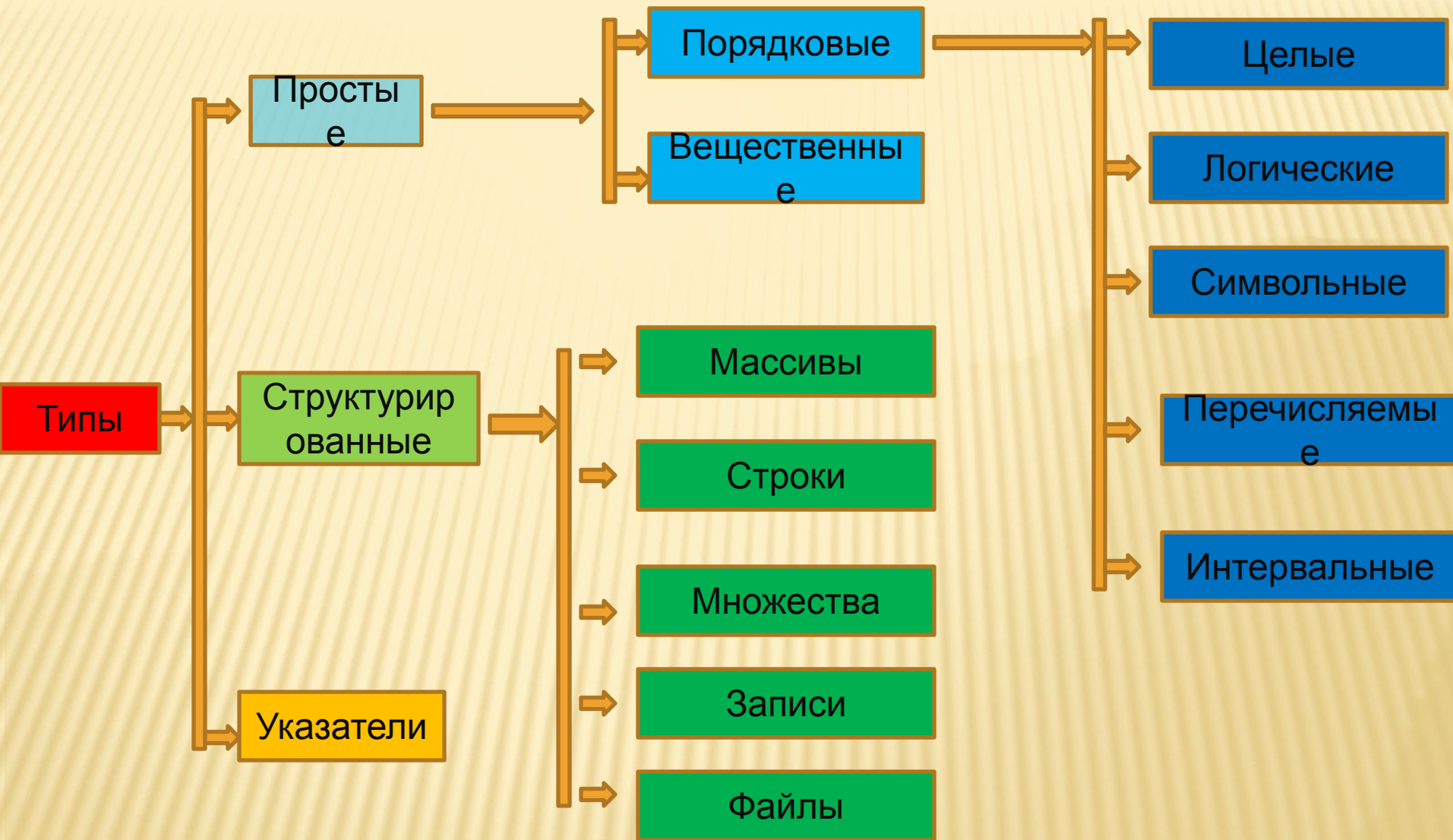


# Управляющие конструкции языка, условный оператор

Учитель информатики  
МОБУ СОШ №1 с.  
Бижбуляк  
Миронов Л В

# Графический способ описания алгоритма (программы)

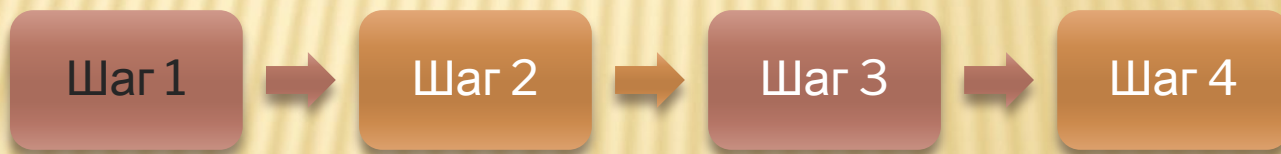
Наименование символа	обозначение	функции
Процесс		Выполнение операций или группы операций, в результате которых изменяются значения, форма представления или расположения данных
Решение		Решение (блок ветвлений) выбор направления решения.
Ввод-вывод		Ввод-вывод данных
Пуск-остановка		Начало или конец алгоритма
Цикл		Организация цикла
Соединение		Указание связи между прерванными линиями



Идентификатор	Длина (байт)	Диапазон значений
<b>Целые типы</b>		
<b>integer</b>	2	-32768..32767
<b>byte</b>	1	0..255
<b>word</b>	2	0..65535
<b>shortint</b>	1	-128..127
<b>longint</b>	4	-2147483648..2147483647
<b>Вещественные типы</b>		
<b>real</b>	6	$2,9 \times 10^{-39} - 1,7 \times 10^{38}$
<b>single</b>	4	$1,5 \times 10^{-45} - 3,4 \times 10^{38}$
<b>double</b>	8	$5 \times 10^{-324} - 1,7 \times 10^{308}$
<b>extended</b>	10	$3,4 \times 10^{-4932} - 1,1 \times 10^{4932}$
<b>Логический тип</b>		
<b>boolean</b>	1	true, false
<b>Символьный тип</b>		
<b>char</b>	1	все символы кода ASCII

