

**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  
Котова А.А.

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 Mock-объектов (необязательное дополнительное задание).

## Текст программы

```
import sys
import cmath
# Получение коэффициентов от пользователя
def get_coef(index, str):
    # попытка считать коэф. из командной строки
    try:
        coef_str = sys.argv[index]
        # ввод коэф. с клавиатуры
    except:
        print(str)
        coef_str = input()
    while True:
        try:
            # перевод строки в действительное число
            float(coef_str)
            break
        except:
            coef_str = input()
    coef = float(coef_str)
    return coef
```

```
def get_roots(a,b,c):
    answr = []
    #расчет дискриминанта
    D = b*b - 4*(a*c)
    #расчет корней
    if D == 0:
        root = -b/(2.0*a)
        answr.append(root)
    elif D > 0.0:
        sqD = cmath.sqrt(D)
        root1 = (-b + sqD)/(2.0*a)
        root2 = (-b - sqD)/(2.0*a)
        answr.append(root1)
        answr.append(root2)
    return answr
```

```
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)
    len_roots = len(roots)
```

```
#Вывод корней
if len_roots == 0:
    print('Нет Корней.')
if len_roots == 1:
    print('Единственный корень уравнения: {}'.format(roots[0]))
if len_roots == 2:
    print('Два корня уравнения: {} и {}'.format(roots[0], roots[1]))
#для запуска кода из командной строки
if __name__ == "__main__":
    main()
```

### Файл 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])
```

### Файл features/ 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]"

## Результаты:

```
When starting function                                # steps/test.py:19
Then we should see "[-2, 0, 2]"                       # steps/test.py:27

Scenario Outline: multiple roots roots -- @1.5        # tests2.feature:14
  Given 1 coef a, coef b is -10 and c is 9            # steps/test.py:8
  When starting function                               # steps/test.py:19
  Then we should see "[-3, 1, 1, 3]"                  # steps/test.py:27

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

