

## Лабораторная работа 4 «Условные конструкции Python»

**Цель работы:** изучить синтаксис условных конструкций языка Python, продемонстрировать возможности конструкций ветвления на примере разработки интерактивных приложений.

### Основное задание

Разработать интерактивную программу «*Quadric Equation*» («Квадратное уравнение») для решения квадратных уравнений вида:  $ax^2 + bx + c = 0$ . Программа должна запрашивать соответствующие параметры  $a$ ,  $b$  и  $c$ , проверять параметры и выдавать результат.

### Индивидуальное задание

В соответствии с заданием своего варианта выполнить задания из Приложения «А» «Разветвляющиеся алгоритмы».

### Дополнительное задание\*

1. Написать программу, которая бы эмулировала игру «*Dice*» (игра в кости). Суть игры заключается в броске двух шестигранных кубиков (костей) и подсчёте общей суммы очков, которые выпали на первой и второй кости. Для генерирования случайного значения на костях воспользоваться функциями из стандартного модуля `random`.

2. Написать программу – симулятор пирожков с «сюрпризом». Программа должна выводить пирожок и один из пяти (можно больше) различных «сюрпризов», который бы выбирался случайно.

3. Написать программу «*Mood Sensor*», которая определяет настроение пользователя в текущий момент времени. Приложение генерирует случайное число, в зависимости от которого выводится одно из псевдографических «лиц», отображающее настроение.

### Требования к выполнению

1. Программа должна обязательно быть снабжена комментариями на английском языке, в которых необходимо указать краткое предназначение программы, номер лабораторной работы и её название, версию программы, ФИО разработчика и дату разработки.

2. Каждая программа должна быть снабжена дружелюбным и интуитивно понятным интерфейсом (да-да, пусть даже пока в консольном варианте).



Теперь Вы знаете условные конструкции языка Python!

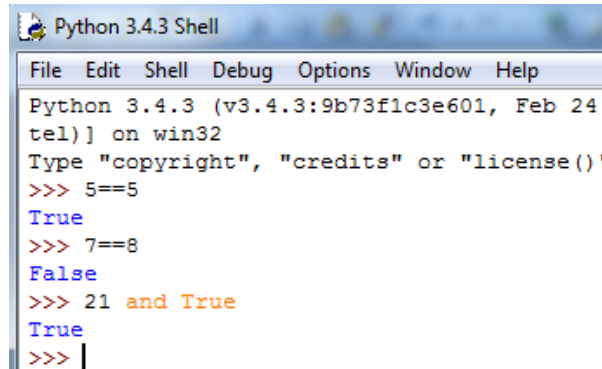
### Контрольные вопросы

1. Перечислить основные управляющие конструкции.
2. Описать основные элементы блок-схем.
3. Как в языке Python реализуется механизм истинности-ложности? Может ли значение быть условием?
4. Описать синтаксис простой условной конструкции `if`. Представить примерную блок-схему конструкции.
5. С помощью каких операторов можно комбинировать в одной условной конструкции `if` несколько условий? Какой механизм оптимизации применяет интерпретатор Python для эффективного вычисления результата комбинированных условных выражений?
6. Описать синтаксис условной конструкции `if-else`. Представить примерную блок-схему конструкции.
7. Описать синтаксис условной конструкции `elif`. Представить примерную блок-схему конструкции.
8. Чем использование `elif` будет отличаться от использования вложенных условных конструкций `if-else`?
9. Как сгенерировать случайную последовательность чисел с использованием функций `randint()` и `randrange()`? Чем они отличаются? Какие ещё есть полезные функции в модуле `random`?

## Базовый синтаксис языка *Python*: условные конструкции

### ЛОГИЧЕСКИЕ ВЫРАЖЕНИЯ

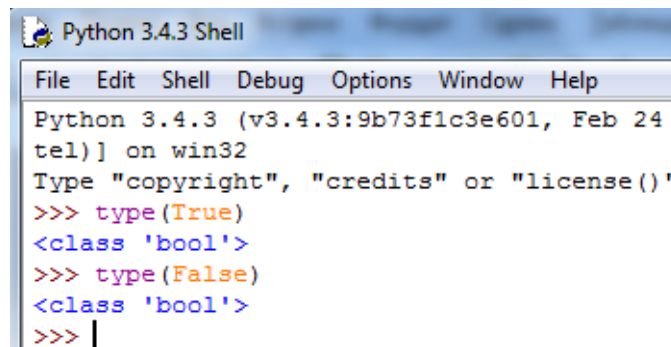
Логическими (*boolean expression*) называются выражения, которые могут принимать одно из двух значений – истина или ложь. Операнды логических операторов должны быть логическими выражениями. В *Python* любое ненулевое число интерпретируется им как «истинное» (рисунок 4.1).



```
Python 3.4.3 Shell
File Edit Shell Debug Options Window Help
Python 3.4.3 (v3.4.3:9b73f1c3e601, Feb 24
tel)] on win32
Type "copyright", "credits" or "license()"
>>> 5==5
True
>>> 7==8
False
>>> 21 and True
True
>>> |
```

Рис. 4.1. Пример логических выражений

*True* и *False* – специальные значения, которые принадлежат к типу *bool*, они не являются строками (рисунок 4.2).



```
Python 3.4.3 Shell
File Edit Shell Debug Options Window Help
Python 3.4.3 (v3.4.3:9b73f1c3e601, Feb 24
tel)] on win32
Type "copyright", "credits" or "license()"
>>> type(True)
<class 'bool'>
>>> type(False)
<class 'bool'>
>>> |
```

Рис. 4.2. *True* и *False*

### УСЛОВНОЕ ИСПОЛНЕНИЕ

Условные инструкции (*conditional statements*) позволяют изменять ход выполнения программы в зависимости от условий.

На рисунке 4.3 представлена блок-схема и код для проверки положительности числа *x* с выводом соответствующего сообщения.

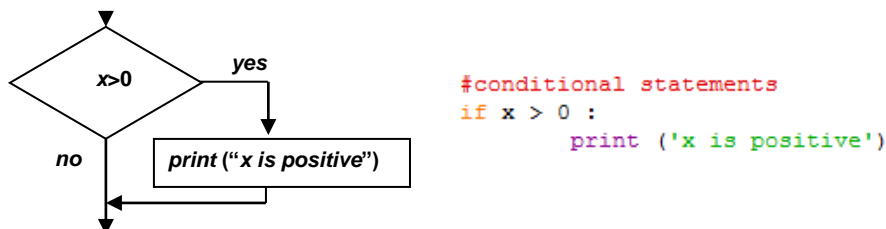


Рис. 4.3. Блок-схема условной конструкции и код на языке *Python*

Логическое выражение после инструкции *if* называется *условием*, далее следует *символ двоеточия* (*:*) и *строка* (строки) с *отступом*. Если логическое условие истинно, то управление получает выражение, записанное с отступами, иначе – выражение пропускается.

Не существует ограничения на число инструкций, которые могут встречаться в теле *if*, но хотя бы одна инструкция там должна быть. Иногда полезно иметь тело *if* без инструкций (обычно оставляют место для кода, который еще не написан). В этом случае можно воспользоваться инструкцией *pass*, которая ничего не делает (рисунок 4.4).

```
File Edit Format Run Options Window Help
#conditional statements
if x < 0 :
    pass # следует обработать
        #отрицательные значения!
```

Рис. 4.4. Инструкция *pass*

## АЛЬТЕРНАТИВНОЕ ИСПОЛНЕНИЕ

Альтернативное исполнение (*alternative execution*) инструкции *if* предполагает два направления выполнения, и условие определяет, какое из них выполнится. Пример (рисунок 4.5) иллюстрирует проверку четности/нечетности числа *x* с выводом соответствующего сообщения.

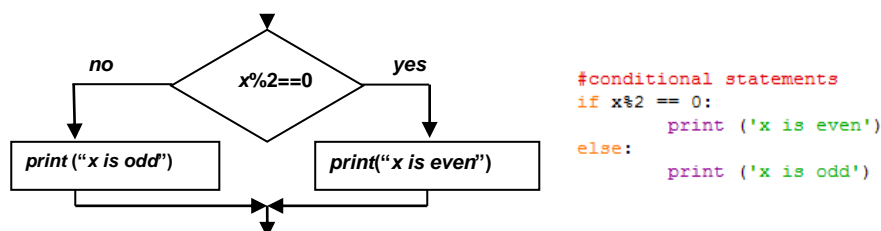


Рис. 4.5. – Инструкция *if*

Альтернативное исполнение предполагает выполнение одного из вариантов, так как условие может быть либо истинным, либо ложным. Варианты называются *ветвями (branches)*.

## ПОСЛЕДОВАТЕЛЬНОСТЬ УСЛОВИЙ

Когда имеется больше двух вариантов выполнения, то необходимо больше двух ветвей. В этом случае, можно воспользоваться *сцепленными условиями (chained conditional)*. На рисунке 4.6 приведен алгоритм сравнения двух чисел *x* и *y*.

