

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

Курс «Базовые компоненты интернет-технологий»  
Лабораторная работа №5 на тему:  
«Модульное тестирование в Python»

Выполнил:  
студент группы ИУ5-34Б:  
Даниелян Алла Армановна  
Подпись и дата:

Проверил:  
преподаватель каф. ИУ5  
Гапанюк Ю.В.  
Подпись и дата:

Москва, 2022 г.

### Постановка задачи:

1. Выберите любой фрагмент кода из лабораторных работ 1 или 2 или 3-4. (выбрали лабораторную работу №1)
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:

#### Тестируемый файл FUNCS.py

```
import math
import sys
def get_coef(index, prompt):
    ...
    Читаем коэффициент из командной строки или вводим с клавиатуры
    Args:
        index (int): Номер параметра в командной строке
        prompt (str): Приглашение для ввода коэффициента
    Returns:
        float: Коэффициент квадратного уравнения
    ...
    flag = 1
    while flag==1:
        try:
            # Пробуем прочитать коэффициент из командной строки
            coef_str = sys.argv[index]
        except:
            # Вводим с клавиатуры
            print(prompt)
            coef_str = input()
            # Переводим строку в действительное число
        try:
            coef = float(coef_str)
            flag = 0
        except:
            print ('Try again, dummy')
            flag = 1
    return coef
def get_sqr_roots(a, b, c):
    ...
    Вычисление корней квадратного уравнения
    Args:
        a (float): коэффициент A
        b (float): коэффициент B
        c (float): коэффициент C
    Returns:
```

```

        list[float]: Список корней
    ...

result = []
if a==0 and b == 0 and c == 0:
    result = [0]*5
    return result
if a==0 and b == 0:
    return result
if a ==0 and c == 0:
    result = [0]
    return result
D = b*b - 4*a*c
if D == 0.0:
    root = -b / (2.0*a)
    result.append(root)
elif D > 0.0:
    sqD = math.sqrt(D)
    root1 = (-b + sqD) / (2.0*a)
    root2 = (-b - sqD) / (2.0*a)
    result.append(root1)
    result.append(root2)
return result

def get_bisqr_roots (a,b,c):
    try:
        root = get_sqr_roots (a,b,c)
    except:
        root = get_sqr_roots(b,0.0,c)
        return root
        #если сделать ретурн здесь, то выйдет ли он из всей функции или только
из исключения??
        #судя по всему, из всей функции хех
l = len(root)
bi_root =[]
for i in range (0,l) :
    if root[i]>0:
        bi_root.append(math.sqrt(root[i]))
        bi_root.append(-math.sqrt(root[i]))
    elif root[i] ==0:
        bi_root.append (0)
return bi_root

def print_roots (root):
l = len(root)
if l == 5:
    print ("Infinity of roots")
elif l ==0:
    print ("No roots")
elif l == 1:
    print ('One root: {}'.format(root[0]))
elif l == 2:
    print ('Two roots: {} and {}'.format(root[0],root[1]))

```

```

elif l == 3:
    print ('Three roots: {}, {} and {}'.format(root[0],root[1],root[2]))
else:
    print ('Four roots: {}, {}, {} and {}'.format(root[0],root[1],root[2],root[3]))
return

```

## Файл first.feature

Feature: Kakoy-to tam feature blin

### Scenario Outline: Four Bisquare roots

Given programm is run

When the user inputs <a>, <b>, <c>

Then result is <r1>, <r2>, <r3>, <r4>

Examples:

a	b	c	r1	r2	r3	r4
1	-5	4	2	-2	1	-1

### Scenario Outline: Two Bisquare roots

Given programm is run

When the user inputs <a>, <b>, <c>

Then result is <r1>, <r2>

Examples:

a	b	c	r1	r2
1	-5	-36	3	-3

### Scenario Outline: One Bisquare Root

Given programm is run

When the user inputs <a>, <b>, <c>

Then result is <r1>

Examples:

a	b	c	r1
0	1	0	0

## Файл test\_bdd.py

```

from pytest_bdd import given, when, then, scenarios, parsers
from my_module.FUNCS import get_bisqr_roots
#Можно и без пункта given, но мне так больше нравится, может
scenarios ("first.feature")
@given("programm is run")
def programm_is_run ():
    pass
@when(parsers.parse("the user inputs {a}, {b}, {c}"), target_fixture = 'coef')
def set_coef (a,b,c):
    l = [a,b,c]
    return l
@then(parsers.parse( "result is {r1}, {r2}, {r3}, {r4}"))
def results(coef, r1, r2, r3, r4):
    coef = list(float (x) for x in coef)
    res = [float(x) for x in (r1,r2,r3,r4)]
    assert get_bisqr_roots (coef[0], coef[1], coef[2]) == res
@then(parsers.parse( "result is {r1}, {r2}"))

```

```
def results(coef, r1, r2):
    coef = list(float (x) for x in coef)
    res = [float(x) for x in (r1,r2)]
    assert get_bisqr_roots (coef[0], coef[1], coef[2]) == res
@then(parsers.parse( "result is {r1}"))
def results(coef, r1):
    coef = list(float (x) for x in coef)
    res = [ float(r1)]
    assert get_bisqr_roots (coef[0], coef[1], coef[2]) == res
```

### Файл test\_tdd.py

```
import unittest
from my_module.FUNCS import get_sqr_roots
#Чтобы запустить тест, напиши в терминале "python -m unittest test_tdd.py"
```

```
class TestLab1_get_sqr_root (unittest.TestCase):
    def test_one_root (self):
        self.assertEqual (get_sqr_roots(0,1,0), [0.0])
    def test_zero_root(self):
        self.assertEqual (get_sqr_roots(1,0,1), [])
    def test_two_roots (self):
        self.assertEqual(get_sqr_roots(1,2,-3), [1.0, -3.0])
```

### Результаты выполнения программы:

#### BDD тест:

```
(env1) PS C:\Users\allad\OneDrive\Документы\lab1_py\env1\steps>
(env1) PS C:\Users\allad\OneDrive\Документы\lab1_py\env1\steps> pytest .\test_bdd.py
===== test session starts =====
platform win32 -- Python 3.10.7, pytest-7.2.0, pluggy-1.0.0
rootdir: C:\Users\allad\OneDrive\Документы\lab1_py\env1\steps
plugins: bdd-6.1.1
collected 3 items

test_bdd.py ... [100%]

===== 3 passed in 0.08s =====
(env1) PS C:\Users\allad\OneDrive\Документы\lab1 py\env1\steps> █
```

#### TDD тест:

```
(env1) PS C:\Users\allad\OneDrive\Документы\lab1_py\env1\steps> python -m unittest .\test_tdd.py
...
-----
Ran 3 tests in 0.001s

OK
(env1) PS C:\Users\allad\OneDrive\Документы\lab1_py\env1\steps> █
```