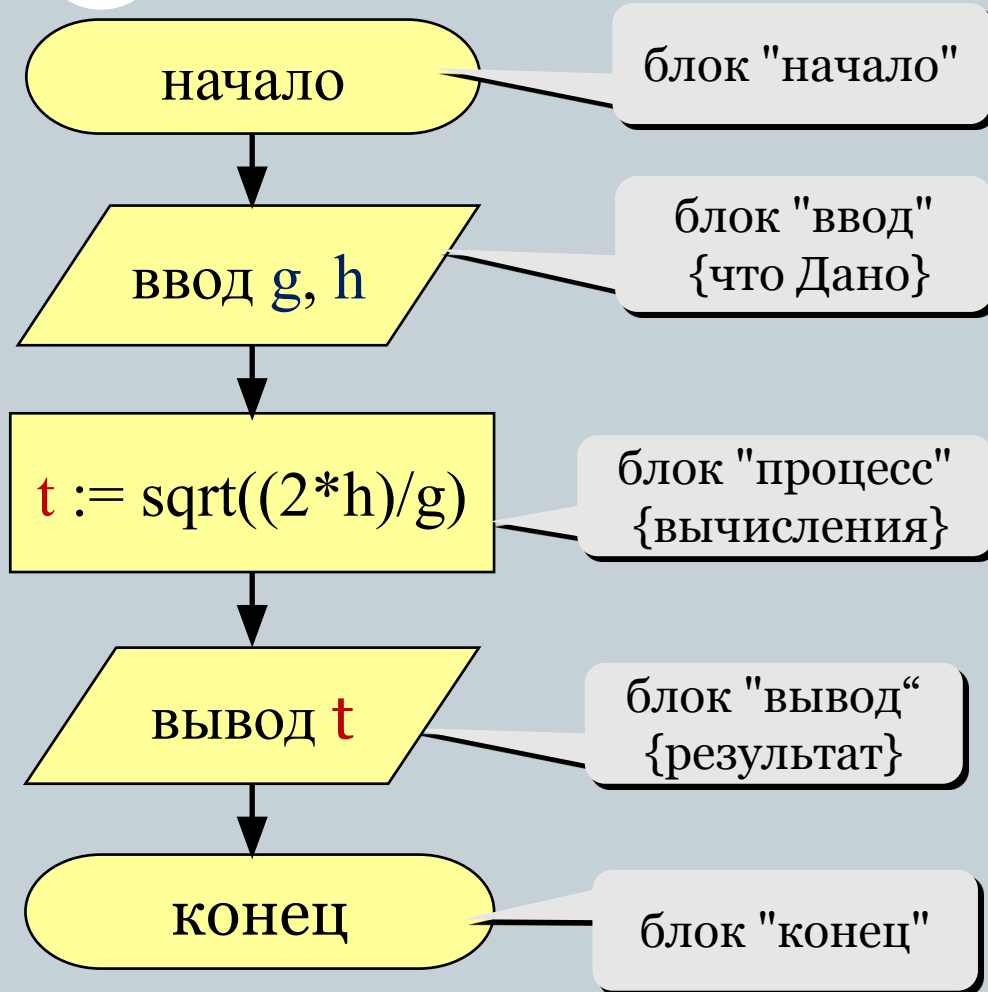
**$g=9,81$  задать const**

## Программа

```
program Lr2;
uses Crt;
const g=9.81;
var h,t: real;
Begin
  clrscr;
  Write( 'Zadaite visoty doma' );
  Readln (h);
  t := Sqrt(2*h/g)....;
  Writeln ( 'Kirpich padaet ', t, ' c' );
  readkey;
End.
```

Вместо многоточия впишите необходимые имена переменных и формулу вычисления

## Блок-схема



**Блок-схему и текст программы переписать в тетрадь**

СОДЕРЖАНИЕ

# Язык программирования Паскаль (Pascal)



**ОПОРНЫЙ КОНСПЕКТ  
«ВЕТВЛЕНИЕ»**

# Ветвление



**Ветвление** – процесс, который реализуется по одному из **двух** направлений, в зависимости от **выполнения определенного условия**.

