



**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 № ИУ5-33Б
Пермяков Дмитрий**

**Проверил:
Преподаватель кафедры ИУ-5
Гапанюк Юрий**

2022 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 A, B, C, вычисляет дискриминант и **ДЕЙСТВИТЕЛЬНЫЕ** корни уравнения (в зависимости от дискриминанта).
- Коэффициенты A, B, C 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 A, B, C 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](#).

```
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 коэффициента

    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 A
        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 A:')
    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

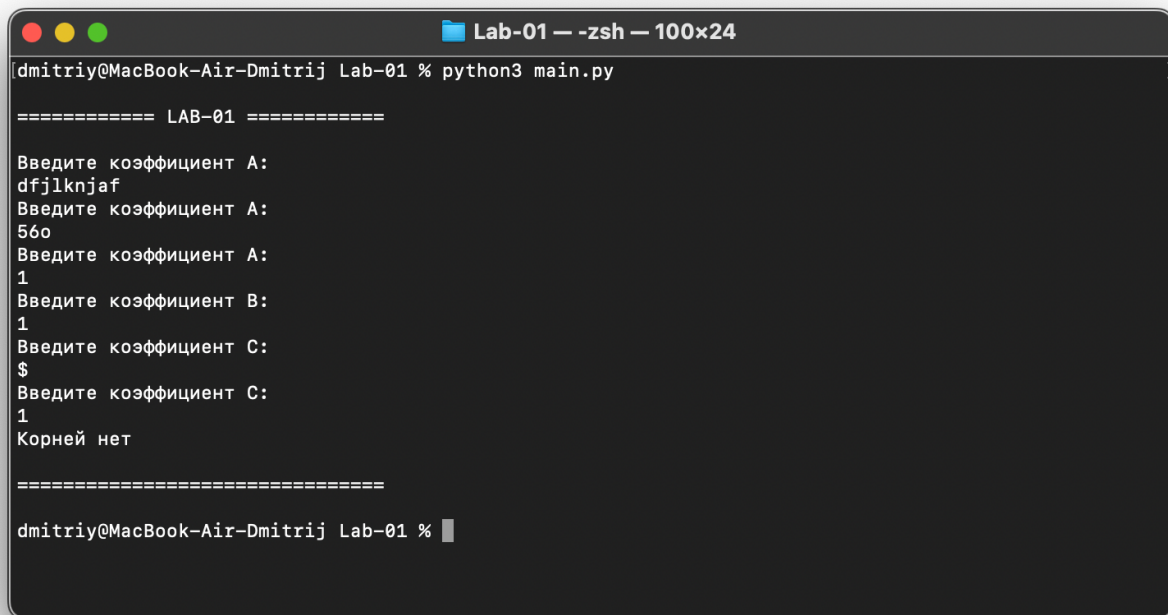
    finally_roots = []
    for el in roots:
        if el > 0 or el == 1:
            finally_roots.append(math.sqrt(el))
            finally_roots.append(-1 * math.sqrt(el))
        elif el == 0:
            finally_roots.append(el)

    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:



```
Lab-01 — -zsh — 100x24  
dmitriy@MacBook-Air-Dmitrij Lab-01 % python3 main.py  
  
===== LAB-01 =====  
Введите коэффициент A:  
dfjlnkja  
Введите коэффициент A:  
560  
Введите коэффициент A:  
1  
Введите коэффициент B:  
1  
Введите коэффициент C:  
$  
Введите коэффициент C:  
1  
Корней нет  
  
=====  
dmitriy@MacBook-Air-Dmitrij Lab-01 %
```

```
Lab-01 — -zsh — 100x24

dmitriy@MacBook-Air-Dmitrij Lab-01 % python3 main.py
[
===== LAB-01 =====
Введите коэффициент A:
число
Введите коэффициент A:
1
Введите коэффициент B:
-5
Введите коэффициент C:
6
Четыре корня: 1.7320508075688772 и -1.7320508075688772 и 1.4142135623730951 и -1.4142135623730951
=====

dmitriy@MacBook-Air-Dmitrij Lab-01 %
```

```
Lab-01 — -zsh — 137x38

dmitriy@MacBook-Air-Dmitrij Lab-01 % python3 main.py 1 -1 4
===== LAB-01 =====
Корней нет
=====

dmitriy@MacBook-Air-Dmitrij Lab-01 % python3 main.py 1 0 0
===== LAB-01 =====
Один корень: -0.0
=====

dmitriy@MacBook-Air-Dmitrij Lab-01 % python3 main.py 1 -4 4
===== LAB-01 =====
Два корня: 1.4142135623730951 и -1.4142135623730951
=====

dmitriy@MacBook-Air-Dmitrij Lab-01 % python3 main.py 1 -2 0.75
===== LAB-01 =====
Четыре корня: 1.224744871391589 и -1.224744871391589 и 0.7071067811865476 и -0.7071067811865476
=====

dmitriy@MacBook-Air-Dmitrij Lab-01 %
```