



**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)**

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

Лабораторная работа №4
по дисциплине «Базовые компоненты интернет-технологий»

Выполнил:
студент группы ИУ5-32Б
Балабанова А.О.

Проверил:
Канев А.И.

2021 г.
Описание задания

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:
 - o TDD - фреймворк.
 - o BDD - фреймворк.
 - o Создание Mock-объектов.

Цель лабораторной работы: изучение реализации шаблонов проектирования и возможностей модульного тестирования в языке Python.

Test_TDD

```
import unittest
import sys
import math
from main import get_roots

s1 = set([0])
s2 = set([-math.sqrt(2), math.sqrt(2)])
s3 = set([0, 2, -2])
class TestGetRoots(unittest.TestCase):
    def test_area(self):
        self.assertEqual(set(get_roots(1, 0, 0)), s1)
    def test_area(self):
        self.assertEqual(set(get_roots(1, 0, -4)), s2)
    def test_area(self):
        self.assertEqual(set(get_roots(1, -4, 0)), s3)
```

Результат

```
C:\Users\Acer\lab4\venv\Scripts\python.exe "C:\Program Files\JetBrains\PyCharm 2021.2.1\plugins\python\helpers\pycharm\_jb_unittest_runner.py" --path C:/Users/Acer/lab4/test_TDD
Testing started at 16:04 ...

Ran 1 test in 0.002s

OK
Launching unittests with arguments python -m unittest C:/Users/Acer/lab4/test_TDD.py in C:\Users\Acer\lab4
```

Test_BDD

```
from behave import *
from test_TDD import *

@given("I have A = {number1:g}, B = {number2:g}, C = {number3:g}")
def have_numbers(step, number1, number2, number3):
    step.context.number1 = number1
    step.context.number2 = number2
```

```

        step.context.number3 = number3

@when("I solve the biquadrate equation")
def solution(step):
    step.context.result = get_roots(step.context.number1,
    step.context.number2, step.context.number3)

@then("I expect to get {result:g}")
def expect_result(step, result):
    assert step.context.result == result

```

Feature

```

Feature: Test
  Scenario: Test my function
    Given I have A = 1, B = 0, C = 0
    When I solve the biquadrate equation
    Then I expect to get 0

```

Test_Mock

```

import unittest
from main import get_roots
from unittest.mock import patch, Mock

s1 = set([0])
s3 = set([0, 2, -2])
class TestGetRoots(unittest.TestCase):
    @patch('main.get_roots', return_value = [0])
    def test(self, s1):
        self.assertEqual(set(get_roots(1, 0, 0)), set([0]))

    @patch('main.get_roots', return_value=[0, 2, -2])
    def test(self, s3):
        self.assertEqual(set(get_roots(1, -4, 0)), set([0, 2, -2]))

```

Результат

```

✓ Tests passed: 1 of 1 test - 0.0ms
C:\Users\Acer\lab4\venv\Scripts\python.exe "C:\Program Files\JetBrains\PyCharm 2021.2.1\plugins\python\helpers\pycharm\jb_unittest_runner.py" --path C:/Users/Acer/lab4/test_Mo
Testing started at 16:08 ...
Launching unittests with arguments python -m unittest C:/Users/Acer/lab4/test_Mock.py in C:\Users\Acer\lab4

Ran 1 test in 0.005s

OK

Process finished with exit code 0

```