Московский государственный технический университет имени Н.Э.Баумана

 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Факультет РТ Радиотехнический

Кафедра ИУ5 Системы обработки информации и управления

**Отчет по лабораторной работе № 4 по курсу**

**Базовые компоненты интернет-технологий**

#### " Шаблоны проектирования и модульное тестирование в Python."

|  |  |  |
| --- | --- | --- |
| Исполнитель |  |  |
| студент группы РТ5-31б | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Титов А.Д. |
|  |  | “\_\_\_” \_\_\_\_\_\_\_\_\_\_ 2021 г. |
|  |  |  |
| Проверил |  |  |
| Доцент кафедры ИУ5 | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Гапанюк Ю.Е. |
|  |  | “\_\_\_” \_\_\_\_\_\_\_\_\_\_ 2021 г. |

Москва -  2021

**Описание задания**

Цель лабораторной работы: изучение реализации шаблонов проектирования и возможностей модульного тестирования в языке Python. В модульных тестах необходимо применить следующие технологии:

* TDD - фреймворк.
* BDD - фреймворк.
* Создание Mock-объектов.

**Текст программы**

**TDD-фреймворк**

import sys

sys.path.insert(0, "C:/Users/Andrew/Desktop/BKIT\_UNIVERSITY/lab1")

import unittest

from qr import get\_roots

class TestGetRoots(unittest.TestCase):

def testGetRoots(self):

self.assertEqual(get\_roots(1, -10, 21), [7, 3])

self.assertEqual(get\_roots(1, 2, 1), [-1])

if \_\_name\_\_ == "\_\_main\_\_":

unittest.main()

**BDD-фреймворк**

Feature: testing the function get\_roots\_b

Scenario: get roots of BQ

Given I put roots [0, 4] into the function

Then I get roots [0, 2, -2]

Scenario: get roots of BQ

Given I put roots [9, 4] into the function

Then I get roots [3, -3, 2, -2]

import sys

sys.path.insert(0, "C:/Users/Andrew/Desktop/BKIT\_UNIVERSITY/lab1")

from behave import \*

from qr import get\_roots\_b

@given('I put roots {roots} into the function')

def step\_impl(context, roots: str):

roots = list(map(

int, roots.replace("[", "")

.replace("]", "")

.split(", ")))

context.result = get\_roots\_b(roots)

@then('I get roots {result}')

def step\_impl(context, result: str):

result = list(map(

float, result.replace("[", "")

.replace("]", "")

.split(", ")))

assert context.result == result

**Mock-объекты**

import sys

sys.path.insert(0, "C:/Users/Andrew/Desktop/BKIT\_UNIVERSITY/lab1")

import unittest

from unittest.mock import Mock

from qr import get\_roots

class TestQr(unittest.TestCase):

def test\_get\_roots(self):

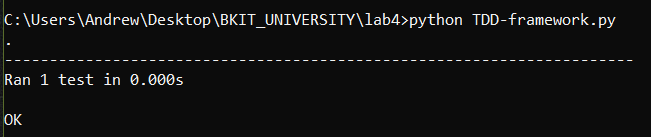
mockRoot = Mock(return\_value = 412)

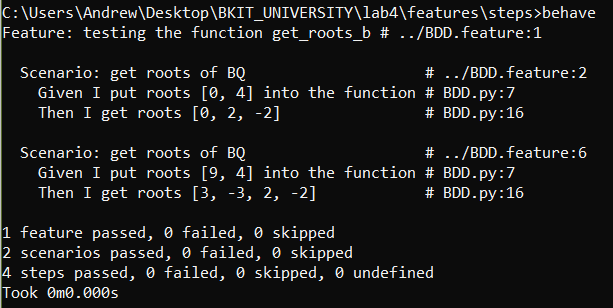
get\_roots(mockRoot(), 2, 1)

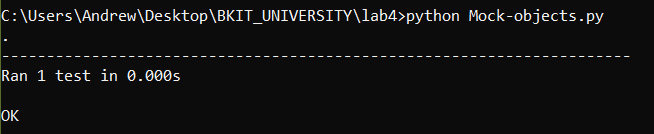
if \_\_name\_\_ == "\_\_main\_\_":

unittest.main()

**Примеры выполнения программы**

****

****

****