

ИХ РЕАЛИЗАЦИЯ НА ЯЗЫКЕ ПРОГРАММИРОВАНИЯ ПАСКАЛЬ.

Яресько Виктория Викторовна
учитель информатики и ИКТ
МБОУ «Гимназия №1» г.о. Мытищи
Московской области

Цели и задачи:

- ❑ Ввести понятие ветвления.
- ❑ Рассмотреть виды ветвлений.
- ❑ Показать, как реализуются ветвящиеся алгоритмы на языке программирования Паскаль.
- ❑ Рассмотреть задачи с ветвящимися алгоритмами.

Понятие ветвления

Ветвление – форма организации действий, при которой в зависимости от условия выполняется одна, либо другая серия действий (команд, операторов).

Ветвление позволяет менять порядок выполнения команд по результатам проверки некоторого условия.

В команде ветвления в качестве условия может использоваться отношение неравенства между величинами.

Понятие ветвления

Ветвление является *структурной командой*. Его исполнение происходит в несколько шагов:

- ❑ проверка условия (выполнение логического выражения);
- ❑ выполнение команд на одной из ветвей.

Синтаксис языка программирования

У каждого человеческого языка есть своя грамматика, включающая в себя правила, по которым должны выстраиваться в цепочку элементы языка, чтобы получилось правильное предложение. Совокупность этих правил образует часть грамматики, называемую СИНТАКСИСОМ.

В языках программирования тоже есть предложения. Такими предложениями являются операторы. Следовательно у языков программирования тоже должен быть свой синтаксис, который описывает правила, по которым записываются операторы языка, и из операторов составляется программа. После того, как человек запускает программу на выполнение, любая среда программирования прежде, чем действительно выполнить её, сначала проверит, нет ли в ней синтаксических ошибок, и если они есть, то программу выполнять не будет, а выдаст сообщение об ошибке.

Операции отношения или сравнения

Знак	Наименование	Пример	Пояснение
$<$	Меньше	$a < 6$	a меньше 6
\leq	Меньше или равно	$b \leq 23$	b меньше или равно 23
$>$	Больше	$x > 5$	x больше 5
\geq	Больше или равно	$y \geq 8$	y больше или равно 8
$=$	Равно	$c = 10$	c равно 10
\neq	Не равно	$d \neq 3$	d не равно 3

Виды ветвлений:

- неполное ветвление (обход),
- полное ветвление (альтернатива);
- вложенные ветвления;
- ветвление по ряду условий.

Неполное ветвление

К неполным ветвлениям относятся алгоритмы, выполняющие следующую структуру логического выражения: «Если ... то ...».



Запись условного оператора на Паскале

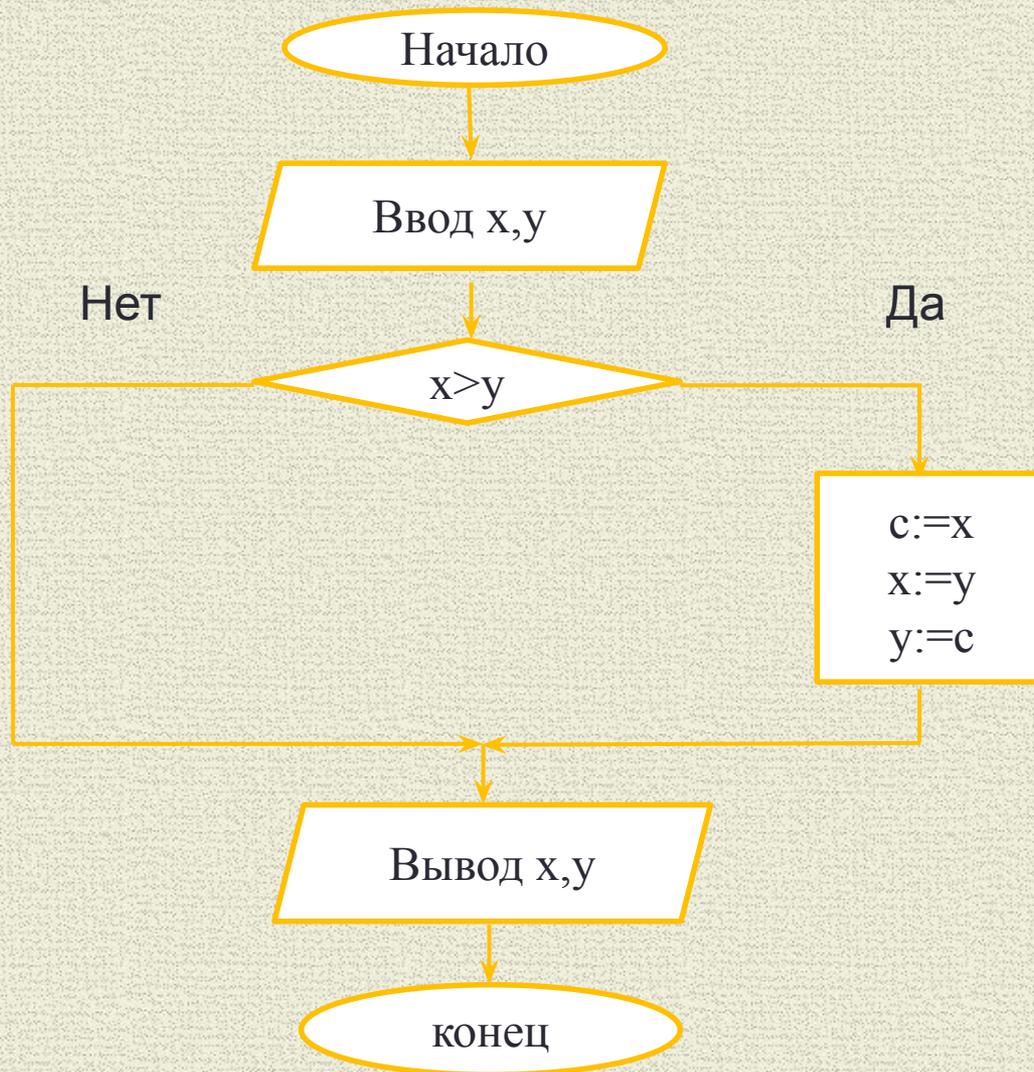
Неполная форма оператора **if**

IF <условие> **THEN** <оператор>

Задача 1

Составьте блок-схему и программу упорядочения значений двух переменных x и y по возрастанию.

Решение задачи 1



Решение задачи 1 (сортировка)

Алгоритм решения задачи:

алг сортировка

вещ x, y, c

нач

 ввод x, y

 если $x > y$

 то $c := x$

$x := y$

$y := c$

кв

