

# СТРУКТУРА ВЕТВЛЕНИЯ

# Линейный алгоритм

- Линейный алгоритм – это набор команд, выполняемых последовательно во времени, друг за другом.



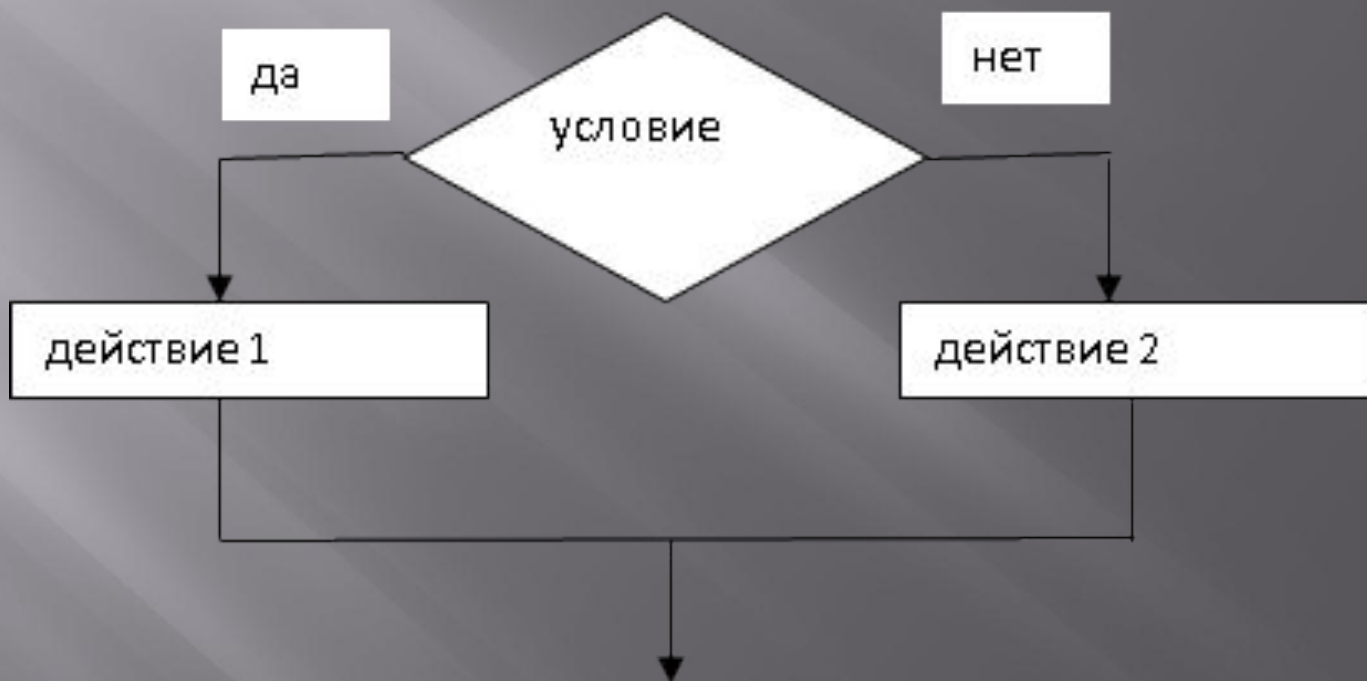
# Алгоритмическая структура «ветвление»

- Алгоритм, содержащий хотя бы одно условие, в результате которого обеспечивается переход на один из двух возможных шагов, называется разветвляющимся.

