

# Лабораторная работа №2

## Условия

### 1. Синтаксис условной инструкции

Все ранее рассматриваемые программы имели линейную структуру: все инструкции выполнялись последовательно одна за одной, каждая записанная инструкция обязательно выполняется.

Допустим мы хотим по данному числу  $x$  определить его абсолютную величину (модуль). Программа должна напечатать значение переменной  $x$ , если  $x > 0$  или же величину  $-x$  в противном случае. Линейная структура программы нарушается: в зависимости от справедливости условия  $x > 0$  должна быть выведена одна или другая величина. Соответствующий фрагмент программы на Питоне имеет вид:

```
x = int(input())
if x > 0:
    print(x)
else:
    print(-x)
```

В этой программе используется условная инструкция `if` (если). После слова `if` указывается проверяемое условие ( $x > 0$ ), завершающееся двоеточием. После этого идет блок (последовательность) инструкций, который будет выполнен, если условие истинно, в нашем примере это вывод на экран величины  $x$ . Затем идет слово `else` (иначе), также завершающееся двоеточием, и блок инструкций, который будет выполнен, если проверяемое условие неверно, в данном случае будет выведено значение  $-x$ .

Итак, условная инструкция в Питоне имеет следующий синтаксис:

```
if Условие:
    Блок инструкций 1
else:
    Блок инструкций 2
```

*Блок инструкций 1* будет выполнен, если *Условие* истинно. Если *Условие* ложно, будет выполнен *Блок инструкций 2*.

В условной инструкции может отсутствовать слово `else` и последующий блок. Такая инструкция называется неполным ветвлением. Например, если дано число  $x$  и мы хотим заменить его на абсолютную величину  $x$ , то это можно сделать следующим образом:

```
x = int(input())
if x < 0:
    x = -x
print(x)
```

В этом примере переменной  $x$  будет присвоено значение  $-x$ , но только в том случае, когда  $x < 0$ . А вот инструкция `print(x)` будет выполнена всегда, независимо от проверяемого условия.

Для выделения блока инструкций, относящихся к инструкции `if` или `else` в языке Питон используются *отступы*. Все инструкции, которые относятся к одному блоку, должны иметь равную величину отступа, то есть *одинаковое число пробелов* в начале строки. Рекомендуется использовать отступ в 4 пробела и не рекомендуется использовать в качестве отступа символ табуляции.

Это одно из существенных отличий синтаксиса Питона от синтаксиса большинства языков, в которых блоки выделяются специальными словами, например, `нц . . . кц` в Кумире, `begin . . . end` в Паскале или фигурными скобками в Си.

## 2. Вложенные условные инструкции

Внутри условных инструкций можно использовать любые инструкции языка Питон, в том числе и условную инструкцию. Получаем вложенное ветвление – после одной развилки в ходе исполнения программы появляется другая развилка. При этом вложенные блоки имеют больший размер отступа (например, 8 пробелов). Покажем это на примере программы, которая по данным ненулевым числам  $x$  и  $y$  определяет, в какой из четвертей координатной плоскости находится точка  $(x,y)$ :

```
x = int(input())
y = int(input())
if x > 0:
    if y > 0:                # x > 0, y > 0
        print("Первая четверть")
    else:                   # x > 0, y < 0
        print("Четвертая четверть")
else:
    if y > 0:                # x < 0, y > 0
        print("Вторая четверть")
    else:                   # x < 0, y < 0
        print("Третья четверть")
```

Входные данные: 2, -3. Выходные данные: Четвертая четверть.

В этом примере мы использовали *комментарии* – текст, который интерпретатор игнорирует. Комментариями в Питоне является символ `#` и весь текст после этого символа до конца строки.

## 3. Операторы сравнения

Как правило, в качестве проверяемого условия используется результат вычисления одного из следующих операторов сравнения:

<	Меньше	Условие верно, если первый операнд меньше второго.
>	Больше	Условие верно, если первый операнд больше второго.
<=	Меньше или равно	
>=	Больше или равно	
==	Равенство	Условие верно, если два операнда равны.
!=	Неравенство	Условие верно, если два операнда неравны.

