

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

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

Курс «ПиКЯП»

Отчет по лабораторной работе №1

«Объектно-ориентированные возможности языка Python.»

Выполнил:

студент группы ИУ5-36Б

Рухлин Алексей

Проверил:

преподаватель каф.  
ИУ5

Нардид А. Н.

Москва, 2024 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

```
import sys
import math
```

```
def get_coefficient(prompt):
    while True:
```

```

    try:
        return float(input(prompt))
    except ValueError:
        print("Некорректный ввод. Попробуйте снова.")

def calculate_discriminant(a, b, c):
    return b**2 - 4*a*c

def solve_biquadratic(a, b, c):
    d = calculate_discriminant(a, b, c)
    if d < 0:
        print("Уравнение не имеет действительных корней.")
        return

    sqrt_d = math.sqrt(d)
    r1 = (-b + sqrt_d) / (2*a)
    r2 = (-b - sqrt_d) / (2*a)

    roots = []
    if r1 >= 0:
        sqrt_r1 = math.sqrt(r1)
        roots.extend([sqrt_r1, -sqrt_r1])
    if r2 >= 0:
        sqrt_r2 = math.sqrt(r2)
        roots.extend([sqrt_r2, -sqrt_r2])

    if not roots:
        print("Уравнение не имеет действительных корней.")
    else:
        print("Действительные корни уравнения:", sorted(set(roots)))

def main():
    if len(sys.argv) == 4:
        try:
            a = float(sys.argv[1])

```

```

        b = float(sys.argv[2])
        c = float(sys.argv[3])
    except ValueError:
        print("Некорректные параметры командной строки. Пожалуйста,
введите значения с клавиатуры.")
        a = get_coefficient("Введите A: ")
        b = get_coefficient("Введите B: ")
        c = get_coefficient("Введите C: ")
    else:
        a = get_coefficient("Введите A: ")
        b = get_coefficient("Введите B: ")
        c = get_coefficient("Введите C: ")

    solve_biquadratic(a, b, c)

if __name__ == "__main__":
    main()
...

---
```

## Экранные формы с примерами выполнения программы

### Пример 1. Пользователь вводит коэффициенты с клавиатуры:

```

Введите A: 1
Введите B: -5
Введите C: 4
Дискриминант: 9.0
Действительные корни уравнения: [-2.0, 2.0, -1.0, 1.0]
```

### Пример 2. Пользователь указывает коэффициенты через командную строку:

```
$ python solve_biquadratic.py 1 -3 2
```

```
Дискриминант: 1.0
```

```
Действительные корни уравнения: [-1.0, 1.0]
```

```
...
```

### **Пример 3. Уравнение не имеет действительных корней:**

```
Введите A: 1
```

```
Введите B: 0
```

```
Введите C: 1
```

```
Дискриминант: -4.0
```

```
Уравнение не имеет действительных корней.
```

### **Заключение**

В рамках лабораторной работы была разработана и протестирована программа для решения биквадратного уравнения. Программа успешно обрабатывает ввод данных, вычисляет дискриминант и выводит корни уравнения, если они существуют. Также реализована обработка некорректного ввода.