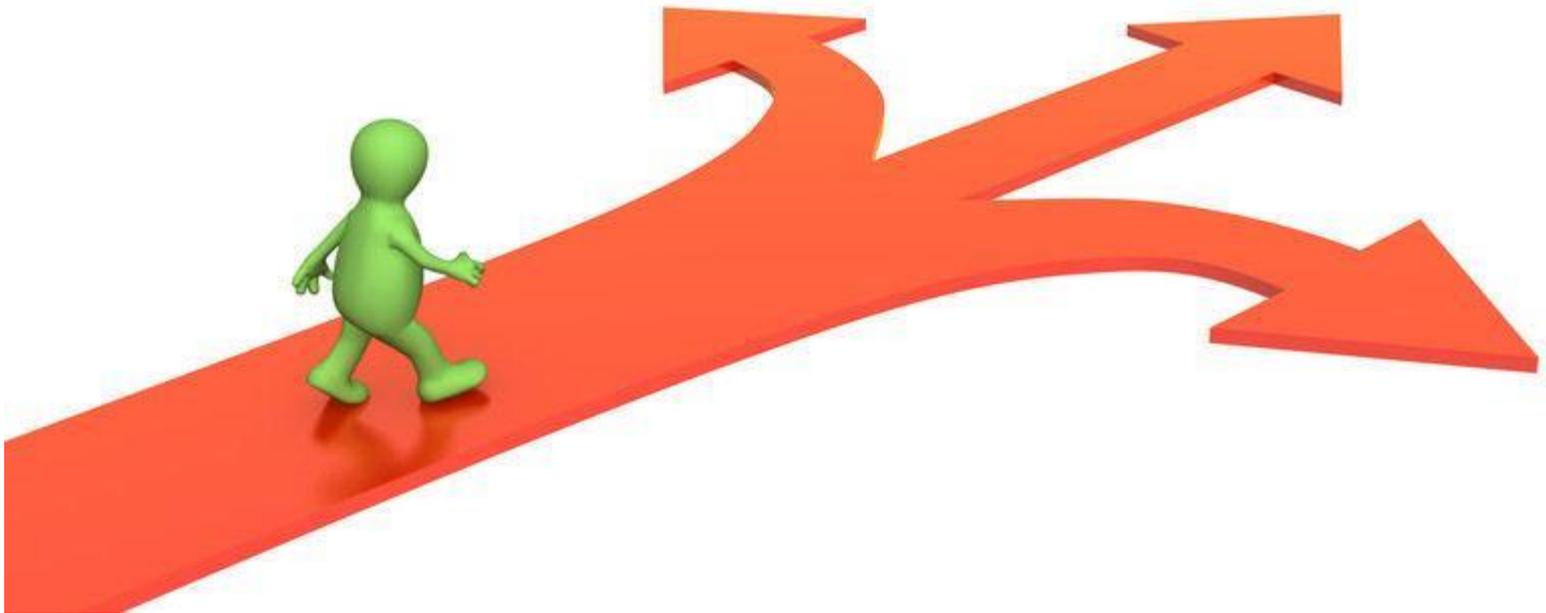


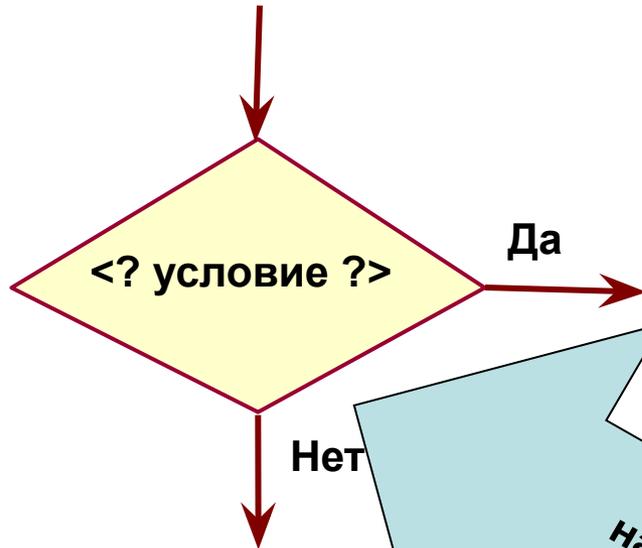
# Лекция 5

## Алгоритмы с разветвлением

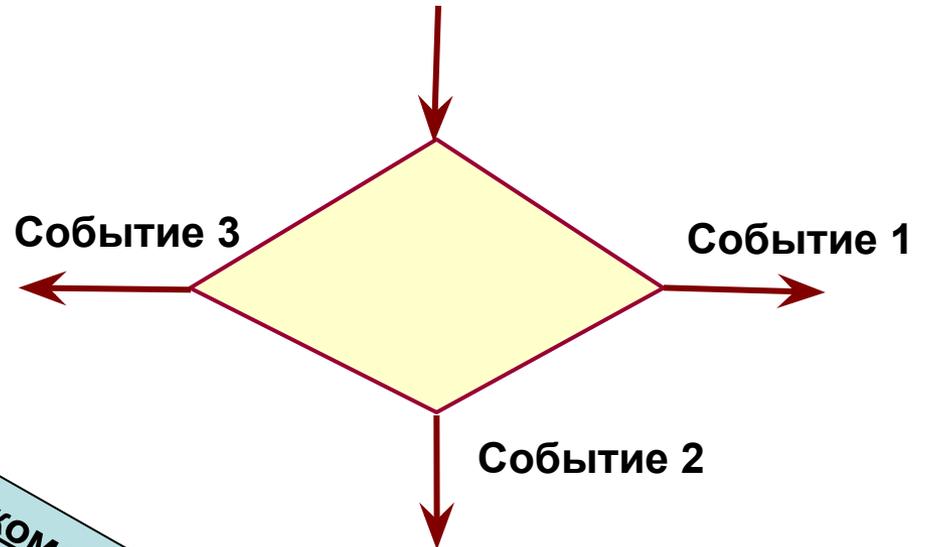


# Блок принятия решения

Простое решение

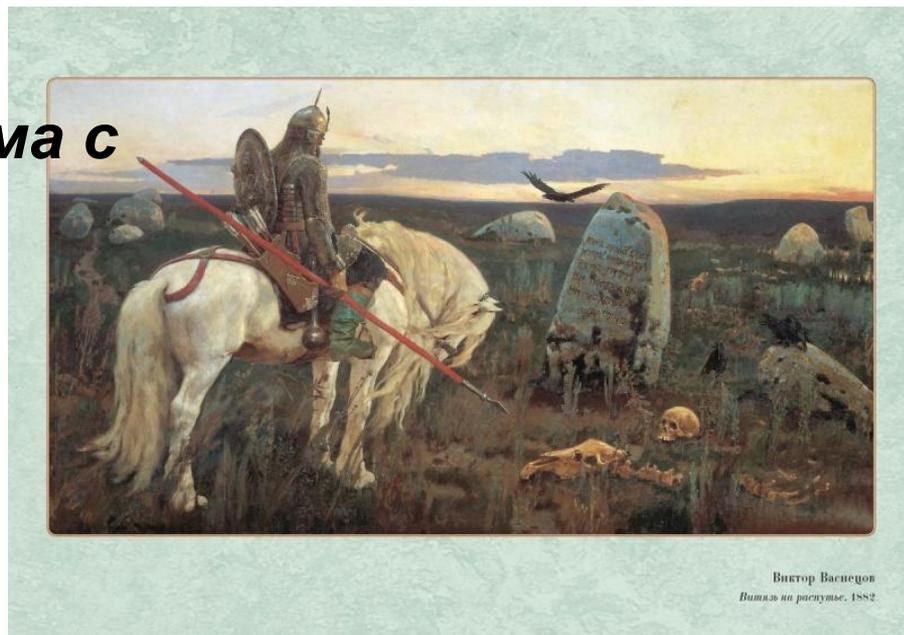


Сложное решение



**Рекомендация:** при отсутствии навыков декомпозиции сложной задачи предпочтение следует отдавать простым решениям

## 5.1. Пример алгоритма с разветвлением

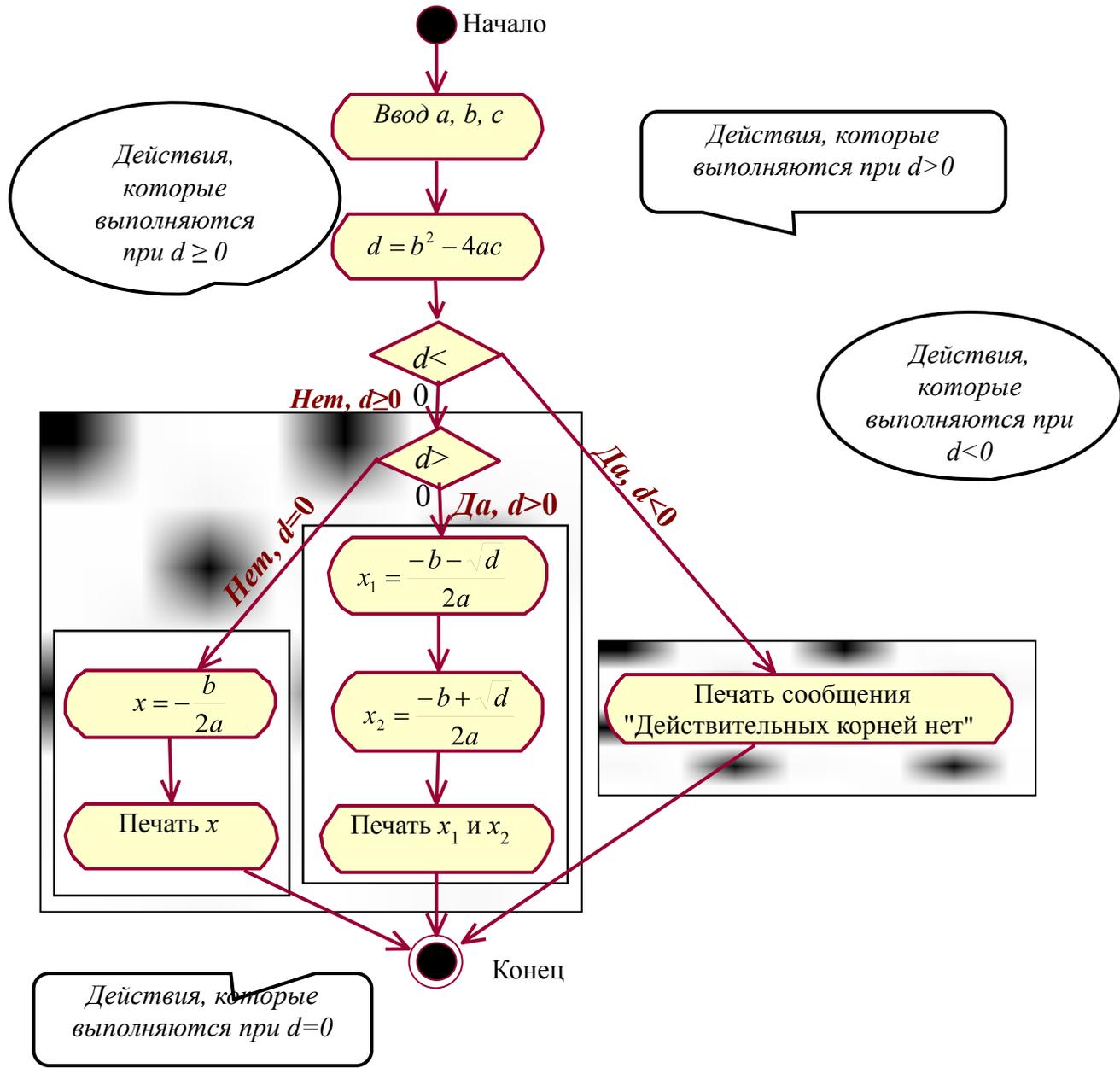


### Задание 5.1:

#### **Постановка задачи:**

$a, b, c$  – действительные числа. Требуется найти действительные корни квадратного уравнения

