**Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования**

**«Московский государственный технический университет**

**имени Н.Э. Баумана (национальный исследовательский университет)»**

**(МГТУ им. Н.Э. Баумана)**

**Факультет «Информатика и системы управления»**

**Кафедра «Системы обработки информации и управления»**

Отчёт по лабораторной работе №1

Выполнил:

студент группы ИУ5-33Б

Александр Саргсян

Проверил:

к.т.н., доц., Ю. Е. Гапанюк

2022 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, C могут быть заданы в виде параметров командной строки. Если они не заданы, то вводятся с клавиатуры в соответствии с пунктом 2.
4. Если коэффициент А, В, 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):

    try:

        coef\_str = sys.argv[index]

        coef = float(coef\_str)

    except:

         while True:

            try:

                print(prompt)

                coef\_str = input()

                coef = float(coef\_str)

                return coef

            except ValueError:

                print ('Попробуйте снова!')

    return coef

def get\_roots(a, b, c):

    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)

    return result

def sec\_get\_roots(a, b, c):

    result = []

    D = b\*b - 4\*a\*c

    if D == 0.0:

        root = -b / (2.0\*a)

        # Проверка на то, является ли корень положительным или равным 0.

        if (root > 0.0):

            result.extend([math.sqrt(root), math.sqrt(root)\*(-1.0)])

        elif root == 0:

            result.append(root)

    elif D > 0.0:

        sqD = math.sqrt(D)

        root1 = (-b + sqD) / (2.0\*a)

        root2 = (-b - sqD) / (2.0\*a)

        if root1 > 0:

            result.extend([math.sqrt(root1), math.sqrt(root1) \* (-1.0)])

        elif root1 == 0:

            result.append(root1)

        if root2 > 0:

            result.extend([math.sqrt(root2), math.sqrt(root2) \* (-1.0)])

        elif root2 == 0:

            result.append(root2)

    return result

def main():

    a = get\_coef(1, 'Введите коэффициент А:')

    b = get\_coef(2, 'Введите коэффициент B:')

    c = get\_coef(3, 'Введите коэффициент C:')

    roots = sec\_get\_roots(a,b,c)

    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]))

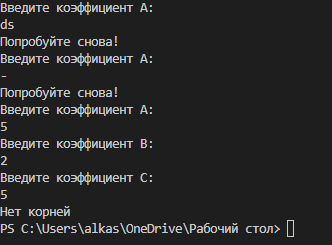
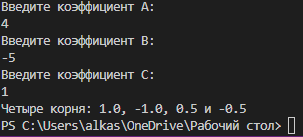
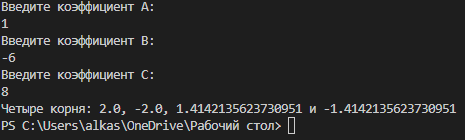
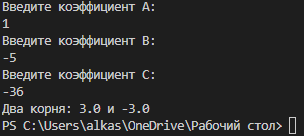
    elif len\_roots == 4:

        print('Четыре корня: {}, {}, {} и {}'.format(roots[0], roots[1], roots[2], roots[3]))

if \_\_name\_\_ == "\_\_main\_\_":

    main()

**Примеры выполнения:**

****