**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 №1

«Основные конструкции языка C#»

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы РТ5-31Б: |  | преподаватель кафедры ИУ5 |
| Кузнецов С. А. |  | Гапанюк Ю. Е. |
| Подпись и дата: |  | Подпись и дата: |

Москва, 2024 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. Если коэффициент А, В, С введен некорректно, то необходимо проигнорировать некорректное значение и вводить коэффициент повторно.
4. Корни уравнения выводятся зелёным цветом. Если корней нет, то сообщение выводится красным цветом.
5. Коэффициенты А, В, С могут быть заданы в виде параметров командной строки. Если они не заданы, то вводятся с клавиатуры в соответствии с пунктом 2. Проверка из пункта 3 в этом случае производится для параметров командной строки без повторного ввода с клавиатуры.
6. Решить с помощью ооп и процедурной парадигмы

Текст программы

Файл «Polinom.py»(ООП)

import math  
import sys  
  
  
class Polinom:  
 def \_\_init\_\_(self):  
 self.rootsArr = None  
 self.a = 0.0  
 self.b = 0.0  
 self.c = 0.0  
  
 def getCoef(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 enterCoefs(self):  
 while True:  
 self.a = self.getCoef(1, "A = ")  
 self.b = self.getCoef(2, "B = ")  
 self.c = self.getCoef(3, "C = ")  
 if self.a == 0:  
 print("invalid data")  
 continue  
 else:  
 break  
  
 def calculate(self):  
  
 self.enterCoefs()  
 d = self.b \*\* 2 - 4 \* self.a \* self.c  
 if d < 0.0:  
 self.rootsArr = []  
 return []  
 elif d == 0.0:  
 x = (-self.b) / (2 \* self.a)  
 self.rootsArr = self.processX(x)  
 else:  
 x1 = (-self.b + math.sqrt(d)) / (2 \* self.a)  
 x2 = (-self.b - math.sqrt(d)) / (2 \* self.a)  
  
 self.rootsArr = self.processX(x1) + self.processX(x2)  
 return self.rootsArr  
  
 @staticmethod  
 def processX(x):  
 if x < 0.0:  
 return []  
 elif x == 0.0:  
 return [0.0]  
 else:  
 return [math.sqrt(x), -math.sqrt(x)]  
  
  
def main():  
 pol = Polinom()  
 root = pol.calculate()  
  
 print(root)  
  
  
# Если сценарий запущен из командной строки  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

файл «Process.py»(процедурная)

import math  
import sys  
  
  
def getCoef(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 is\_valid(a):  
 if a == 0:  
 return False  
 else:  
 return True  
  
  
def process\_roots(x):  
 if x > 0:  
 return [math.sqrt(x), -math.sqrt(x)]  
 elif x == 0:  
 return [0]  
 else:  
 return []  
  
  
def get\_roots(a, b, d):  
 if d < 0:  
 return []  
 elif d == 0:  
 x = -b / (2 \* a)  
 return process\_roots(x)  
 elif d > 0:  
 x1 = (-b + math.sqrt(d)) / (2 \* a)  
 x2 = (-b - math.sqrt(d)) / (2 \* a)  
 return process\_roots(x1) + process\_roots(x2)  
  
  
def calculate\_roots():  
 a = 0.0  
 b = 0.0  
 c = 0.0  
 while not is\_valid(a):  
 a = getCoef(1, "A = ")  
 b = getCoef(1, "B = ")  
 c = getCoef(1, "C = ")  
 d = (b \*\* 2) - 4 \* a \* c  
 return get\_roots(a, b, d)  
  
  
def main():  
 print(calculate\_roots())  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 main()

Экранные формы с примерами выполнения программы

Изображение выглядит как снимок экрана, Шрифт, дизайн

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, Шрифт, дизайн

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, черный

Автоматически созданное описание