**Министерство науки и высшего образования Российской Федерации** **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования** **«Московский государственный технический университет** **имени Н.Э. Баумана**

**(национальный исследовательский университет)»**

**(МГТУ им. Н.Э. Баумана)**

**Факультет «Информатика и системы управления»**

**Кафедра ИУ5 «Системы обработки информации и управления»**

Курс «Базовые компоненты интернет-технологий»

Отчет по лабораторной работе №1

«Основные конструкции языка Python»

Выполнил:

студент группы ИУ5-33Б

Юрова Е.О.

Проверил:

2021 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. 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 sys  
import math  
  
  
def get\_coef(index, prompt):  
 try:  
 coef\_str = sys.argv[index]  
 except:  
 print(prompt)  
 try:  
 coef\_str = input()  
 except:  
 return get\_coef(index,prompt)  
 coef = float(coef\_str)  
 if index ==1 and coef ==0.0:  
 print ("Enter the correct coefficient")  
 return get\_coef(index, prompt)  
 return coef  
  
  
def get\_roots(a, b, c):  
 result = []  
 D = b \* b - 4 \* a \* c  
 if D == 0.0:  
 root = -b/(2.0\*a)  
 root1 = math.sqrt(-b/(2.0\*a))  
 D = b\*b - 4\*a\*c  
 if D == 0.0:  
 root = -b / (2.0\*a)  
 if root == 0:  
 result.append(0)  
 else:  
 result.append(root1)  
 result.append(-root1)  
  
 elif D > 0.0:  
 sqD = math.sqrt(D)  
 if (-b + sqD) / (2.0\*a) >= 0 and (-b - sqD) / (2.0\*a) >= 0:  
 root1 = math.sqrt((-b + sqD) / (2.0\*a))  
 root3 = -math.sqrt((-b + sqD) / (2.0\*a))  
 root2 = math.sqrt((-b - sqD) / (2.0\*a))  
 root4 = -math.sqrt((-b - sqD) / (2.0\*a))  
 if root1 == 0 and root2 != 0:  
 result.append(0)  
 result.append(root2)  
 result.append(root4)  
 elif root1 != 0 and root2 == 0:  
 result.append(0)  
 result.append(root1)  
 result.append(root3)  
 elif root2 == root4 and root2 != 0:  
 result.append(root1)  
 result.append(root2)  
 else:  
 result.append(root1)  
 result.append(root2)  
 result.append(root3)  
 result.append(root4)  
 elif (-b + sqD) / (2.0\*a) >= 0 and (-b - sqD) / (2.0\*a) < 0:  
 root1 = math.sqrt((-b + sqD) / (2.0\*a))  
 root3 = -math.sqrt((-b + sqD) / (2.0\*a))  
 result.append(root1)  
 result.append(root3)  
 elif (-b + sqD) / (2.0\*a) < 0 and (-b - sqD) / (2.0\*a) >= 0:  
 root2 = math.sqrt((-b - sqD) / (2.0\*a))  
 root4 = -math.sqrt((-b - sqD) / (2.0\*a))  
 result.append(root2)  
 result.append(root4)  
 return result  
  
  
  
  
  
  
def main():  
 a = get\_coef(1, 'Vvedite koef A:')  
 b = get\_coef(2, 'vvedite koef B:')  
 c = get\_coef(3, 'vvedite koef C:')  
  
 roots = get\_roots(a, b, c)  
 len\_roots = len(roots)  
 if len\_roots == 0:  
 print('No roots')  
 elif len\_roots == 1:  
 print('one roots: {}'.format(roots[0]))  
 elif len\_roots == 2:  
 print('two roots: {} and {}'.format(roots[0], roots[1]))  
 elif len\_roots == 3:  
 print('three roots: {} and {} and {}'.format(roots[0], roots[1], roots[2]))  
 elif len\_roots == 4:  
 print('four roots: {} and {} and {} and {}'.format(roots[0], roots[1], roots[2], roots[3]))  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()  
  
  
# qr.py 1 0 -4

**Экранные формы с примерами выполнения задания:**