 вывод x, y

кон

Составной оператор

Этот пример иллюстрирует следующее правило Паскаля:

Если на какой-то из ветвей оператора ветвления находится несколько последовательных операторов, то их нужно записать между служебными словами `begin` и `end`.

Конструкция такого вида:

`begin` <последовательность операторов> `end`

называется **составным оператором**.

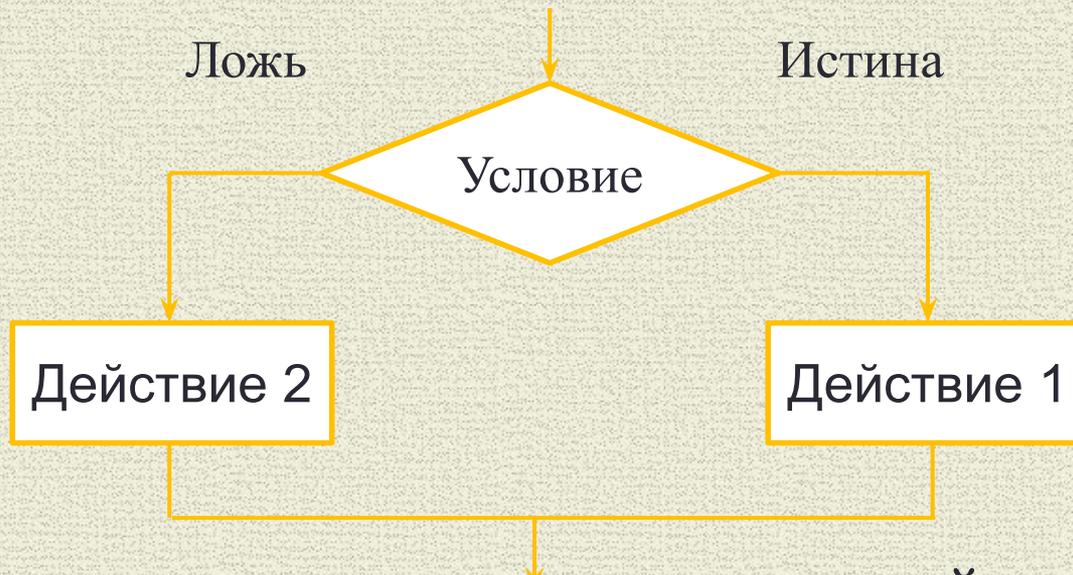
Решение задачи (сортировка)

