

**Федеральное государственное бюджетное образовательное
учреждение высшего образования «Московский государственный
технический университет имени Н.Э.Баумана
(национальный исследовательский университет)»**



**Факультет «Информатика и системы управления»
Курс «Базовые компоненты интернет-технологий»**

Лабораторная работа №5 «Модульное тестирование в Python»

Выполнил:

студент группы ИУ5-33Б Николай Горкунов

подпись: _____, дата: _____

Проверил:

преподаватель Юрий Гапанюк

подпись: _____, дата: _____

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).

Исходный код.

Тестируемая программа tdd методологией:

```
import json
import sys
from process_data_to_test.print_result import print_result
import random

@print_result
def f1(arg):
    return list(map(lambda x: x[1], sorted(list(map(lambda y:
(y.casefold(), y), list(map(lambda x: x.get('job-name'),
arg)))))))

@print_result
def f2(arg):
    return list(filter(lambda x: x.casefold().find('программист')
== 0, arg))

@print_result
def f3(arg):
    return list(map(lambda z: z + ' с опытом Python', arg))

@print_result
def f4(arg):
    return list(zip(arg, list(map(lambda w: 'зарплата {}
руб.'.format(random.randint(100000, 200000)), arg))))
```

Тестируемая программа bdd методологией:

```
import random

def gen_random(num_count, begin, end):
    for i in range(num_count):
        yield random.randint(begin, end)

def main():
    rndGen = gen_random(5, 1, 3)
    for i in rndGen:
        print(i, end = " ")
    print()

if __name__ == "__main__":
    main()
```

Код tdd тестирования (pytest модуль):

```
import json
import pytest
import re
import sys
from process_data_to_test.process_data import f1, f2, f3, f4

f1_to_f2 = ["преподаватель языка C", "программист на языке A",
            "Программист на языке B", "программист на языке C"]
f2_to_f3 = ["программист на языке A", "Программист на языке B",
            "программист на языке C"]
f3_to_f4 = ["программист на языке A с опытом Python", "Программист
на языке B с опытом Python", "программист на языке C с опытом
Python"]
path = 'process_data_to_test/data.json'
with open(path) as f:
    data = json.load(f)

def test_f1():
    assert f1(data) == f1_to_f2

def test_f2():
    assert f2(f1_to_f2) == f2_to_f3

def test_f3():
    assert f3(f2_to_f3) == f3_to_f4

def test_f4():
    ans = f4(f3_to_f4)
    assert len(ans) == len(f3_to_f4)
    for i in range(len(f3_to_f4)):
        assert ans[i][0] == f3_to_f4[i]
        assert re.match(r"зарплата\s+\d+\s+руб\.", ans[i][1])
        assert 100000 <= int(re.search(r"\d+", ans[i][1]).group())
<= 200000

def main():
    test_f1()
    test_f2()
    test_f3()
    test_f4()
```

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

Код bdd тестирования (pytest-bdd модуль):

```
import pytest  
from pytest_bdd import scenario, given, when, then, parsers  
from process_data_to_test.gen_random import gen_random  
  
@scenario('gen_random.feature', 'Giving count, begin and end')  
def test_gen_random():  
    pass  
  
@given(parsers.parse("{num_count:d} numbers to generate from  
{begin:d} to {end:d}"))  
def given_to_gen_random(num_count, begin, end):  
    pytest.num_count = num_count  
    pytest.begin = begin  
    pytest.end = end  
    pytest.tmp = gen_random(num_count, begin, end)  
  
@when("using gen_random generator")  
def generate():  
    pytest.tmp_list = [i for i in pytest.tmp]  
  
@then("I should get these numbers")  
def check():  
    assert len(pytest.tmp_list) == pytest.num_count  
    for i in pytest.tmp_list:  
        assert pytest.begin <= i <= pytest.end
```

feature файл:

```
@gen_random  
Feature: gen_random  
    Generator function to generate random integer.
```

```
Scenario: Giving count, begin and end  
    Given 5 numbers to generate from 2 to 2  
    When using gen_random generator  
    Then I should get these numbers  
  
    Given 5 numbers to generate from 1 to 3  
    When using gen_random generator
```

Then I should get these numbers

Given 5 numbers to generate from -3 to -1

When using `gen_random` generator

Then I should get these numbers

Пример выполнения.

Тестирование tdd:

```
(virtualenv) nop@nopc:~/Projects/bmstu_3sem/BKIT_2022$ python tdd.py
f1
преподаватель языка C
программист на языке A
Программист на языке B
программист на языке C
f2
программист на языке A
Программист на языке B
программист на языке C
f3
программист на языке A с опытом Python
Программист на языке B с опытом Python
программист на языке C с опытом Python
f4
('программист на языке A с опытом Python', 'зарплата 103505 руб.')
('Программист на языке B с опытом Python', 'зарплата 103676 руб.')
('программист на языке C с опытом Python', 'зарплата 169807 руб.')
(virtualenv) nop@nopc:~/Projects/bmstu_3sem/BKIT_2022$ █
```

Тестирование bdd:

```
(virtualenv) nop@nopc:~/Projects/bmstu_3sem/BKIT_2022$ pytest
===== test session starts =====
platform linux -- Python 3.8.10, pytest-7.2.0, pluggy-1.0.0
rootdir: /home/nop/Projects/bmstu_3sem/BKIT_2022, configfile: pytest.ini
plugins: bdd-6.1.1
collected 1 item

test_gen_random.py . [100%]

===== 1 passed in 0.02s =====
(virtualenv) nop@nopc:~/Projects/bmstu_3sem/BKIT_2022$ █
```