**Московский государственный технический**

**университет им. Н.Э. Баумана**

Факультет «Радиотехнический»

Кафедра «Системы обработки информации и управления»

Курс «Парадигмы и конструкции языков программирования»

Отчет по лабораторной работе №2

«Решение биквадратного уравнения на языке Python с использование ООП»

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы РТ5-31Б: |  | преподаватель кафедры ИУ5 |
| Шарафутдинов М.Э. |  | Гапанюк Ю.Е. |
| Подпись и дата: |  | Подпись и дата: |

Москва, 2024 г.

Постановка задачи

Разработать программу для решения биквадратного уравнения в объектно-ориентированной парадигме.

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). Если они не заданы, то вводятся с клавиатуры в соответствии с пунктом 2. Описание работы с параметрами командной строки.
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

import math

class Bqe:

    def \_\_init\_\_(self):

        self.a = 0.0

        self.b = 0.0

        self.c = 0.0

        self.D = -1

        self.roots = []

    def get\_coef(self, prompt):

        print(prompt)

        coef\_str = input()

        while (True):

            try:

                coef = float(coef\_str)

                break

            except:

                print(prompt)

                coef\_str = input()

        return coef

    def get\_coefs(self):

        while(self.D < 0):

            self.a = self.get\_coef('Введите коэффициент A: ')

            self.b = self.get\_coef('Введите коэффициент B: ')

            self.c = self.get\_coef('Введите коэффициент C: ')

            self.D = self.b\*self.b - 4\*self.a\*self.c

            if (self.D < 0):

                print('Дискриминант D =', self.D, 'меньше нуля')

    def calculate\_roots(self):

        D = math.sqrt(self.D)

        root1 = (-self.b + D) / (2 \* self.a)

        root2 = (-self.b - D) / (2 \* self.a)

        if (root1 >= 0):

            self.roots.append(math.sqrt(root1))

            self.roots.append(-self.roots[-1])

        if (root2 >= 0 and root2 != root1):

            self.roots.append(math.sqrt(root2))

            self.roots.append(-self.roots[-1])

    def print\_roots(self):

        if (len(self.roots) == 0):

            print('Нет корней')

        else:

            print('Корни уравнения: ', end = '')

            for j in self.roots:

                print(j, end = ' ')

            print('')

def main():

    r = Bqe()

    r.get\_coefs()

    r.calculate\_roots()

    r.print\_roots()

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

    main()

Результат выполнения программы

