## Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования

**«Московский государственный технический университет имени Н.Э. Баумана**

## (национальный исследовательский университет МГТУ им.

**Н.Э. Баумана)**

## Факультет «Информатика и системы управления» Кафедра «Системы обработки информации и управления»

Лабораторная работа №1

## «Основные конструкции языка Python»

по предмету

«Базовые компоненты интернет-технологий»

Выполнил: студент группы № ИУ5-31Б

Радченко Дмитрий

Проверил: Преподаватель кафедры ИУ-5

Гапанюк Юрий

2023 г.

# Задание

Разработать программу для решения биквадратного уравнения.

* Программа должна быть разработана в виде консольного приложения на языке Python.
* Программа осуществляет ввод с клавиатуры коэффициентов А, В, С, вычисляет дискриминант и ДЕЙСТВИТЕЛЬНЫЕ корни уравнения (в зависимости от дискриминанта).
* Коэффициенты А, В, С могут быть заданы в виде параметров командной строки ( вариант задания параметров приведен в конце файла с примером кода ). Если они не заданы, то вводятся с клавиатуры в соответствии с пунктом 2. Описание работы с параметрами командной строки.
* Если коэффициент А, В, С введен или задан в командной строке некорректно, то необходимо проигнорировать некорректное значение и вводить коэффициент повторно пока коэффициент не будет введен корректно. Корректно заданный коэффициент - это коэффициент, значение которого может быть без ошибок преобразовано в действительное число.

# Листинг программы.

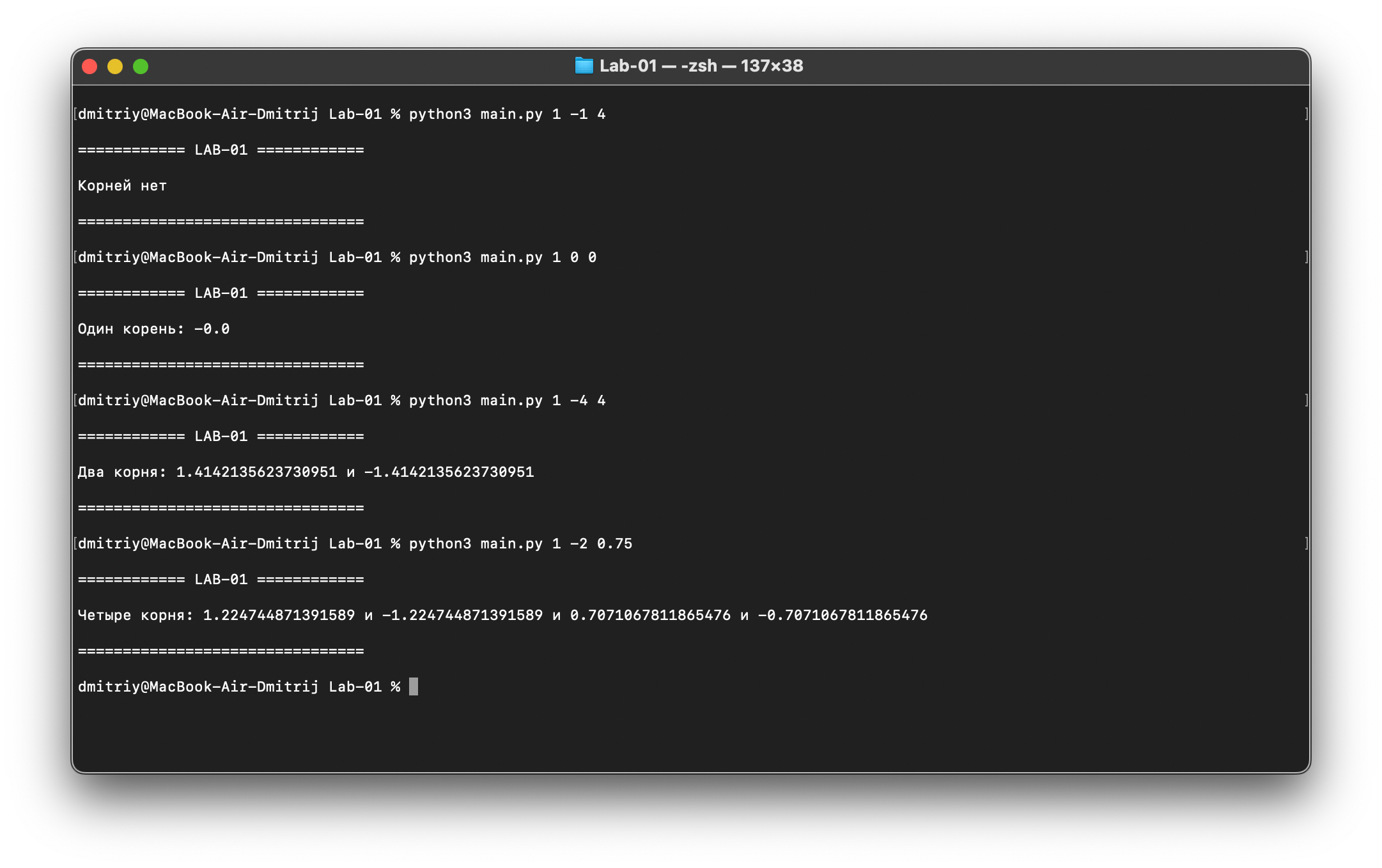
Код можно увидеть на GitHub по [ссылке](https://github.com/IU5-IT/IU5-IT/tree/main/Term-3/BKIT-2022/Lab-01/Lab-01).

# Copyright © 2022 mightyK1ngRichard <dimapermyakov55@gmail.com>  
  
*import* sys  
*import* math  
  
  
*def* get\_coef(index, prompt) -> float:  
 *'''  
 Читаем коэффициент из командной строки или вводим с клавиатуры  
  
 Args:  
 index (int): Номер параметра в командной строке  
 prompt (str): Приглашение для ввода коэффицента  
  
 Returns:  
 float: Коэффициент квадратного уравнения  
 '''  
 try*:  
 # Пробуем прочитать коэффициент из командной строки  
 *try*:  
 coef\_str = sys.argv[index]  
 coef = float(coef\_str)  
 *return* coef  
 *except* ValueError:  
 # Если неверный формат, повторяем ввод.  
 *pass  
 except*:  
 # Вводим с клавиатуры  
 *while True*:  
 *try*:  
 print(prompt)  
 coef\_str = input()  
 coef = float(coef\_str)  
 *return* coef  
 *except* ValueError:  
 # Если неверный формат, повторяем ввод.  
 *pass  
  
  
def* get\_roots(a, b, c) -> list:  
 *'''  
 Вычисление корней квадратного уравнения  
  
 Args:  
 a (float): коэффициент А  
 b (float): коэффициент B  
 c (float): коэффициент C  
  
 Returns:  
 list[float]: Список корней  
 '''* result = []  
 D = b \* b - 4 \* a \* c  
 *if* D == 0.0:  
 root = -b / (2.0 \* a)  
 result.append(root)  
 *elif* D > 0.0:  
 sqD = math.sqrt(D)  
 root1 = (-b + sqD) / (2.0 \* a)  
 root2 = (-b - sqD) / (2.0 \* a)  
 result.append(root1)  
 result.append(root2)  
  
 *return* result  
  
  
*def* main():  
 a = get\_coef(1, 'Введите коэффициент А:')  
 b = get\_coef(2, 'Введите коэффициент B:')  
 c = get\_coef(3, 'Введите коэффициент C:')  
 roots = get\_roots(a, b, c)  
  
 count = 0  
 *for* el *in* roots:  
 *if* el > 0:  
 count += 1  
  
 finally\_roots = []  
 *for* el *in* roots:  
 *if* el > 0 *or* el == 1:  
 finally\_roots.append(math.sqrt(el))  
 finally\_roots.append(-1 \* math.sqrt(el))  
 *elif* el == 0:  
 finally\_roots.append(el)  
  
 len\_roots = len(finally\_roots)  
 *if* len\_roots == 0:  
 print('Корней нет')  
 *elif* len\_roots == 1:  
 print(f'Один корень: {finally\_roots[0]}')  
 *elif* len\_roots == 2:  
 print('Два корня: {} и {}'.format(finally\_roots[0], finally\_roots[1]))  
 *elif* len\_roots == 3:  
 print('Три корня: {} и {} и {}'.format(finally\_roots[0], finally\_roots[1], finally\_roots[2]))  
 *else*:  
 print('Четыре корня: {} и {} и {} и {}'.format(finally\_roots[0], finally\_roots[1], finally\_roots[2],  
 finally\_roots[3]))  
  