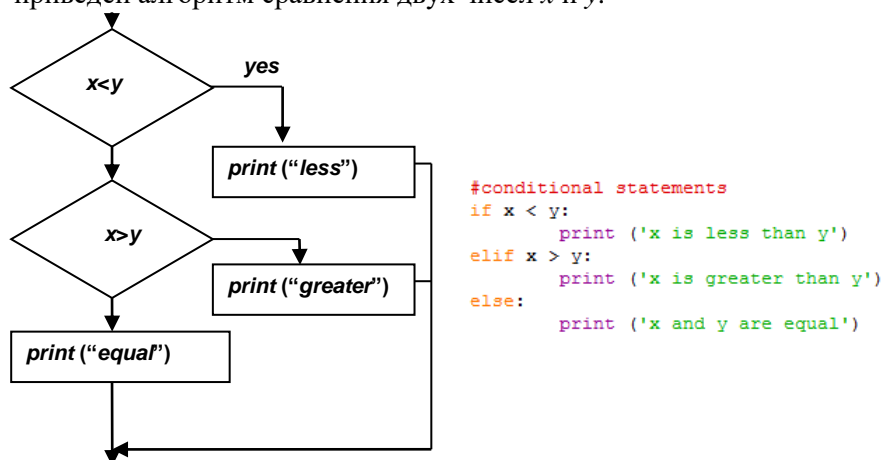


Рис. 4.6. Сцепленные условия

Инструкция *elif* – аббревиатура от «*else if*». Оператор *else* должен быть в конце инструкции, но может и отсутствовать.

Каждое условие проверяется в порядке расположения. Если первое условие ложно, то проверяется следующее и так далее. Если одно из условий истинно, то выполняется соответствующая ветка, и инструкция завершается.

## ВЛОЖЕННЫЕ УСЛОВИЯ

Условия могут быть вложенными. Пример *трихотомии*, иллюстрирующий сравнение чисел *x* и *y*, представлен рисунке 4.7.

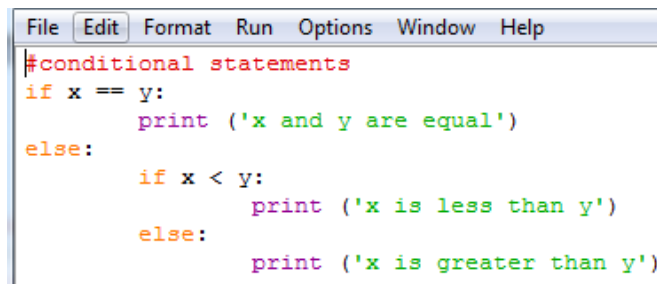


Рис. 4.7. Код трихотомии

Хотя отступ инструкций делает структуру более очевидной, *вложенные условия (nested conditionals)* усложняют чтение кода. Их следует по возможности избегать.

### ПЕРЕХВАТ ИСКЛЮЧЕНИЙ С ПОМОЩЬЮ TRY И EXCEPT

Структура условного выполнения «try/except» обрабатывает типы ожидаемых и неожиданных ошибок. Идея try и except заключается в следующем. Пусть некоторая последовательность инструкций может иметь ошибки. Нужно добавить дополнительные инструкции (блок except), которые выполняются в случае возникновения ошибки, и игнорируются, если ошибка не произошла.

Общий вид конструкции «try/except»:

```

try:
    instruction 1
except:
    instruction 2

```

Выполнение начинается с последовательности инструкций в блоке try. Если все выполняется без ошибок, блок except пропускается. Если произошло исключение в блоке try, то блок try покидается и выполняет последовательность инструкций внутри блока except. Обработка исключения с помощью инструкции try называется *перехватом (catching)* исключения.

Пример кода и результат программы ввода с учетом некорректного ввода представлен на рисунке 4.8.

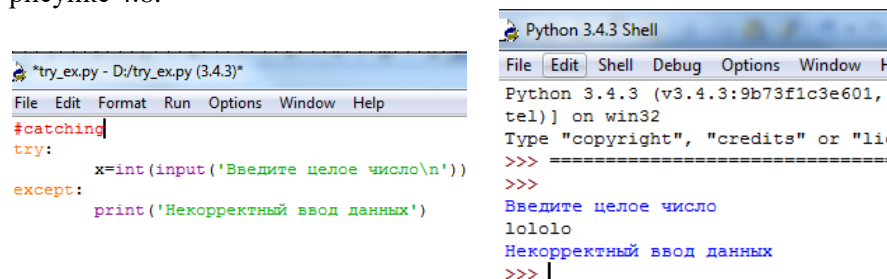


Рис. 4.8. Учет некорректного ввода целого числа с клавиатуры

### СЛУЧАЙНЫЕ ЧИСЛА

Программы, генерирующие одни и те же выходные значения для одинаковых входных значений, называются *детерминированными*. Для многих приложений, например игр, характерна непредсказуемость в поведении. Для создания недетерминированных программ можно использовать *алгоритмы генерации псевдослучайных чисел*.

