

**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»**

Выполнила:  
студентка группы ИУ5-31Б  
Гапеева О.Р.

Проверил:  
Гапанюк Е.Ю.

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).
  - Создание Моск-объектов (необязательное дополнительное задание).

## Текст программы

Файлы:

```
test.py
features/
    tests2.feature
    steps/
        test.py
```

```
test.py
import unittest
from lab1 import get_roots

class SquareEqSolverTestCase(unittest.TestCase):
    def test_no_root(self):
        res = get_roots(1, 11, 10)
        self.assertEqual(len(res), 0)

    def test_single_root(self):
        res = get_roots(10, 0, 0)
        self.assertEqual(len(res), 1)
        self.assertEqual(res, [0])

    def test_two_roots(self):
        res = get_roots(1, -2, -8)
        self.assertEqual(len(res), 2)
        self.assertEqual(res, [-2, 2])

    def test_three_roots(self):
        res = get_roots(-4, 16, 0)
        self.assertEqual(len(res), 3)
        self.assertEqual(res, [-2, 0, 2])

    def test_four_roots(self):
        res = get_roots(1, -10, 9)
        self.assertEqual(len(res), 4)
```

```
self.assertEqual(res, [-3, -1, 1, 3])
```

tests2.feature

Feature: testing roots

Scenario Outline: multiple roots roots

Given <a> coef a, coef b is <b> and c is <c>

When starting function

Then we should see <result>

Examples:

a	b	c	result
1	10	11	"[]"
10	0	0	"[0]"
1	-2	-8	"[-2, 2]"
4	16	0	"[-2, 0, 2]"
1	-10	9	"[-3, 1, 1, 3]"

test.py

```
from behave import *
```

```
import sys
```

```
sys.path.append("../..")
```

```
from lab1 import get_roots
```

```
@given(u'{given_a} coef a, coef b is {given_b} and c is {given_c}')
```

```
def step_impl(context, given_a, given_b, given_c):
```

```
    global a
```

```
    global b
```

```
    global c
```

```
    a = int(given_a)
```

```
    b = int(given_b)
```

```
    c = int(given_c)
```

```
    return True
```

```
@When("starting function")
```

```
def step_impl(context):
```

```
    global result
```

```
    result = get_roots(a, b, c)
```

```
    return True
```

```
@Then("we should see {given_result}")
```

```
def step_impl(context, given_result):
```

```
    try:
```

```
        assert(result == given_result)
```

```
        return True
```

```
    except:
```

```
        return False
```

# Экранные формы с примерами выполнения программы

test.py (TDD)

```
(env) PS C:\Users\olgap\Documents\BKIT\lab5> python -m unittest
64
.81
.0
.256
.36
.
```

-----  
Ran 5 tests in 0.002s

OK

test.py (BDD)

```
(env) PS C:\Users\olgap\Documents\BKIT\lab5\features\steps> python -m behave
Feature: testing roots # ../tests2.feature:1
```

```
Scenario Outline: multiple roots roots -- @1.1 # ../tests2.feature:10
  Given 1 coef a, coef b is 10 and c is 11 # test.py:8
  When starting function # test.py:19
  Then we should see "[]" # test.py:26
```

```
Scenario Outline: multiple roots roots -- @1.2 # ../tests2.feature:11
  Given 10 coef a, coef b is 0 and c is 0 # test.py:8
  When starting function # test.py:19
  Then we should see "[0]" # test.py:26
```

```
Scenario Outline: multiple roots roots -- @1.3 # ../tests2.feature:12
  Given 1 coef a, coef b is -2 and c is -8 # test.py:8
  When starting function # test.py:19
  Then we should see "[-2, 2]" # test.py:26
```

```
Scenario Outline: multiple roots roots -- @1.4 # ../tests2.feature:13
  Given 4 coef a, coef b is 16 and c is 0 # test.py:8
  When starting function # test.py:19
  Then we should see "[-2, 0, 2]" # test.py:26
```

```
Scenario Outline: multiple roots roots -- @1.5 # ../tests2.feature:14
  Given 1 coef a, coef b is -10 and c is 9 # test.py:8
  When starting function # test.py:19
  Then we should see "[-3, 1, 1, 3]" # test.py:26
```

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

---