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



## 5.2. Условный оператор

$\langle \text{условный оператор} \rangle ::=$

*if*  $\langle \text{логическое выражение} \rangle$  *then*

$\| \langle \text{оператор 1} \rangle \|$

$\|$  *else*

$\| \langle \text{оператор 2} \rangle \| \|$

Если  $\langle \text{логическое выражение} \rangle$  истинно ( $\langle \text{логическое выражение} \rangle = \text{True}$ ), то управление передается на  $\langle \text{оператор 1} \rangle$ , если же  $\langle \text{логическое выражение} \rangle$  не выполнилось ( $\langle \text{логическое выражение} \rangle = \text{False}$ ), то начинает работать  $\langle \text{оператор 2} \rangle$ . После того, как один из этих двух операторов отработал (или отсутствует), программа продолжает выполняться линейно: начинает работать та ее часть, которая находится непосредственно за условным оператором.

# Контрольные примеры:

**Контрольный пример 1:  $2x^2 - 8x + 6 = 0$**

Исходные данные:

A=2 B= -8 C=6

Ожидаемый результат:

**$x_1=1$   $x_2=3$**

**Контрольный пример 2:  $2x^2 - 8x + 16 = 0$**

Исходные данные:

A=2 B= -8 C=16

Ожидаемый результат:

**Действительных корней  
нет**

**Контрольный пример 3:  $2x^2 - 8x + 8 = 0$**

Исходные данные:

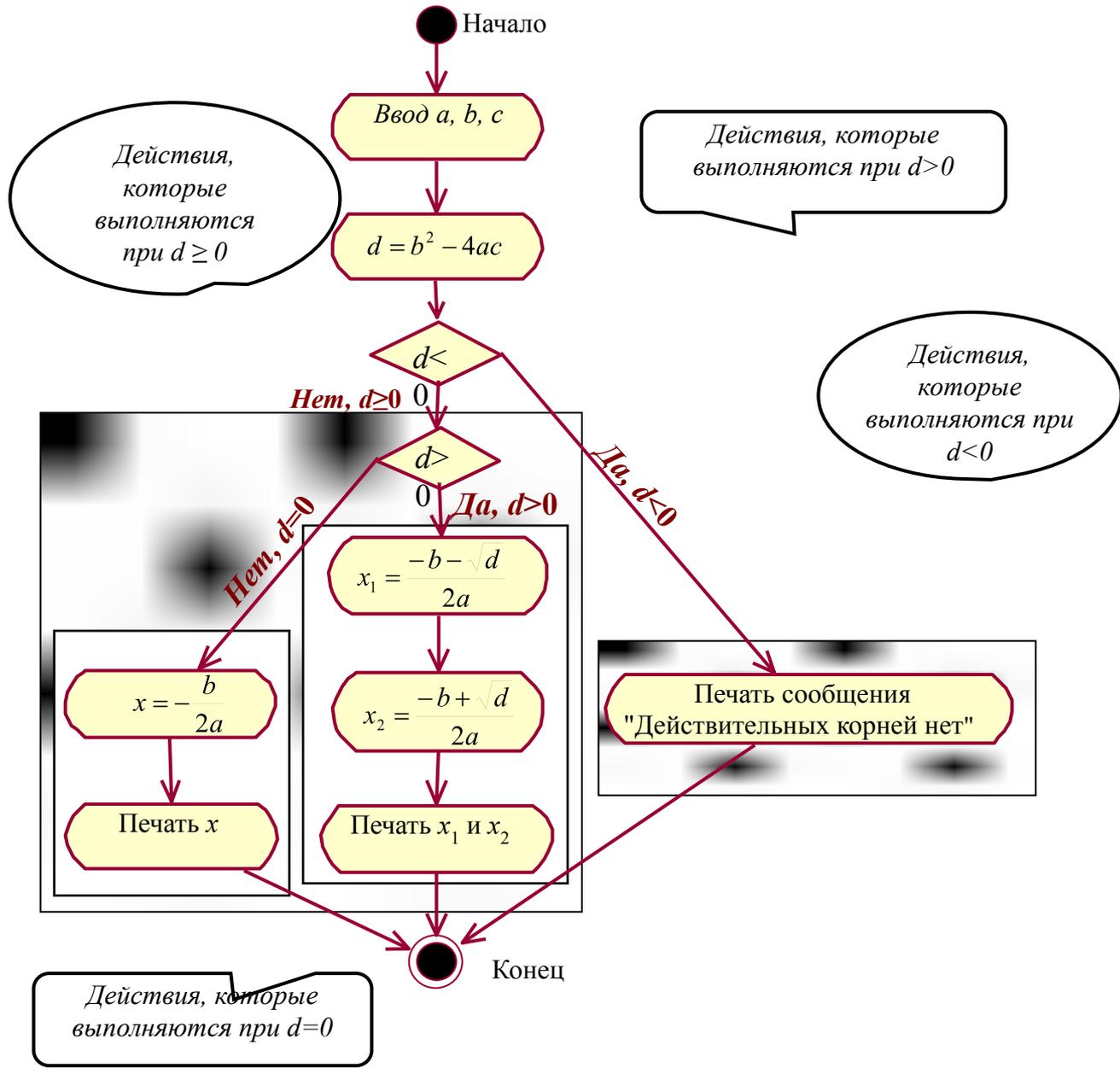
A=2 B= -8 C=8

Ожидаемый результат:

**$x=2$**

№ п.п.	Фрагмент программы	Комментарии
1.	<pre>X:=0; <b>if</b> X&gt;0 <b>then</b>;</pre>	<p>Синтаксически верная конструкция оператора <code>if</code>, тем не менее, не приводящая ни к каким внешне проявляющимся активным действиям.</p>
2.	<pre>X:=0; <b>if</b> X&gt;0 <b>then else</b>;</pre>	<p>Синтаксически верная конструкция оператора <code>if</code>, тем не менее, не приводящая ни к каким внешне проявляющимся активным действиям.</p>
3.	<pre>X:=5; <b>if</b> X&gt;0 <b>then</b>     X:=-10; WriteLn('Результат X=', X);</pre>	<p>Т.к. логическое выражение <math>X &gt; 0</math> истинно, управление будет передано оператору <math>X := -10</math> и переменная <math>X</math> поменяет свое значение. На экран будет выведено сообщение</p> <p style="text-align: center;">Результат X=-10</p>
4.	<pre>X:=-15; <b>if</b> X&gt;0 <b>then</b>     X:=-10; WriteLn('Результат X=', X);</pre>	<p>Т.к. логическое выражение <math>X &gt; 0</math> ложно, управление будет передано оператору печати, следующему непосредственно за условным оператором и на экран будет выведено сообщение</p> <p style="text-align: center;">Результат X=-15</p>

№ п.п.	Фрагмент программы	Комментарии
5.	<pre>X:=5; <b>if</b> X&lt;=0 <b>then</b> <b>else</b>   X:=-10; WriteLn('Результат X=', X);</pre>	<p>Синтаксически верная конструкция, но, тем не менее, грамотной ее не назовешь. По конечному результату она полностью эквивалентна конструкции, приведенной в примере 3.</p>
6.	<pre>X:=5; <b>if</b> X&lt;=0 <b>then</b>   X:=1 <b>else</b>   X:=-10; WriteLn ('Результат X=', X);</pre>	<p>Т.к. логическое выражение <math>X \leq 0</math> ложно, то оператор <math>X := 1</math> будет пропущен, выполнится оператор присваивания <math>X := -10</math>, на печать поступит сообщение Результат <math>X = -10</math></p>
7.	<pre>X:=-15; <b>if</b> X&lt;=0 <b>then</b>   X:=1 <b>else</b>   X:=-10; WriteLn('Результат X=', X);</pre>	<p>Т.к. логическое выражение <math>X \leq 0</math> истинно, то выполнится оператор <math>X := 1</math>, оператор присваивания <math>X := -10</math> <b>будет пропущен и на печать поступит сообщение</b> Результат <math>X = 1</math></p>



## 5.3. Составной оператор

- **Простым оператором** является такой оператор, который не содержит в себе других операторов.
- **Структурные операторы** строятся из других операторов, порядок выполнения которых должен быть последовательным (составные операторы и операторы над записями), определяемым условной передачей управления (условные операторы) или повторяющимся (операторы цикла).
- **Составной оператор** — конструкция языка программирования, состоящая из нескольких простых команд (операторов) языка программирования, но участвующая в программе в качестве единого оператора.



