

**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»

Отчет по лабораторной работе №5
«Модульное тестирование в Python»

Выполнил:
студент группы ИУ5-34Б

Новиков Богдан

Проверил:
преподаватель каф.
ИУ5

Гапанюк Ю. Е.

2022 г.

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).

Текст программы.

Tdd:

test_equations.py

```
import pytest, sys
from lab1 import get_roots, get_coef

def test_get_roots():
    assert get_roots(1, -10, 9) == [-3.0, -1.0, 1.0, 3.0]
    assert get_roots(1, -2, 1) == [-1.0, 1.0]
    assert get_roots(1, 2, 3) == []
```

```
def test_get_coef():
    assert get_coef(1, "Write number 1") == 1
```

```
def test_type():
    try:
        get_roots(12, "B", 4)
        assert False
    except TypeError:
        assert True
```

bdd:

test.py

```
from behave import Given, When, Then
from lab1 import get_roots

@Given("equation with coef A {A} B {B} C {C}")
def given_increment(context, A: str, B: str, C: str):
```

```
context.A = int(A)
context.B = int(B)
context.C = int(C)
```

```
@When("we calculate roots")
def given_increment(context):
    context.results = get_roots(context.A, context.B, context.C)

@Then("we should see root1 {root1} root2 {root2} root3 {root3} root4 {root4}")
def then_results(context, root1: str, root2: str, root3: str, root4: str):
    if root1 == "empty":
        assert context.results == []
    elif root2 == "empty":
        assert context.results == [float(root1)]
    elif root3 == "empty":
        assert context.results == [float(root1), float(root2)]
    elif root4 == "empty":
        assert context.results == [float(root1), float(root2), float(root3)]
    else:
        assert context.results == [float(root1), float(root2), float(root3), float(root4)]
```

test.feature

Feature: Testing the Equation

Scenario: Test calculate 4 roots

Given equation with coef A 1 B -10 C 9

When we calculate roots

Then we should see root1 -3.0 root2 -1.0 root3 1.0 root4 3.0

Scenario: Test calculate 2 roots

Given equation with coef A 1 B -2 C 1

When we calculate roots

Then we should see root1 -1.0 root2 1.0 root3 empty root4 empty

Scenario: Test calculate 0 roots

Given equation with coef A 1 B 2 C 3

When we calculate roots

Then we should see root1 empty root2 empty root3 empty root4 empty behave

Результаты выполнения программы:

```
+ lab5 git:(lab5) x python -m pytest test -s
===== test session starts =====
platform linux -- Python 3.10.7, pytest-7.2.0, pluggy-1.0.0
rootdir: /home/bogdan/Projects/bkit/lab5
plugins: anyio-3.6.1
collected 3 items

test/test_equations.py .Write number 1
1
..
===== 3 passed in 3.76s =====
```

```
+ lab5 git:(lab5) x workon bddtest
(bddtest) + lab5 git:(lab5) x behave
Feature: Testing the Equation # features/steps/test.feature:1

Scenario: Test calculate 4 roots # features/steps/test.feature:2
  Given equation with coef A 1 B -10 C 9 # features/steps/test.py:5 0.000s
  When we calculate roots # features/steps/test.py:12 0.000s
  Then we should see root1 -3.0 root2 -1.0 root3 1.0 root4 3.0 # features/steps/test.py:17 0.000s

Scenario: Test calculate 2 roots # features/steps/test.feature:7
  Given equation with coef A 1 B -2 C 1 # features/steps/test.py:5 0.000s
  When we calculate roots # features/steps/test.py:12 0.000s
  Then we should see root1 -1.0 root2 1.0 root3 empty root4 empty # features/steps/test.py:17 0.000s

Scenario: Test calculate 0 roots # features/steps/test.feature:12
  Given equation with coef A 1 B 2 C 3 # features/steps/test.py:5 0.000s
  When we calculate roots # features/steps/test.py:12 0.000s
  Then we should see root1 empty root2 empty root3 empty root4 empty # features/steps/test.py:17 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 0m0.001s
```