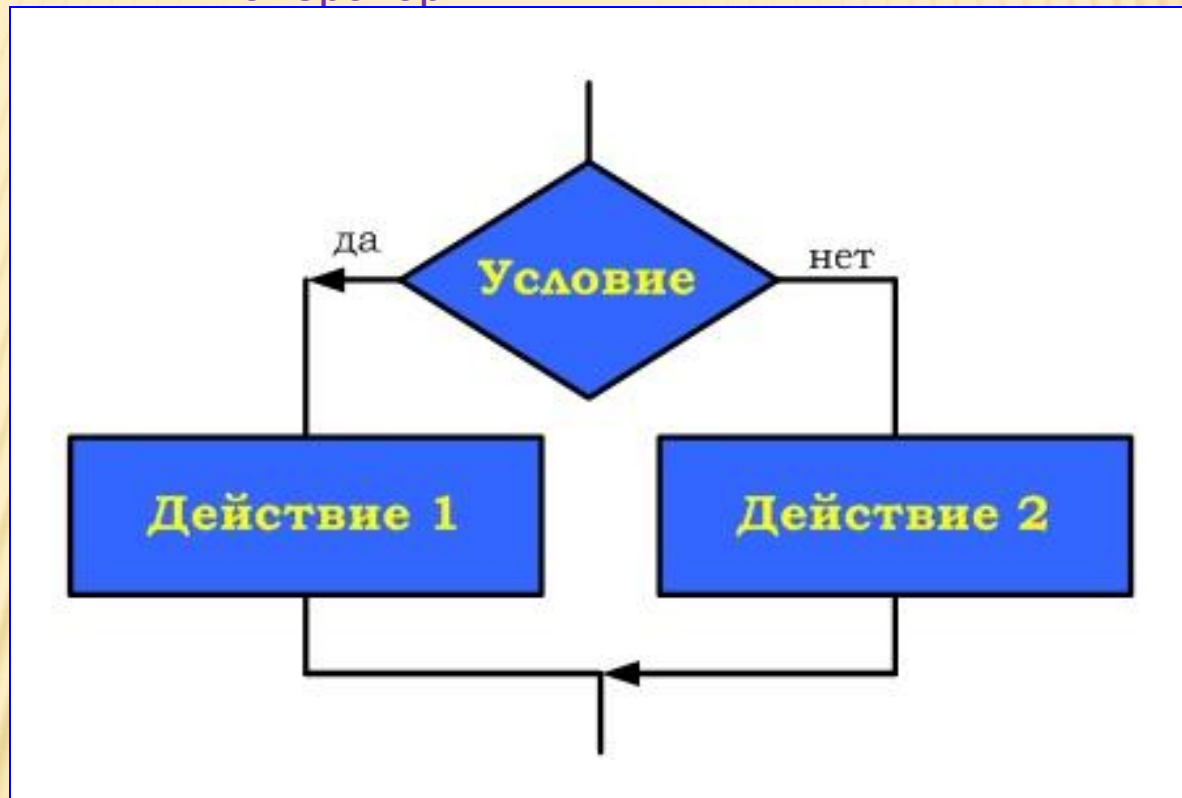
# ЛИНЕЙНЫЙ АЛГОРИТМ

Все действия в программе выполняются последовательно одно за другим



# Управляющие конструкции

Ветвление. Условный  
оператор



**Условный оператор** – реализует алгоритмическую конструкцию "развилка" (ветвление) и изменяет порядок выполнения операторов в зависимости от истинности или ложности некоторого условия.

Существует два варианта оператора:

Полная форма:

```
if <условие> then<оператор 1> else<оператор  
2>;
```

и краткая форма:

```
if <условие> then<оператор 1>;
```

**Если (IF)** условие истинно, **тогда (THEN)** выполняется оператор1, **иначе (ELSE)** выполняется оператор2.

**Составной оператор** – представляет собой группу операторов ограниченных так называемыми операторными скобками Begin и End. В этом случае данная группа операторов, состоящая из произвольного количества любых операторов, воспринимается компилятором как один оператор.

Begin

Write ('Текущее сообщение');

z:=l;

d:=d+l;

End;

```
if x>=0 then
```

```
begin
```

```
    write ('Контрольное число
```

```
положительно');
```

```
    z:=z+l;
```

```
    d:=d-1;
```

```
    x:=x+10;
```

```
end
```

```
else
```

```
begin
```

```
    write ('Контрольное число  
отрицательно');
```

```
    z:=z-l;
```

```
    d:=d+l;
```

```
    x:=x-10;
```

```
end.
```



# Программа для решения квадратного уравнения.

$$ax^2 + bx + c = 0$$

$$D = b^2 - 4ac$$

$D > 0$

$$x_1 = \frac{-b + \sqrt{D}}{2a}$$

$$x_2 = \frac{-b - \sqrt{D}}{2a}$$

$D = 0$

$$x = \frac{-b}{2a}$$

$D < 0$  уравнение корней не имеет

```
Program Kvor;  
Var  
a, b, c: real;  
x1, x2: real;  
D: real;  
Begin  
  Writeln('Решение квадратного уравнения');  
  Write('введите коэффициент a:', a);  
  Readln(a);  
  Write('введите коэффициент b:', b);  
  Readln(b);  
  Write('введите коэффициент c:', c);  
  Readln(c);  
  D:=b*b-4*a*c;  
  If D>0 then  
  begin  
    x1:=(-b+sqrt(D))/(2*a);  
    x2:=(-b-sqrt(D))/(2*a);  
    Writeln('Первый корень уравнения:', x1);  
    Writeln('Второй корень уравнения:', x2);  
  end;  
  If D=0 then  
  begin  
    x1:=-b/(2*a);  
    Writeln('Уравнение имеет один корень:', x1);  
  end;  
  If D<0 then  
  Writeln('Уравнение корней не имеет');  
  Readln;  
End.
```