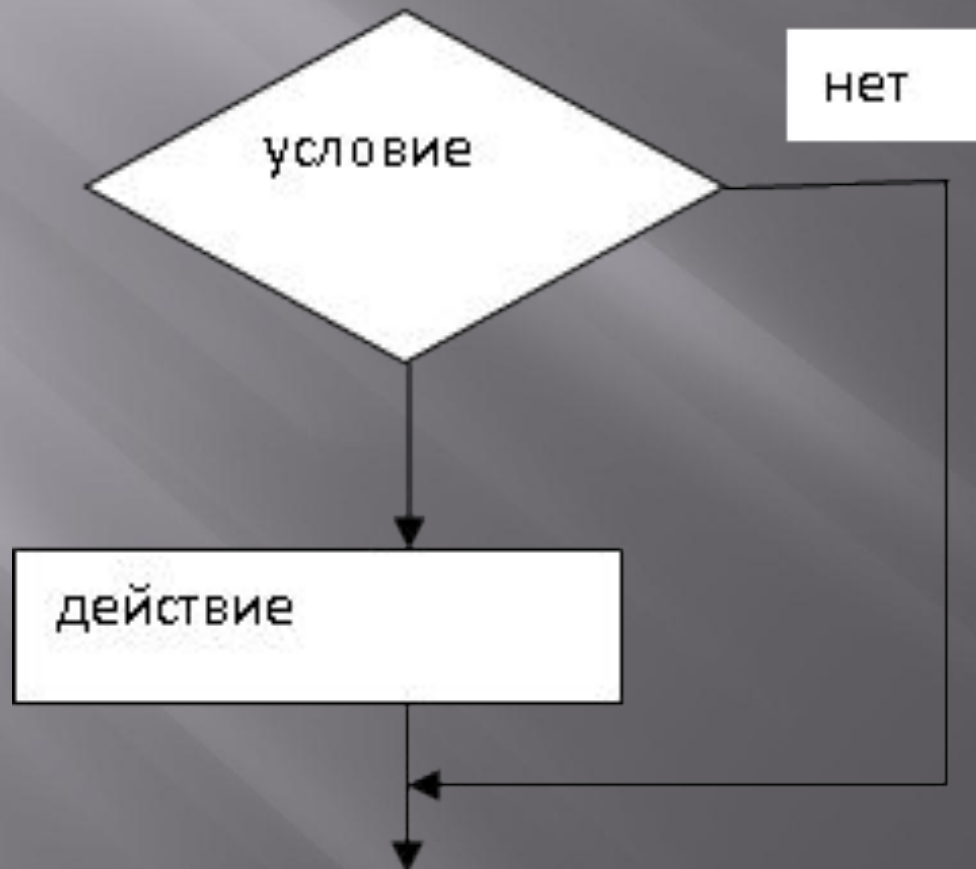
# Пример разветвляющегося алгоритма



# Полная форма



# Неполная форма



Общий вид словесной записи алгоритма ветвления в полной форме выглядит так:

*Если условие выполняется, то выполняется действие1, иначе выполняется действие2.*

Неполная форма выглядит так:

*Если условие выполняется, то выполняется действие.*

# Операции сравнения

$A < B$	A меньше B
$A \leq B$	A меньше или равно B
$A = B$	A равно B
$A > B$	A больше B
$A \geq B$	A больше или равно B
$A \neq B$ или $A > B$	A не равно B



# Оператор перехода

имеет вид

**GOTO <метка>**

Он позволяет передать управление непосредственно на нужный оператор программы. Перед этим оператором должна располагаться метка, отделенная от него двоеточием.

# Условный оператор IF

Условный оператор на языке программирования имеет следующий общий вид

```
IF <условие> Then <оператор1> Else  
<оператор2>;
```

где <условие>- выражение логического типа;  
<оператор1> выполняется, если условие верно,  
<оператор2> выполняется, если условие неверно.

# Вложенные условия

*Пример.*

**IF a>0**

**Then writeln('число положительное')**

**Else if a=0**

**Then writeln('число 0')**

**Else writeln('число положительное');**

# Операторные скобки **BEGIN...END**

объединяют несколько операторов в один операторный блок.

# Пример программы

```
Program primer2;  
  
    label 1;  
  
    var m,n:real;  
  
begin  
  
    clrscr;  
  
    Write('Введите вес мешковm,n=');  
  
    readln(m,n);  
  
    if m>n then begin  
        writeln('Вес 1 мешка больше, его вес:', m:2:2);  
        goto 1;  
    end;  
  
    if m<n then  
        writeln('Вес 2 мешка больше, его вес:', n:2:2)  
        else writeln('Вес мешков одинаков.', m:2:2);  
  
    1:readln;  
  
end.
```

# Итоги

- ▣ Алгоритм, в котором выполнение действия зависит от поставленного условия, называется алгоритмом ветвления.
- ▣ Алгоритмы ветвления могут быть полными и неполными.
- ▣ Условный оператор изменяет порядок действий в зависимости от некоторого условия, оператор может быть полным или неполным.
- ▣ Полный условный оператор состоит из условия IF и двух ветвей: положительной – THEN – и отрицательной – ELSE. В каждой из ветвей можно поместить по одному вложенному оператору.
- ▣ Неполный условный оператор состоит из условия IF и положительной ветви THEN.
- ▣ Операторные скобки BEGIN...END объединяют несколько операторов в один операторный блок.
- ▣ Операторный блок воспринимается как один оператор.