МГТУ им. Н.Э. Баумана

# Отчёт по лабораторной работе N5
# по курсу «Базовые компоненты и интернет-технологии»

Преподаватель
Гапанюк Ю.Е.
29.10.2022

Студент группы ИУ5-34Б
Лавренов М.А.
29.10.2022

2022 г.

**Полученное 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:**

Файл gen_random.feature:

```
Feature: Func create array with random counts
  Scenario: New array will contain random numbers from users interval
    Given Having numbers 20, 2, 4
    When Array got created with func gen_random
    Then Result should be an array with random numbers which set will be
[2,3,4] (not 100% chance)
```

Файл main.py:

```
import random


def gen_random(amount, start, stop):
    res = []
    for i in range(0,amount):
        res.append(random.randrange(start, stop+1))
    return(res)


def squared_cont(start,stop):
    mas = []
    for i in range(start,stop+1):
        mas.append(i+i**2)
    return mas


def sort(data):
    res = sorted(data, key=abs, reverse=False)
    return res
```

Файл sort.feature:

```
Feature: Func is sorting array
  Scenario: Array will be sorted by abs
    Given Some array
    When Array get sorted with sort
    Then Array is sorted
```

Файл squared_cont.feature:

```
Feature: Func create array with counts for rule: i + i**2
  Scenario: New array will contain modified numbers from users interval
    Given Having interval 2-4
    When Array got created with func squared_cont
    Then Result should be an array with numbers (i + i**2) for each i in given
interval
```

Файл TDD.py:

```
import unittest
import sys,os

sys.path.append(os.getcwd()) #current working directory
from main import *
```

```
class TestGenRandom(unittest.TestCase):
    def test_gen_random1(self):
        self.assertEqual(list(set(gen_random(20,2,4))),[2,3,4])

class TestSquaredCont(unittest.TestCase):
    def test_squared_cont(self):
        self.assertEqual(squared_cont(2,4),[6,12,20])


class TestSort(unittest.TestCase):
    def test_sort(self):
        self.assertEqual(sort([1,2,56,-20]),[1,2,-20,56])



if __name__=='__main__':
    unittest.main()
```

Файл test_gen_random.py:

```
from pytest_bdd import scenario, given, when, then

from main import gen_random


@scenario("C:/Users/1/PycharmProjects/BKIT5/gen_random.feature",'New array
will contain random numbers from users interval')
def testing_gen_random():
    pass

@given('Having numbers 20, 2, 4',target_fixture='parameters')
def parameters():
    return 20,2,4

@when('Array got created with func gen_random',target_fixture='resarray')
def resarray():
    return list(set(gen_random(20,2,4)))

@then('Result should be an array with random numbers which set will be [2,3,4]
(not 100% chance)')
def resarray(resarray):
    assert resarray == [2,3,4]
```

Файл test_sort.py:

```
from pytest_bdd import scenario, given, when, then

from main import sort


@scenario("C:/Users/1/PycharmProjects/BKIT5/sort.feature",'Array will be
sorted by abs')
def testing_sort():
    pass

@given('Some array',target_fixture='array')
def array():
    return [2,1,-20,56]

@when('Array get sorted with sort',target_fixture='sorting')
def sorting(array):
```

```
    return sort(array)

@then('Array is sorted')
def sorting(sorting):
    assert sorting == [1,2,-20,56]
```

Файл test_squared_cont.py:

```
from pytest_bdd import scenario, given, when, then

from main import squared_cont


@scenario("C:/Users/1/PycharmProjects/BKIT5/squared_cont.feature",'New array
will contain modified numbers from users interval')
def testing_squared_cont():
    pass

@given('Having interval 2-4',target_fixture='parameters')
def parameters():
    return 2,4

@when('Array got created with func squared_cont',target_fixture='resarray')
def resarray():
    return squared_cont(2,4)

@then('Result should be an array with numbers (i + i**2) for each i in given
interval')
def resarray(resarray):
    assert resarray == [6,12,20]
```

Результаты выполнения:

TDD:

```
============================= test session starts ==============================
collecting ... collected 3 items

TDD.py::TestGenRandom::test_gen_random1 PASSED                           [ 33%]
TDD.py::TestSquaredCont::test_squared_cont PASSED                        [ 66%]
TDD.py::TestSort::test_sort PASSED                                       [100%]

============================== 3 passed in 0.01s ===============================

Process finished with exit code 0
```

BDD:

```
============================= test session starts =============================
collecting ... collected 1 item

test_squared_cont.py::testing_squared_cont <- venv\lib\site-packages\pytest_bdd\scenario.py PASSED [100%]

============================== 1 passed in 0.02s ==============================

Process finished with exit code 0
```

```
Launching pytest with arguments C:/Users/1/PycharmProjects/BKIT5/test_sort.py --no-header --no-su

============================= test session starts =============================
collecting ... collected 1 item

test_sort.py::testing_sort <- venv\lib\site-packages\pytest_bdd\scenario.py PASSED [100%]

============================== 1 passed in 0.02s ==============================
```

```
============================= test session starts =============================
collecting ... collected 1 item

test_gen_random.py::testing_gen_random <- venv\lib\site-packages\pytest_bdd\scenario.py PASSED [100%]

============================== 1 passed in 0.02s ==============================
```