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 ИУ5-34Б

Угрюмов Михаил

Проверил:
преподаватель каф. ИУ5

Гапанюк Ю. Е.

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).

Текст программы.

sort.py

```
def my_sort_with_lambda(data):
    return sorted(data, key = lambda x: abs(x))

def my_sort_without_lambda(data):
    return sorted(data, key = abs)
```

field.py

```
def field(items, *args):
    assert len(args) > 0

    if len(args) == 1:
        answer = []
        for i in range(len(items)):
            answer.append(items[i].get(args[0]))
    else:
        answer = [{} for i in range(len(items))]
        for i in range(len(items)):
            for key in args:
                curr = items[i].get(key)
                if curr is not None:
                    answer[i][key] = curr

    return answer
```

unique.py

```
class Unique(object):
    def __init__(self, items, **kwargs):
        self.cnt = -1
        if len(kwargs) == 0:
            self.ignore_case = False
        else:
            value = kwargs['ignore_case']
            self.ignore_case = value
        self.items = []
        for elem in items:
            if isinstance(elem, str):
                if self.ignore_case == True:
                    elem_ = elem.lower()
                    if elem_ not in self.items:
                        self.items.append(elem_)
                else:
                    if elem not in self.items:
                        self.items.append(elem)
            else:
                if elem not in self.items:
                    self.items.append(elem)

    def __next__(self):
```

```
        if self.cnt < len(self.items) - 1:
            self.cnt += 1
            return self.items[self.cnt]
        else:
            raise StopIteration

    def __iter__(self):
        return self

    def __repr__(self):
        return str(self.items)
```

TDD_sort.py

```
from functions.sort import my_sort_with_lambda, my_sort_without_lambda
import unittest as ut

class TestSort(ut.TestCase):
    def setUp(self):
        self.data = [4, -30, 30, 100, -100, 123, 1, 0, -1, -4]
        self.answer = [0, 1, -1, 4, -4, -30, 30, 100, -100, 123]

    def test_1(self):
        self.assertEqual(my_sort_with_lambda(self.data), self.answer)

    def test_2(self):
        self.assertEqual(my_sort_without_lambda(self.data), self.answer)

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

TDD_field.py

```
from functions.field import field
import unittest as ut

class TestField(ut.TestCase):
    def setUp(self):
        self.goods = [
            {'title': 'Ковер', 'price': 2000, 'color': 'green'},
            {'title': 'Диван для отдыха', 'price': 5300, 'color': 'black'}
        ]
        self.answer_1 = ['Ковер', 'Диван для отдыха']
        self.answer_2 = [
            {'title': 'Ковер', 'price': 2000},
            {'title': 'Диван для отдыха', 'price': 5300}
        ]

    def test_1(self):
        self.assertEqual(field(self.goods, 'title'), self.answer_1)

    def test_2(self):
        self.assertEqual(field(self.goods, 'title', 'price'), self.answer_2)

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

TDD_unique.py

```
from functions.unique import Unique
import unittest as ut

class TestUnique(ut.TestCase):
    def setUp(self):
        self.symbols = ['a', 'A', 'b', 'B', 'a', 'A', 'b', 'B']
        self.numbers = [1, 2, 2, 1, 1, 3, 2, 3, 5, 4, 1]
        self.answer_1 = [1, 2, 3, 4, 5]
        self.answer_2 = ['a', 'A', 'b', 'B']
```

```
        self.answer_3 = ['a', 'b']

    def test_1(self):
        self.assertEqual(set(Unique(self.numbers)), set(self.answer_1))

    def test_2(self):
        self.assertEqual(set(Unique(self.symbols)), set(self.answer_2))

    def test_3(self):
        self.assertEqual(set(Unique(self.symbols, ignore_case=True)),
set(self.answer_3))

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

my_sort_with_lambda.feature

```
Feature: sorting by ABS with lambda
Scenario: sorting by ABS with lambda
Given data
When sorting data: my_sort_with_lambda
Then data was sorted
```

my_sort_without_lambda.feature

```
Feature: sorting by ABS without lambda
Scenario: sorting by ABS without lambda
Given data
When sorting data: my_sort_without_lambda
Then data was sorted
```

BDD_sort.py

```
from functions.sort import my_sort_with_lambda, my_sort_without_lambda
from pytest_bdd import scenario, given, when, then

@scenario('features\\my_sort_with_lambda.feature','sorting by ABS with
lambda')
def testing_my_sort_with_lambda():
    pass

@scenario('features\\my_sort_without_lambda.feature','sorting by ABS without
lambda')
def testing_my_sort_without_lambda():
    pass

@given('data', target_fixture='data')
def data():
    return [4, -30, 30, 100, -100, 123, 1, 0, -1, -4]

@when('sorting data: my_sort_with_lambda', target_fixture='answer')
def sort_with_lambda(data):
    return my_sort_with_lambda(data)

@when('sorting data: my_sort_without_lambda', target_fixture='answer')
def sort_without_lambda(data):
    return my_sort_without_lambda(data)

@then('data was sorted')
def check_answer(answer):
    assert answer == [0, 1, -1, 4, -4, -30, 30, 100, -100, 123]
```

field_1.feature

