

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

**Факультет «Радиотехнический»
Кафедра ИУ5 «Системы обработки информации и управления»**

Курс «Базовые компоненты интернет-технологий»

Отчёт по лабораторной работе № 1

Выполнил:

студент группы РТ5-31Б
Сафонов Федор

Подпись и дата:

Проверил:

преподаватель каф. ИУ5
Гапанюк Ю.Е.

Подпись и дата:

Постановка задачи

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.

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:
        # Попробуем прочитать коэффициент из командной строки
        coef_str = sys.argv[index]
        coef = float(coef_str)
    except:
        # Вводим с клавиатуры
        while True:
            try:
                print(prompt)
                coef = float(input())
            except:
                continue
            else:
                break
    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)
```

```
    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)
```

```
result = list(filter(lambda x: x >= 0, result))
```

```
result_bi = []
```

```
for elem in result:
```

```
    if (math.fabs(elem) == 0):
```

```
        result_bi.append(0)
```

```
        continue
```

```
    x = math.sqrt(elem)
```

```
    result_bi.append(x)
```

```
    result_bi.append(-x)
```

```
return result_bi
```

```
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('Три корня: {} и {}'.format(roots[0], roots[1]))
```

```
elif len_roots == 3:
```

```
    print('Три корня: {} и {} и {}'.format(roots[0], roots[1], roots[2]))
```

```
elif len_roots == 4:
```

```
    print('4 корня: ', ', '.join(list(map(str, roots))))
```

```
# Если сценарий запущен из командной строки
```

```
if __name__ == "__main__":
```

```
    main()
```

```
# Пример запуска  
# qg.py 1 0 -4
```

Результаты

```
Введите коэффициент A:  
3  
Введите коэффициент B:  
6  
Введите коэффициент C:  
-1  
Введите коэффициент C:  
0  
Три корня: 0.5773502691896257 и -0.5773502691896257 и 0  
  
Process finished with exit code 0  
|
```