

# **Алгоритмические конструкции.**

## **Ветвление**

# Что такое алгоритм?

*Алгоритм – точное и понятное описание действий над заданными объектами, позволяющее получить конечный результат*

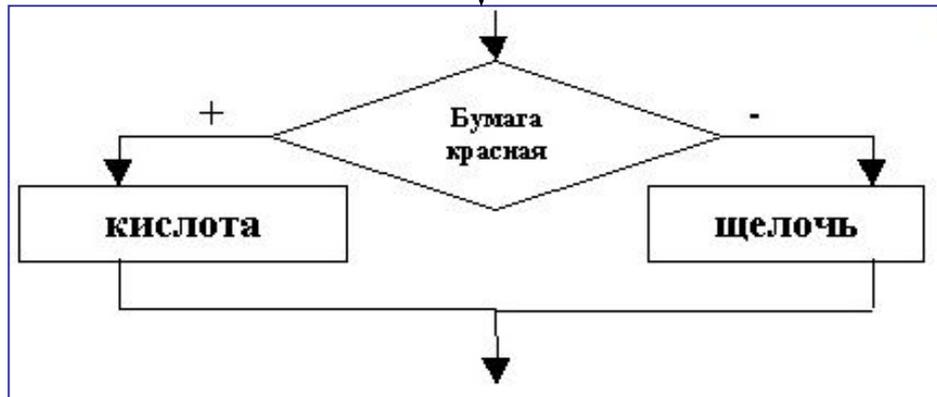
## Способы записи алгоритмов

- 1. Словесный**
- 2. Блок-схема**
- 3. Программа**

# Словесный      Блок-схема      Программа

если бумажка покраснела  
то раствор кислотный  
иначе раствор щелочной

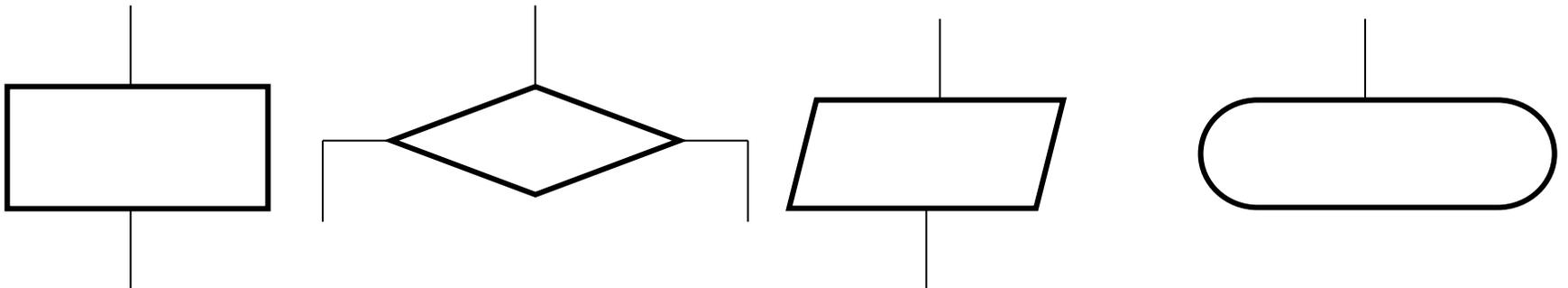
```
Program pr;  
Var a, b,s: real;  
Begin  
  Writeln ('Введите числа b и c');  
  Readln( b,c);  
  A=b+c;  
  Writeln ( 'a=',a);  
End.
```



# Блок-схемы

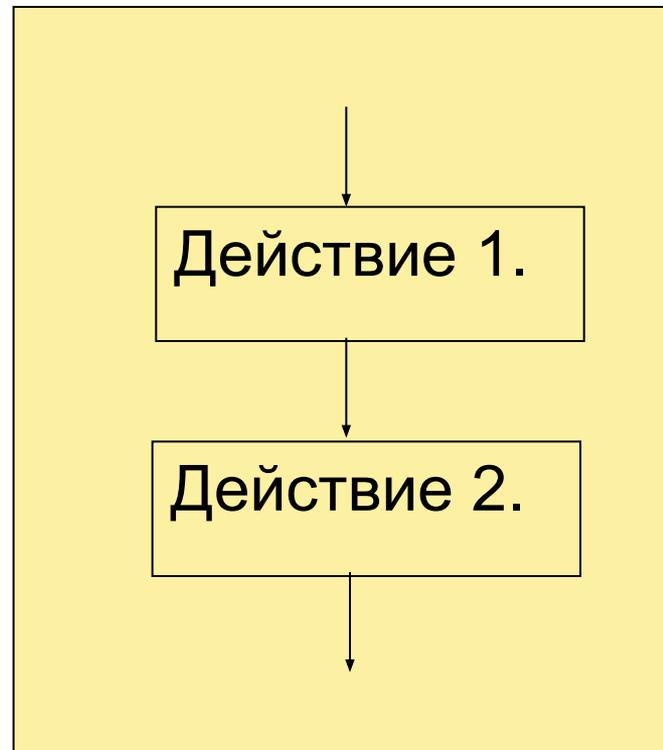
**Блок-схемы** являются графическим способом представления алгоритмов.

**Блок-схема** состоит из блоков, соединенных линиями.



## Какой алгоритм называется линейным?

Это группа шагов алгоритма, выполняемых последовательно друг за другом без каких-либо условий



# Что такое программа?

Алгоритм, записанный с помощью операторов языка программирования

## Программа на Турбо Паскаль

```
Program pr;  
Var a, b,s: real;  
Begin  
    Writeln ('Введите числа b и  
    c');  
    Readln( b,c);  
    A=b+c;  
    Writeln ( 'a=',a);  
End.
```

# СТРУКТУРА ПРОГРАММЫ НА ЯЗЫКЕ ТУРБО ПАСКАЛЬ

```
Program Name;      { название программы без пробелов }  
  
{ разделы описаний }  
  const      {раздел описания констант}  
  var        {раздел описания переменных (используется чаще всего)}  
  
{ основная программа }  
  Begin  
    { тело основной программы }  
    .....  
  End.
```

## ОПЕРАТОР ВЫВОДА ДАННЫХ

***WRITE (a1, a2, ... , an)***

***WRITELN (a1, a2, ... , an)***

## ОПЕРАТОР ВВОДА ДАННЫХ

***READ (a1, a2, ... , an)***

***READLN (a1, a2, ... , an)***

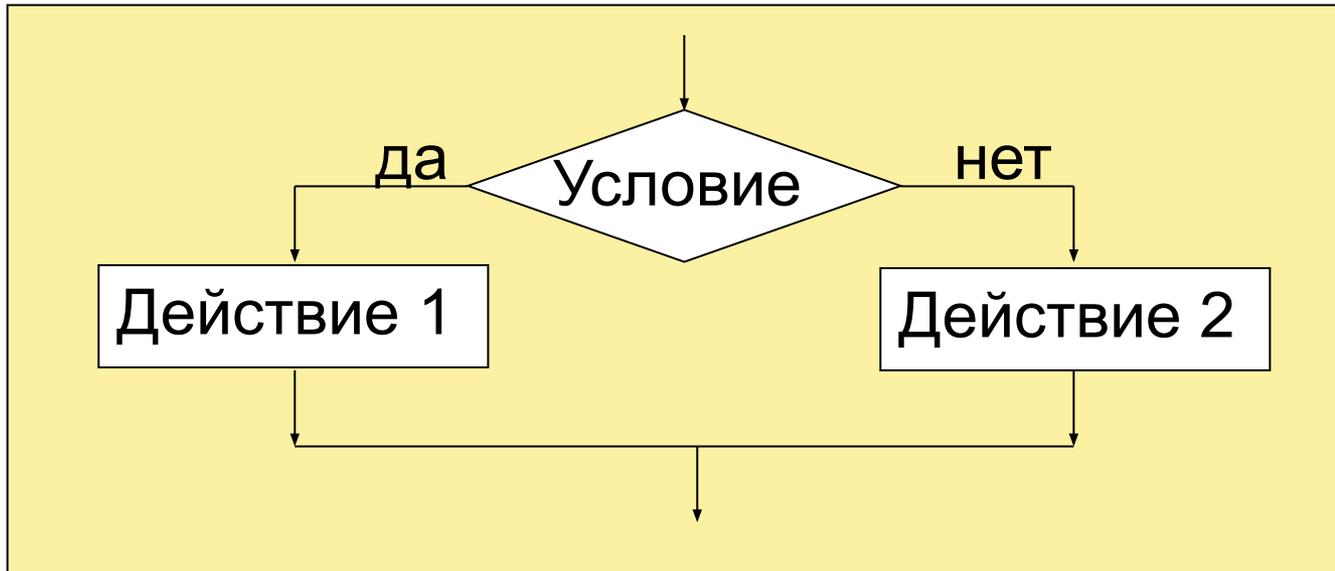
## ОПЕРАТОР ПРИСВАИВАНИЯ

**“ := ”**

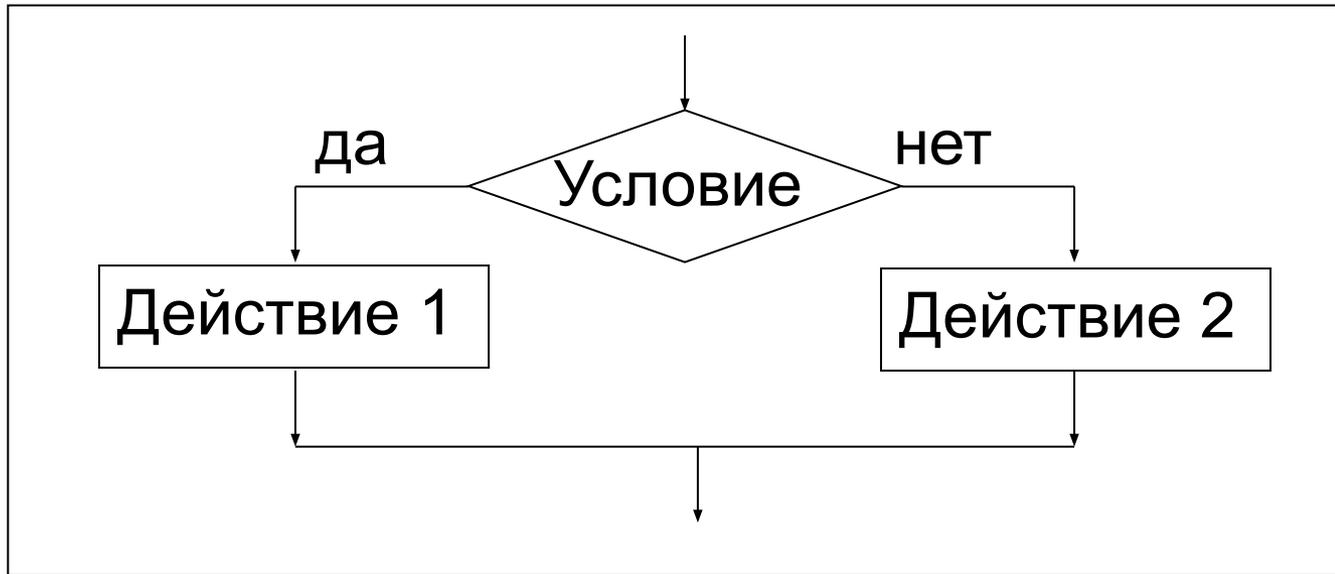
**A:=B;**

# Ветвление

**Ветвление** представляет собой алгоритмическую конструкцию, в которой выполнение того или иного шага зависит от **ИСТИННОСТИ** условия

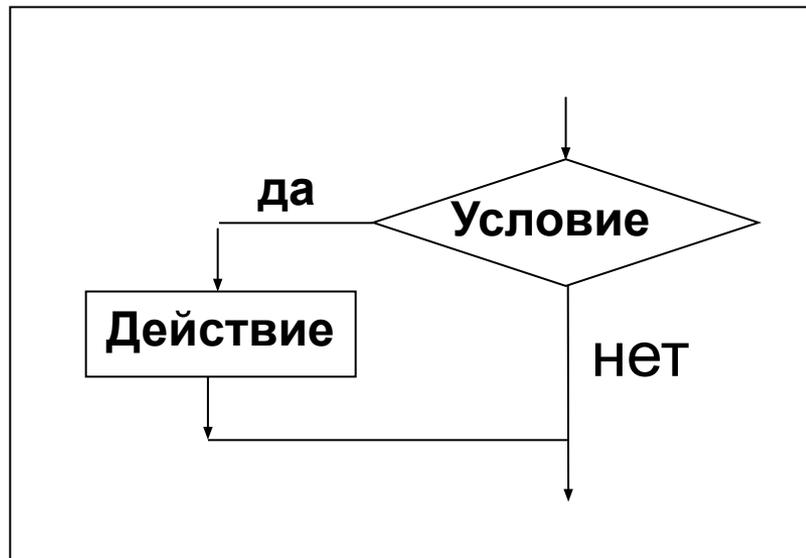


**блок-схема ветвления в полной форме**

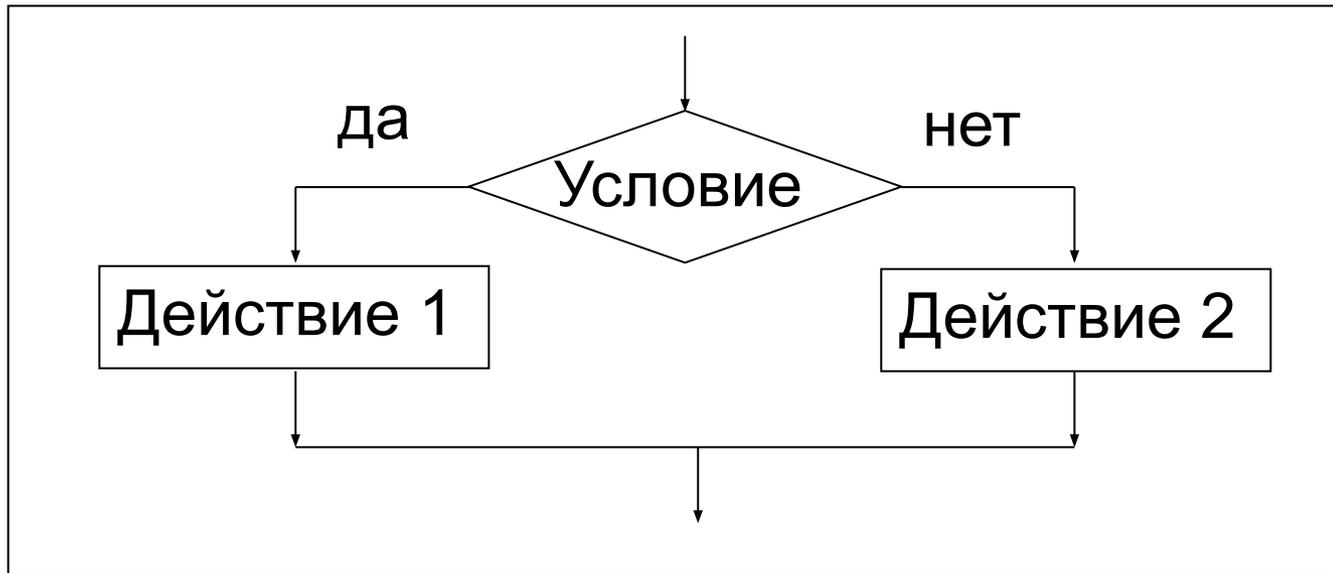


Конструкция **ветвления в полной форме** реализуется следующим образом. Если условие истинно, то выполняется **действие 1**, если условие ложно, то выполняется **действие 2**

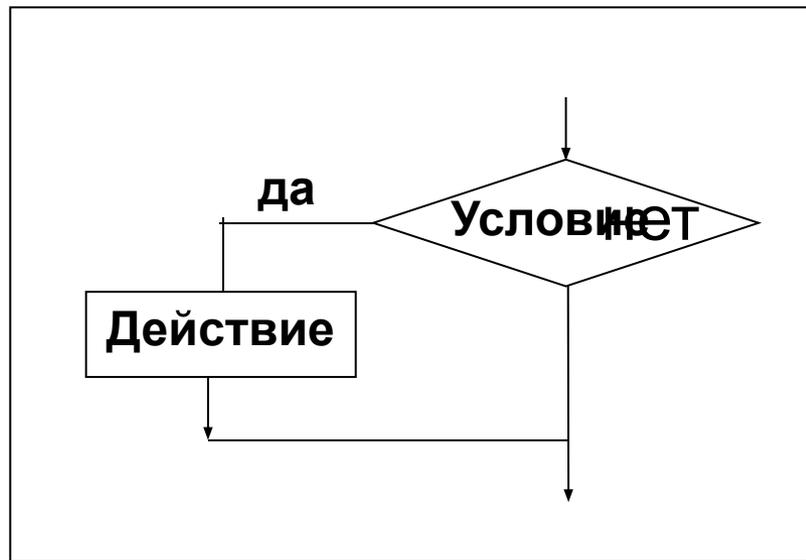
Если в ветвлении присутствуют действия только для истинности условия, то говорят, что конструкция записана в неполной (в сокращенной) форме



Если выбрано направление, в котором отсутствует действие, то конструкция ветвления **не выполняется** и управление получает конструкция, следующая за ветвлением.



**IF** <условие> **THEN** <оператор1> **ELSE** <оператор2>



**IF** <условие> **THEN** <оператор>

# АЛГОРИТМ РАБОТЫ УСЛОВНОГО ОПЕРАТОРА:

1. Вычисляется **<условие>**.
2. Если оно **TRUE** (истина), то выполняется **<оператор 1>**, а **<оператор 2>** пропускается.
3. Если результат - **FALSE** (ложь), наоборот **<оператор 1>** пропускается, а выполняется **<оператор 2>**.

