Московский государственный технический университет им. Н.Э. Баумана

Факультет «Информатика и системы управления»
Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Базовые компоненты интернет-технологий»

Лабораторная работа №4

Выполнил:
студент группы ИУ5-34Б
Ковыршин Павел
Подпись и дата:

Проверил:
Гапанюк Ю.Е.
Подпись и дата:

Москва, 2021 г.

Постановка задачи.

1. Необходимо для произвольной предметной области реализовать от одного до трех шаблонов проектирования: один порождающий, один структурный и один поведенческий. В качестве справочника шаблонов можно использовать следующий каталог. Для сдачи лабораторной работы в минимальном варианте достаточно реализовать один паттерн.
2. Вместо реализации паттерна Вы можете написать тесты для своей программы решения биквадратного уравнения. В этом случае, возможно, Вам потребуется доработать программу решения биквадратного уравнения, чтобы она была пригодна для модульного тестирования.
3. В модульных тестах необходимо применить следующие технологии:
    - TDD - фреймворк.
    - BDD - фреймворк.
    - Создание Mock-объектов.

Текст программы.
prog.py

```python
def is_real(n):
    try:
        result = n == float(n)
        return result
    except ValueError:
        return False

def biquad_roots(a, b = 0, c = 0):
    assert is_real(a)
    assert is_real(b)
    assert is_real(c)
    D = b**2 - 4 * a * c
    qroots = []
    if D < 0 or a == 0:
        return []
    else:
        qroots.append((-b + D**0.5) / (2 * a))
        qroots.append((-b - D**0.5) / (2 * a))
        biqroots = []
        for qroot in qroots:
            if qroot < 0:
                continue
            biqroot = qroot**0.5
            if biqroot not in biqroots:
                biqroots.append(biqroot)
            if -biqroot not in biqroots:
                biqroots.append(-biqroot)
        return sorted(biqroots)

if __name__ == '__main__':
```

```
    print(biquad_roots(1))
```

testTDD.py

```
from prog import biquad_roots
from prog import is_real
import unittest

class biquad_test(unittest.TestCase):
    def test1(self):
        self.assertEqual(biquad_roots(1), [0])
    def test2(self):
        self.assertEqual(biquad_roots(1, -1, 0), [-1, 0, 1])
    def test3(self):
        self.assertEqual(biquad_roots(1, 1, -6), [-2**0.5, 2**0.5])
    def test4(self):
        self.assertEqual(biquad_roots(-1, 4, -3), [-3**0.5, -1, 1, 3**0.5])
    def test5(self):
        self.assertEqual(biquad_roots(2, 0, 1), [])
    def test6(self):
        self.assertEqual(biquad_roots(0, 1, 1), [])

    def test7(self):
        self.assertFalse(is_real('5'))
    def test8(self):
        self.assertFalse(is_real('-4.5'))
    def test9(self):
        self.assertFalse(is_real('route'))
    def test10(self):
        self.assertTrue(is_real(33))
    def test11(self):
        self.assertTrue(is_real(-6.3))

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

steps/stepsBDD.py

```
from behave import given, when, then
from prog import biquad_roots

@given("coefficients {a:g}, {b:g}, {c:g}")
def given_c(context, a, b, c):
    context.a = a
    context.b = b
    context.c = c

@when("the roots get calculated")
def calculation(context):
    context.result = biquad_roots(context.a, context.b, context.c)
```

```python
@then("they are [{result}]")
def get_result(context, result):
    result_list = []
    for value in result.split(', '):
        try:
            result_list.append(float(value))
        except ValueError:
            pass
    assert context.result == result_list
```

featureBDD.feature

```
Feature:
    Calculating the roots of a biquadratic equation
Scenario: first equation
    Given coefficients 1, 0, 0
    When the roots get calculated
    Then they are [0]

Scenario: first equation
    Given coefficients 1, -1, 0
    When the roots get calculated
    Then they are [-1, 0, 1]

Scenario: first equation
    Given coefficients 2, 0, 1
    When the roots get calculated
    Then they are [ ]
```

Результат выполнения программы.

testTDD.py

```
C:\Users\pahan\python_prog\lab-4>python testTDD.py
...........
----------------------------------------------------------------------
Ran 11 tests in 0.002s

OK
```

behave

```
C:\Users\pahan\python_prog\lab-4>behave
Feature:  # featureBDD.feature:1
  Calculating the roots of a biquadratic equation
  Scenario: first equation        # featureBDD.feature:4
    Given coefficients 1, 0, 0    # steps/stepsBDD.py:4
    When the roots get calculated # steps/stepsBDD.py:10
    Then they are [0]             # steps/stepsBDD.py:14

  Scenario: second equation       # featureBDD.feature:9
    Given coefficients 1, -1, 0   # steps/stepsBDD.py:4
    When the roots get calculated # steps/stepsBDD.py:10
    Then they are [-1, 0, 1]      # steps/stepsBDD.py:14

  Scenario: third equation        # featureBDD.feature:14
    Given coefficients 2, 0, 1    # steps/stepsBDD.py:4
    When the roots get calculated # steps/stepsBDD.py:10
    Then they are [ ]             # steps/stepsBDD.py:14

1 feature passed, 0 failed, 0 skipped
3 scenarios passed, 0 failed, 0 skipped
9 steps passed, 0 failed, 0 skipped, 0 undefined
Took 0m0.005s
```