Составим программу:

```
Program sorting;  
var x,y,c:real;  
begin  
  writeln ('Введи два числа');  
  readln (x,y);  
  if x>y then  
    begin  
      c:=x;  
      x:=y;  
      y:=c;  
    end;  
  writeln (x,',',y);  
end.
```

Полное ветвление

Полное ветвление - алгоритм, в котором выполняется одно из двух действий, в зависимости от истинности условия.



Если условие истинно, то выполняется действие 1, а иначе выполняется действие 2.

Запись условного оператора на Паскале

Полная форма оператора IF

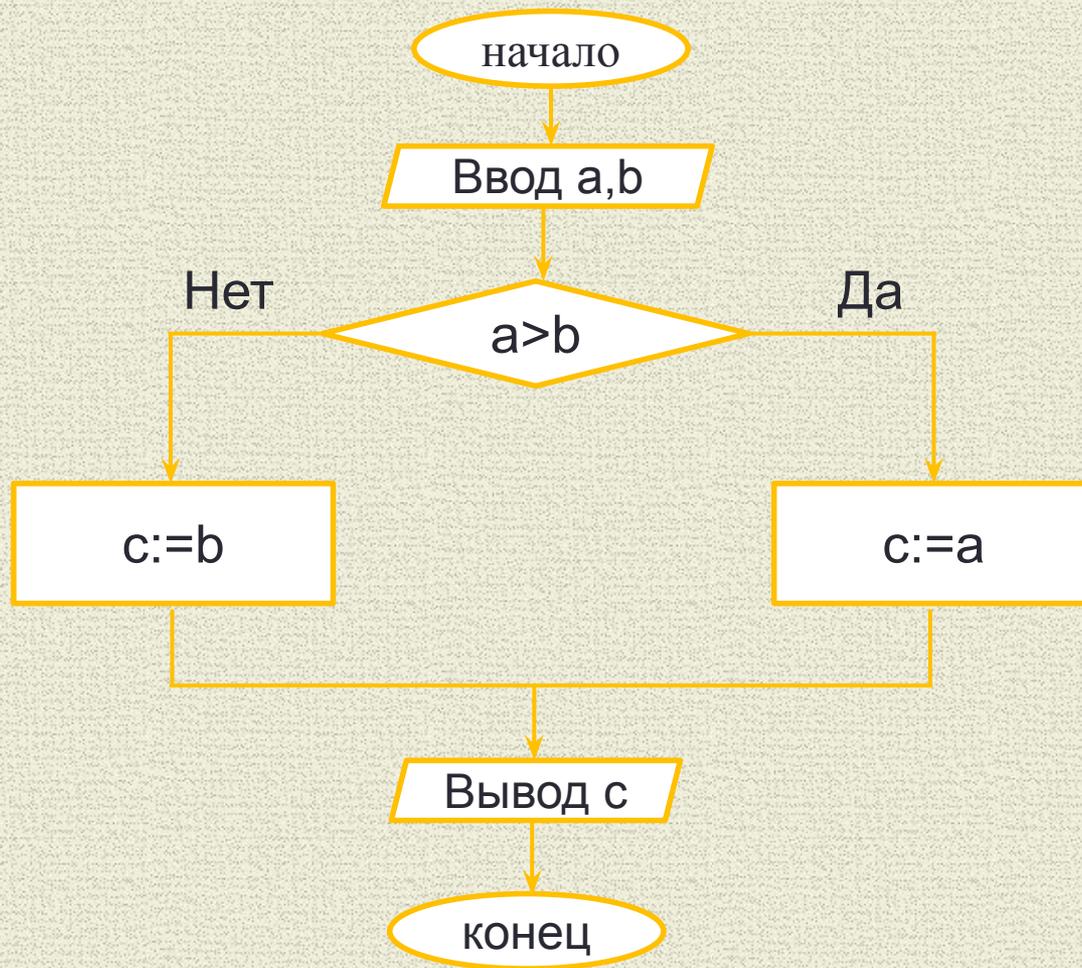
IF *<условие>* THEN *<оператор>* ELSE
<оператор>