```
Feature: get goods` fields_1
Scenario: get goods` fields_1
Given goods_1, *args_1
```

```
When processing_1
Then answer_1 got
```

field_2.feature

```
Feature: get goods` fields_2
Scenario: get goods` fields_2
Given goods_2, *args_2
When processing_2
Then answer_2 got
```

BDD_field.py

```
from functions.field import field
from pytest_bdd import scenario, given, when, then

@scenario('features\\field_1.feature','get goods` fields_1')
def test_1():
    pass

@scenario('features\\field_2.feature','get goods` fields_2')
def test_2():
    pass

@given('goods_1, *args_1', target_fixture='data_1')
def goods():
    goods = [
        {'title': 'Ковер', 'price': 2000, 'color': 'green'},
        {'title': 'Диван для отдыха', 'price': 5300, 'color': 'black'}
    ]
    args = ['title']
    data_1 = [goods, args]
    return data_1

@given('goods_2, *args_2', target_fixture='data_2')
def goods():
    goods = [
        {'title': 'Ковер', 'price': 2000, 'color': 'green'},
        {'title': 'Диван для отдыха', 'price': 5300, 'color': 'black'}
    ]
    args = ['title', 'price']
    data_2 = [goods, args]
    return data_2

@when('processing_1', target_fixture='answer_1')
def processing(data_1):
    return field(data_1[0], *data_1[1])

@when('processing_2', target_fixture='answer_2')
def processing(data_2):
    return field(data_2[0], *data_2[1])

@then('answer_1 got')
def check_answer(answer_1):
    assert answer_1 == ['Ковер', 'Диван для отдыха']

@then('answer_2 got')
def check_answer(answer_2):
    assert answer_2 == [
        {'title': 'Ковер', 'price': 2000},
        {'title': 'Диван для отдыха', 'price': 5300}
    ]
```

Результаты выполнения программы:

TDD_sort.py

```
C:\Users\Миша\Desktop\БКИТ\laba5_new\venv\Scripts\python.exe "C:/Program Files/Je
Testing started at 17:27 ...
Launching pytest with arguments C:\Users\Миша\Desktop\БКИТ\laba5_new\TDD_sort.py

============================= test session starts =============================
collecting ... collected 2 items

TDD_sort.py::TestSort::test_1 PASSED                                     [ 50%]
TDD_sort.py::TestSort::test_2 PASSED                                     [100%]

============================== 2 passed in 0.02s ==============================

Process finished with exit code 0
```

TDD_field.py

```
C:\Users\Миша\Desktop\БКИТ\laba5_new\venv\Scripts\python.exe "C:/Program Files/J
Testing started at 17:28 ...
Launching pytest with arguments TDD_field.py::TestField --no-header --no-summary

============================= test session starts =============================
collecting ... collected 2 items

TDD_field.py::TestField::test_1 PASSED                                   [ 50%]
TDD_field.py::TestField::test_2 PASSED                                   [100%]

============================== 2 passed in 0.02s ==============================

Process finished with exit code 0
```

TDD_unique.py

```
C:\Users\Миша\Desktop\БКИТ\laba5_new\venv\Scripts\python.exe "C:/Program Files/J
Testing started at 17:28 ...
Launching pytest with arguments C:\Users\Миша\Desktop\БКИТ\laba5_new\TDD_unique.

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

TDD_unique.py::TestUnique::test_1 PASSED                                 [ 33%]
TDD_unique.py::TestUnique::test_2 PASSED                                 [ 66%]
TDD_unique.py::TestUnique::test_3 PASSED                                 [100%]

============================== 3 passed in 0.02s ==============================

Process finished with exit code 0
```

BDD_sort.py

```
C:\Users\Миша\Desktop\БКИТ\laba5_new\venv\Scripts\python.exe "C:/Program Files/JetBrains/PyCharm Community
Testing started at 17:28 ...
Launching pytest with arguments C:\Users\Миша\Desktop\БКИТ\laba5_new\BDD_sort.py --no-header --no-summary -

============================= test session starts =============================
collecting ... collected 2 items

BDD_sort.py::testing_my_sort_with_lambda <- venv\Lib\site-packages\pytest_bdd\scenario.py PASSED [ 50%]
BDD_sort.py::testing_my_sort_without_lambda <- venv\Lib\site-packages\pytest_bdd\scenario.py PASSED [100%]

============================== 2 passed in 0.02s ==============================

Process finished with exit code 0
```

BDD_field.py

```
C:\Users\Миша\Desktop\БКИТ\laba5_new\venv\Scripts\python.exe "C:/Program Files/JetBr
Testing started at 17:29 ...
Launching pytest with arguments C:\Users\Миша\Desktop\БКИТ\laba5_new\BDD_field.py --

============================= test session starts =============================
collecting ... collected 2 items

BDD_field.py::test_1 <- venv\Lib\site-packages\pytest_bdd\scenario.py PASSED [ 50%]
BDD_field.py::test_2 <- venv\Lib\site-packages\pytest_bdd\scenario.py PASSED [100%]

============================== 2 passed in 0.02s ==============================

Process finished with exit code 0
```