## Условный оператор

Различают два вида оператора:

- **полный условный оператор;**
- **неполный условный оператор.**

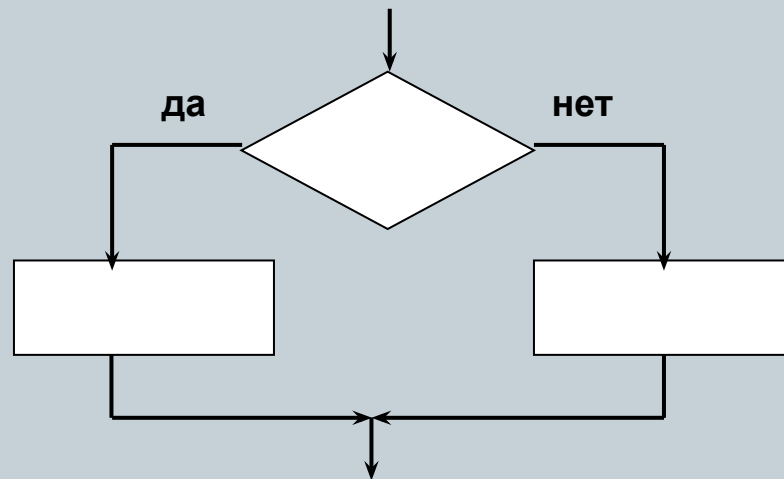
## Выбор

# Ветвление

Формат неполного условного перехода:



Представление в виде блок-схемы:



Формат оператора:

линейная форма

**IF условие Then оператор1 Else оператор2 ;**

Пример записи:

Если заданное число положительно, то удвоить его, иначе увеличить в пять раз.

**If a>0 then s:=a\*2 else s:=a\*5;**  
(если) (тогда) (иначе)

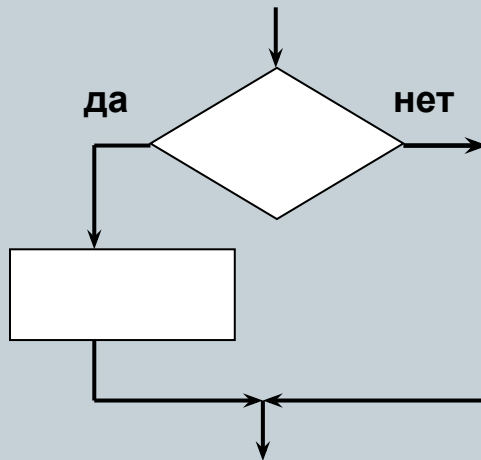
блочная форма

**IF условие Then**  
**begin**  
**y:=sqr(a);** } операторы  
**Writeln (y);** } ;  
**end**  
**Else begin**  
**операторы ;**  
**end ;**

# Ветвление

Формат ПОЛНОГО УСЛОВНОГО оператора:

Представление в виде блок-схемы:



Формат оператора:

**IF** условие **Then** оператор ; {выполняется только истинность условия}

# Ветвление

Для записи условий используются следующие знаки

отношений:

=	– равно;	<>	– неравно (возможен вариант ><);
>	– больше;	>=	– больше или равно;
<	– меньше;	<=	– меньше или равно.

Помимо простых условий можно в программе использовать и **составные** (сложные) условия.

Для их записи используются связки:

**AND** – и;

**OR** – или;

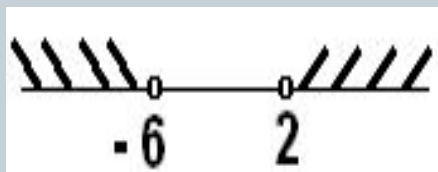
**NOT** – нет.



условие  $-6 < x \leq 2$  в математике - можно записать:

$-6 < x$  **и**  $x \leq 2$ ;

на языке Pascal: **IF**  $(-6 < x)$  **and**  $(x \leq 2)$  **Then ...**



в математике :  $x < -6$  **или**  $x > 2$ ;

на языке Pascal: **IF**  $(x < -6)$  **or**  $(x > 2)$  **Then ...**

# Выбор

Для ситуаций, где имеется несколько (три и более) альтернатив



Формат оператора выбора:

```
Case переменная of  
Значение 1: оператор1 ;  
Значение 2: оператор2 ;  
...  
Else оператор4;  
end ;
```

Пример записи:

Задать число (от 1 до 4-х). По порядковому номеру числа, определить , какое время года. (1-зима; 2-весна; 3-лето; 4-осень).

```
Case x of  
1: writeln ('Зима') ;  
2: writeln ('Весна') ;  
3: writeln ('Лето') ;  
4: writeln ('Осень') ;  
end;
```

# Язык программирования Паскаль (Pascal)



**ОПОРНЫЙ КОНСПЕКТ  
«ЦИКЛЫ С УСЛОВИЕМ»**



# Цикл с предусловием



```
while условие do begin  
  {тело цикла}  
end;
```

- **МОЖНО ИСПОЛЬЗОВАТЬ СЛОЖНЫЕ УСЛОВИЯ:**

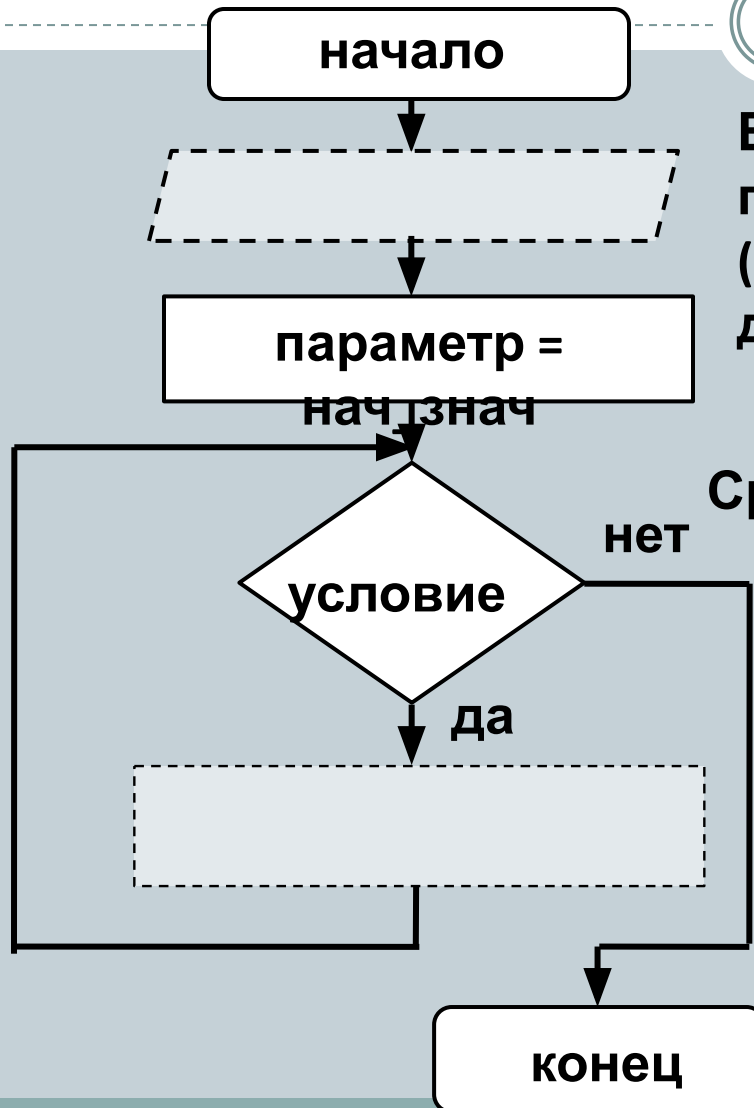
```
while (a<b) and (b<c) do begin  
  {тело цикла}  
end;
```

- **если в теле цикла только один оператор, слова begin и end можно не писать:**

```
while a < b do a := a + 1;
```

# Цикл с предусловием

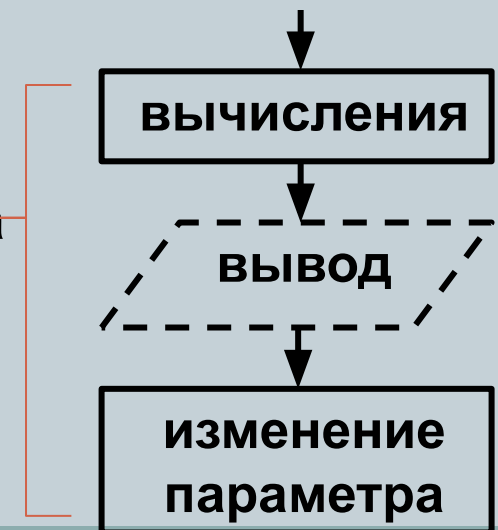
Блок-схема



Ввод неизменяющихся переменных (при наличии в исходных данных)

Сравнение параметра с конечным значением

Тело цикла



# Цикл с предусловием



## Особенности:

- условие пересчитывается каждый раз при входе в цикл
- если условие на входе в цикл ложно, цикл не выполняется ни разу

```
a := 4; b := 6;  
while a > b do a := a - b;
```

- если условие никогда не станет ложным, программа зацикливается

```
a := 4; b := 6;  
while a < b do d := a + b;
```

