

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего
образования «Московский государственный технический университет имени
Н.Э. Баумана (национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

**Отчет по лабораторной работе № 4 по курсу
Базовые компоненты интернет-технологий
«Шаблоны проектирования и модульное тестирование в
Python»**

Выполнил:
студент группы ИУ5-34Б
Братухин Александр
Подпись и дата:
13.12.21

Проверил:

Подпись и дата:

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.

Текст программы

Файл main.py

```
from math import *
def is_real_number(tmp):
    try:
        out = float(tmp)
        if tmp == 0:
            return float(0)
        return out
    except:
        return False

def reshen_bqr(a, b=0, c=0):
    assert is_real_number(a)
    if b != 0:
        assert is_real_number(b)
    if c != 0:
        assert is_real_number(c)
    Discr = b**2 - 4 * a * c
    roots_t = []
    if Discr < 0 or a == 0:
        return []
    else:
        roots_t.append((-b + Discr**0.5) / (2 * a))
        roots_t.append((-b - Discr**0.5) / (2 * a))
        bi_roots = []
        for p in roots_t:
            if p < 0:
                continue
            tmp = pow(p, 0.5)
            if tmp not in bi_roots:
                bi_roots.append(tmp)
            if -tmp not in bi_roots:
                bi_roots.append(-tmp)
        bi_roots = sorted(bi_roots)
        return bi_roots

if __name__ == '__main__':
    t = reshen_bqr(1, -5, 6)
    print(t)
# Пример запуска
# qr.py 1 0 -4
```

Файл TDD.py

```
import unittest
from main import reshen_bqr, is_real_number

class b_test(unittest.TestCase):
    def test1(self):
        self.assertEqual(reshen_bqr(1, 0, -4), [-1.4142135623730951,
1.4142135623730951])
    def test2(self):
        self.assertEqual(reshen_bqr(2, 0, 1), [])
    def test3(self):
        self.assertEqual(reshen_bqr(123, 42, -4), [-0.27857139469058545,
0.27857139469058545])
    def test4(self):
        self.assertEqual(reshen_bqr(1, -5, 6), [-1.7320508075688772, -
1.4142135623730951, 1.4142135623730951, 1.7320508075688772])
    def test5(self):
        self.assertEqual(reshen_bqr(1), [0])
    def test6(self):
        self.assertTrue(is_real_number(5))
    def test6(self):
```

```

        self.assertTrue(is_real_number(2.231243412))

    def test7(self):
        self.assertFalse(is_real_number('2.a231243412'))

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

```

Файл stepsBDD.py

```

from behave import given, when, then
from main import reshen_bqr

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

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

@then("He summoned {result}")
def get_result(context, result):
    context.result = result

```

Файл featureBDD.feature

```

from behave import given, when, then
from main import reshen_bqr

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

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

@then("He summoned {result}")
def get_result(context, result):
    assert context.result == result

```

Результаты тестов

```
PS D:\Лабы и прога 3 сем\lab_4> behave
Feature: # featureBDD.feature:1
  One day a wizard came
  Scenario: first spell # featureBDD.feature:4
    Given spell 1, 0, 0 # steps/stepsBDD.py:4
    When Magick!!!! # steps/stepsBDD.py:10
    Then He summoned 0 # steps/stepsBDD.py:14

  Scenario: the second spell # featureBDD.feature:9
    Given spell 1, -2, -8 # steps/stepsBDD.py:4
    When Magick!!!! # steps/stepsBDD.py:10
    Then He summoned -2, 2 # steps/stepsBDD.py:14

  Scenario: the third spell # featureBDD.feature:14
    Given spell 2, 0, 1 # steps/stepsBDD.py:4
    When Magick!!!! # steps/stepsBDD.py:10
    Then He summoned None # 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 0m0.003s
PS D:\Лабы и прога 3 сем\lab_4>
```

Ran 7 tests in 0.003s

OK

Launching unittests with arguments python -m unittest D:/Лабы и прога 3 сем/lab_4/TDD.py in D:\Лабы и прога 3 сем\lab_4

Process finished with exit code 0