



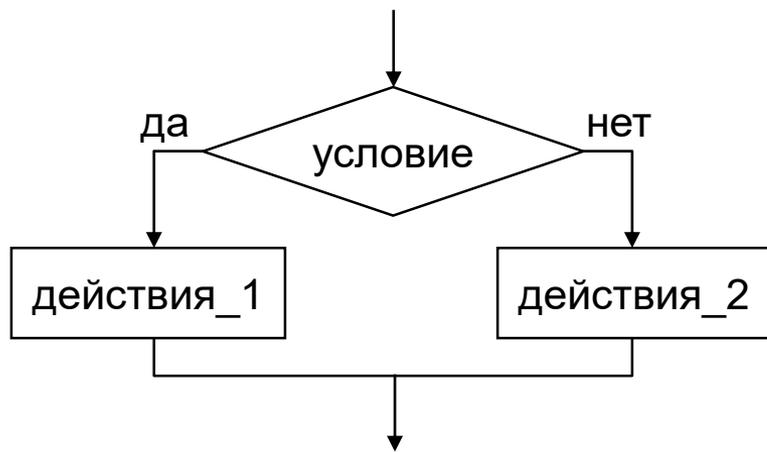
Язык программирования Python

Операторы ветвления в языке Python

МБОУ «Средняя общеобразовательная школа №11»
Учитель Информатики и ИКТ
Папулова Анна Сергеевна

Операторы ветвления

Полная форма ветвления (условного оператора):



```
if <условие>:  
    <блок_операторов_1>  
else:  
    <блок_операторов_2>
```

Если условие истинно, то выполняется блок_1, **иначе** – блок_2.

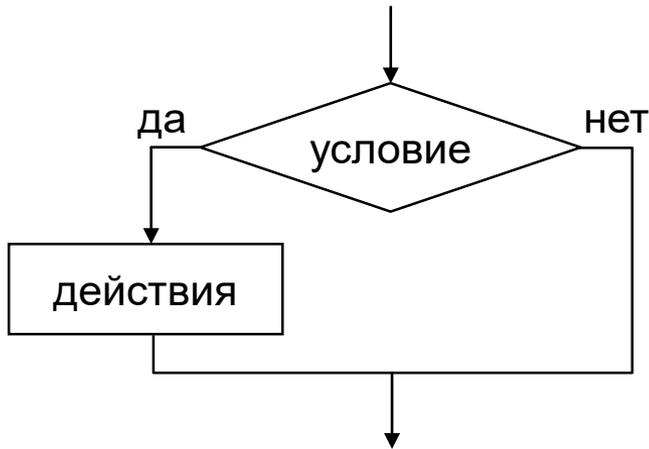
*В языке Python важную роль играют отступы операторов от левой границы текста программы. Начало и конец блоков операторов определяется этими сдвигами. Задаются отступы пробелами или клавишей **Tab**.*

Если в блоке всего один оператор, то допустима такая форма записи:

```
if <условие>: <оператор_1>  
else: <оператор_2>
```

Операторы ветвления

Неполная форма ветвления (условного оператора):



```
if <условие>:  
    <блок_операторов>
```

Если условие истинно, то выполняется блок операторов.
В противном случае – переход к следующему оператору программы.

Если в блоке всего один оператор, то допустима такая форма записи:

```
if <условие>: <оператор>
```

Записать в тетрадь!

| Название | Python |
|------------------|------------------------|
| равно | <code>x == y</code> |
| не равно | <code>x != y</code> |
| меньше | <code>x < y</code> |
| больше | <code>x > y</code> |
| меньше или равно | <code>x <= y</code> |
| больше или равно | <code>x >= y</code> |

Логические операторы

- *Логическое И* является бинарным оператором, имеет вид `and`. Оператор `and` возвращает `True` тогда и только тогда, когда оба его операнда имеют значение `True`.
- *Логическое ИЛИ* является бинарным оператором и возвращает `True` тогда и только тогда, когда хотя бы один операнд равен `True`. Оператор “логическое ИЛИ” имеет вид `or`.
- *Логическое НЕ* (отрицание) является унарным (то есть с одним операндом) оператором и имеет вид `not`, за которым следует единственный операнд. Логическое `НЕ` возвращает `True`, если операнд равен `False` и наоборот.

Решение квадратных уравнений на языке программирования

Входные данные: коэффициенты (старший, второй и свободный член)

Три условия:

1. Дискриминант меньше нуля
2. Дискриминант равен нулю
3. Дискриминант больше нуля

Команды по три человека, каждый выполняет свое условие.

Проверка работы программы по следующим наборам входных данных:

| 1 набор | 2 набор | 3 набор | 4 набор |
|----------|------------|----------|----------|
| $a = 1$ | $a = 4$ | $a = 1$ | $a = 1$ |
| $b = -2$ | $b = 2$ | $b = -5$ | $b = -2$ |
| $c = 1$ | $c = 0,25$ | $c = 6$ | $c = 10$ |