

Урок №27



Программирование разветвляющихся алгоритмов. Условный оператор. Составной оператор. Многообразие способов записи ветвления

НАЧАЛА ПРОГРАММИРОВАНИЯ

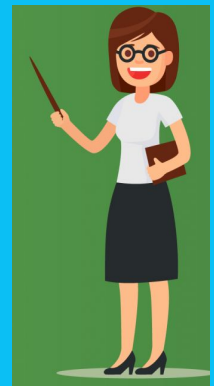
8 класс



ИЗДАТЕЛЬСТВО

БИНОМ

Не забудьте
записать в тетради
число, классная
работа, тема урока



Ключевые слова

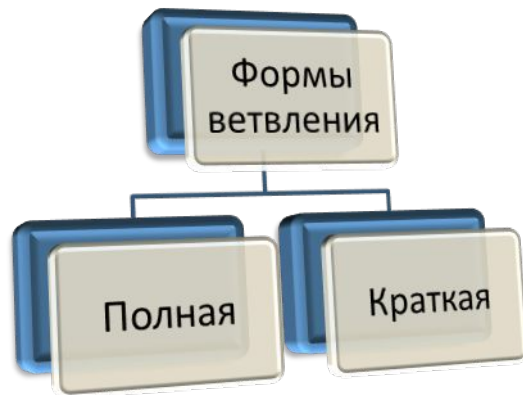
- **условный оператор**
- **полная и краткая формы**
- **составной оператор**
- **вложенные ветвления**



Ветвление

Ветвление - алгоритмическая конструкция, в которой в зависимости от результата проверки условия («да» или «нет») предусмотрен выбор одной из двух последовательностей действий (ветвей).

Алгоритмы, в основе которых лежит структура «ветвление», называют **разветвляющимися**.

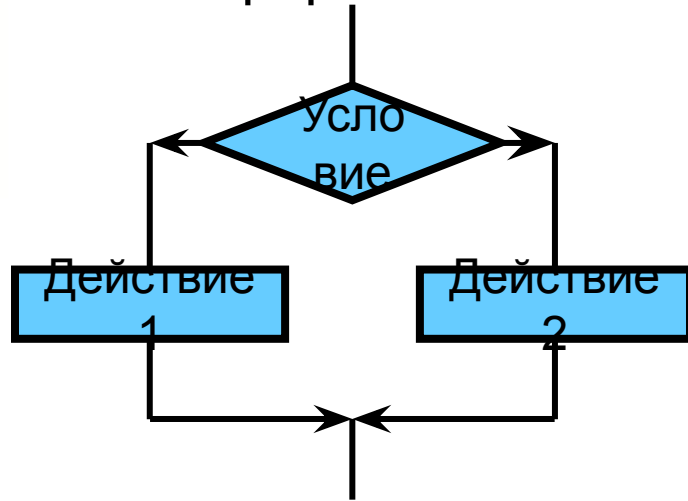


В программе ветвление реализуется условным оператором.

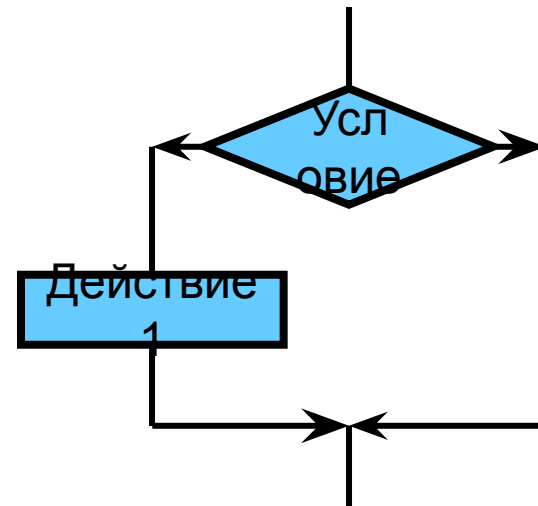


Общий вид условного оператора

Полная форма



Краткая форма



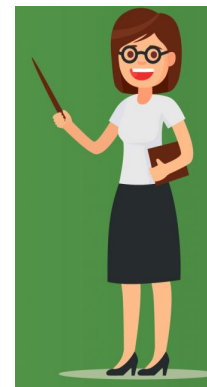
```
if <условие>  
  then  
    <оператор_1>  
  else  
    <оператор_2>;
```

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

! Перед **else** знак «;» не ставится.