Например, условие  $(x * x < 1000)$  означает “значение  $x * x$  меньше 1000”, а условие  $(2 * x != y)$  означает “удвоенное значение переменной  $x$  не равно значению переменной  $y$ ”.

Операторы сравнения в Питоне можно объединять в цепочки (в отличие от большинства других языков программирования, где для этого нужно использовать логические связки), например,  $a == b == c$  или  $1 <= x <= 10$ .

## 4. Тип данных bool

Операторы сравнения возвращают значения специального логического типа bool. Значения логического типа могут принимать одно из двух значений: True(истина) или False (ложь). Если преобразовать логическое True к типу int, то получится 1, а преобразование False даст 0. При обратном преобразовании число 0 преобразуется в False, а любое ненулевое число в True. При преобразовании str в bool пустая строка преобразовывается в False, а любая непустая строка в True.

#### 4.1. Логические операторы

Иногда нужно проверить одновременно не одно, а несколько условий. Например, проверить, является ли данное число четным можно при помощи условия  $(n \% 2 == 0)$  (остаток от деления  $n$  на 2 равен 0), а если необходимо проверить, что два данных целых числа  $n$  и  $m$  являются четными, необходимо проверить справедливость обоих условий:  $n \% 2 == 0$  и  $m \% 2 == 0$ , для чего их необходимо объединить при помощи оператора and (логическое И):  $n \% 2 == 0$  and  $m \% 2 == 0$ .

В Питоне существуют стандартные логические операторы: логическое И, логическое ИЛИ, логическое отрицание.

*Логическое И* является бинарным оператором (то есть оператором с двумя операндами: левым и правым) и 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.

Пример. Проверим, что хотя бы одно из чисел  $a$  или  $b$  оканчивается на 0:

```
a = int(input())
b = int(input())
if a % 10 == 0 or b % 10 == 0:
    print('YES')
else:
    print('NO')
```

Проверим, что число  $a$  — положительное, а  $b$  — неотрицательное:

```
if a > 0 and not (b < 0):
```

Или можно вместо not ( $b < 0$ ) записать ( $b \geq 0$ ).

#### 5. Каскадные условные инструкции

Пример программы, определяющий четверть координатной плоскости, можно переписать используя “каскадную” последовательность операций if... elif... else:

```
x = int(input())
y = int(input())
if x > 0 and y > 0:
    print("Первая четверть")
elif x > 0 and y < 0:
```

```
print("Четвертая четверть")
elif y > 0:
    print("Вторая четверть")
else:
    print("Третья четверть")
```

В такой конструкции условия `if`, ..., `elif` проверяются по очереди, выполняется блок, соответствующий первому из истинных условий. Если все проверяемые условия ложны, то выполняется блок `else`, если он присутствует.

## 6. Вычисления

### 6.1 Целочисленная арифметика

Для целых чисел определены операции `+`, `-`, `*` и `**`. Операция деления `/` для целых чисел возвращает вещественное число (значение типа `float`). Также функция возведения в степень возвращает значение типа `float`, если показатель степени — отрицательное число.

Но есть и специальная операция целочисленного деления, выполняющегося с отбрасыванием дробной части, которая обозначается `//` (она соответствует операции `div` в Паскале). Она возвращает целое число: целую часть частного. Другая близкая ей операция — это операция взятия остатка от деления, обозначаемая `%` (она соответствует операции `mod` в Паскале). Например:

```
print(17 / 3)    # выведет 5.666666666667
print(17 // 3)   # выведет 5
print(17 % 3)    # выведет 2
```

### 6.2 Действительные числа

В этом разделе речь пойдет о действительных числах, имеющих тип `float`.

Обратите внимание, что если вы хотите считать с клавиатуры действительное число, то результат, возвращаемый функцией `input()` необходимо преобразовывать к типу `float`:

```
x = float(input())
print(x)
```

Действительные (вещественные) числа представляются в виде чисел с десятичной точкой (а не запятой, как принято при записи десятичных дробей в русских текстах). Для записи очень больших или очень маленьких по модулю чисел используется так называемая запись «с плавающей точкой» (также называемая «научная» запись). В этом случае число представляется в виде некоторой десятичной дроби, называемой мантиссой, умноженной на целочисленную степень десяти (порядок). Например, расстояние от Земли до Солнца равно  $1.496 \cdot 10^{11}$ , а масса молекулы воды  $2.99 \cdot 10^{-23}$ .

Числа с плавающей точкой в программах на языке Питон, а также при вводе и выводе записываются так: сначала пишется мантисса, затем пишется буква `e`, затем пишется порядок. Пробелы внутри этой записи не ставятся. Например, указанные выше константы можно записать в виде `1.496e11` и `2.99e-23`. Перед самым числом также может стоять знак минус.

Напомним, что результатом операции деления `/` всегда является действительное число (`float`), в то время как результатом операции `//` является целое число (`int`).

Преобразование действительных чисел к целому производится с округлением в сторону нуля, то есть `int(1.7) == 1`, `int(-1.7) == -1`.

### 6.3. Библиотека `math`

Для проведения вычислений с действительными числами язык Питон содержит много дополнительных функций, собранных в библиотеку (модуль), которая называется `math`.

Для использования этих функций в начале программы необходимо подключить математическую библиотеку, что делается командой

```
import math
```

Например, пусть мы хотим округлять вещественные числа до ближайшего целого числа вверх. Соответствующая функция `ceil` от одного аргумента вызывается, например, так: `math.ceil(x)` (то есть явно указывается, что из модуля `math` используется функция `ceil`). Вместо числа `x` может быть любое число, переменная или выражение. Функция возвращает значение, которое можно вывести на экран, присвоить другой переменной или использовать в выражении:

```
import math
x = math.ceil(4.2)
y = math.ceil(4.8)
print(x)
print(y)
```

Другой способ использовать функции из библиотеки `math`, при котором не нужно будет при каждом использовании функции из модуля `math` указывать название этого модуля, выглядит так:

```
from math import ceil
```

```
x = 7 / 2
y = ceil(x)
print(y)
```

или так:

```
from math import *

x = 7 / 2
y = ceil(x)
print(y)
```

Ниже приведен список основных функций модуля `math`. Более подробное описание этих функций можно найти на сайте с документацией языка Питон.

Некоторые из перечисленных функций (`int`, `round`, `abs`) являются стандартными и не требуют подключения модуля `math` для использования.

Функция	Описание
<b>Округление</b>	
<b><code>int(x)</code></b>	Округляет число в сторону нуля. Это стандартная функция, для ее использования не нужно подключать модуль <code>math</code> .
<b><code>round(x)</code></b>	Округляет число до ближайшего целого. Если дробная часть числа равна 0.5, то число округляется до ближайшего четного числа.
<b><code>round(x, n)</code></b>	Округляет число <code>x</code> до <code>n</code> знаков после точки. Это стандартная функция, для ее использования не нужно подключать модуль <code>math</code> .
<b><code>floor(x)</code></b>	Округляет число вниз («пол»), при этом <code>floor(1.5) == 1</code> , <code>floor(-1.5) == -2</code>
<b><code>ceil(x)</code></b>	Округляет число вверх («потолок»), при этом <code>ceil(1.5) == 2</code> , <code>ceil(-1.5) == -1</code>
<b><code>abs(x)</code></b>	Модуль (абсолютная величина). Это — стандартная функция.
<b>Корни, логарифмы</b>	
<b><code>sqrt(x)</code></b>	Квадратный корень. Использование: <code>sqrt(x)</code>
<b><code>log(x)</code></b>	Натуральный логарифм. При вызове в виде <code>log(x, b)</code> возвращает логарифм по основанию <code>b</code> .
<b><code>e</code></b>	Основание натуральных логарифмов <code>e = 2,71828...</code>

Тригонометрия	
<b>sin (x)</b>	Синус угла, задаваемого в радианах
<b>cos (x)</b>	Косинус угла, задаваемого в радианах
<b>tan (x)</b>	Тангенс угла, задаваемого в радианах
<b>asin (x)</b>	Арксинус, возвращает значение в радианах
<b>acos (x)</b>	Арккосинус, возвращает значение в радианах
<b>atan (x)</b>	Арктангенс, возвращает значение в радианах
<b>atan2 (y, x)</b>	Полярный угол (в радианах) точки с координатами (x, y).
<b>degrees (x)</b>	Преобразует угол, заданный в радианах, в градусы.
<b>radians (x)</b>	Преобразует угол, заданный в градусах, в радианы.
<b>pi</b>	Константа $\pi = 3.1415...$

## 7. Практические задания

1. Определить, в какую из областей — I или II (рис. 1) — попадает точка с заданными координатами. Для простоты принять, что точка не попадает на границу областей.

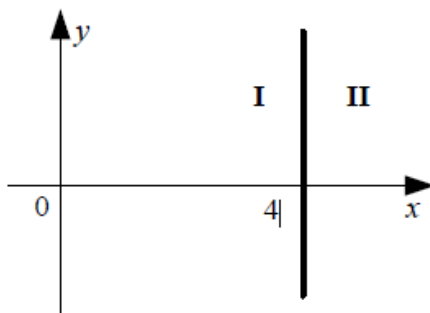


Рис.1

2. Определить, в какую из областей — I или II (рис. 2) — попадает точка с заданными координатами. Для простоты принять, что точка не попадает на границу областей.

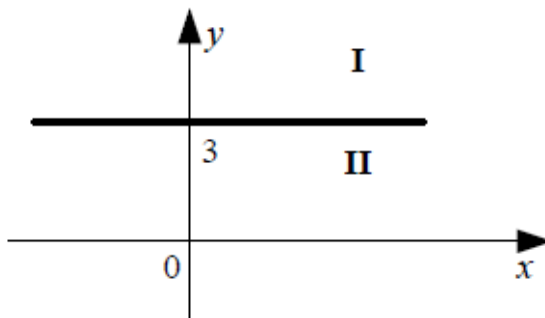


Рис. 2

3. Для функций, заданных графически (рис. 3), определить значение  $y$  при заданном значении  $x$ .

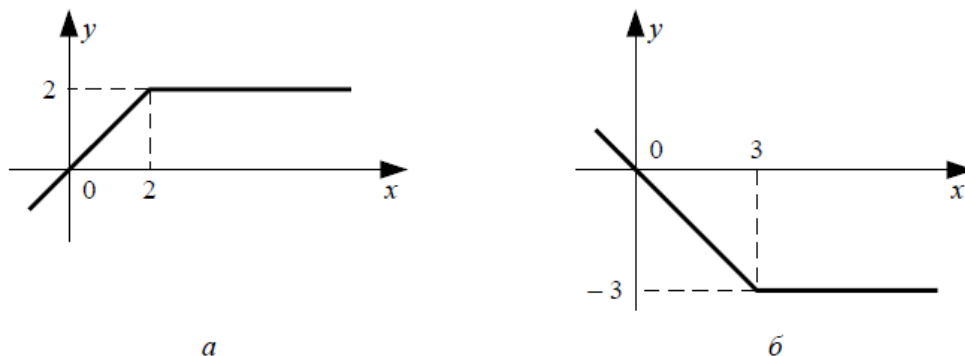


Рис. 3

4. Даны два различных вещественных числа. Определить: а) какое из них больше; б) какое из них меньше.
5. Определить максимальное и минимальное значения из двух различных вещественных чисел.
6. Известны год и номер месяца рождения человека, а также год и номер месяца сегодняшнего дня (январь — 1 и т. д.). Определить возраст человека (число полных лет). В случае совпадения указанных номеров месяцев считать, что прошел полный год.
7. Рассчитать значение  $y$  при заданном значении  $x$ :

$$y = \begin{cases} \sin^2 x & \text{при } x > 0, \\ 1 - 2 \sin x^2 & \text{в противном случае.} \end{cases}$$

8. Рассчитать значение  $y$  при заданном значении  $x$ :

$$y = \begin{cases} \sin x^2 & \text{при } x > 0, \\ 1 + 2 \sin^2 x & \text{в противном случае.} \end{cases}$$

9. Определить, является ли число  $a$  делителем числа  $b$ ?
10. Дано натуральное число. Определить: а) является ли оно четным; б) оканчивается ли оно цифрой 7.
11. Дано двузначное число. Определить: а) какая из его цифр больше: первая или вторая; б) одинаковы ли его цифры.

12. Дано трехзначное число. Выяснить, является ли оно палиндромом ("перевертышем"), т. е. таким числом, десятичная запись которого читается одинаково слева направо и справа налево.
13. Дано трехзначное число. а) Верно ли, что все его цифры одинаковые? б) Определить, есть ли среди его цифр одинаковые.
14. Дано натуральное число. а) Верно ли, что оно заканчивается четной цифрой? б) Верно ли, что оно заканчивается нечетной цифрой?
15. \*Работа светофора для пешеходов запрограммирована следующим образом: в начале каждого часа в течение трех минут горит зеленый сигнал, затем в течение двух минут — красный, в течение трех минут — опять зеленый и т. д. Дано вещественное число  $t$ , означающее время в минутах, прошедшее с начала очередного часа. Определить, сигнал какого цвета горит для пешеходов в этот момент.
16. Дана масса в килограммах. Найти число полных тонн в ней.
17. Дан прямоугольник с размерами 543 130 мм. Сколько квадратов со стороной 130 мм можно отрезать от него?
18. С начала 1990 года по некоторый день прошло  $n$  месяцев и 2 дня. Присвоить целочисленной величине  $x$  значение 1, 2, ..., 11 или 12 в зависимости от того, каким месяцем (январем, февралем и т. п.) является месяц этого дня. Например, при  $n$  3 значение  $x$  равно 4.
19. Дано трехзначное число. Найти число, полученное при прочтении его цифр справа налево.
20. Дано трехзначное число. В нем зачеркнули первую слева цифру и приписали ее в конце. Найти полученное число.
21. Дано трехзначное число. В нем зачеркнули последнюю справа цифру и приписали ее в начале. Найти полученное число.
22. Дано трехзначное число. Найти число, полученное при перестановке первой и второй цифр заданного числа.
23. Дано трехзначное число. Найти число, полученное при перестановке второй и третьей цифр заданного числа.
24. Дано трехзначное число, в котором все цифры различны. Получить шесть чисел, образованных при перестановке цифр заданного числа.



