Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждениевысшего образования

«Московский государственный технический университетимени Н.Э. Баумана (национальный исследовательский университет)»(МГТУ им. Н.Э. Баумана)

Факультет «Информатика и системы управления»

Кафедра «Системы обработки информации и управления»

Лабораторная работа №5

«Модульное тестирование в Python»

по предмету

Базовые компоненты интернет-технологий

Выполнил:

студент группы № ИУ5-33Б

Попов Степан Дмитриевич

Проверил:

Преподаватель кафедры ИУ-5

Гапанюк Юрий

2022 г.

Задание

- Выберите любой фрагмент кода из лабораторных работ 1 или 2 или 3-4.

- Модифицируйте код таким образом, чтобы он был пригоден для модульного тестирования.

Листинг кода

unit_test.py:

```
import unittest
from unittest.mock import patch


from lab_python_fp.process_data import f1
from lab_python_fp.unique import Unique


# testing f1 from process_data.py
class Test_Process_Data_F1(unittest.TestCase):
    def setUp(self):
        self.data = [
            {"job-name" : "Street-photographer"},
            {"job-name" : "programmist"} ,
            {"job-name" : "programmist"},
            {"job-name" : "Tatoo-master"}
        ]
        self.supposed_result = [
            "programmist",
            "Street-photographer",
            "Tatoo-master"
        ]
    def test_without_mock(self):
        res = f1(self.data)
        self.assertEqual(res, self.supposed_result)

    @patch('lab_python_fp.process_data.field')
    @patch('lab_python_fp.process_data.Unique')
    def test_with_mock(self, mock_Unique, mock_field):
        mock_field.return_value = ["Street-photographer", "programmist",
"programmist", "Tatoo-master"]
        mock_Unique.return_value = (i for i in ["Street-photographer",
"programmist", "Tatoo-master"] )

        res = f1(self.data)

        mock_field.assert_called_once_with(self.data, 'job-name')
        mock_Unique.assert_called_once_with(mock_field.return_value,
ignore_case=True)
        self.assertEqual(res, self.supposed_result)

# testing Unique from unique.py
class Test_Unique(unittest.TestCase):
    def test_same_object(self):
        data = ['a', 'a', 'a', 'a', 'a', 'a', 'a', 'a', 'a', 'a']
        res = list(Unique(data))
        supposed_res = ['a']
        self.assertEqual(res, supposed_res)
    def test_base(self):
        data = [1, 1, 1, 1, 1, 2, 2, 1, 1, 2, 3, 'A', 3, 'a', '1567', 1567, 'yooo']
        res = list(Unique(data))
        supposed_res = [1, 2, 3, 'A', 'a', '1567', 1567, 'yooo']
        self.assertEqual(res, supposed_res)

    def test_ignore_case(self):
        data = ['aBC', 'ABc', 'aBc', 'abc', 'ABC']
        res = list(Unique(data, ignore_case=True))
        supposed_res = ['aBC']
        self.assertEqual(res, supposed_res)

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

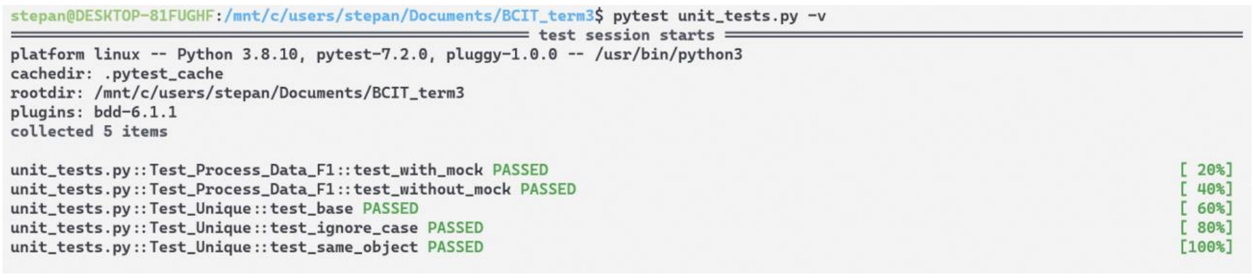
sort.feature

```
Feature: Sorting
    sorting abc
    Scenario: Sort_abs with original sort
        Given There is an array which is [4, -30, 30, 100, -100, 123, 1, 0, -1, -4]
        When I sort this array originally
        Then Array is [123, 100, -100, -30, 30, 4, -4, 1, -1, 0]

