**Московский государственный технический**

**университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»

Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Электротехника»

Отчет по Лабораторной работе 1.

| Выполнил: |  | Проверил: |
| --- | --- | --- |
| студент группы ИУ5-33Б |  | преподаватель каф. ИУ5 |
| Комаров Дмитрий |  | Белодедов М.В. |
| Подпись и дата: |  | Подпись и дата: |

Москва, 2022 г.

**Задание:**

Разработать программу для решения биквадратного уравнения.

Программа должна быть разработана в виде консольного приложения на языке Python.

Программа осуществляет ввод с клавиатуры коэффициентов А, В, С, вычисляет дискриминант и ДЕЙСТВИТЕЛЬНЫЕ корни уравнения (в зависимости от дискриминанта).

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.

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.

**Код программы:**

**import sys**

**import math**

**class SquareRoots:**

**def \_\_init\_\_(self):**

**self.coef\_A = 0.0**

**self.coef\_B = 0.0**

**self.coef\_C = 0.0**

**self.num\_roots = 0**

**self.roots\_list = []**

**def get\_coef(self, index, prompt):**

**try:**

**coef\_str = sys.argv[index]**

**except:**

**while(True):**

**try:**

**print(prompt)**

**coef\_str = input()**

**coef = float(coef\_str)**

**return coef**

**except ValueError:**

**pass**

**return float(coef\_str)**

**def get\_coefs(self):**

**self.coef\_A = self.get\_coef(1, 'Введите коэффициент А:')**

**self.coef\_B = self.get\_coef(2, 'Введите коэффициент B:')**

**self.coef\_C = self.get\_coef(3, 'Введите коэффициент C:')**

**def calculate\_roots(self):**

**a = self.coef\_A**

**b = self.coef\_B**

**c = self.coef\_C**

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

**if D == 0.0:**

**root1 = -b / (2.0\*a)**

**if (root1 > 0):**

**root1 = (root1)\*\*0.5**

**root2 = -root1**

**self.num\_roots = 2**

**self.roots\_list.append(root1, root2) #!!!!!!!!!!!!!!!!**

**elif(root1 == 0):**

**self.num\_roots = 1**

**self.roots\_list.append(0)**

**else:**

**self.num\_roots = 0**

**elif D > 0.0:**

**sqD = math.sqrt(D)**

**root1 = (-b + sqD) / (2.0\*a)**

**root3 = (-b - sqD) / (2.0\*a)**

**if (root1 > 0):**

**root1 = (root1)\*\*0.5**

**root2 = -root1**

**self.num\_roots += 2**

**self.roots\_list.append(root1)**

**self.roots\_list.append(root2)**

**elif(root1 == 0):**

**self.num\_roots += 1**

**self.roots\_list.append(0)**

**else:**

**self.num\_roots = 0**

**if (root3 > 0):**

**root3 = (root3)\*\*0.5**

**root4 = -root3**

**self.num\_roots += 2**

**self.roots\_list.append(root3)**

**self.roots\_list.append(root4)**

**elif(root3 == 0):**

**self.num\_roots += 1**

**self.roots\_list.append(0)**

**else:**

**self.num\_roots = 0**

**def print\_roots(self):**

**if self.num\_roots != len(self.roots\_list):**

**print(('Ошибка. Уравнение содержит {} действительных корней, ' +\**

**'но было вычислено {} корней.').format(self.num\_roots, len(self.roots\_list)))**

**else:**

**if self.num\_roots == 0:**

**print('Нет корней')**

**elif self.num\_roots == 2:**

**print(('Два кореня: {} и {}').format(self.roots\_list[0], self.roots\_list[1]))**

**elif self.num\_roots == 4:**

**print(('Четыре корня: {} , {} , {} и {}').format(self.roots\_list[0],**

**self.roots\_list[1],**

**self.roots\_list[2],**

**self.roots\_list[3]))**

**def main():**

**r = SquareRoots()**

**r.get\_coefs()**

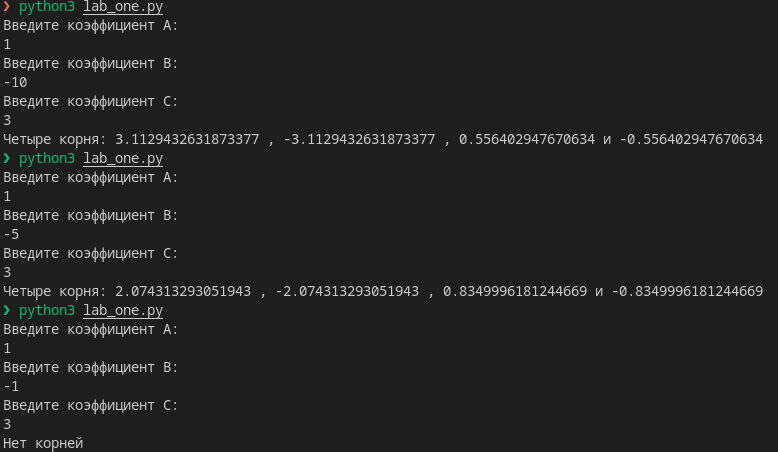
**r.calculate\_roots()**

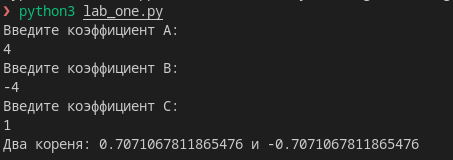
**r.print\_roots()**

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

**main()**

**Тесты:**

****

****