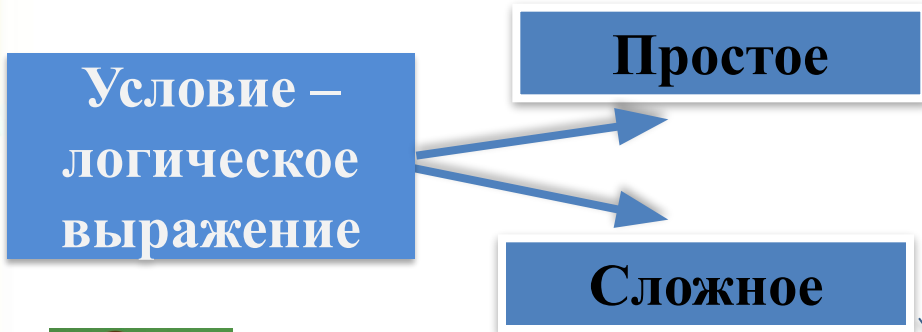
Не забудьте
делать
записи в
тетради

Словесно данный алгоритм можно сформулировать так:
Если условие **истинно**, то выполняется **Действие 1**,
если условие **ложно**, то выполняется **Действие 2**



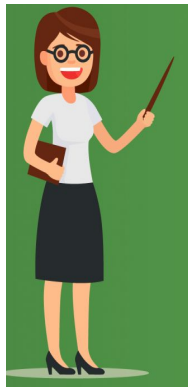
Общий вид условного оператора

Теперь разберемся в деталях!



Константы, переменные, выражения, связанные между собой **знаками операций отношения**
=, <> (не равно), >(больше), <(меньше), >=(больше или равно), <=(меньше или равно)

Простые логические выражения, связанные **логическими операциями**
OR – дизъюнкция, ИЛИ, логическое сложение
AND – конъюнкция, И, логическое умножение
NOT – отрицание, НЕ, логическое отрицание
Например, NOT (Условие1)
(Условие 1) OR (Условие 2)
(Условие 1) AND (Условие 2)



Не забудьте записать в тетрадь



Например,
`readln(A);`

Например,
`writeln('D=', D);`

Например,
`Y:=A+5;`

Примеры решения задач (записать в

тетрадь)

Задача 1. Принадлежит ли точка x отрезку $[a, b]$?

Разберемся в условии задачи:

Что дано?

