

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

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

Курс «Парадигмы и конструкции языков программирования»

Отчет по ЛР№1

Выполнил:
студент группы ИУ5-34Б
Кожевников М. А.
Подпись и дата:

Проверил:
преподаватель каф. ИУ5
Гапанюк Ю.Е.
Подпись и дата:

Москва, 2023 г.

Задача:

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.
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

def get_params():
    try:
        params = [float(i) for i in sys.argv[1:]]
    except:
        print("Ошибка при взятии аргументов из cmd")
    else:
        if len(params) == 3 and params[0] != 0:
            print("Взяты параметры из cmd")
            return params
        else:
            print("Ошибка при взятии аргументов из cmd")

    while len(params) != 3 or params[0] == 0:
        try:
            prompt = "Введите 3 действительных числа через пробел (А, В, С): "
            params = list(map(float, input(prompt).split()))
        except:
            print("Error")
        else:
            if len(params) != 3 or params[0] == 0:
                print("Error")
    return params

def find_roots(arr):
    roots = []
    d = float(arr[1] ** 2 - 4 * arr[0] * arr[2])
    if d == 0:
        x = -arr[1] / (2 * arr[0])
        roots.append(x)
    elif d > 0:
        x1 = (-arr[1] + d ** 0.5) / (2 * arr[0])
        x2 = (-arr[1] - d ** 0.5) / (2 * arr[0])
        if x1 > 0:
            roots.append(x1 ** 0.5)
            roots.append(-(x1 ** 0.5))
        if x2 > 0:
            roots.append(x2 ** 0.5)
            roots.append(-(x2 ** 0.5))
    return roots

def main():
    params = get_params()
    roots = find_roots(params)
    if len(roots) >= 1:
        for el in roots:
            print("X = {:.6}".format(el))
    else:
        print("Корней нет")

if '__main__' == __name__:
    main()
```

Результат работы программы:

```
Введите 3 действительных числа через пробел (A, B, C): 1 -10 24
X = 2.44949
X = -2.44949
X = 2.0
X = -2.0
```

```
Введите 3 действительных числа через пробел (A, B, C): 1 n 4
Еггор
Введите 3 действительных числа через пробел (A, B, C): 1 2 3
Корней нет
```