**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 лабораторной работе №1

«**Основные конструкции языка Python.**»

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-31Б |  | преподаватель каф. ИУ5 |
| Ларкин Б. В. |  | Гапанюк Ю. Е. |
| Подпись и дата: |  | Подпись и дата: |
|  |  |  |

Москва, 2023 г.

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 А, В, С, вычисляет дискриминант и ДЕЙСТВИТЕЛЬНЫЕ корни уравнения (в зависимости от дискриминанта).
3. Коэффициенты А, В, С могут быть заданы в виде параметров командной строки ( [вариант задания параметров приведен в конце файла с примером кода](https://github.com/ugapanyuk/BKIT_2022/blob/main/code/lab1_code) ). Если они не заданы, то вводятся с клавиатуры в соответствии с пунктом 2. [Описание работы с параметрами командной строки.](https://realpython.com/python-command-line-arguments/#the-command-line-interface)
4. Если коэффициент А, В, С 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

**\_\_main\_\_.py**

import sys

import math

def get\_coef(index, prompt):

    '''

    Reads a coefficient, be it KBM or console

    Args:

        index (int): Number of the parameter in the console string

        prompt (str): Suggests an input

    Returns:

        float: coefficient of the biquadratic equation

    '''

    try: #to get a coeff from cmd

        coef\_str = sys.argv[index]

    except:

        # KBM input

        print(prompt)

        coef\_str = input()

    coef = float(coef\_str)

    return coef

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

    '''

    Solving a biquadratic equation

    Args:

        a (float): coef A

        b (float): coef B

        c (float): coef C

    Returns:

        list[float]: root list

    '''

    q\_roots = [] #Roots of the quadratic eq

    D = b\*b - 4\*a\*c

    if D == 0.0:

        root = -b / (2.0\*a)

        q\_roots.append(root)

    elif D > 0.0:

        sqD = math.sqrt(D)

        root1 = (-b + sqD) / (2.0\*a)

        root2 = (-b - sqD) / (2.0\*a)

        q\_roots.append(root1)

        q\_roots.append(root2)

    biq\_roots = []

    for i in q\_roots:

        if i == 0.0:

            biq\_roots.append(0) #-0 = +0

        elif i>0: #if i<0 the biquadratic eq roots become imaginary, which is forbidden

            biq\_roots.append(math.sqrt(i))

            biq\_roots.append(-math.sqrt(i))

    return biq\_roots

def main():

    '''

    Main function

    '''

    a = get\_coef(1, 'Enter coefficient A:')

    b = get\_coef(2, 'Enter coefficient B:')

    c = get\_coef(3, 'Enter coefficient C:')

    # Calculating roots

    roots = get\_roots(a,b,c)

    # Roots output

    len\_roots = len(roots)

    if len\_roots == 0:

        print('No roots')

    elif len\_roots == 1:

        print('One root: {}'.format(roots[0]))

    elif len\_roots == 2:

        print('Two roots: {} & {}'.format(roots[0], roots[1]))

    elif len\_roots == 3:

        print('Three roots: {} & {} & {}'.format(roots[0], roots[1], roots[2], ))

    elif len\_roots == 4:

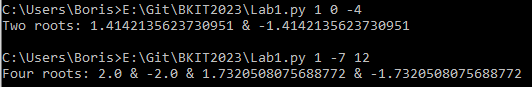
        print('Four roots: {} & {} & {} & {}'.format(roots[0], roots[1], roots[2], roots[3]))

# If executed from cmd

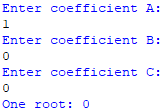
if \_\_name\_\_ == "\_\_main\_\_":

    main()

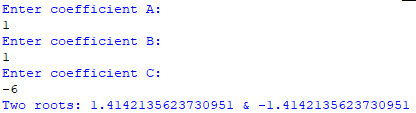
Анализ результатов

Запуск через консоль:

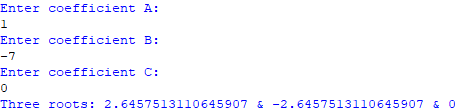
1 корень:



2 корня:



3 корня:



4 корня:

