
§ 1.4

Программирование разветвляющихся алгоритмов

Ключевые слова:

- условный оператор
- неполный условный оператор
- составной оператор
- каскадное ветвление

1.4.1. Условный оператор

При записи на языке Python разветвляющихся алгоритмов используют **условный оператор**. Его общий вид:

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

Слова **if** и **else** начинаются на одном уровне, а все команды внутренних блоков сдвинуты относительно этого уровня вправо на одно и то же расстояние. Начало и конец блока, который выполняется при истинности (ложности) условия, определяется именно этими сдвигами.

Обратите внимание! В Python сдвиги (отступы) операторов относительно левой границы влияют на работу программы. Для отступа можно использовать пробелы (обычно не меньше двух) или символы табуляции (вставляются при нажатии на клавишу *Tab*).

Если при истинности (ложности) какого-то условия предполагается выполнить по одному действию, то условный оператор может быть записан в одну строку:

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

Для записи неполных ветвлений используется неполная форма условного оператора:

```
if <условие>:  
    <оператор>
```

В качестве условий используются логические выражения:

- простые — записанные с помощью операций отношения (<, >, >=, <=, != (не равно), == (равно));
- составные — записанные с помощью логических операций (**and**, **or**, **not**).

В языке Python разрешены двойные неравенства, например $A < B < C$.

Пример. Напишем на языке Python рассмотренный в п. 2.4.2 (пример 8) учебника для 8 класса алгоритм определения принадлежности точки x отрезку $[a, b]$.

```
# Программа 9
print('Определение принадлежности точки отрезку')
a = int(input('Введите a: '))
b = int(input('Введите b: '))
x = int(input('Введите x: '))
if x >= a and x <= b:
    print('Точка принадлежит отрезку')
else:
    print('Точка не принадлежит отрезку')
```

Пример. Воспользуемся неполным условным оператором для записи на языке Python рассмотренного в п. 2.4.2 (пример 9) учебника для 8 класса алгоритма присваивания переменной y значения наибольшей из трёх величин a , b и c .

```
# Программа 10
print ('Нахождение наибольшей из трёх величин')
a = int(input('Введите a: '))
b = int(input('Введите b: '))
c = int(input('Введите c: '))
y = a
if b > y:
    y = b
if c > y:
    y = c
print('y=', y)
```

Дополните эту программу так, чтобы её выполнение приводило к нахождению минимального значения из четырёх величин a , b , c и d .



Пример. Уравнение вида $ax^2 + bx + c = 0$, где x — переменная, a , b и c — некоторые числа, причём $a \neq 0$, называется квадратным уравнением. Алгоритм решения квадратного уравнения вам хорошо известен. Запишем соответствующую программу на языке Python:

```
# Программа 11
from math import *
print('Решение квадратного уравнения')
print('Введите коэффициенты a, b, c>>')
a = float(input('a='))
b = float(input('b='))
c = float(input('c='))
d = b * b - 4 * a * c
if d < 0:
    print ('Корней нет')
if d == 0:
    x = - b / (2 * a)
    print('Корень уравнения x=', "{:6.4f}".format(x))
if d > 0:
    x1 = (-b + sqrt(d)) / (2 * a)
    x2 = (-b - sqrt(d)) / (2 * a)
    print('Корни уравнения:')
    print('x1=', "{:6.4f}".format(x1))
    print('x2=', "{:6.4f}".format(x2))
```