    Scenario: Sort_abs with lambda sort
        Given There is an array which is [4, -30, 30, 100, -100, 123, 1, 0, -1, -4]
        When I sort this array with lambda
        Then Array is [123, 100, -100, -30, 30, 4, -4, 1, -1, 0]
```

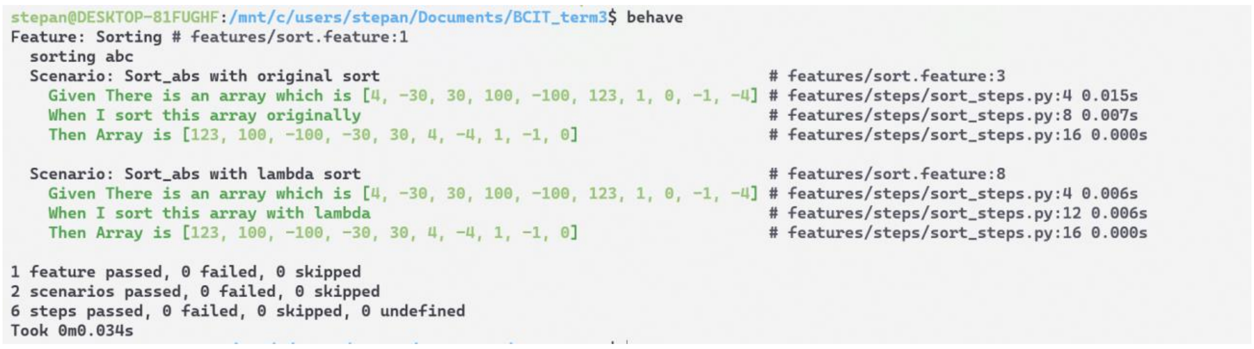
sort_steps.py

```
from behave import given, when, then

from lab_python_fp.sort import sort1, sort2


@given("There is an array which is [{array}]")

def step_have_array(context, array):

   context.array = [int(i) for i in array.split(', ')]


@when("I sort this array originally")

def step_sort1_array(context):

   context.sorted_array = sort1(context.array)


@when("I sort this array with lambda")

def step_sort2_array(context):

   context.sorted_array = sort2(context.array)


@then("Array is [{array}]")

def step_expect_result(context, array):

   result = [int(i) for i in array.split(', ')]

   assert context.sorted_array == result
```

Примеры работы программы:


```
stepan@DESKTOP-81FUGHF:/mnt/c/users/stepan/Documents/BCIT_term3$ pytest unit_tests.py -v
=================================== test session starts ====================================
platform linux -- Python 3.8.10, pytest-7.2.0, pluggy-1.0.0 -- /usr/bin/python3
cachedir: .pytest_cache
rootdir: /mnt/c/users/stepan/Documents/BCIT_term3
plugins: bdd-6.1.1
collected 5 items

unit_tests.py::Test_Process_Data_F1::test_with_mock PASSED                           [ 20%]
unit_tests.py::Test_Process_Data_F1::test_without_mock PASSED                        [ 40%]
unit_tests.py::Test_Unique::test_base PASSED                                         [ 60%]
unit_tests.py::Test_Unique::test_ignore_case PASSED                                  [ 80%]
unit_tests.py::Test_Unique::test_same_object PASSED                                  [100%]
```

BDD:


```
stepan@DESKTOP-81FUGHF:/mnt/c/users/stepan/Documents/BCIT_term3$ behave
Feature: Sorting # features/sort.feature:1
  sorting abc
  Scenario: Sort_abs with original sort                                      # features/sort.feature:3
    Given There is an array which is [4, -30, 30, 100, -100, 123, 1, 0, -1, -4] # features/steps/sort_steps.py:4 0.015s
    When I sort this array originally                                        # features/steps/sort_steps.py:8 0.007s
    Then Array is [123, 100, -100, -30, 30, 4, -4, 1, -1, 0]                 # features/steps/sort_steps.py:16 0.000s

  Scenario: Sort_abs with lambda sort                                        # features/sort.feature:8
    Given There is an array which is [4, -30, 30, 100, -100, 123, 1, 0, -1, -4] # features/steps/sort_steps.py:4 0.006s
    When I sort this array with lambda                                       # features/steps/sort_steps.py:12 0.006s
    Then Array is [123, 100, -100, -30, 30, 4, -4, 1, -1, 0]                 # features/steps/sort_steps.py:16 0.000s

1 feature passed, 0 failed, 0 skipped
2 scenarios passed, 0 failed, 0 skipped
6 steps passed, 0 failed, 0 skipped, 0 undefined
Took 0m0.034s
```