

**Московский государственный технический
университет им. Н.Э. Баумана.**

Факультет «Информатика и управление»

Кафедра ИУ5. Курс «Базовые компоненты интернет-технологий»

Отчет по лабораторной работе №1
**«Основные конструкции языка Python.
Решение биквадратного уравнения»**

Выполнил:
студент группы ИУ5-32Б
Казицин Алексей

Проверил:
преподаватель кафедры
Гапанюк Ю.Е.

Подпись и дата:

Подпись и дата:

Москва, 2022 г.

Цель лабораторной работы: изучение основных конструкций языка Python.

Задание:

Разработать программу для решения биквадратного уравнения.

1. Программа должна быть разработана в виде консольного приложения на языке Python.
2. Программа осуществляет ввод с клавиатуры коэффициентов A , B , C , вычисляет дискриминант и **ДЕЙСТВИТЕЛЬНЫЕ** корни уравнения (в зависимости от дискриминанта).
3. Коэффициенты A , B , C могут быть заданы в виде параметров командной строки. Если они не заданы, то вводятся с клавиатуры в соответствии с пунктом 2.
4. 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.

Текст программы:

```
import sys
import math
```

```
def get_coef(index, prompt):
```

```
    '''
```

```
    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
```

```
        coef_str = sys.argv[index]
```

```
    except:
```

```
        # Вводим с клавиатуры
```

```
        print(prompt)
```

```
        coef_str = input()
```

```
    # Переводим строку в действительное число
```

```
    flag = False
```

```
    while not flag:
```

```
        try:
```

```
            coef = float(coef_str)
```

```
            if index == 1:
```

```
                if abs(coef) != 0.0:
```

```
                    pass
```

```
                else:
```

```
                    raise TypeError
```

```
            flag = True
```

```
        except:
```

```
            print('Введено некорректное значение коэффициента ' + prompt[-2] + '.')
```

```
            print(prompt)
```

```
            coef_str = input()
```

```
    return coef
```

```
def get_roots(a, b, c):
```

```
    '''
```

```
    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)
```

```
        if root == -0.0:
```

```
            root = abs(root)
```

```
        result.append(root)
```

```
    elif D > 0.0:
```

```

    sqD = math.sqrt(D)
    root1 = (-b + sqD) / (2.0 * a)
    root2 = (-b - sqD) / (2.0 * a)
    if root1 == -0.0:
        root1 = abs(root1)
    if root2 == -0.0:
        root2 = abs(root2)
    result.append(root1)
    result.append(root2)
return result

```

```
def main():
```

```
    '''
```

```
    Основная функция
```

```
    '''
```

```
    a = get_coef(1, 'Введите коэффициент A:')
    b = get_coef(2, 'Введите коэффициент B:')
    c = get_coef(3, 'Введите коэффициент C:')
    # Вычисление корней
    roots = get_roots(a, b, c)
    # Вывод корней
    len_roots = len(roots)
    if len_roots == 0:
        print('Нет корней')
    elif len_roots == 1:
        print('Один корень: {}'.format(roots[0]))
    elif len_roots == 2:
        print('Два корня: {} и {}'.format(roots[0], roots[1]))

# Если сценарий запущен из командной строки
if __name__ == "__main__":
    main()

# Пример запуска
# qr.py 1 0 -4

```

Примеры выполнения программы с различными входными данными:

#1

```
C:\Users\lesha\AppData\Local\Progr
Введите коэффициент A:
1
Введите коэффициент B:
0
Введите коэффициент C:
-4
Два корня: 2.0 и -2.0

Process finished with exit code 0
```

#2

```
C:\Users\lesha\AppData\Local\Programs\Python\P
Введите коэффициент A:
0
Введено некорректное значение коэффициента A.
Введите коэффициент A:
sup
Введено некорректное значение коэффициента A.
Введите коэффициент A:
1
Введите коэффициент B:
0
Введите коэффициент C:
-4
Два корня: 2.0 и -2.0

Process finished with exit code 0
```

#3

```
C:\Users\lesha\AppData\Local\Programs\Python\Python37
Введите коэффициент A:
1
Введите коэффициент B:
фыафыб
Введено некорректное значение коэффициента B.
Введите коэффициент B:
20
Введите коэффициент C:
sa
Введено некорректное значение коэффициента C.
Введите коэффициент C:
10
Два корня: -0.5131670194948619 и -19.486832980505138

Process finished with exit code 0
```

#4

Configuration	Logs
Script path:	C:\Users\lesha\PycharmProjects\lab1.py
Parameters:	1 0 -4

```
C:\Users\lesha\AppData\Local\Programs\Pyt
Два корня: 2.0 и -2.0

Process finished with exit code 0
```

#5

Configuration

Logs

Script path:

Parameters:

C:\Users\lesha\AppData\Local\Programs\Python\Python39\python.exe C:\Users\lesha\PycharmProjects\lab1.py 0 0 0
Введено некорректное значение коэффициента A.
Введите коэффициент A:
1
Один корень: 0.0

#6

Configuration

Logs

Script path:

Parameters:

C:\Users\lesha\AppData\Local\Programs\Python\Python39\python.exe C:\Users\lesha\PycharmProjects\lab1.py 1 sup sup
Введено некорректное значение коэффициента B.
Введите коэффициент B:
aas
Введено некорректное значение коэффициента B.
Введите коэффициент B:
0
Введено некорректное значение коэффициента C.
Введите коэффициент C:
-4
Два корня: 2.0 и -2.0