Модуль random предоставляет функции, которые генерируют псевдослучайные числа (далее «случайные числа»):

- random() возвращает случайное число с плавающей точкой в интервале от 0.0 до 1.0 (включая 0.0 и не включая 1.0);
- randint(low, high) возвращает целочисленное значение в интервале от low до high включительно;
- randrange(start, stop, step) возвращает число от start включительно до stop не включительно с шагом step;
- для выбора случайного элемента из последовательности, можно воспользоваться функцией choice().

Модуль random также включает функции, которые генерируют значения, принадлежащие показательному закону распределения, распределению Гаусса, Гамма и другие.

## ПРИЛОЖЕНИЕ А «Разветвляющиеся алгоритмы»

**Задание 1.** В соответствии с заданием своего варианта составить программу для вычисления значений составной функции  $G(x)$ . Параметр  $a$  вводится с клавиатуры. Функция  $f(x)$  может принимать одно из двух значений ( $e^x$  или  $x^2$ ) по выбору пользователя. Выполнить расчет для всех вариантов ветвления. Результат представить в виде:

$x$	$a$	$f(x)$	$G(x)$

**Задание 2.** В соответствии с заданием своего варианта составить программу, которая распределяет целые числа, вводимые пользователем с клавиатуры по четырем группам. Результат представить в виде: число 49 относится к группам А и В.

Таблица А 1. Индивидуальные задания

Вариант	Задание 1.	Вариант	Задание 2.
<b>1</b>	$G(x) = \begin{cases} \sin(f(x) + a)^2 - \sqrt{ f(x) }, & ax > 0; \\ (f(x) + 2a)^2 - \sqrt{ f(x)a }, & ax < 0; \\ (f(x) + a)^2 + 1, & ax = 0; \end{cases}$	<b>13</b>	$G(x) = \begin{cases} 2e^{af(x)}, & 1 < ax < 11; \\ \sqrt{ f(x) + a }, & 11 \leq ax < 55; \\ \sqrt{ a^3 f(x) }, & \text{в остальных случаях}; \end{cases}$
<b>2</b>	$G(x) = \begin{cases} \ln( f(x)  + 5) + (f(x))^3, & \frac{x}{a} > 0; \\ \ln\left(\left \frac{f(x)}{a}\right  + 2\right) + a^3, & \frac{x}{a} < 0; \\ (f(x)^2 + a)^3, & x = 0; \\ 0, & a = 0; \end{cases}$	<b>14</b>	$G(x) = \begin{cases} a^5 + 5e^{af(x)}, & a - \text{четное}, x \geq 0; \\ 7\sqrt{ f(x) - a }, & a - \text{нечетное}, x < 0; \\ \sqrt{ a^3 }, & \text{в остальных случаях}; \end{cases}$
<b>3</b>	$G(x) = \begin{cases} f(x)^2 a^2 + \sin(a), & x - a = 0; \\ (f(x) - a)^2 + a, & x - a > 0; \\ (af(x))^2 + 15a, & x - a < 0; \end{cases}$	<b>15</b>	$G(x) = \begin{cases} \sin(a) + \ln( \sin(f(x))  + 1), & x - a = 0; \\ f(x)^2 +  a , & x - a > 0; \\ \sqrt{ a^2 + f(x)^2 }, & x - a < 0; \end{cases}$
<b>4</b>	$G(x) = \begin{cases} \arctg(f(x)), & x > a; \\ (af(x))^3 + \cos(a + 1), & x < a; \\ (a + f(x))^3 + 25, & x = a; \end{cases}$	<b>16</b>	$G(x) = \begin{cases} \cos(a + 1) + \ln( \sin(f(x)^2) + 1 ), & \frac{x}{a} > 0; \\ f(x)^2 + a, & \frac{x}{a} < 0; \\ \sqrt{ a^2 + f(x)^2 }, & x = 0; \\ x + \cos a, & a = 0 \end{cases}$
<b>5</b>	$G(x) = \begin{cases} a\sqrt{f(x)}, & a - \text{четное}, x > 0; \\ \sin(2a), & a - \text{нечетное}, x < 0; \\ \sqrt{ af(x) }, & \text{в остальных случаях}; \end{cases}$	<b>17</b>	$G(x) = \begin{cases} \cos(a + 1) + e^{f(x)}, & \frac{x}{a} > 0; \\ f(x)^2 - f(a), & \frac{x}{a} < 0; \\ \sqrt{ a^2 - f(x)^2 }, & x = 0; \\  x  + \cos a, & a = 0 \end{cases}$