$\langle \text{составной оператор} \rangle ::=$

*begin*

$\{ \langle \text{оператор} \rangle ; \}$

*end*

Фрагмент программы	Комментарии
<pre> X:=5; if X&gt;0 then   begin     A:='круть';     B:='верть';   end else   begin     A:='верть';     B:='круть';   end; WriteLn(A,'-',B); </pre>	<p>Т.к. логическое выражение <math>X&gt;0</math> истинно, управление передается составному оператору</p> <pre> begin   A:='круть';   B:='верть'; end. </pre> <p>В результате работы приведенного фрагмента программы на экран поступит сообщение</p> <p>круть-верть</p>
<pre> X:=-5; if X&gt;0 then   begin     A:='круть';     B:='верть';   end else   begin     A:='верть';     B:='круть';   end; WriteLn(A,'-',B); </pre>	<p>Т.к. логическое выражение <math>X&gt;0</math> ложно, управление передается составному оператору</p> <pre> begin   A:='верть';   B:='круть'; end; </pre> <p>В результате работы приведенного фрагмента программы на экран поступит сообщение</p> <p>верть-круть</p>

```
program Quadratic;  
var  
  A,B,C,D,X,X1,X2: Real;
```

```
begin
```

```
  Write('Введите коэффициенты квадратного  
уравнения:');
```

```
  ReadLn(A,B,C);
```

```
  D:=Sqr(B)-4*A*C;
```

```
  if D<0 then
```

```
    WriteLn(' Действительных корней нет')
```

```
  else
```

```
    if D=0 then
```

```
      begin
```

```
        X:=-B/(2*A);
```

```
        WriteLn('X=', X:5:1);
```

```
      end
```

```
    else
```

```
      begin
```

```
        X1:=(-B-Sqr(D))/(2*A);
```

```
        X2:=(-B+Sqr(D))/(2*A);
```

```
        WriteLn('X1=', X1:5:1, ' X2=', X2:5:1);
```

```
      end;
```

```
  ReadLn;
```

```
end.
```

В случае невыполнения условия  $d < 0$   
управление передается условному  
оператору со сложной структурой

## **Контрольный пример 1: $2x^2-8x+6=0$**

*Исходные данные:*

$$A=2 \quad B=-8 \quad C=6$$

*Ожидаемый результат:*

$$x_1=1 \quad x_2=3$$

### **Протокол 5.1**

**Работа программы Quadratic при решении квадратного уравнения  $2x^2-8x+6=0$**

1. Печать сообщения

Введите коэффициенты квадратного уравнения:

2. Программа останавливается и ожидает, когда пользователь наберет строку ввода:

2€-8€6↵

В результате  $A=2$ ,  $B=-8$ ,  $C=6$ .

3. Вычисление дискриминанта  $D:=\text{Sqr}(B)-4*A*C=(-8)^2-4*2*6=16$ .

4.  $(D<0)=(16<0)=\text{False}$

5.  $(D=0)=(16=0)=\text{False}$

6.  $X1:=(-B-\text{Sqr}(D))/(2*A)=1$

7.  $X2:=(-B+\text{Sqr}(D))/(2*A)=3$

8. Печать сообщения « $X1=€€€1.0 \quad X2=€€€3.0$ ».

9. Конец работы программы.

## **Контрольный пример 2: $2x^2-8x+16=0$**

*Исходные данные:*

A=2 B= -8 C=16

*Ожидаемый результат:*

Действительных корней нет

### **Протокол 5.2**

#### **Работа программы Quadratic при решении квадратного уравнения $2x^2-8x+16=0$**

1. Печать сообщения

Введите коэффициенты квадратного уравнения:

Программа останавливается и ожидает, когда пользователь наберет строку ввода:

2€-8€16↵

В результате A=2, B=-8, C=16.

2. Вычисление дискриминанта  $D:=\text{Sqr}(B)-4*A*C=(-8)^2-4*2*16= -64$ .

3. Печать сообщения «Действительных корней нет».

4.  $(D<0)=(16<0)=\text{False}$

5. Конец работы программы.

### **Контрольный пример 3: $2x^2-8x+16=0$**

*Исходные данные:*

$$A=2 \quad B=-8 \quad C=8$$

*Ожидаемый результат:*

$$x=2$$

#### **Протокол 5.2**

#### **Работа программы Quadratic при решении квадратного уравнения $2x^2-8x+8=0$**

1. Печать сообщения

Введите коэффициенты квадратного уравнения:.

Программа останавливается и ожидает, когда пользователь наберет строку ввода:

2€-8€8↵

В результате  $A=2$ ,  $B=-8$ ,  $C=8$ .

2. Вычисление дискриминанта  $D:=\text{Sqr}(B)-4*A*C=(-8)^2-4*2*8=0$ .

3.  $(D<0)=(0<0)=\text{False}$

4.  $(D=0)=(16=0)=\text{True}$

5.  $X:=-B/(2*A)=8/(2*2)=2$

6. Печать сообщения « $X=€€€2.0$ ».

7. Конец работы программы.