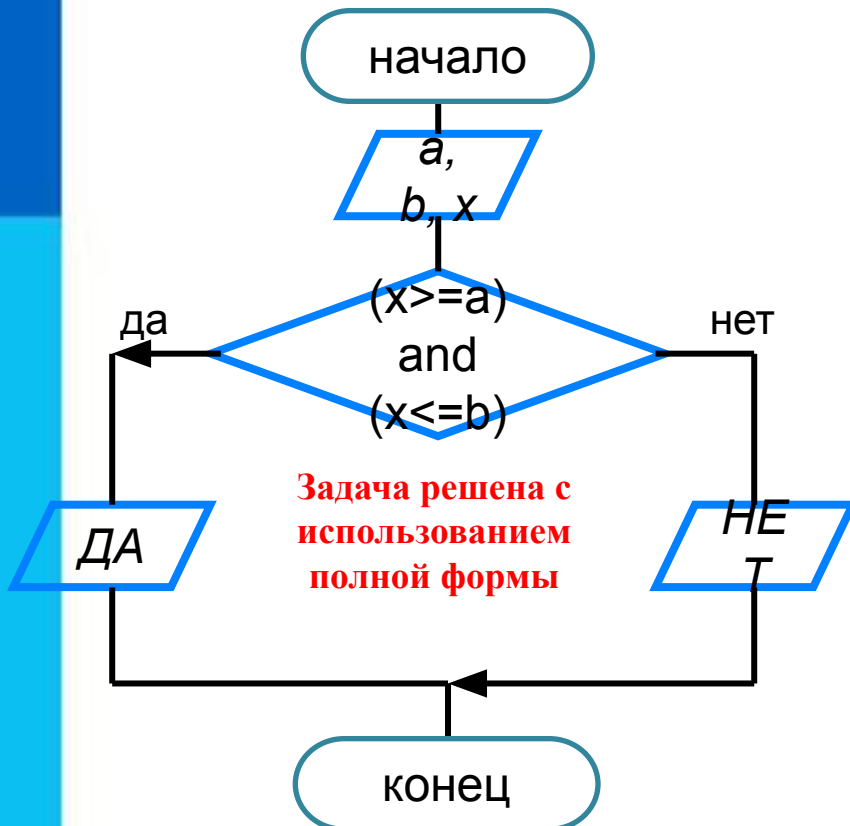
ОТВЕТ: x, a, b

В каком виде нужно получить результат?

ОТВЕТ: Да (принадлежит)

или Нет(не принадлежит)

Для решения нужно проверить условие $a \leq x \leq b$ (условие сложное!)



program Z1;

var

x, a, b : real;

begin

writeln ('Точка принадлежит отрезку?');

write ('Введите a, b>>');

readln (a, b);

write ('Введите x>>');

readln (x);

if (x >= a) **and** (x <= b)

then writeln ('Принадлежит отрезку')

else writeln ('Не принадлежит отрезку');

readln;

end.

Примеры решения задач (записать в тетрадь)

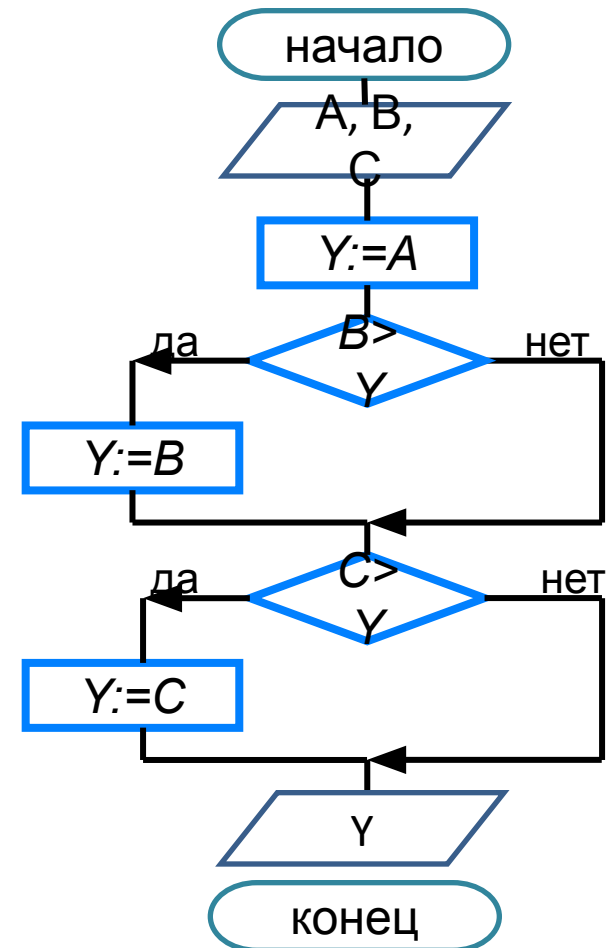
Задача 2. Даны три величины A , B , C . Переменной Y присвоить значение большей из них.