<b>6</b>	$G(x) = \begin{cases} e^{f(x)}, 1 < ax < 2; \\ \sin(a) + \sqrt{ f(x) }, 2 \leq ax \leq 10; \\ \sqrt{ a^3 }, \text{ в остальных случаях}; \end{cases}$	<b>18</b>	$G(x) = \begin{cases} 1 + (f(x) + a)^2, ax > 0; \\ f(x)^2 - \sqrt{ f(a) }, ax < 0; \\ \sqrt{ a^2 + f(x)^2 }, ax = 0; \end{cases}$
<b>7</b>	$G(x) = \begin{cases} e^{f(x)+2a}, 0 < ax < 1; \\ 2\sqrt{ f(x) } - \cos a, 2 \leq ax \leq 10; \\ \sqrt{ f(x) }, \text{ в остальных случаях}; \end{cases}$	<b>19</b>	$G(x) = \begin{cases} \sqrt{1 + (f(x) + a)^2},  ax  > 1; \\ f(x)^2 + e^a,  ax  < 1; \\ \sqrt{ 2a^2 + f(x)^2 },  ax  = 1; \end{cases}$
<b>8</b>	$G(x) = \begin{cases} \sin(5f(x) + 3ae^{f(x)}), a < x; \\ \cos(f(x) + \sqrt{ f(x) }), a > x; \\ \sqrt{ a^2 + f(x)^2 }, a = x; \end{cases}$	<b>20</b>	$G(x) = \begin{cases} \sqrt{20 + (f(x) + a)^2},  a + x  > 1; \\ f(x) - e^a,  a + x  < 1; \\ a - \sqrt{ 2a^2 + f(x) },  a + x  = 1; \end{cases}$
<b>9</b>	$G(x) = \begin{cases} 2\sin(f(x)) + 3a^2, x >  a ; \\  f(x) - a , x <  a ; \\ \sqrt{ a^2 + f(x)^2 }, x =  a ; \end{cases}$	<b>21</b>	$G(x) = \begin{cases} \sqrt{21 + (f(x) + a)^2},  a - x  > 1; \\ \sin(f(x) + a),  a - x  < 1; \\ a + \sqrt{ 2a^2 - f(x) },  a - x  = 1; \end{cases}$
<b>10</b>	$G(x) = \begin{cases} \ln( f(x)  + 1) + a, 10 >  ax ; \\  f(x) + a , 10 <  ax ; \\ \sqrt{ a^2 + f(x)^2 },  ax  = 10; \end{cases}$	<b>22</b>	$G(x) = \begin{cases} \sqrt{e^a + (f(x) + a)^2},  ax  > 5; \\ f(x) + \cos(e^a),  ax  < 5; \\ a - \sqrt{ 2a^2 f(x) },  ax  = 5; \end{cases}$
<b>11</b>	$G(x) = \begin{cases} \sin(4\sin(f(x)) + 3a), a > x; \\ \cos( f(x) + a ), a < x; \\ \sqrt{ a^2 + f(x)^2 }, a = x; \end{cases}$	<b>23</b>	$G(x) = \begin{cases} \sqrt{5 + (f(x) + a)^2},  ax  > 2; \\ 2f(x) - e^a,  ax  < 2; \\ \sin(2a) - \sqrt{ 2a^4 - f(x) },  ax  = 2; \end{cases}$
<b>12</b>	$G(x) = \begin{cases} e^{f(x)-a}, 0.7 < ax < 11; \\ \sqrt{\cos^2( f(x) )}, 0 < ax \leq 0.7; \\ a, \text{ в остальных случаях}; \end{cases}$	<b>24</b>	$G(x) = \begin{cases} \sqrt{20 + f(x)^2},  a + x  > 10; \\ f(x) + e^a,  a + x  < 10; \\ 24a - \sqrt{ 2a^2 + f(x) },  a + x  = 10; \end{cases}$
<b>25</b>	$G(x) = \begin{cases} \sqrt{25 + (f(x) + a)^2},  a + x  > 25; \\ f(x)^2 - e^a,  a + x  < 25; \\ \cos a + \sqrt{ 2a^2 - 2f(x) },  a + x  = 25; \end{cases}$		