

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

Факультет «Информатика и системы управления»  
Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Разработка Интернет-Приложений»

Отчет по лабораторной работе №1  
«Решение биквадратного уравнения»

Выполнил:  
студент группы ИУ5-53Б  
Ветошкин Артём

Подпись и дата:  
30.09.21

Проверил:  
Юрий Евгеньевич Гапанюк

Подпись и дата:

Москва, 2021 г.

### 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 А, В, С, вычисляет дискриминант и **ДЕЙСТВИТЕЛЬНЫЕ** корни уравнения (в зависимости от дискриминанта).
3. Коэффициенты А, В, С могут быть заданы в виде параметров командной строки. Если они не заданы, то вводятся с клавиатуры в соответствии с пунктом 2. Описание работы с параметрами командной строки.
4. 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.

### Текст программы:

```
import sys
import math

def force_conv_number(string):
    try:
        return float(string)
    except:
        return None

def _get_coef(index, prompt):
    coef_str = ''
    if len(sys.argv) > index:
        coef_str = sys.argv[index]
    else:
        print(prompt)
        coef_str = input()

    return force_conv_number(coef_str)

def get_coef(index, prompt):
    tmp = None
    while tmp == None:
        tmp = _get_coef(index, prompt)
    return tmp
```

```

def complex_sqrt(number):
    return pow(number, 0.5)

def get_roots(a, b, c):
    result = []
    D = b*b - 4*a*c
    sqD = complex_sqrt(complex(D, 0))
    root1 = (-b + sqD) / (2.0*a)
    root2 = (-b - sqD) / (2.0*a)

    result.append(complex_sqrt(root1))
    result.append(-complex_sqrt(root1))
    result.append(complex_sqrt(root2))
    result.append(-complex_sqrt(root2))
    return result

def print_roots(roots):
    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]))
    elif len_roots == 3:
        print('Три корня: {}, {} и {}'.format(roots[0], roots[1], roots[2]))
    elif len_roots == 4:
        print('Четыре корня: {} и {}'.format(roots[0], roots[1], roots[3], roots[4]))

def input_coef():
    a = get_coef(1, 'Введите коэффициент A:')
    b = get_coef(2, 'Введите коэффициент B:')
    c = get_coef(3, 'Введите коэффициент C:')

    return [a, b, c]

def main():
    [a, b, c] = input_coef()

    roots = get_roots(a,b,c)

    real_roots = list(map(lambda x: x.real, filter(lambda number: number.imag == 0,
roots)))

    print_roots(real_roots)

if __name__ == "__main__":
    main()

```

## Пример работы:

```
PS C:\Users\vet_v\Desktop\RIP> & c:/Users/vet_v/Desktop/RIP/lab_4/.venv/Scripts/python.exe c:/Users/vet_v/Desktop/RIP/lab_1/main.py
Введите коэффициент A:
2
Введите коэффициент B:
6
Введите коэффициент C:
7
Нет корней
PS C:\Users\vet_v\Desktop\RIP> & c:/Users/vet_v/Desktop/RIP/lab_4/.venv/Scripts/python.exe c:/Users/vet_v/Desktop/RIP/lab_1/main.py
Введите коэффициент A:
1
Введите коэффициент B:
-5
Введите коэффициент C:
-6
Два корня: 2.449489742783178 и -2.449489742783178
PS C:\Users\vet_v\Desktop\RIP>
```