**Московский государственный технический**

**Университет им. Н.Э. Баумана**

**Факультет «Радиотехничский»**

**Кафедра «Системы обработки информации и управления»**

Курс «Парадигмы и конструкции языков программирования»

Отчет по лабораторной работе №1

«Изучение основных конструкций языка Python»

Выполнил:

студент группы РТ5-31Б

Фешин Д.А.

Проверил:

Гапанюк Е.Ю.

2024 г.

**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.

Дополнительное задание 1 (\*). Разработайте две программы на языке Python - одну с применением процедурной парадигмы, а другую с применением объектно-ориентированной парадигмы.

Дополнительное задание 2 (\*). Разработайте две программы - одну на языке Python, а другую на любом другом языке программирования (кроме С++).

**Текст программы**

Python

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

'''

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):

'''

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:

root = -b / (2.0\*a)

D\_2 = -4\*(-root)

if D\_2 == 0.0:

root1 = 0

self.num\_roots = 1

self.roots\_list.append(root1)

elif D\_2 > 0:

sqD = math.sqrt(D\_2)

root1 = sqD / 2.0

root2 = -sqD / 2.0

self.num\_roots = 2

self.roots\_list.append(root1)

self.roots\_list.append(root2)

elif D > 0.0:

sqD = math.sqrt(D)

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

root2 = (-b - sqD) / (2.0\*a)

D\_2 = -4 \* (-root1)

D\_3 = -4 \* (-root2)

if (D\_2 == 0) and (D\_3 > 0):

root3 = 0

sqD = math.sqrt(D\_3)

root4 = sqD / 2.0

root5 = -sqD / 2.0

self.num\_roots = 3

self.roots\_list.append(root3)

self.roots\_list.append(root4)

self.roots\_list.append(root5)

elif (D\_2 > 0) and (D\_3 == 0):

root3 = 0

sqD = math.sqrt(D\_2)

root4 = sqD / 2.0

root5 = -sqD / 2.0

self.num\_roots = 3

self.roots\_list.append(root3)

self.roots\_list.append(root4)

self.roots\_list.append(root5)

elif (D\_2 > 0) and (D\_3 > 0):

sqD\_1 = math.sqrt(D\_2)

root3 = sqD\_1 / 2.0

root4 = -sqD\_1 / 2.0

sqD\_2 = math.sqrt(D\_3)

root5 = sqD\_2 / 2.0

root6 = -sqD\_2 / 2.0

self.num\_roots = 4

self.roots\_list.append(root3)

self.roots\_list.append(root4)

self.roots\_list.append(root5)

self.roots\_list.append(root6)

elif (D\_2 < 0) and (D\_3 > 0):

sqD = math.sqrt(D\_3)

root4 = sqD / 2.0

root5 = -sqD / 2.0

self.num\_roots = 2

self.roots\_list.append(root4)

self.roots\_list.append(root5)

elif (D\_2 > 0) and (D\_3 < 0):

sqD = math.sqrt(D\_2)

root4 = sqD / 2.0

root5 = -sqD / 2.0

self.num\_roots = 2

self.roots\_list.append(root4)

self.roots\_list.append(root5)

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()

C#

using System;

class QuadraticEquationSolver

{

private double a, b, c;

public QuadraticEquationSolver(double a, double b, double c)

{

this.a = a;

this.b = b;

this.c = c;

}

public void Solve()

{

if (Math.Abs(a) < double.Epsilon)

{

Console.WriteLine("Это не квадратное уравнение.");

return;

}

double discriminant = b \* b - 4 \* a \* c;

if (discriminant > 0)

{

double sqrtDiscriminant = Math.Sqrt(discriminant);

double x1 = (-b + sqrtDiscriminant) / (2 \* a);

double x2 = (-b - sqrtDiscriminant) / (2 \* a);

Console.WriteLine($"Два корня: {x1} и {x2}");

}

else if (discriminant == 0)

{

double x = -b / (2 \* a);

Console.WriteLine($"Один корень: {x}");

}

else

{

Console.WriteLine("Корней нет (в области действительных чисел).");

}

}

}

class Program

{

static void Main()

{

Console.Write("Введите коэффициент a: ");

double a = double.Parse(Console.ReadLine());

Console.Write("Введите коэффициент b: ");

double b = double.Parse(Console.ReadLine());

Console.Write("Введите коэффициент c: ");

double c = double.Parse(Console.ReadLine());

QuadraticEquationSolver solver = new QuadraticEquationSolver(a, b, c);

solver.Solve();

}

}