**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 ИУ5-34Б |  | преподаватель каф. ИУ5 |
| Бромберг Е.А. |  | Нардид А.Н. |
|  |  |  |

Москва, 2024 г.

**Постановка задачи**

**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, 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**

Main.py

import sys  
import math  
  
class SquareRoots:  
  
 def \_\_init\_\_(self):  
 *'''  
 Конструктор класса  
 '''* # Объявление коэффициентов  
 self.coef\_A = 0.0  
 self.coef\_B = 0.0  
 self.coef\_C = 0.0  
 # Количество корней  
 self.num\_roots = 0  
 # Список корней  
 self.roots\_list = []  
  
 def get\_coef(self, index, prompt):  
 *'''  
 Читаем коэффициент из командной строки или вводим с клавиатуры  
 Args:  
 index (int): Номер параметра в командной строке  
 prompt (str): Приглашение для ввода коэффицента  
 Returns:  
 float: Коэффициент квадратного уравнения  
 '''* try:  
 # Пробуем прочитать коэффициент из командной строки  
 coef\_str = sys.argv[index]  
 except:  
 # Вводим с клавиатуры  
 print(prompt)  
 coef\_str = input()  
 # Переводим строку в действительное число  
 coef = float(coef\_str)  
 return coef  
  
 def get\_coefs(self):  
 *'''  
 Чтение трех коэффициентов  
 '''* while True:  
 try:  
 self.coef\_A = self.get\_coef(1, 'Введите коэффициент А:')  
 if self.coef\_A == 0:  
 print("Коэффициент A не должен равняться 0!")  
 sys.argv = []  
 continue  
 self.coef\_B = self.get\_coef(2, 'Введите коэффициент B:')  
 self.coef\_C = self.get\_coef(3, 'Введите коэффициент C:')  
 break  
 except:  
 print("Введите числа!")  
 sys.argv = []  
  
  
 def calculate\_roots(self):  
 *'''  
 Вычисление корней квадратного уравнения  
 '''* a = self.coef\_A  
 b = self.coef\_B  
 c = self.coef\_C  
 # Вычисление дискриминанта и корней  
 D = b\*b - 4\*a\*c  
 if D == 0.0:  
 if (-b / (2.0\*a)) > 0.0:  
 root1 = -math.sqrt(-b / (2.0\*a))  
 root2 = math.sqrt(-b / (2.0\*a))  
 self.num\_roots = 2  
 self.roots\_list.append(root1)  
 self.roots\_list.append(root2)  
 if (-b / (2.0\*a)) == 0.0:  
 root1 = 0.0  
 self.num\_roots = 1  
 self.roots\_list.append(root1)  
 elif D > 0.0:  
 sqD = math.sqrt(D)  
 if ((-b - sqD) / (2.0\*a)) > 0.0:  
 self.num\_roots = 2  
 root1 = -math.sqrt((-b - sqD) / (2.0\*a))  
 root2 = math.sqrt((-b - sqD) / (2.0\*a))  
 self.roots\_list.append(root1)  
 self.roots\_list.append(root2)  
 if ((-b - sqD) / (2.0\*a)) == 0.0:  
 self.num\_roots = 1  
 root1 = 0.0  
 self.roots\_list.append(root1)  
 if ((-b + sqD) / (2.0\*a)) > 0.0:  
 self.num\_roots += 2  
 root1 = -math.sqrt((-b + sqD) / (2.0\*a))  
 root2 = math.sqrt((-b + sqD) / (2.0\*a))  
 self.roots\_list.append(root1)  
 self.roots\_list.append(root2)  
 if ((-b + sqD) / (2.0\*a)) == 0.0:  
 self.num\_roots += 1  
 root1 = 0.0  
 self.roots\_list.append(root1)  
  
 def print\_roots(self):  
 # Проверка отсутствия ошибок при вычислении корней  
 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 == 1:  
 print('Один корень: {}'.format(self.roots\_list[0]))  
 elif self.num\_roots == 2:  
 print('Два корня: {} и {}'.format(self.roots\_list[0], \  
 self.roots\_list[1]))  
 elif self.num\_roots == 3:  
 print('Три корня: {}, {} и {}'.format(self.roots\_list[0], \  
 self.roots\_list[1], self.roots\_list[2]))  
 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():  
 *'''  
 Основная функция  
 '''* # Создание объекта класса  
 r = SquareRoots()  
 # Последовательный вызов необходимых методов  
 r.get\_coefs()  
 r.calculate\_roots()  
 r.print\_roots()  
  
# Если сценарий запущен из командной строки  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

**Экранные формы с примерами выполнения программы**