Внимание! Перед ELSE
точка с запятой не ставится!

Задача 2

Даны два числа a , b . Выберите большее из них.

Задача 2



Задача 2

Алгоритм выбора большего из двух чисел, реализующий полное ветвление:

```
алг БИД
вещ a,b,c
нач
    ввод a,b
    если a>b
        то c:=a
        иначе c:=b
    кв
    вывод c
кон
```

Задача 2

Составим программу:

```
program bid;  
var a,b,c: real;  
begin  
    writeln ('введите значения переменных a,b');  
    readln (a,b);  
    if a>b then c:=a else c:=b;  
    writeln ('Большее число',' ',c:4:2)  
end.
```

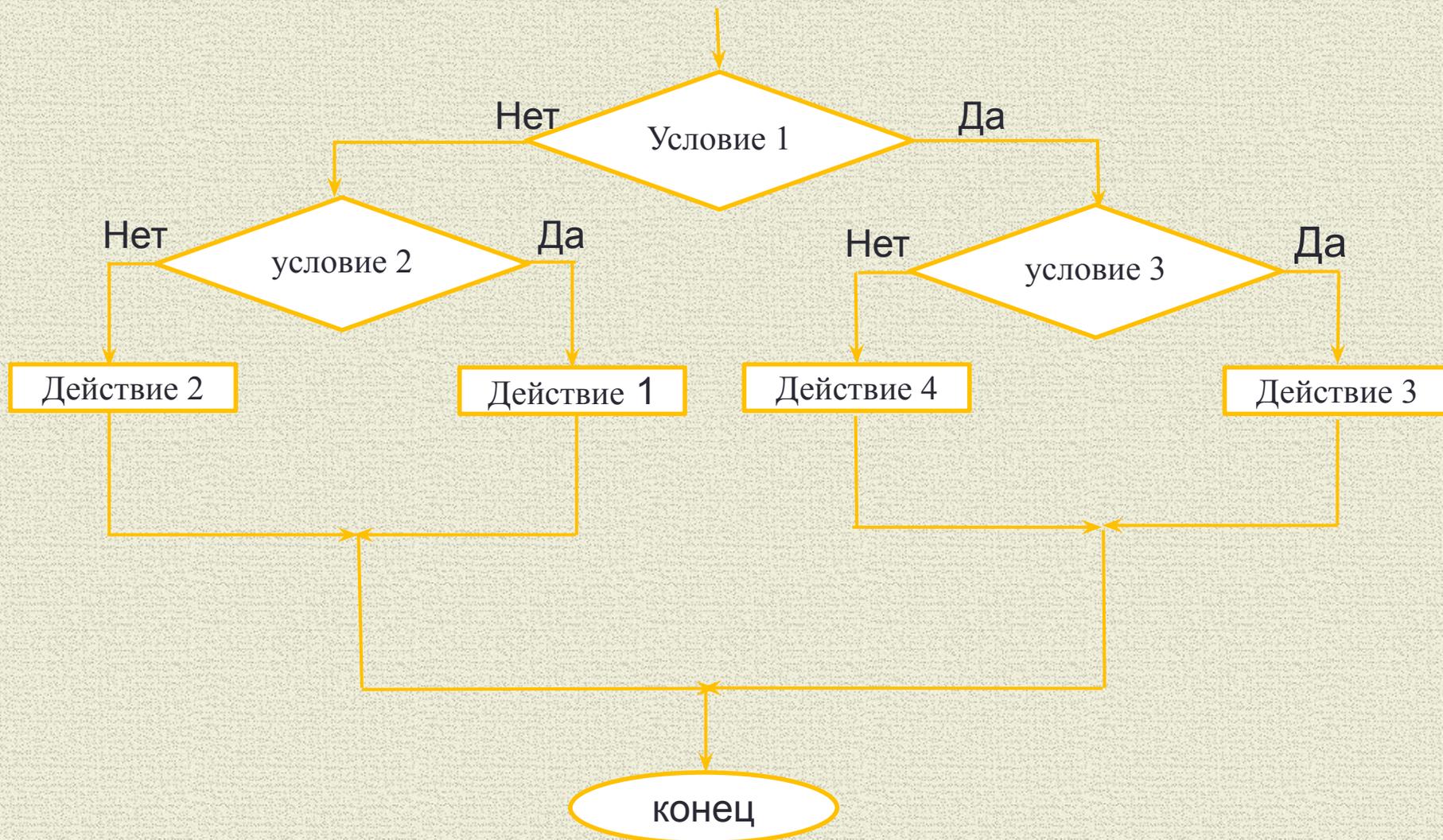
[Обратно](#)



Вложенное ветвление

В структуре вложенного ветвления следующая особенность: одна или обе ветви условия могут продолжаться не блоками вычислительных операций или ввода - вывода, а дополнительным блоком условия. Один из видов такого ветвления представлен на рисунке.

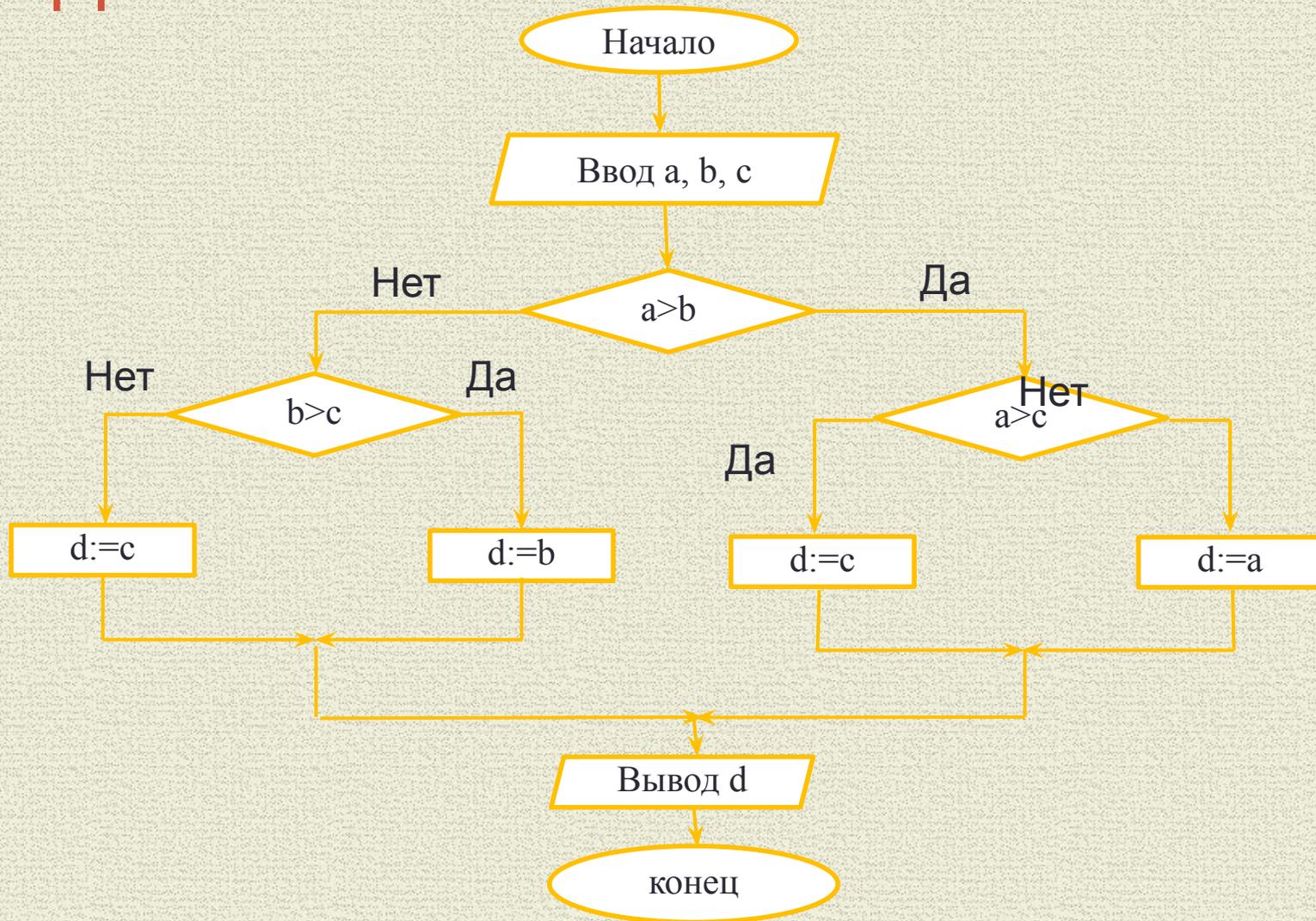
Вложенное ветвление



Задача 3

Определите большее из трех чисел a , b , c .

Задача 3



Задача 3

Структура этого алгоритма – вложенные ветвления:

алг БИТ

вещ a, b, c, d

нач

ввод a, b, c

если $a > b$

то если $a > c$ то $d := a$ иначе $d := c$ кв

иначе если $b > c$ то $d := b$ иначе $d := c$ кв

кв

вывод d

кон

Задача 3

Составим программу, используя вложенные ветвления:

```
program bit;
var
  a,b,c,d: real;
begin
  writeln ('введите значения переменных a,b,c');
  readln (a,b,c);
  if a>b
    then if a>c then d:=a else d:=b
    else if b>c then d:=b else d:=c;
  writeln ('Большее число', ' ',d)
end.
```

[Обратно](#)



Сложные логические выражения

Условие, содержащее логические связки (и, или, нет), называется сложным условием.

Условие, не содержащее логических связок, называется простым, или элементарным условием.

С логическими связками (операциями) вы встречались, когда работали с базами данных и электронными таблицами.