Для улучшения  
читабельност  
и программ  
стандарт  
структурного  
программиров  
ания  
предписывает  
оформлять их  
шрифтом  
Courier New  
с отступами.  
Принцип  
расстановки  
отступов :

1. В одной  
строке  
печатаются  
не более 1  
простого  
оператора.
2. Если  
оператор

```
program Quadratic;
var
  A,B,C,D,X,X1,X2: Real;
begin
  Write('Введите коэффициенты квадратного уравнения:');
  ReadLn(A,B,C);
  D:=Sqr(B)-4*A*C;
  if D<0 then
    WriteLn('Действительных корней нет')
  else
    if D=0 then
      begin
        X:=-B/(2*A);
        WriteLn('X=', X:5:1);
      end
    else
      begin
        X1:=(-B-Sqr(D))/(2*A);
        X2:=(-B+Sqr(D))/(2*A);
        WriteLn('X1=',X1:5:1,' X2=', X2:5:1);
      end;
    ReadLn;
  end.
```

```
program Quadratic;  
var  
  A,B,C,D,X,X1,X2: Real;  
begin  
  Write('Введите коэффициенты квадратного уравнения:');  
  ReadLn(A,B,C);  
  D:=Sqr(B)-4*A*C;  
  if D<0 then  
    WriteLn(' Действительных корней нет')  
  else  
    if D=0 then  
      begin  
        X:=-B/(2*A);  
        WriteLn('X=', X:5:1);  
      end  
    else  
      begin  
        X1:=(-B-Sqr(D))/(2*A);  
        X2:=(-B+Sqr(D))/(2*A);  
        WriteLn('X1=',X1:5:1,' X2=', X2:5:1);  
      end;  
    ReadLn;  
end.
```

```

program Quadratic;
var
  A,B,C,D,X,X1,X2: Real;
begin
  Write('Введите коэффициенты квадратного уравнения:');
  ReadLn(A,B,C);
  D:=Sqr(B)-4*A*C;
  if D<0 then
    WriteLn('Действительных корней нет')
  else
    if D=0 then
      begin
        X:=-B/(2*A);
        WriteLn('X=', X:5:1);
      end
    else
      begin
        X1:=(-B-Sqr(D))/(2*A);
        X2:=(-B+Sqr(D))/(2*A);
        WriteLn('X1=',X1:5:1,' X2=', X2:5:1);
      end;
    ReadLn;
  end.

```

```

program Quadratic;
var
  A,B,C,D,X,X1,X2: Real;
begin
  Write('Введите коэффициенты квадратного уравнения:');
  ReadLn(A,B,C);
  D:=Sqr(B)-4*A*C;
  if D<0 then
    WriteLn('Действительных корней нет')
  else
    if D=0 then
      begin
        X:=-B/(2*A);
        WriteLn('X=', X:5:1);
      end
    else
      begin
        X1:=(-B-Sqr(D))/(2*A);
        X2:=(-B+Sqr(D))/(2*A);
        WriteLn('X1=',X1:5:1,' X2=', X2:5:1);
      end;
    ReadLn;
  end.

```

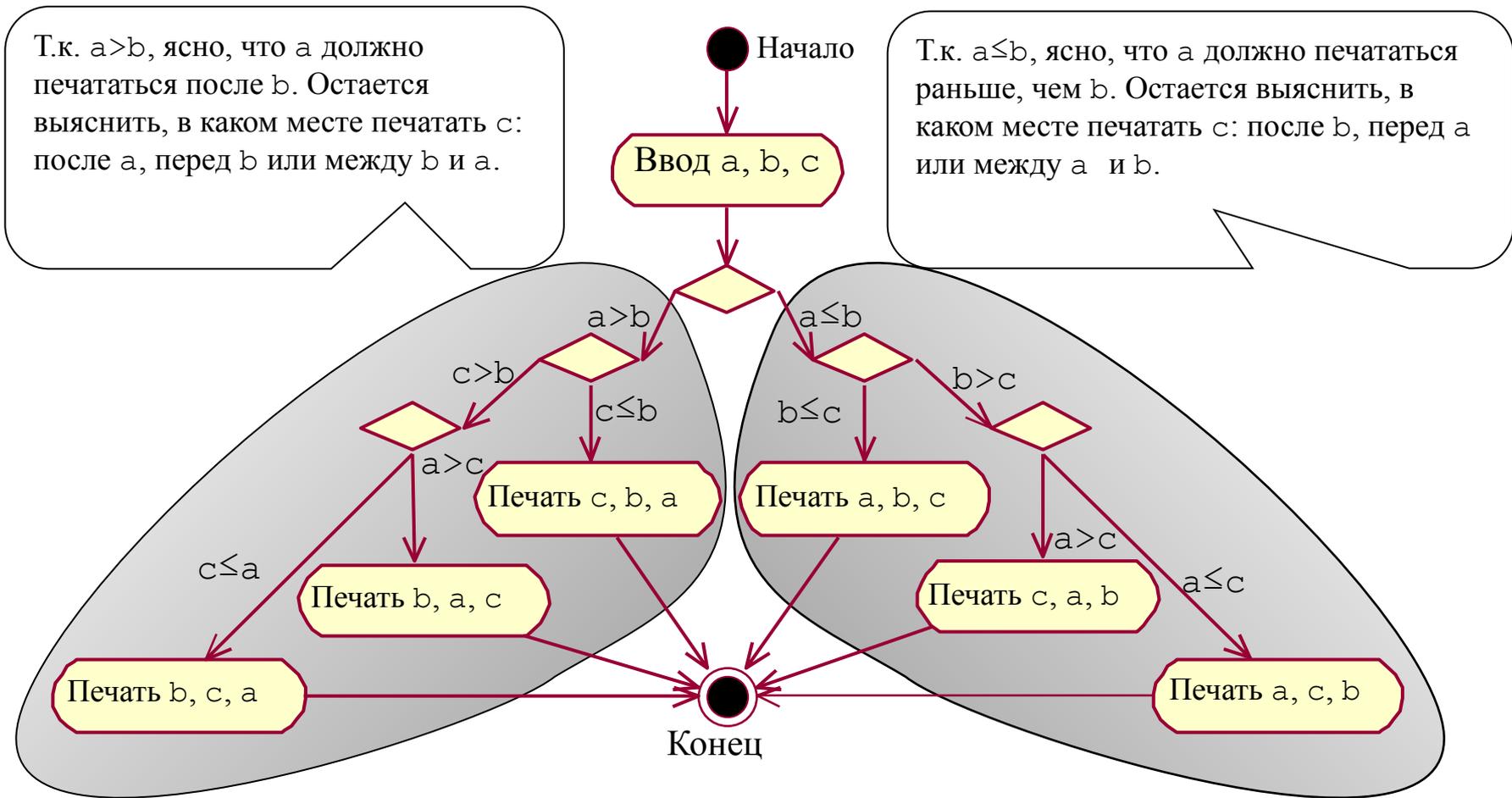
## Задание 5.2:

### **Постановка задачи:**

Заданы три **целых числа**  $a$ ,  $b$ ,  $c$ . Требуется вывести их на экран дисплея в порядке возрастания.



<b>№ варианта</b>	<b>Порядок вывода на печать</b>	<b>Условие</b>
1.	$a, b, c$	$a \leq b \leq c$
2.	$a, c, b$	$a \leq c \leq b$
3.	$b, a, c$	$b \leq a \leq c$
4.	$b, c, a$	$b \leq c \leq a$
5.	$c, a, b$	$c \leq a \leq b$
6.	$c, b, a$	$c \leq b \leq a$



*Рисунок – Блочная схема алгоритма сортировки трех чисел*

```

program Sorting;
var
  A,B,C: Integer;
begin
  Write('Введите три целых числа: ');
  ReadLn(A,B,C);
  if A<=B then
    if B<=C then
      WriteLn(A:3,B:3,C:3)           { вариант
1}
    else
      if A<=C then
        WriteLn(A:3,C:3,B:3)       { вариант
2}
      else
        WriteLn(C:3,A:3,B:3)       { вариант
5}
    else
      if C<=B then
        WriteLn(C:3,B:3,A:3)       { вариант
6}
      else
        if C<=A then
          WriteLn(B:3,C:3,A:3)     { вариант
4}
        else
          WriteLn(B:3,A:3,C:3);    { вариант
3}
    ReadLn;
end.

```

## **Работа программы Sorting при значениях A=7, B=5, C=8**

1. Печать сообщения

Введите три целых числа:

2. Программа останавливается и ожидает, когда пользователь наберет строку ввода:

7 5 8 ↵

В результате A=7, B=5, C=8.

3.  $(A \leq B) = (7 \leq 5) = \text{False}$  {Возможными остаются варианты 3, 4, 6}.

4.  $(C \leq B) = (8 \leq 5) = \text{False}$  {Возможными остаются варианты 3, 4}.

5.  $(C \leq A) = (8 \leq 7) = \text{False}$  {Остается вариант 3}.

6. Печать C, B, A. На экран выводится сообщение «578».

7. Конец работы программы.

## 5.4. Простейшие логические операции и выражения



*fn*

© 2010 Сергей Тарасов www.sergeytarasov.ru

<i>Логическое И (and)</i>		
<b>x</b>	<b>y</b>	<b>x and y</b>
True	True	True
True	False	False
False	True	False
False	False	False

*Логическое ИЛИ (or)*

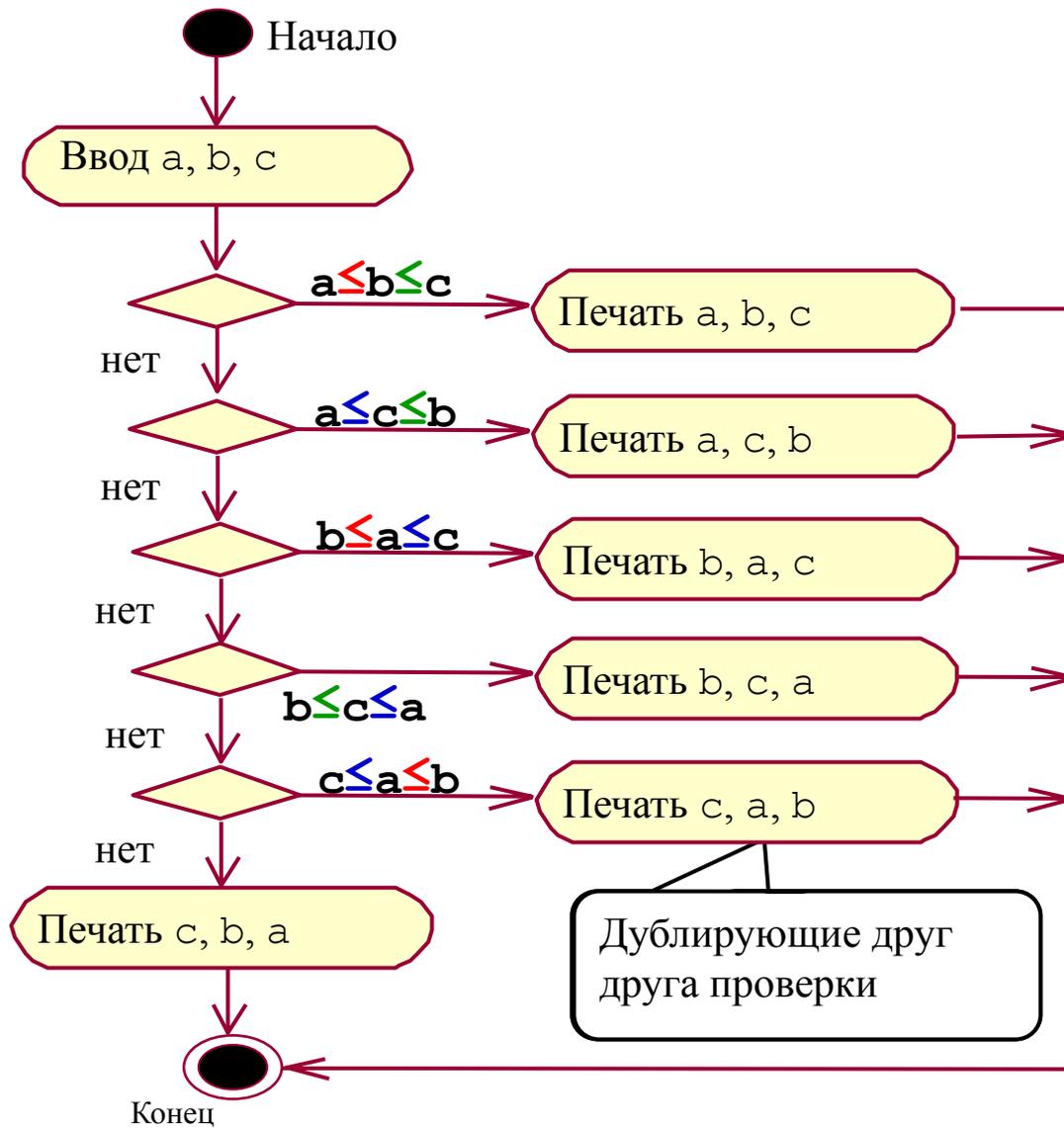
<b>x</b>	<b>y</b>	<b>x or y</b>
True	True	True
True	False	True
False	True	True
False	False	False

*Логическое НЕ (not)*

<b>X</b>	<b>not X</b>
True	False
False	True

Фрагмент программы	Комментарии
<pre> X:=5; <b>if (X&gt;0) and (X&lt;10) then</b>     A:=1 <b>else</b>     A:=13 WriteLn('Результат A=', A); </pre>	<p>Первое условие <math>X&gt;0=True</math> (истинно), второе <math>X&lt;10=True</math> (истинно), поэтому и все сложное логическое выражение принимает значение True. Управление передается оператору <math>A:=1</math>. На экран поступит сообщение «Результат A=1».</p>
<pre> X:=15; <b>if (X&gt;0) and (X&lt;10) then</b>     A:=1 <b>else</b>     A:=13 WriteLn('Результат A=', A); </pre>	<p>Условие <math>X&gt;0=True</math> (истинно), но выражение <math>X&lt;10=False</math> (ложно), поэтому и все сложное логическое выражение принимает значение False. Управление передается оператору <math>A:=13</math>. В итоге на экран поступит сообщение «Результат A=13».</p>
<pre> X:=15; <b>if (X&gt;10) or (X&lt;0) then</b>     A:=120 <b>else</b>     A:=130 WriteLn('Результат A=', A); </pre>	<p>Условие <math>X&gt;10=True</math> (истинно), выражение <math>X&lt;0=False</math> (ложно), поэтому и все сложное логическое выражение, включающее логическое <i>ИЛИ</i>, принимает значение True. Управление передается оператору <math>A:=120</math>. В итоге на экран поступит сообщение «Результат A=120». Аналогичный результат получится, например, при <math>X=-10</math>.</p>

Фрагмент программы	Комментарии
<pre> X:=5; <b>if (X&gt;10) or (X&lt;0) then</b>     A:=120 <b>else</b>     A:=130 WriteLn('Результат A=', A); </pre>	<p>Оба условия ложны (<math>X &gt; 10 = \text{False}</math> и <math>X &lt; 0 = \text{False}</math>), поэтому и все сложное логическое выражение включающее логическое <i>ИЛИ</i> принимает значение <code>False</code>. Управление передается оператору <code>A:=130</code>. В итоге на экран поступит сообщение «Результат A=130». Аналогичный результат будет получен, например, при любых <math>X \in [0; 10]</math>.</p>
<pre> X:=5; <b>if not(X&gt;10) then</b>     A:=22 <b>else</b>     A:=33 WriteLn('Результат A=', A); </pre>	<p>Условие <math>X &gt; 10</math> ложно (<math>X &gt; 10 = \text{False}</math>), поэтому все сложное логическое выражение включающее логическое <i>НЕ</i> истинно (принимает значение <code>True</code>). Управление передается оператору <code>A:=22</code>. В итоге на экран поступит сообщение «Результат A=22».</p>



*Блочная схема алгоритма сортировки трех чисел*