

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

Факультет «Радиотехнический»  
Кафедра «Системы обработки информации и управления»

Курс «Парадигмы и конструкции языков программирования»

Отчет по лабораторной работе № 1  
«Основные конструкции языка Python»  
Вариант № 13

Выполнил:  
студент группы РТ5-31Б  
Романов Ю.И.  
Подпись и дата:

Проверил:  
к.т.н., доц. каф. ИУ5  
Гапанюк Ю.Е.  
Подпись и дата:

Москва, 2025 г.

## Описание задания

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 ( 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.
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.
5. Дополнительное задание 1 (\*). Разработайте две программы на языке Python - одну с применением процедурной парадигмы, а другую с применением объектно-ориентированной парадигмы.
6. Дополнительное задание 2 (\*). Разработайте две программы - одну на языке Python, а другую на любом другом языке программирования (кроме C++).

## Текст программы

main.py:

```
import sys
import math

def get_coef(index, prompt):
    try:
        if len(sys.argv) > index:
            return float(sys.argv[index])
```

```
except (IndexError, ValueError):  
    pass
```

```
while True:  
    try:  
        return float(input(prompt))  
    except ValueError:  
        print("Ошибка! Коэффициент должен быть действительным числом")
```

```
def get_biquadratic_roots(a, b, c):  
    result = []
```

```
    if a == 0:  
        print("В биквадратном уравнении a не может равняться 0")  
        return result
```

```
    D = b*b - 4*a*c
```

```
    if D < 0.0:  
        return result  
    elif D == 0.0:  
        t = -b / (2.0*a)  
        if t > 0:  
            root = math.sqrt(t)  
            result.append(root)  
            result.append(-root)  
        elif t == 0:  
            result.append(0.0)
```

```
    else:  
        sqrt_D = math.sqrt(D)  
        t1 = (-b + sqrt_D) / (2.0*a)  
        t2 = (-b - sqrt_D) / (2.0*a)
```

```
        if t1 > 0:  
            root = math.sqrt(t1)  
            result.append(root)  
            result.append(-root)  
        elif t1 == 0:  
            result.append(0.0)
```

```
        if t2 > 0:  
            root = math.sqrt(t2)  
            result.append(root)  
            result.append(-root)  
        elif t2 == 0:
```

```

        result.append(0.0)

    result = sorted(set(result))
    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_biquadratic_roots(a, b, c)

    len_roots = len(roots)
    if len_roots == 0:
        print('Действительных корней нет')
    elif len_roots == 1:
        print(f'Один действительный корень: {roots[0]}')
    elif len_roots == 2:
        print(f'Два действительных корня: {roots[0]} и {roots[1]}')
    elif len_roots == 3:
        print(f'Три действительных корня: {roots[0]}, {roots[1]} и {roots[2]}')
    elif len_roots == 4:
        print(f'Четыре действительных корня: {roots[0]}, {roots[1]}, {roots[2]} и {roots[3]}')

if __name__ == "__main__":
    main()

```

## Экранные формы с примерами выполнения программы

```

● yuriy@192 lab1 % /usr/local/bin/python3
Введите коэффициент a: 1
Введите коэффициент b: 2
Введите коэффициент c: 3
Действительных корней нет

```

```

● yuriy@192 lab1 % python3 main.py 1 0 -1
Два действительных корня: -1.0 и 1.0

```

```

● yuriy@192 lab1 % /usr/local/bin/python3 "/Users/yu
Введите коэффициент a: 1
Введите коэффициент b: -5
Введите коэффициент c: 4
Четыре действительных корня: -2.0, -1.0, 1.0 и 2.0

```