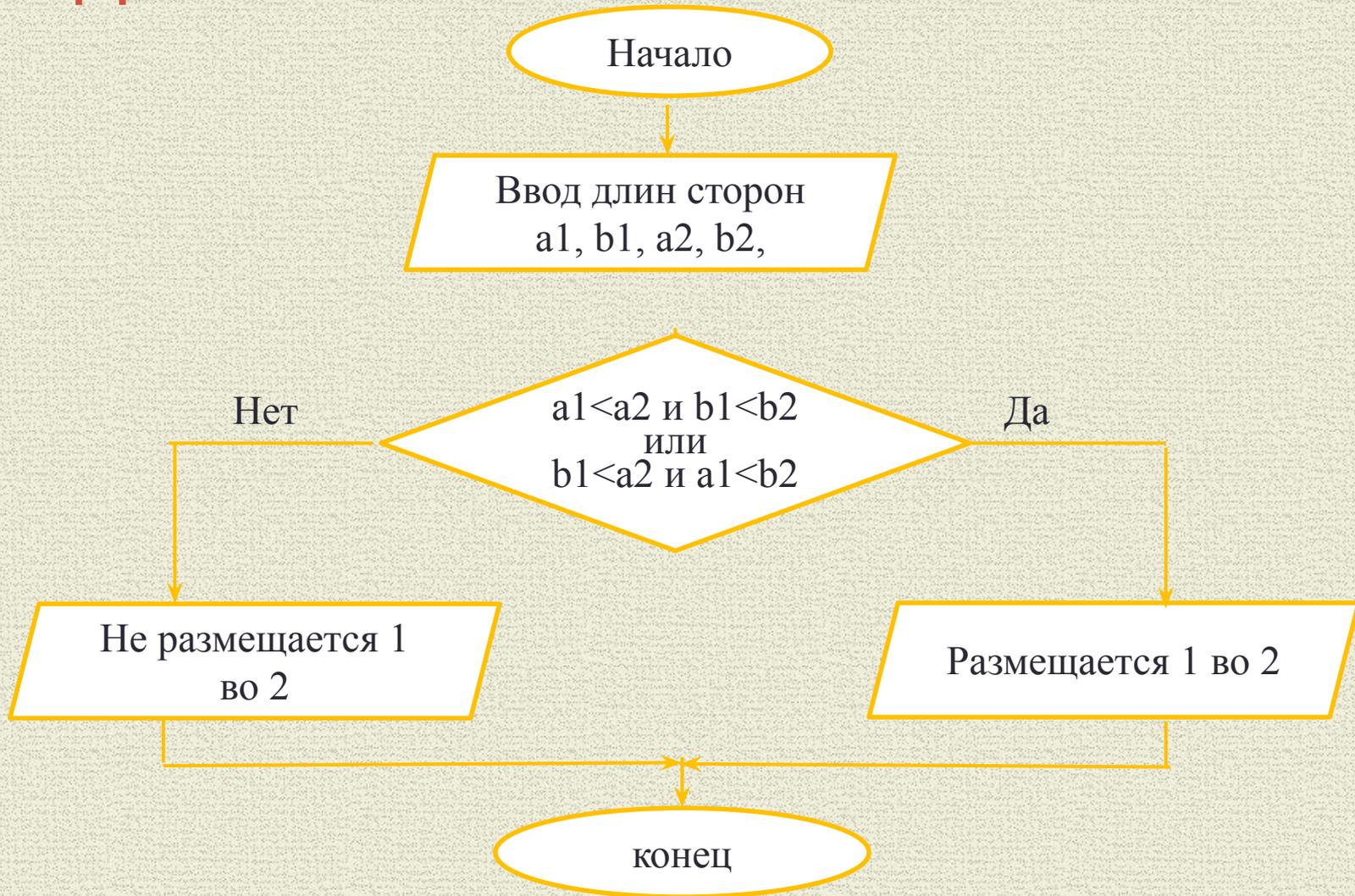
Логические операции

Логическая операция	Ее значение	Примеры записи	Значение примера
AND	Логическое "И"	$(x < 7) \text{ and } (x > 3)$	x меньше 7 И x больше 3 $(3 < x < 7)$
OR	Логическое "ИЛИ"	$(y > 100) \text{ or } (y < 10)$	y больше 100 ИЛИ y меньше 10 $(y < 10, y > 100)$
NOT	Логическое "НЕ"	$\text{not } (x = 2)$	НЕ x равно 2

Задача 4

Два прямоугольника заданы длинами сторон. Написать программу, после выполнения которой выясняется, можно ли первый прямоугольник целиком разместить во втором. (Рассмотреть только случай, когда соответствующие стороны прямоугольников параллельны.)

Задача 4



Задача 4

Пользуясь блок-схемой, составим программу, в которой должно быть реализовано полное ветвление и сложное условие (см. блок-схему):

```
Program Pryanoug;
```

```
  var
```

```
    a1, b1, a2, b2 : real;
```

```
  begin
```

```
    write ('Введите длину и ширину первого прямоугольника ');
```

```
    readln (a1, b1);
```

```
    write ('Введите длину и ширину второго прямоугольника ');
```

```
    readln (a2, b2);
```

```
    if ((a1 < a2) and (b1 < b2)) or ((b1 < a2) and (a1 < b2))
```

```
      then writeln('Первый прямоугольник размещается во втором')
```

```
      else writeln('Первый прямоугольник не размещается во втором')
```

```
  end.
```

Ветвление по ряду условий (оператор варианта case)

Условный оператор (**If**) позволяет сделать выбор из двух вариантов: да/нет (истина/ложь). Для организации выбора из нескольких вариантов приходится использовать вложенные условные операторы (**If**), тогда алгоритм и программа могут оказаться очень сложными, или оператор выбора **case**.

