Московский государственный технический

университет им. Н.Э. Баумана

Отчёт по лабораторной работе №6
по курсу «Парадигмы и конструкции языков программирования»

Проверил:
преподаватель каф. ИУ5
Гапанюк Ю. Е.

Выполнил:
Студент группы РТ5-31Б
Иванов А. А.

Москва, 2024 г.

Задание:

Цель лабораторной работы: изучение возможностей модульного тестирования в языке Python.

Требования к отчету:

Отчет по лабораторной работе должен содержать:

1. титульный лист;
2. описание задания;
3. текст программы;
4. экранные формы с примерами выполнения программы.

Задание:

1. Выберите любой фрагмент кода из лабораторных работ 1 или 2 или 3-4.
2. Модифицируйте код таким образом, чтобы он был пригоден для модульного тестирования.
3. Разработайте модульные тесты. В модульных тестах необходимо применить следующие технологии:
    - TDD - фреймворк (не менее 3 тестов).
    - BDD - фреймворк (не менее 3 тестов).

Код программы:

```
import math

def get_coeff(name, value=None):
    while True:
        if value is not None:
            try:
                coeff = float(value)
                return coeff
            except ValueError:
                raise ValueError(f"Некорректное значение для коэфф. {name}:
'{value}'. Нужно число.")


def solv(A, B, C):
    if A == 0:
        raise ValueError("Коэфф. A не должен быть = 0.")

    D = B**2 - 4*A*C
    if D < 0:
        return []

    y1 = (-B + math.sqrt(D)) / (2*A)
    y2 = (-B - math.sqrt(D)) / (2*A)
```

```
    roots = set()

    for y in [y1, y2]:
        if y < 0:
            continue
        elif y == 0:
            roots.add(0.0)
        else:
            sqrt_y = math.sqrt(y)
            roots.add(sqrt_y)
            roots.add(-sqrt_y)
    return sorted(roots)

if __name__ == "__main__":
    try:
        A = get_coeff("A", input("Введите коэфф. A: "))
        B = get_coeff("A", input("Введите коэфф. B: "))
        C = get_coeff("A", input("Введите коэфф. C: "))

        roots = solv(A, B, C)
        if not roots:
            print("Нет корней")
        else:
            print("Корни:")
            for root in roots:
                print(root)
    except ValueError as e:
        print(e)
import unittest
from main import get_coeff, solv

class Test(unittest.TestCase):

    def test_1(self):
        with self.assertRaises(ValueError):
            get_coeff("abc", "A")

    def test_2(self):
        self.assertEqual(solv(1, 1, 1), [])

    def test_3(self):
        self.assertEqual(solv(1, 1, -2), [-1.0, 1.0])

    def test_4(self):
        self.assertEqual(solv(1, -10, 9), [-3.0, -1.0, 1.0, 3.0])

if __name__ == "__main__":
    unittest.main()
```

Вывод:

```
....
----------------------------------------------------------------------
Ran 4 tests in 0.001s

OK
```