# Сколько раз выполняется цикл, определите значение $a$ или $b$ ?

```
a := 4; b := 6;  
while a < b do a := a + 1;
```

**2 раза**  
 $a = 6$

```
a := 4; b := 6;  
while a < b do a := a + b;
```

**1 раз**  
 $a = 10$

```
a := 4; b := 6;  
while a > b do a := a + 1;
```

**0 раз**  
 $a = 4$

```
a := 4; b := 6;  
while a < b do b := a - b;
```

**1 раз**  
 $b = -2$

```
a := 4; b := 6;  
while a < b do a := a - 1;
```

**зацикливание**

# Замена for на while

```
for i:=1 to 10 do  
  begin  
    {тело цикла}  
  end;
```

```
for i:=a downto b do  
  begin  
    {тело цикла}  
  end;
```

```
i:= 1;  
while i <= 10 do begin  
  {тело цикла}  
  i:= i + 1;  
end;
```

```
i:= a;  
while i >= b do begin  
  {тело цикла}  
  i:= i - 1;  
end;
```

Замена цикла **for** на **while** возможна **всегда**.

Замена **while** на **for** возможна только тогда, когда можно заранее **рассчитать число шагов цикла**.

# Цикл с постусловием

**Цикл с постусловием** – это цикл, в котором проверка условия выполняется в конце цикла.

```
repeat
    {тело цикла}
until условие;
```

## Особенности:

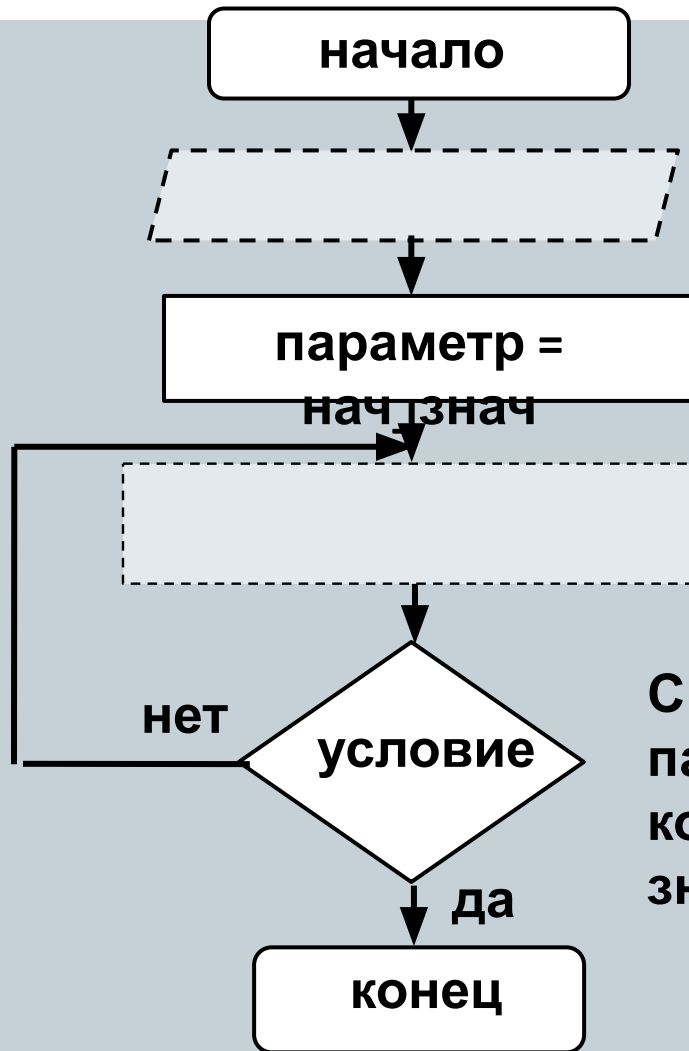
- тело цикла всегда выполняется хотя бы один раз
- после слова `until` ("до тех пор, пока не...") ставится условие **ВЫХОДА** из цикла

```
repeat
    writeln('Введите положительное число');
    read(n);
until n > 0;
```

условие ВЫХОДА

# Цикл с постусловием

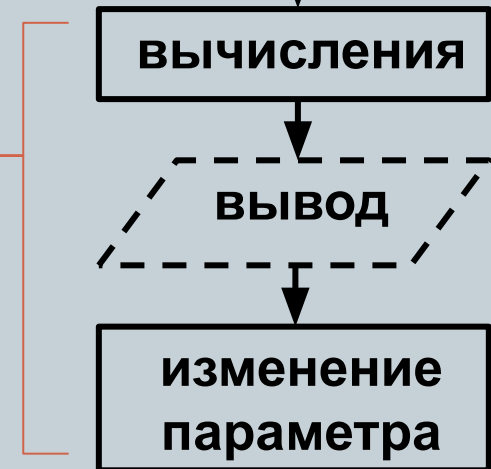
## Блок-схема



Ввод неизменяющихся переменных (при наличии в исходных данных)