  
*if* \_\_name\_\_ == "\_\_main\_\_":  
 print('\n============ LAB-01 ============\n')  
 main()  
 print('\n================================\n')

**Примеры работы программы:**







|  |
| --- |
| import sys |
|  | import math |
|  |  |
|  |  |
|  | def get\_coef(index, prompt) -> float: |
|  | ''' |
|  | Читаем коэффициент из командной строки или вводим с клавиатуры |
|  |  |
|  | Args: |
|  | index (int): Номер параметра в командной строке |
|  | prompt (str): Приглашение для ввода коэффицента |
|  |  |
|  | Returns: |
|  | float: Коэффициент квадратного уравнения |
|  | ''' |
|  | try: |
|  | # Пробуем прочитать коэффициент из командной строки |
|  | try: |
|  | coef\_str = sys.argv[index] |
|  | coef = float(coef\_str) |
|  | return coef |
|  | except ValueError: |
|  | # Если неверный формат, повторяем ввод. |
|  | pass |
|  | except: |
|  | # Вводим с клавиатуры |
|  | while True: |
|  | try: |
|  | print(prompt) |
|  | coef\_str = input() |
|  | coef = float(coef\_str) |
|  | return coef |
|  | except ValueError: |
|  | # Если неверный формат, повторяем ввод. |
|  | pass |
|  |  |
|  |  |
|  | def get\_roots(a, b, c) -> list: |
|  | ''' |
|  | Вычисление корней квадратного уравнения |
|  |  |
|  | Args: |
|  | a (float): коэффициент А |
|  | b (float): коэффициент B |
|  | c (float): коэффициент C |
|  |  |
|  | Returns: |
|  | list[float]: Список корней |
|  | ''' |
|  | result = [] |
|  | D = b \* b - 4 \* a \* c |
|  | if D == 0.0: |
|  | root = -b / (2.0 \* a) |
|  | result.append(root) |
|  | elif D > 0.0: |
|  | sqD = math.sqrt(D) |
|  | root1 = (-b + sqD) / (2.0 \* a) |
|  | root2 = (-b - sqD) / (2.0 \* a) |
|  | result.append(root1) |
|  | result.append(root2) |
|  |  |
|  | return result |
|  |  |
|  |  |
|  | def main(): |
|  | a = get\_coef(1, 'Введите коэффициент А:') |
|  | b = get\_coef(2, 'Введите коэффициент B:') |
|  | c = get\_coef(3, 'Введите коэффициент C:') |
|  | roots = get\_roots(a, b, c) |
|  |  |
|  | count = 0 |
|  | for el in roots: |
|  | if el > 0: |
|  | count += 1 |
|  |  |
|  | finally\_roots = [] |
|  | for el in roots: |
|  | if el > 0 or el == 1: |
|  | finally\_roots.append(math.sqrt(el)) |
|  | finally\_roots.append(-1 \* math.sqrt(el)) |
|  | elif el == 0: |
|  | finally\_roots.append(el) |
|  |  |
|  | len\_roots = len(finally\_roots) |
|  | if len\_roots == 0: |
|  | print('Корней нет') |
|  | elif len\_roots == 1: |
|  | print(f'Один корень: {finally\_roots[0]}') |
|  | elif len\_roots == 2: |
|  | print('Два корня: {} и {}'.format(finally\_roots[0], finally\_roots[1])) |
|  | elif len\_roots == 3: |
|  | print('Три корня: {} и {} и {}'.format(finally\_roots[0], finally\_roots[1], finally\_roots[2])) |
|  | else: |
|  | print('Четыре корня: {} и {} и {} и {}'.format(finally\_roots[0], finally\_roots[1], finally\_roots[2], |
|  | finally\_roots[3])) |
|  |  |
|  |  |
|  | if \_\_name\_\_ == "\_\_main\_\_": |
|  | print('\n============ LAB-01 ============\n') |
|  | main() |
|  | print('\n================================\n') |