Группа: ИУ5-31Б

Студент: Болотин Ю.А.

**Постановка задачи**

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**

import sys  
import math  
  
  
def read\_coef\_from\_input(name):  
 while True:  
 try:  
 coef = float(input(f"Введите коэффициент {name}: "))  
 except ValueError:  
 print("Ошибка. Введите действительное число")  
 else:  
 break  
 return coef  
  
  
def read\_coef(index, name):  
 try:  
 coef = float(sys.argv[index])  
 except IndexError:  
 coef = read\_coef\_from\_input(name)  
 return coef  
  
  
def get\_coefs():  
 a = read\_coef(1, "A")  
 b = read\_coef(2, "B")  
 c = read\_coef(3, "C")  
 return a, b, c  
  
  
def get\_roots(a, b, c):  
 if a == 0:  
 if b == 0:  
 return []  
 else:  
 return [-1 \* c / b, ]  
 result = []  
  
 d = b\*\*2 - 4\*a\*c  
 print(f"Дискриминант: {d}")  
 if d > 0:  
 d\_sqrt = math.sqrt(d)  
 root1 = (-b + d\_sqrt) / (2.0 \* a)  
 root2 = (-b - d\_sqrt) / (2.0 \* a)  
 if root1 > 0:  
 result.append(math.sqrt(root1))  
 result.append(-math.sqrt(root1))  
 elif root1 == 0:  
 result.append(root1)  
 if root2 > 0:  
 result.append(math.sqrt(root2))  
 result.append(-math.sqrt(root2))  
 elif root2 == 0:  
 result.append(math.fabs(root2))  
 elif d == 0:  
 root = -b / (2.0 \* a)  
 if root > 0:  
 result.append(math.sqrt(root))  
 result.append(-math.sqrt(root))  
 elif root == 0:  
 result.append(0)  
  
 return sorted(result)  
  
  
def print\_roots(roots):  
 roots\_number = len(roots)  
 if roots\_number == 0:  
 print("Нет корней")  
 elif roots\_number == 1:  
 print(f"Один корень: {roots[0]}")  
 elif roots\_number == 2:  
 print(f"Два корня: {roots[0]}, {roots[1]}")  
 elif roots\_number == 3:  
 print(f"Три корня: {roots[0]}, {roots[1]}, {roots[2]}")  
 elif roots\_number == 4:  
 print(f"Четыре корня: {roots[0]}, {roots[1]}, {roots[2]}, {roots[3]}")  
  
  
def main():  
 a, b, c = get\_coefs()  
 roots = get\_roots(a, b, c)  
 print\_roots(roots)  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

**Вывод:**

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание