**Объектно-ориентированные возможности языка Python.**

Студент: Шебеко Николай

Группа: ИУ5-33Б  
Дата: 15.11.24г.

Разработать программу для решения [биквадратного уравнения.](https://ru.wikipedia.org/wiki/%D0%A3%D1%80%D0%B0%D0%B2%D0%BD%D0%B5%D0%BD%D0%B8%D0%B5_%D1%87%D0%B5%D1%82%D0%B2%D1%91%D1%80%D1%82%D0%BE%D0%B9_%D1%81%D1%82%D0%B5%D0%BF%D0%B5%D0%BD%D0%B8#%D0%91%D0%B8%D0%BA%D0%B2%D0%B0%D0%B4%D1%80%D0%B0%D1%82%D0%BD%D0%BE%D0%B5_%D1%83%D1%80%D0%B0%D0%B2%D0%BD%D0%B5%D0%BD%D0%B8%D0%B5)

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](https://github.com/ugapanyuk/BKIT_2022/blob/main/code/lab1_code) ). Если они не заданы, то вводятся с клавиатуры в соответствии с пунктом 2. [Описание работы с параметрами командной строки.](https://realpython.com/python-command-line-arguments/#the-command-line-interface)
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.
5. Дополнительное задание 1 (\*). Разработайте две программы на языке Python - одну с применением процедурной парадигмы, а другую с применением объектно-ориентированной парадигмы.
6. Дополнительное задание 2 (\*). Разработайте две программы - одну на языке Python, а другую на любом другом языке программирования (кроме С++).

Текст программы:

**main.py:**

import math  
import sys  
  
class BiquadraticEquation:  
 def \_\_init\_\_(self, a, b, c):  
 self.a = a  
 self.b = b  
 self.c = c  
  
 def discriminant(self):  
 return self.b \*\* 2 - 4 \* self.a \* self.c  
  
 def roots(self):  
 d = self.discriminant()  
 if d >= 0:  
 return (-self.b + math.sqrt(d)) / (2 \* self.a), (-self.b - math.sqrt(d)) / (2 \* self.a)  
 else:  
 return None  
  
  
def main():  
 if len(sys.argv) == 4:  
 a, b, c = float(sys.argv[1]), float(sys.argv[2]), float(sys.argv[3])  
 else:  
 while True:  
 try:  
 a = float(input("Введите коэффициент A: "))  
 b = float(input("Введите коэффициент B: "))  
 c = float(input("Введите коэффициент C: "))  
 break  
 except ValueError:  
 print("Некорректный ввод. Введите действительное число.")  
  
 equation = BiquadraticEquation(a, b, c)  
 roots = equation.roots()  
 if roots is not None:  
 print("Корни уравнения:", roots)  
 else:  
 print("Уравнение не имеет вещественных корней.")  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

Снимки экрана:  
Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описаниеИзображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание