Группа: ИУ5-31Б
Студент: До Тхе Фу

Постановка задачи

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 A, B, C, вычисляет дискриминант и ДЕЙСТВИТЕЛЬНЫЕ корни уравнения (в зависимости от дискриминанта).
3. Коэффициенты A, B, C 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 A, B, C 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 C++).

Текст программы

```
import sys
import math


class QuadraticEquationSolver:
    def __init__(self):
        self.a = None
        self.b = None
```

```
        self.c = None

    def read_coef_from_input(self, 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(self, index, name):
        try:
            coef = float(sys.argv[index])
        except IndexError:
            coef = self.read_coef_from_input(name)
        return coef

    def get_coefs(self):
        self.a = self.read_coef(1, "A")
        self.b = self.read_coef(2, "B")
        self.c = self.read_coef(3, "C")

    def get_roots(self):
        if self.a == 0:
            if self.b == 0:
                return []
            else:
                return [-1 * self.c / self.b, ]

        result = []
        d = self.b ** 2 - 4 * self.a * self.c
        print(f"Дискриминант: {d}")

        if d > 0:
            d_sqrt = math.sqrt(d)
```

```
            root1 = (-self.b + d_sqrt) / (2.0 * self.a)
            root2 = (-self.b - d_sqrt) / (2.0 * self.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 = -self.b / (2.0 * self.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(self, 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 solve(self):
        self.get_coefs()
```

```
        roots = self.get_roots()
        self.print_roots(roots)


if __name__ == "__main__":
    solver = QuadraticEquationSolver()
    solver.solve()
```