Ветвление по ряду условий (оператор варианта case)

Формат записи оператора case :

case <выражение порядкового типа> **of**

<значение1> : <оператор1>;

...

<значениеN> : <операторN>;

else <оператор>

End

Внимание! Единственный случай, когда перед словом ELSE можно ставить точку с запятой (;) это в операторе CASE!

Задача 5

Напишите программу, которая запрашивает у пользователя номер месяца и выводит соответствующее название времени года. В случае, если пользователь укажет недопустимое число, программа должна вывести сообщение «Ошибка ввода данных. Число должно быть от 1 до 12. Повторите ввод. ».

Задача 5

```
Program vremya_goda;  
var  
  m:integer;  
begin  
  writeln('Введите номер месяца (число от 1 до 12) m=');  
  readln (m);  
  Case m of  
    1,2,12: writeln ('Время года - зима');  
    3..5:writeln ('Время года - весна');  
    6..8:writeln ('Время года - лето');  
    9..11: writeln ('Время года - осень');  
    else writeln ('Ошибка ввода данных. Число должно быть  
от 1 до 12. Повторите ввод.');
```

end;
end.

Закрепление материала.

Контрольные вопросы:

1. Как схематически выглядит алгоритм с неполным ветвлением?
2. Как схематически выглядит алгоритм с вложенным ветвлением?
3. Как схематически выглядит алгоритм с полным ветвлением?
4. Как записывается условный оператор в полной форме?
5. Как записывается условный оператор в неполной форме?
6. Как записывается общий вид оператора case?

Задача 6

Напишите программу, которая считывает три целых числа (каждое с отдельной строки) и печатает 1, если среди них есть хотя бы одно число, большее удвоенной суммы двух других. Если таких чисел нет, то программа печатает 0. Программа должна выводить только 1 или 0. Известно, что каждое из исходных чисел по абсолютной величине не превосходит 1000.

Задача 6 (ДР в формате ЕГЭ)

```
Program zadacha_6;  
var a, b, c : integer;  
begin  
  writeln ('Введите значение a=');  
  readln(a);  
  writeln ('Введите значение b=');  
  readln(b);  
  writeln ('Введите значение c=');  
  readln(c);  
  if (a>(b+c)*2)or (b>(a+c)*2)or(c>(a+b)*2) then writeln(1)  
    else writeln(0);  
end.
```

[Обратно](#) 

Домашнее задание

§ 36 читать.

Придумать пример разветвляющегося алгоритма.

Индивидуальные задания:

Составить алгоритм нахождения наименьшего из двух элементов

Составить алгоритм нахождения наименьшего из трёх элементов

Составить алгоритм нахождения наибольшего из двух элементов

Составить алгоритм нахождения наибольшего из трёх элементов

Использованная литература и источники информации

1. И.Г. Семакин и др. «Информатика и ИКТ», учебник для 9 класса, БИНОМ, Москва, 2011;
2. Житкова О.А. Кудрявцева Е.К. «Справочные материалы по программированию на языке Паскаль», «ИНТЕЛЛЕКТ – ЦЕНТР», Москва, 2005;
3. С.Н. Лукин, «Turbo Pascal 7.0», самоучитель для начинающих, «ДИАЛОГ – МИФИ», Москва, 2005;
4. С.В. Вольский, П.А. Дмитриев «Turbo Pascal 7.0 для студентов и школьников», Наука и Техника, Санкт-Петербург, 2007;
5. Н.Культин «Turbo Pascal 7.0 в задачах и примерах», БХВ – Петербург, Санкт-Петербург, 2005;
6. <http://yf.kemsu.ru>