Тело цикла

Сравнение параметра с конечным значением



# Сколько раз выполняется цикл, определите значение $a$ или $b$ ?

```
a := 4; b := 6;  
repeat a := a + 1; until a > b;
```

3 раза  
 $a = 7$

```
a := 4; b := 6;  
repeat a := a + b; until a > b;
```

1 раз  
 $a = 10$

```
a := 4; b := 6;  
repeat a := a + b; until a < b;
```

заикливание

```
a := 4; b := 6;  
repeat b := a - b; until a < b;
```

2 раза  
 $b = 6$

```
a := 4; b := 6;  
repeat a := a + 2; until a < b;
```

заикливание



# Язык программирования Паскаль (Pascal)



**ПРИМЕР УЧЕБНОЙ ПРОГРАММЫ  
«ЦИКЛЫ С ПРЕД- И ПОСТУСЛОВИЕМ»**

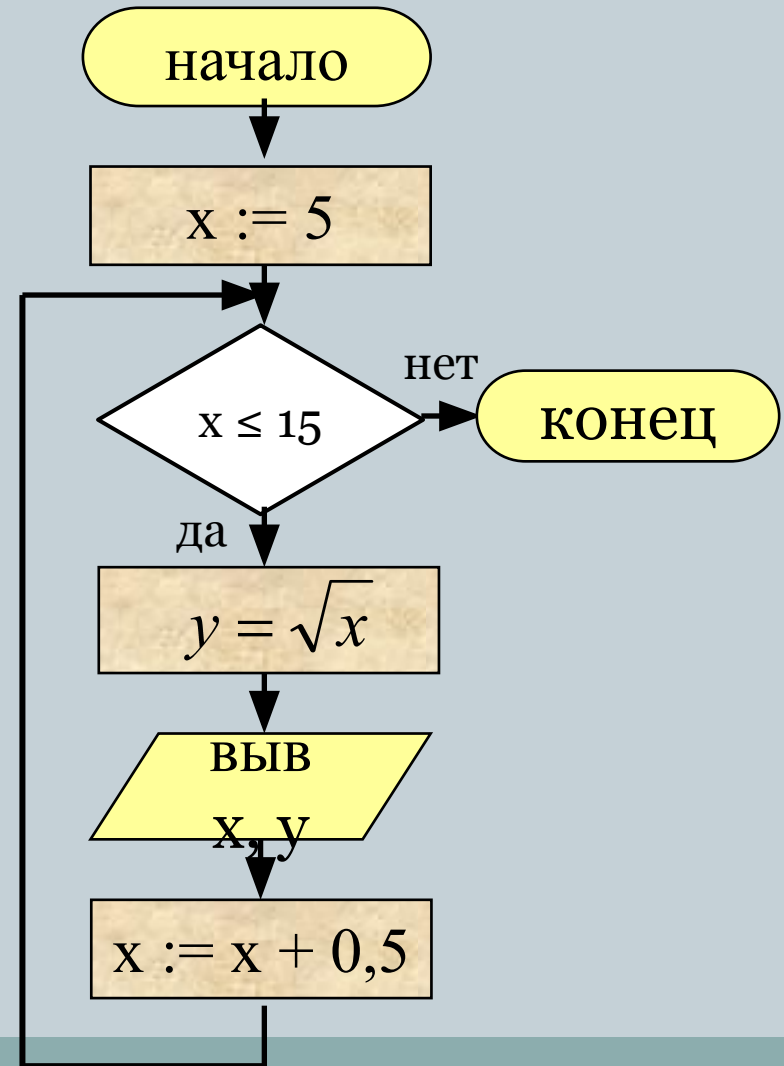
Вычислить значение :  $y = \sqrt{x}$  , если  $x$  изменяется от 5 до 15 с шагом 0,5.

Цикл с предусловием

Блок-схема

## Программа

```
program Lr9;  
uses Crt;  
var x, y: real;  
Begin  
clrscr;  
x:= 5;  
While x<=15 do  
begin  
y = sqrt(x);  
Writeln ('Pri x= ', x:0:1, ' y= ', y:0:3);  
x:= x + 0.5;  
end;  
readkey;  
End.
```



Вычислить значение :  $y = \sqrt{x}$  , если  $x$  изменяется от 5 до 15 с шагом 0,5.

Цикл с постусловием

Блок-схема

## Программа

```
program Lr10;  
  uses Crt;  
  var x, y: real;  
Begin  
  clrscr;  
  x := 5;  
  Repeat  
    y = sqrt(x);  
    Writeln ('Pri x= ', x:0:1, ' y=', y:0:3);  
    x := x + 0.5;  
  Until x > 15;  
  readkey;  
End.
```