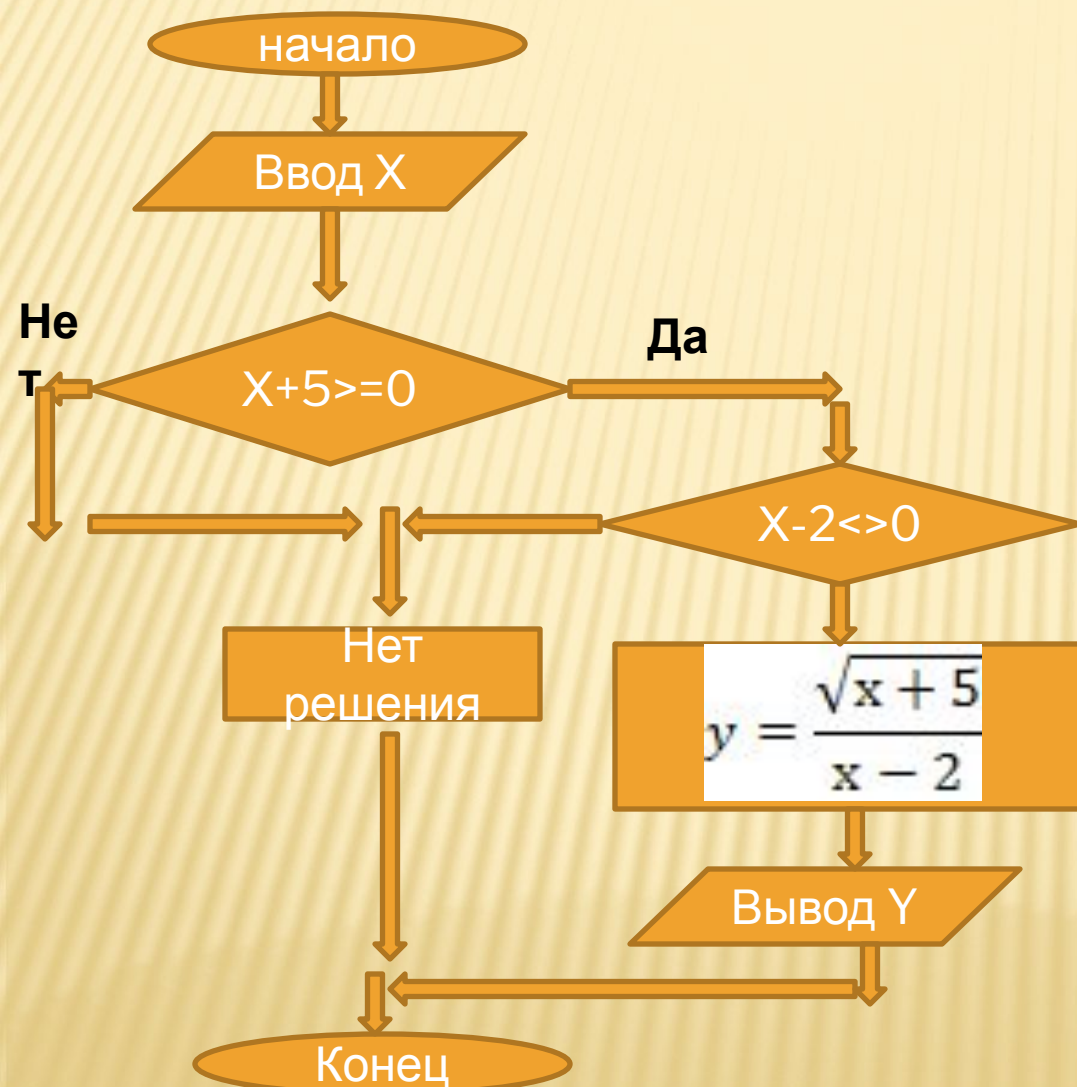
Решить  
уравнение:

$$y = \frac{\sqrt{x+5}}{x-2}$$

Условие:

$$x+5 \geq 0$$

$$x-2 \neq 0$$



```
program trt;  
var x,y:real;  
Begin  
read(x);  
if x+5>=0 then  
if x-2<>0 then  
begin  
y:=sqrt(x+5)/(x-2);  
write(y);  
end  
else writeln('нет  
решения')  
else writeln('нет  
решения')  
End.
```



Домашнее  
задание.

1. Решить уравнение

$$y = \frac{x+2}{\sqrt{x+5}}$$

2. Выучить теорию