# Задача 1

*Найти большее из двух чисел  $a$  и  $b$*

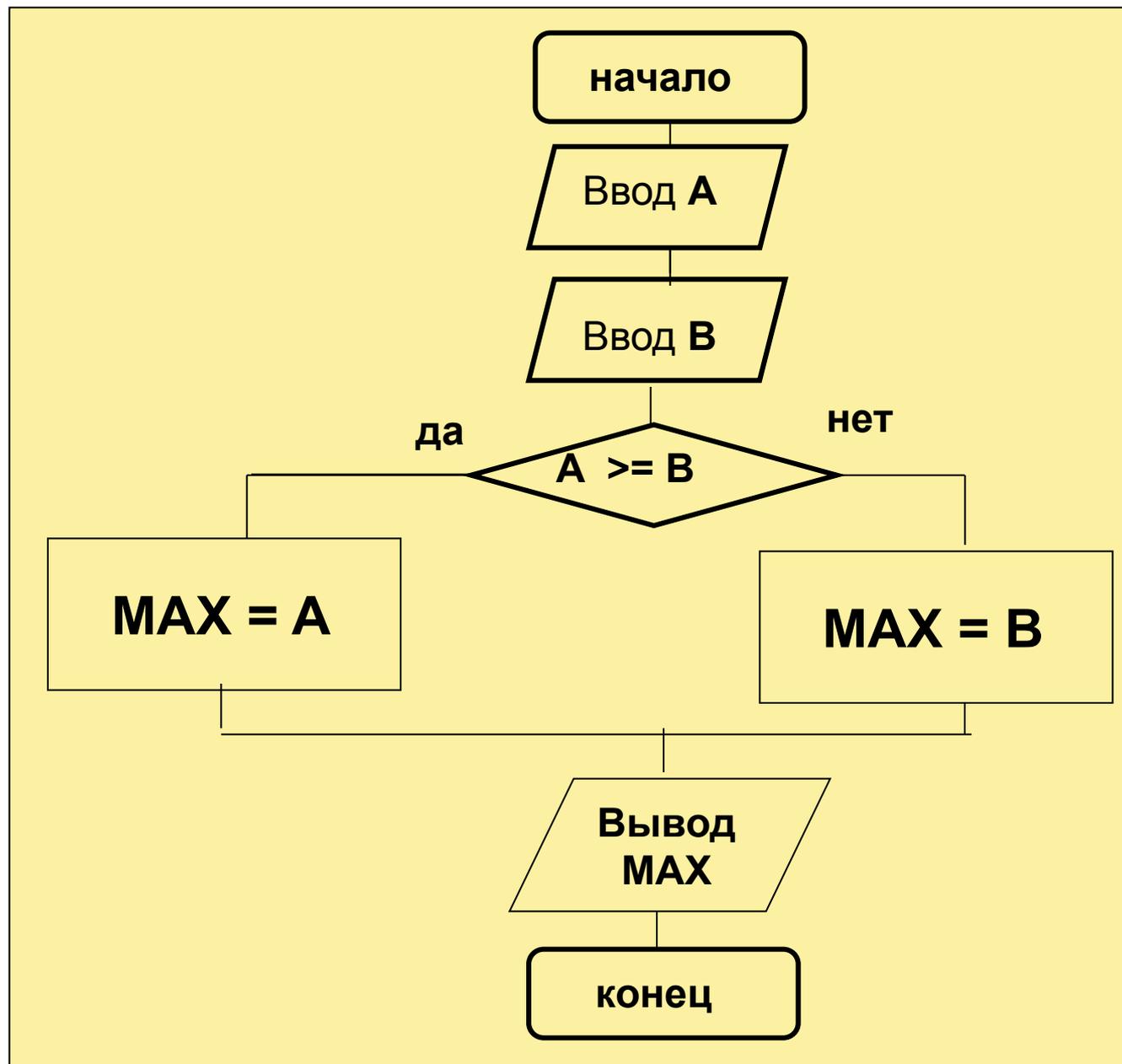
## Решение

В переменные  $A$  и  $B$  с клавиатуры вводятся числа. После их сравнения в переменной  $MAX$  должно находиться большее число.

## Алгоритм:

1. Ввод  $A$  и  $B$
2. ЕСЛИ  $A \geq B$  ТО  $MAX = A$   
ИНАЧЕ  $MAX = B$   
Конец ветвления
3. Конец

# Блок-схема



```
PROGRAM prim_1;  
  VAR a,b,max: integer;  
BEGIN
```

```
  write('Введите целое число a=');  
  readln(a);
```

```
  write('Введите целое число b=');  
  readln(b);
```

```
  IF a>=b THEN max:=a { если условие =true, то max:=a }  
    ELSE max:=b; { если условие =false, то max:=b }  
  writeln('Max=',max);
```

```
END.
```

условие

Оператор\_1

Оператор\_2

# Домашнее задание

- 1. Найти большее из трех чисел.**  
*(Составить блок-схему и программу для решения задачи).*
- 2. Определить, будет ли квадрат числа равняться сумме квадратов его цифр.**  
*(Составить блок-схему и программу для решения задачи).*