Разберемся в условии задачи:

Понятно, что значения нужно сравнить. Предположим, что наибольшим является значение A – сохраним его в переменной Y . Затем сравним с Y переменную B . Если B больше по значению – сохрани его в Y . Аналогично поступим с переменной C . Выводим значение Y .

```
program Z2;  
  var  
    y, a, b, c: integer;  
begin  
  writeln ('Нахождение max(A, B, C)');  
  write ('Введите a, b, c>>');  
  readln (a, b, c);  
  y:=a;  
  if (b>y)  
    then y:=b;  
  if (c>y)  
    then y:=c;  
  writeln ('y=', y);  
  readln;
```

Задача решена с использованием двух кратких форм условного оператора



Составной оператор

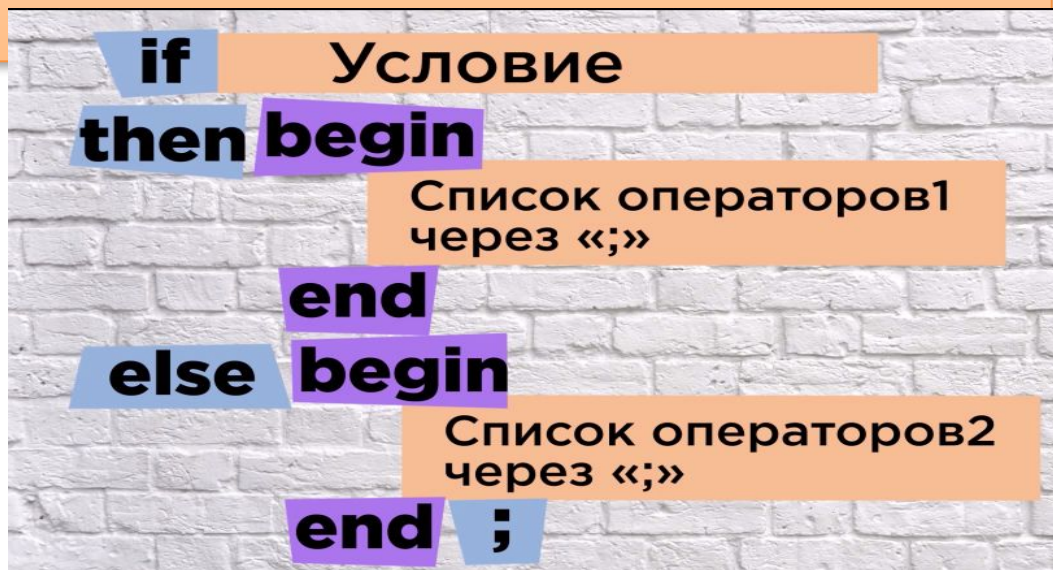
В условном операторе и после **then**,
и после **else** можно использовать
только один оператор.

Если в условном операторе после **then** или после **else**
нужно выполнить **несколько операторов**,
то используют **составной оператор**

begin

список операторов через ;

end



Примеры решения задач (записать в

тетрадь)

Задача 3. Даны три величины. Можно ли построить треугольник с заданными сторонами. Если можно, то найти A, B, C его периметр, иначе вывести сообщение «Данные не корректны».

Разберемся в условии задачи:

Треугольник можно построить если величины положительны, т.е. $A, B, C \geq 0$ и если длина каждой из сторон меньше суммы длин двух других. Таким образом, нужно проверить условие $(A \geq 0) \text{ and } (B \geq 0) \text{ and } (C \geq 0) \text{ and } (A > B + C) \text{ and } (B > A + C) \text{ and } (C > B + A)$

```
program Z3;
```

```
var
```

```
  P, A, B, C: integer;
```

```
begin
```

```
  write ('Введите a, b, c >>');
```

```
  readln (a, b, c);
```

```
  if (A >= 0) and (B >= 0) and (C >= 0) and (A > B + C)
  and (B > A + C) and (C > B + A)
```

```
  then
```

