

**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:

студентка группы ИУ5-34Б
Федотова Анастасия

Подпись и дата:

Проверил:

Гапанюк Ю. Е.

Подпись и дата:

Москва, 2021 г.

Постановка задачи

Задание:

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

featureBDD.feature

```
Feature:
    One day a math came

Scenario: first test
    Given coef 1.2, -1, -0.2
    When Something!!!!
    Then Result -1, 1

Scenario: second test
    Given coef 1, 0, -4
    When Something!!!!
    Then Result -1.4142135623730951, 1.4142135623730951

Scenario: third test
    Given coef 2, 0, 1
    When Something!!!!
    Then Result None
```

test.py

```
from main import *
import unittest

class TestOr(unittest.TestCase):
    def setUp(self):
        self.get_coef = get_coef
        self.get_roots = get_roots

    def test_roots(self):
        self.assertTrue(len(self.get_roots(0, 0, 0)) > 4)
        self.assertEqual(self.get_roots(1, 1, 1), [])
        self.assertEqual(self.get_roots(-1, -1, -1), [])
        self.assertEqual(self.get_roots(0, 0, 1), [])
        self.assertEqual(self.get_roots(1, 0, 0), [0.5])
        self.assertEqual(self.get_roots(0, 1, 0), [0.5])
        self.assertEqual(self.get_roots(1, 0, 1), [])
        self.assertEqual(self.get_roots(0, 1, 1), [])
        self.assertEqual(self.get_roots(10, 25, 0), [0.5])

        self.assertEqual(set(self.get_roots(10, -15, 0)), set([-6 ** 0.5 / 2, 0, 6 ** 0.5 / 2]))
        self.assertEqual(set(self.get_roots(0, 1, -16)), set([-4, 4]))
        self.assertEqual(set(self.get_roots(1, 0, -4)), set([-2 ** 0.5, 2 ** 0.5]))
        self.assertEqual(set(self.get_roots(1, -5, -36)), set([-3, 3]))
        self.assertEqual(set(self.get_roots(1, 14, 48)), set([]))
        self.assertEqual(set(self.get_roots(1, 1, -20)), set([-2, 2]))
        self.assertEqual(set(self.get_roots(1, -5, 4)), set([-2, -1, 1, 2]))

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

stepsBDD.py

```
from behave import given, when, then
from main import get_roots

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

@when("Something!!!!")
def calculation(context):
    context.result = get_roots(context.a, context.b, context.c)

@then("Result {result}")
def get_result(context, result):
    if context.result == result: return True
    else: return False
```

main.py

```
import sys
import math

def get_coef(index, prompt):
    try:
        coef_str = sys.argv[index]
    except:
        coef_str = input(prompt)
    coef = float(coef_str)
    return coef

def d_positive(result, a, b, c, D):
    sqD = math.sqrt(D)
    sq1 = (-b - sqD) / (2.0 * a)
    sq2 = (-b + sqD) / (2.0 * a)
    if sq1 >= 0.0:
        root1 = -math.sqrt(sq1)
        root2 = math.sqrt(sq1)
        result.append(root1)
        if root1 != root2: result.append(root2)
    if sq2 >= 0.0:
        root3 = -math.sqrt(sq2)
        root4 = math.sqrt(sq2)
        result.append(root3)
        if root3 != root4: result.append(root4)
    return result
```

```

def d_zero(result, a, b, c):
    root1 = math.sqrt(-b / (2.0 * a))
    root2 = -math.sqrt(-b / (2.0 * a))
    result.append(root1)
    if root1 != root2: result.append(root2)
    return result

def get_roots(a, b, c):
    result = []
    if a == 0.0:
        if b == 0.0:
            if c == 0.0:
                result = [0.0] * 5
                return result
            else: return result
        else:
            sq = -c / b
            if sq >= 0:
                root1 = -math.sqrt(sq)
                root2 = math.sqrt(sq)
                result.append(root1)
                if root1 != root2: result.append(root2)
            else:
                D = b * b - 4 * a * c
                if D == 0.0: result = d_zero(result, a, b, c)
                elif D > 0.0: result = d_positive(result, a, b, c, D)
    return result

```

```

def main():
    a = get_coef(1, 'Введите коэффициент A: ')
    b = get_coef(2, 'Введите коэффициент B: ')
    c = get_coef(3, 'Введите коэффициент C: ')
    roots = get_roots(a, b, c)
    if not roots: roots = []
    len_roots = len(roots)
    if len_roots > 1: roots.sort()
    if len_roots == 0:
        print('Нет корней')
    elif len_roots == 1:
        print('Один корень: {:.3f}'.format(roots[0]))
    elif len_roots == 2:
        print('Два корня: {:.3f} и {:.3f}'.format(roots[0], roots[1]))
    elif len_roots == 3:
        print('Четыре корня: {:.3f}, {:.3f} и {:.3f}'.format(roots[0], roots[1], roots[2]))
    elif len_roots == 4:
        print('Четыре корня: {:.3f}, {:.3f}, {:.3f} и {:.3f}'.format(roots[0], roots[1], roots[2], roots[3]))
    elif len_roots == 5:
        print('Бесконечное кол-во корней')
    else:
        print('Unexpected result')

if __name__ == "__main__":
    main()

```

Результат выполнения тестов

```
test x
"/Users/mac/Desktop/БКИТ Федотова/lab_4/bin/python" "/Users/mac/Desktop/БКИТ Федотова/lab_4/test.py"
.
-----
Ran 1 test in 0.000s

OK

Process finished with exit code 0
```

```
(lab_4) mac@MacBook-Pro-mac lab_4 % behave
Feature: # featureBDD.feature:1
  One day a math came
  Scenario: first test # featureBDD.feature:4
    Given coef 1.2, -1, -0.2 # steps/stepsBDD.py:4 0.000s
    When Something!!!! # steps/stepsBDD.py:10 0.000s
    Then Result -1, 1 # steps/stepsBDD.py:14 0.000s

  Scenario: second test # featureBDD.feature:9
    Given coef 1, 0, -4 # steps/stepsBDD.py:4 0.000s
    When Something!!!! # steps/stepsBDD.py:10 0.000s
    Then Result -1.4142135623730951, 1.4142135623730951 # steps/stepsBDD.py:14 0.001s

  Scenario: third test # featureBDD.feature:14
    Given coef 2, 0, 1 # steps/stepsBDD.py:4 0.000s
    When Something!!!! # steps/stepsBDD.py:10 0.000s
    Then Result None # steps/stepsBDD.py:14 0.000s

1 feature passed, 0 failed, 0 skipped
3 scenarios passed, 0 failed, 0 skipped
9 steps passed, 0 failed, 0 skipped, 0 undefined
Took 0m0.003s
```

Version Control TODO Problems Terminal Python Packages Python Console