25. Дано натуральное число  $n$  ( $n > 9$ ). Найти: а) число единиц в нем; б) число десятков в нем.
26. Из трехзначного числа  $x$  вычли его последнюю цифру. Когда результат разделили на 10, а к частному слева приписали последнюю цифру числа  $x$ , то получилось число 237. Найти число  $x$ .
27. Из трехзначного числа  $x$  вычли его последнюю цифру. Когда результат разделили на 10, а к частному слева приписали последнюю цифру числа  $x$ , то получилось число  $n$ . Найти число  $n$ . По заданному  $n$  найти число  $x$  (значение  $n$  вводится с клавиатуры,  $10 \leq n \leq 999$  и при этом число десятков в  $n$  не равно нулю).
28. В трехзначном числе  $x$  зачеркнули первую цифру. Когда полученное число умножили на 10, а произведение сложили с первой цифрой числа  $x$ , то получилось число  $n$ . По заданному  $n$  найти число  $x$  (значение  $n$  вводится с клавиатуры,  $1 \leq n \leq 999$ ).
29. В трехзначном числе  $x$  зачеркнули его вторую цифру. Когда к образованному при этом двузначному числу слева приписали вторую цифру числа  $x$ , то получилось число  $n$ . По заданному  $n$  найти число  $x$  (значение  $n$  вводится с клавиатуры,  $10 \leq n \leq 999$  и при этом число десятков в  $n$  не равно нулю).
30. \*Даны целые числа  $h, m, s$  ( $0 < h \leq 23, 0 \leq m \leq 59, 0 \leq s \leq 59$ ), указывающие момент времени: " $h$  часов,  $m$  минут,  $s$  секунд". Определить угол (в градусах) между положением часовой стрелки в начале суток и в указанный момент времени.
31. \*С начала суток часовая стрелка повернулась на  $y$  градусов ( $0 \leq y < 360, y$  — вещественное число). Определить число полных часов и число полных минут, прошедших с начала суток.
32. \*Часовая стрелка образует угол  $y$  с лучом, проходящим через центр и через точку, соответствующую 12 часам на циферблате,  $0 < y \leq 2$ . Определить значение угла для минутной стрелки, а также количество полных часов и полных минут.
33. Записать условие, которое является истинным, когда:
- а) каждое из чисел  $A$  и  $B$  больше 100;
  - б) только одно из чисел  $A$  и  $B$  четное;
  - в) хотя бы одно из чисел  $A$  и  $B$  положительно;
  - г) каждое из чисел  $A, B, C$  кратно трем;
  - д) только одно из чисел  $A, B$  и  $C$  меньше 50;
  - е) хотя бы одно из чисел  $A, B, C$  отрицательно.
34. Шахматная ладья ходит по горизонтали или вертикали. Даны две различные клетки шахматной доски, определите, может ли ладья попасть с первой клетки на вторую одним ходом. Программа получает на вход четыре числа от 1 до 8 каждое, задающие номер столбца и номер строки сначала для первой клетки, потом для второй клетки.

Программа должна вывести YES, если из первой клетки ходом ладьи можно попасть во вторую или NO в противном случае.

35. Записать условие, которое является истинным, когда точка с координатами  $(x, y)$  попадает в заштрихованные участки плоскости, включая их границы (рис. 4).

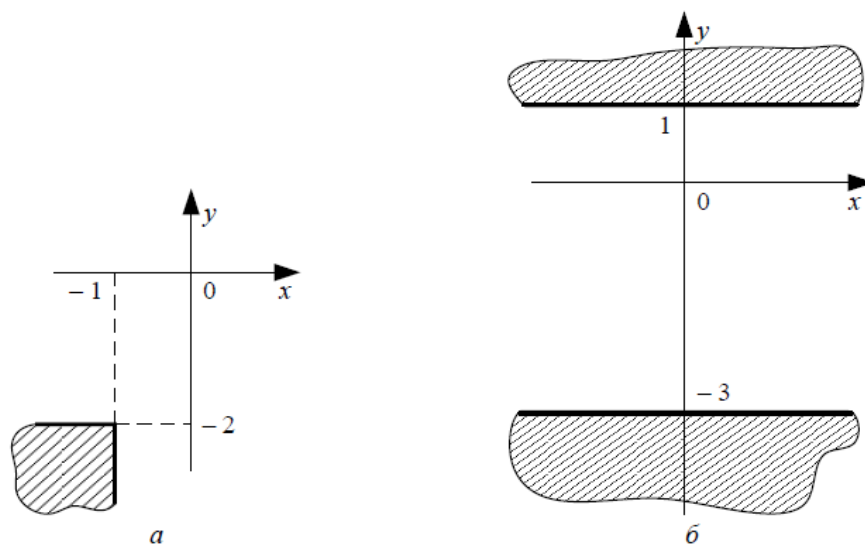


Рис. 4