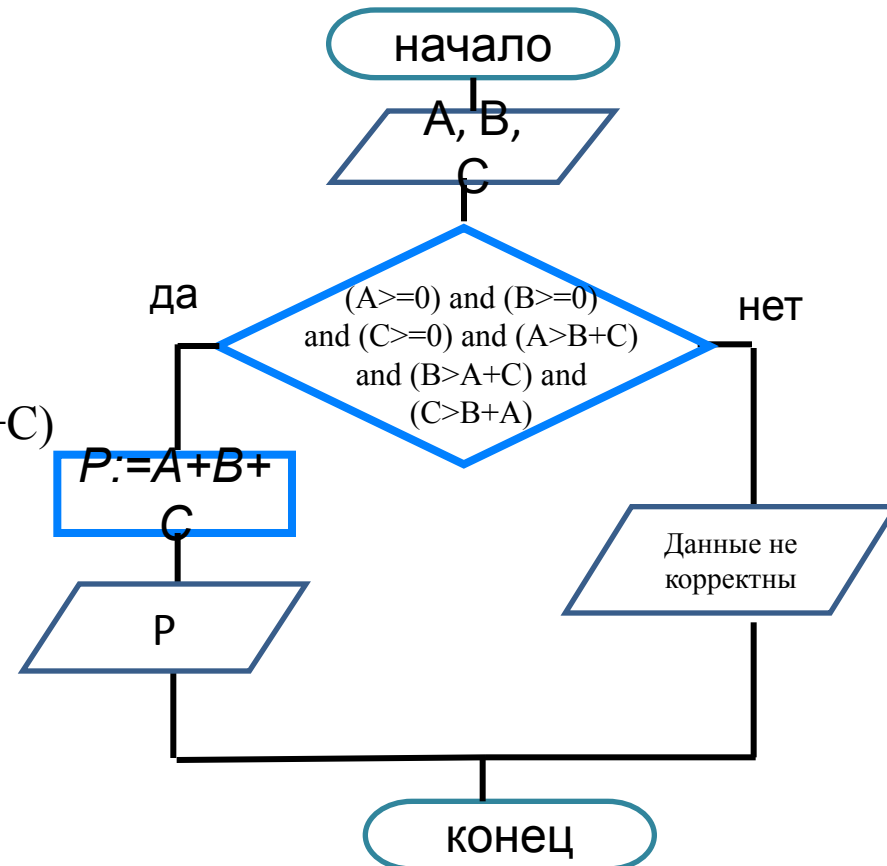
```
    begin
```

```
      P := a + b + c;
```

```
      writeln ('Периметр -', P);
```

```
    end
```

```
  else
```



Вложенные ветвления

```
if <условие1>  
  then  
    if <условие2>  
      then <оператор1>  
      else <оператор2>  
    else <оператор3>;
```

! **else** всегда относится к ближайшему оператору **if**

Решение линейного уравнения

Задача 4. Составим алгоритм решения линейного уравнения $ax+b=0$ (переменные вещественные).

!!!Алгоритм решения задачи представлен на блок-схеме. Какие элементы в блок-схеме отсутствуют? Добавьте их!

Program Z4;

var

a, b, x: real;

begin

writeln ('Решение линейного уравнения');

write ('Введите коэффициенты a , b>>');

readln (a, b);

if a<>0

then

begin

x:=-b/a;

writeln ('Корень уравнения x=', x:9:3)

end

else

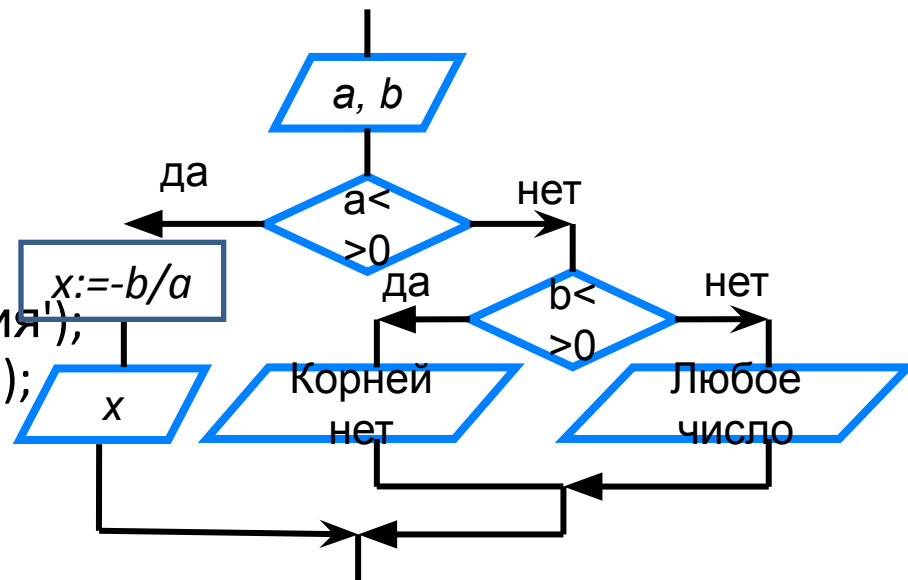
if b<>0

then writeln ('Корней нет')

else writeln ('x – любое число');

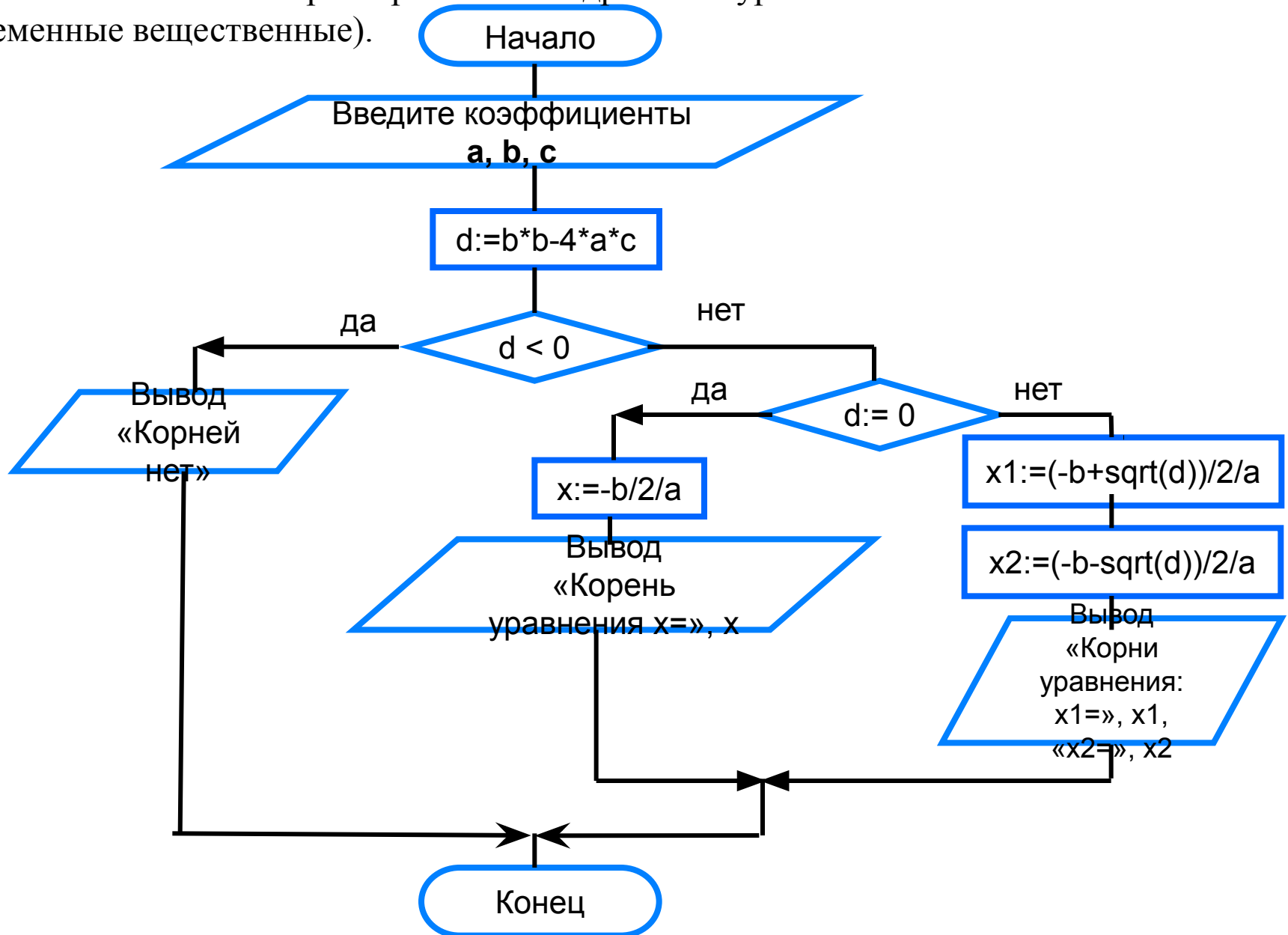
readln;

end.



Блок-схема решения КВУР

Задача 5. Составим алгоритм решения квадратного уравнения $ax^2+bx+c=0$ (переменные вещественные).



```
program Z5;
```

```
var a, b, c: real;
```

```
var d: real;
```

```
var x, x1, x2: real;
```

```
begin
```

```
writeln ('Решение квадратного уравнения');
```

```
write ('Введите коэффициенты a, b, c >>');
```

```
readln (a, b, c);
```

```
d:=b*b-4*a*c;
```

```
if d<0 then writeln ('Корней нет')
```

```
else
```

```
if d=0 then
```

```
begin
```

```
x:=-b/2/a;
```

```
writeln ('Корень уравнения x=', x:9:3)
```

```
end
```

```
else
```

```
begin
```

```
x1:=(-b+sqrt(d))/2/a;
```

```
x2:=(-b-sqrt(d))/2/a;
```

```
writeln ('Корни уравнения:');
```

```
writeln ('x1=', x1:9:3);
```

```
writeln ('x2=', x2:9:3)
```

```
end
```

```
end.
```

Самое главное

При записи на языке Паскаль разветвляющихся алгоритмов используют условный оператор:

```
if <условие>  
  then <оператор_1>  
  else <оператор_2>;
```

Для записи неполных ветвлений используется неполный условный оператор:

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

Если при некотором условии требуется выполнить несколько операторов, то их объединяют в один составной оператор, имеющий вид:

```
begin  
  <список операторов>;  
end
```



Вопросы и задания

(оформляются с тетради: число, домашняя работа, фото конспекта классной и домашней работы в личку или на e.l.meznikova@mail.ru)

1. Запишите предложенные конструкции структурировано, исправьте найденные ошибки.

а) if $x < y$ then $x := 0$ else read (y)

б) if $x \geq y$ then $x := 0$; $y := 0$ else write (z)

в) if $x < y < z$ then $a := a + 1$

2. Используя составной оператор, упростите следующий фрагмент программы:

if $a > b$ then $c := 1$;

if $a > b$ then $d := 2$;

if $a \leq b$ then $c := 3$;

if $a \leq b$ then $d := 4$

Вопросы и задания

(оформляются с тетради: число, домашняя работа, фото конспекта классной и домашней работы в личку или на e.l.meznikova@mail.ru)

3. Составьте блок-схему и программу

Даны две точки в плоской прямоугольной системе координат. Напишите программу, определяющую, которая из точек находится ближе к началу координат.

Пример входных данных	Пример выходных данных
Координаты 1-й точки >> 1, 2 Координаты 2-й точки >> 3, 4	Первая точка ближе

РЕКОМЕНДАЦИИ: найти расстояние от каждой точки до начала координат $L = \sqrt{x^2 + y^2}$, сравнить расстояния. За основу можно взять решение задачи 1

Опорный конспект

Условный оператор

Полная форма

if <условие> *then* <оператор_1> *else* <оператор_2>

Краткая форма

if <условие> *then* <оператор>

Составной оператор

begin <последовательность операторов> *end*

Надеюсь,
что изложенный материал будет
понятен.
Прочитайте несколько раз и Вы увидите,
что все не так сложно



Лифт к успеху не работает.
Используйте ступеньки.
Шаг за шагом.