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»

Выполнил:
студент группы ИУ5-32
Вольвач Михаил
Подпись и дата: 29.12.2022

Проверил:
преподаватель каф. ИУ5
Гапанюк Ю. Е.
Подпись и дата: 29.12.2022

Москва, 2022

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

Файл sort.py

```
def num_sort(data, rev=False):
    return sorted(data, key=int.__abs__, reverse=rev)


def num_lambda_sort(data, rev=False):
    return sorted(data, key=lambda elem: abs(elem), reverse=rev)


def sort(data, rev=False):
    return sorted(data, reverse=rev)
```

Файл tdd/sort_test.py

```
import unittest
from lab_python_fp.sort import sort, num_sort


class SortTest(unittest.TestCase):
    def test1(self):
        self.assertEqual(num_sort([1, 2, 1, 1, 3, 2, 1, 2], rev=False), [1, 1, 1,
1, 2, 2, 2, 3])

    def test2(self):
        self.assertEqual(num_sort([1, 2, 1, 1, 3, 2, 1, 2], rev=True), [3, 2, 2, 2,
1, 1, 1, 1])


class NumSortTest(unittest.TestCase):
    def test1(self):
        self.assertEqual(sort(['1', '2', '1', '1', '3', '2', '1', '2'], rev=False),
['1', '1', '1', '1', '2', '2', '2', '3'])

    def test2(self):
        self.assertEqual(sort(['1', '2', '1', '1', '3', '2', '1', '2'], rev=True),
['3', '2', '2', '2', '1', '1', '1', '1'])


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

Файл features/SortTest.feature

```
Feature: Test of sort.py with number list
  descr

  Scenario: SortTestNum.py without reverse of the result
    Given the list is [1, 2, 1, 1, 3, 2, 1, 2] of type Integer and reverse is
False
    When I sort the list
```

```
    Then I expect the result to be [1, 1, 1, 1, 2, 2, 2, 3]

  Scenario: SortTestNum.py with reverse of the result
    Given the list is [1, 2, 1, 1, 3, 2, 1, 2] of type Integer and reverse is
True
    When I sort the list
    Then I expect the result to be [3, 2, 2, 2, 1, 1, 1, 1]

  Scenario: SortTestNum.py without reverse of the result
    Given the list is [1, 2, 1, 1, 3, 2, 1, 2] of type String and reverse is
False
    When I sort the list
    Then I expect the result to be [1, 1, 1, 1, 2, 2, 2, 3]

  Scenario: SortTestNum.py with reverse of the result
    Given the list is [1, 2, 1, 1, 3, 2, 1, 2] of type String and reverse is
True
    When I sort the list
    Then I expect the result to be [3, 2, 2, 2, 1, 1, 1, 1]
```

Файл features/steps/SortTest.py

```python
from lab_python_fp.sort import num_sort
from lab_python_fp.sort import sort
from behave import *


@given("the list is [{list}] of type {type} and reverse is {reverse}")
def have_info(context, list, type, reverse):

    if type == "string" or type == "String":
        context.type = str
        context.list = [i for i in list.split(', ')]
    elif type == "integer" or type == "Integer":
        context.type = int
        context.list = [int(i) for i in list.split(', ')]
    context.reverse = False if reverse == "False" or reverse == "false" else True


@when("I sort the list")
def find_unique(context):
    context.result = num_sort(context.list, rev=context.reverse) if context.type ==
int else sort(context.list, rev=context.reverse)


@then("I expect the result to be [{list}]")
def expect_result(context, list):
    result = []
    if context.type == str:
        result = [i for i in list.split(', ')]
    elif context.type == int:
        result = [int(i) for i in list.split(', ')]
    assert context.result == result
```

Пример выполнения программы

TDD:

```
/usr/local/bin/python3.10 /Applications/PyCharm.app/Contents/plugins/python/helpers/pycharm/_jb_pytest_runner.py --path /Users/mikhail/Documents/BKIT/Lab1/tdd/sort_test.py
Testing started at 10:34 ...
Launching pytest with arguments /Users/mikhail/Documents/BKIT/Lab1/tdd/sort_test.py --no-header --no-summary -q in /Users/mikhail/Documents/BKIT/Lab1/tdd

============================= test session starts ==============================
collecting ... collected 4 items

sort_test.py::SortTest::test1 PASSED                                     [ 25%]
sort_test.py::SortTest::test2 PASSED                                     [ 50%]
sort_test.py::NumSortTest::test1 PASSED                                  [ 75%]
sort_test.py::NumSortTest::test2 PASSED                                  [100%]

============================== 4 passed in 0.02s ===============================

Process finished with exit code 0
```

BDD:

```
/usr/local/bin/python3.10 /Applications/PyCharm.app/Contents/plugins/python/helpers/pycharm/behave_runner.py
Testing started at 10:41 ...

Process finished with exit code 0
```