|  |  |
| --- | --- |
|  | **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 «Системы обработки информации и управления»**

Вопияшин Никита ИУ5-35Б

Парадигмы и конструкции языков программирования

**ОТЧЁТ ПО**

**Лабораторной работе №1**

Москва

2023

**Задание.**

Написать консольное приложение на Python для решения биквадратного уравнения. Программа должна выполнять следующие шаги:

* Получить коэффициенты A, B и C от пользователя через ввод с клавиатуры или параметры командной строки.
* Проверить, что введенные коэффициенты могут быть корректно преобразованы в действительные числа. Если какой-то коэффициент введен некорректно, программа должна запросить ввод повторно, пока значения не будут корректными.
* Вычислить дискриминант и вывести корни уравнения в зависимости от значения дискриминанта.

Дополнительное задание: реализовать две версии программы на Python - одну с использованием процедурной парадигмы и другую с использованием объектно-ориентированной парадигмы.

**Текст программы.**

Процедурный подход:

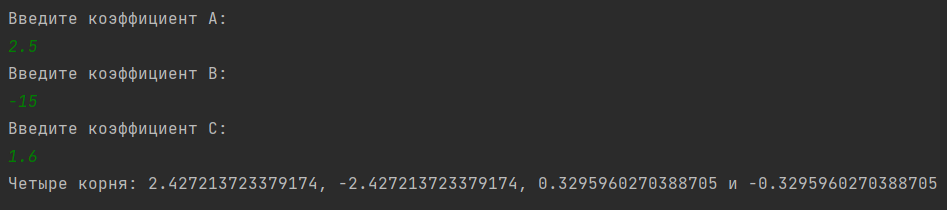
import sys  
import math  
  
def get\_koeff(index):  
 if index == 1:  
 letter = 'A'  
 elif index == 2:  
 letter = 'B'  
 else:  
 letter = 'C'  
 try:  
 koeff\_str = sys.argv[index]  
 except:  
 while True:  
 print('Введите коэффициент {}: '.format(letter))  
 koeff\_str = input()  
 try:  
 koeff = float(koeff\_str)  
 return koeff  
 except ValueError:  
 print('Введено неверное значение. Попробуйте снова.\n')  
  
def get\_roots(a, b, c):  
 result = []  
 D = b\* b - 4 \* a \* c  
 if D == 0.0:  
 root = math.sqrt(-b / (2.0 \* a))  
 if (root == 0.0):  
 result.append(abs(root))  
 else:  
 result.append(root)  
 result.append(-root)  
 elif D > 0.0:  
 sqD = math.sqrt(D)  
 r1 = (-b + sqD) / (2.0 \* a)  
 r2 = (-b - sqD) / (2.0 \* a)  
 if r1 == 0.0:  
 result.append(r1)  
 if r2 == 0.0 and r1 != 0.0:  
 result.append(r2)  
 if r1>0.0:  
 root1 = math.sqrt(r1)  
 result.append(root1)  
 result.append(-root1)  
 if r2>0.0:  
 root2 = math.sqrt(r2)  
 result.append(root2)  
 result.append(-root2)  
 return result  
  
  
def main():  
 a = get\_koeff(1)  
 b = get\_koeff(2)  
 c = get\_koeff(3)  
  
 roots = get\_roots(a, b, c)  
  
 len\_roots = len(roots)  
 if len\_roots == 0:  
 print('Нет корней')  
 elif len\_roots == 1:  
 print('Один корень: {}'.format(roots[0]))  
 elif len\_roots == 2:  
 print('Два корня: {} и {}'.format(roots[0], roots[1]))  
 elif len\_roots == 3:  
 print('Три корня: {}, {} и {}'.format(roots[0], roots[1], roots[2]))  
 elif len\_roots == 4:  
 print('Четыре корня: {}, {}, {} и {}'.format(roots[0], roots[1], roots[2], roots[3]))  
  
  
# Если сценарий запущен из командной строки  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

Объектно-ориентированный подход:

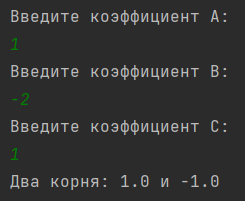
import sys  
import math  
  
  
class Equation:  
 def \_\_init\_\_(self):  
 self.A = 0.0  
 self.B = 0.0  
 self.C = 0.0  
 self.solutions = []  
  
  
 def get\_koeff(self, index):  
 if index == 1:  
 letter = 'A'  
 elif index == 2:  
 letter = 'B'  
 else:  
 letter = 'C'  
 try:  
 koeff\_str = sys.argv[index]  
 except:  
 while True:  
 print('Введите коэффициент {}: '.format(letter))  
 koeff\_str = input()  
 try:  
 koeff = float(koeff\_str)  
 return koeff  
 except ValueError:  
 print('Введено неверное значение. Попробуйте снова.\n')  
 def get\_koeffs(self):  
 self.A = self.get\_koeff(1)  
 self.B = self.get\_koeff(2)  
 self.C = self.get\_koeff(3)  
  
 def get\_roots(self):  
 a = self.A  
 b = self.B  
 c = self.C  
 result = []  
 D = b\*b - 4 \* a \* c  
 if D == 0.0:  
 root = math.sqrt(-b / (2.0 \* a))  
 if (root == 0.0):  
 result.append(abs(root))  
 else:  
 result.append(root)  
 result.append(-root)  
 elif D > 0.0:  
 sqD = math.sqrt(D)  
 r1 = (-b + sqD) / (2.0 \* a)  
 r2 = (-b - sqD) / (2.0 \* a)  
 if r1 == 0.0:  
 result.append(r1)  
 if r2 == 0.0 and r1 != 0.0:  
 result.append(r2)  
 if r1>0.0:  
 root1 = math.sqrt(r1)  
 result.append(root1)  
 result.append(-root1)  
 if r2>0.0:  
 root2 = math.sqrt(r2)  
 result.append(root2)  
 result.append(-root2)  
 return result  
  
  
def main():  
 eq = Equation()  
 eq.get\_koeffs()  
  
 roots = eq.get\_roots()  
  
 len\_roots = len(roots)  
 if len\_roots == 0:  
 print('Нет корней')  
 elif len\_roots == 1:  
 print('Один корень: {}'.format(roots[0]))  
 elif len\_roots == 2:  
 print('Два корня: {} и {}'.format(roots[0], roots[1]))  
 elif len\_roots == 3:  
 print('Три корня: {}, {} и {}'.format(roots[0], roots[1], roots[2]))  
 elif len\_roots == 4:  
 print('Четыре корня: {}, {}, {} и {}'.format(roots[0], roots[1], roots[2], roots[3]))  
  
  
# Если сценарий запущен из командной строки  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

**Экранные формы:**

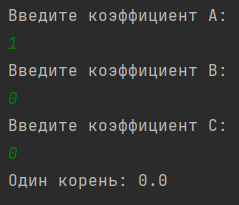
Четыре корня:



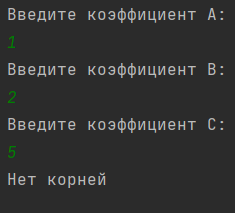
Два корня:



Один корень:



Нет корней:



Некорректный ввод из командной строки:

