Московский государственный технический университет им. Н.Э. Баумана.

Факультет «Информатика и системы управления»

Кафедра ИУ5. Курс «Парадигмы и конструкции языков программирования»

Отчет по лабораторной работе №5
«Модульное тестирование в Python»

Выполнил:

Макеева Е. А.
ИУ5-31Б

Подпись и дата:

Проверил:

Нардид А. Н.

Подпись и дата:

Постановка задачи

Выберите любой фрагмент кода из лабораторных работ 1 или 2 или 3-4.
Модифицируйте код таким образом, чтобы он был пригоден для модульного тестирования.
Разработайте модульные тесты. В модульных тестах необходимо применить следующие технологии:
TDD - фреймворк (не менее 3 тестов).
BDD - фреймворк (не менее 3 тестов).

TDD - фреймворк

```
import unittest
from field_module import field


class TestField(unittest.TestCase):

    def test_single_field(self):
        items = [{'title': 'Ковер', 'price': 2000, 'color': 'green'},
            {'title': 'Диван для отдыха', 'color': 'black'}]
        result = list(field(items, 'title'))
        self.assertEqual(result, ['Ковер', 'Диван для отдыха'])

    def test_multiple_fields(self):
        items = [{'title': 'Ковер', 'price': 2000, 'color': 'green'},
            {'title': 'Диван для отдыха', 'color': 'black'}]
        result = list(field(items, 'title', 'price'))
        self.assertEqual(result, [{'title': 'Ковер', 'price': 2000},
{'title': 'Диван для отдыха'}])

    def test_empty_input(self):
        items = []
        result = list(field(items, 'title'))
        self.assertEqual(result, [])


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

```
D:\3_sem\venv\Scripts\python.exe "C:\Program Files\JetBrains\PyCharm Community Edition 213.3714.452\plugins\python-ce\helpers\pycharm\_jb_pytest_r
Testing started at 9:45 ...
Launching pytest with arguments D:/3_sem/test_field_tdd.py --no-header --no-summary -q in D:\3_sem

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

test_field_tdd.py::TestField::test_empty_input PASSED                    [ 33%]
test_field_tdd.py::TestField::test_multiple_fields PASSED                [ 66%]
test_field_tdd.py::TestField::test_single_field PASSED                   [100%]

============================== 3 passed in 0.05s ==============================

Process finished with exit code 0
```

BDD - фреймворк

```python
from behave import given, when, then
from field_module import field


@given('a list of goods')
def step_given_list_of_goods(context):
    context.goods = [
        {'title': 'Ковер', 'price': 2000, 'color': 'green'},
        {'title': 'Диван для отдыха', 'color': 'black'}
    ]


@when('I get the "{field_name}" field')
def step_when_get_single_field(context, field_name):
    context.result = list(field(context.goods, field_name))


@when('I get the "{field1}" and "{field2}" fields')
def step_when_get_multiple_fields(context, field1, field2):
    context.result = list(field(context.goods, field1, field2))


@when('I try to get an empty field list')
def step_when_try_empty_field_list(context):
    try:
        context.result = list(field(context.goods))
    except AssertionError as e:
        context.result = str(e)


@then('I should get the result {expected_result}')
def step_then_check_result(context, expected_result):
    expected_result = eval(expected_result)
    assert context.result == expected_result


@then('I should receive an error "{error_message}"')
def step_then_check_error(context, error_message):
    assert context.result == error_message
```

```
PS D:\3_sem> behave
Feature: Проверка функции field # features/goods.feature:1

  Scenario: Получение одного поля из списка товаров           # features/goods.feature:3
    Given a list of goods                                     # features/steps/test_field_bdd.py:5
    When I get the "title" field                              # features/steps/test_field_bdd.py:13
    Then I should get the result ["Ковер", "Диван для отдыха"] # features/steps/test_field_bdd.py:31

  Scenario: Получение нескольких полей                                                                  # features/goods.feature:8
    Given a list of goods                                                                               # features/steps/test_field_bdd.py:5
    When I get the "title" and "price" fields                                                           # features/steps/test_field_bdd.py:18
    Then I should get the result [{"title": "Ковер", "price": 2000}, {"title": "Диван для отдыха"}] # features/steps/test_field_bdd.py:31

  Scenario: Передача пустых аргументов                                  # features/goods.feature:13
    Given a list of goods                                               # features/steps/test_field_bdd.py:5
    When I try to get an empty field list                               # features/steps/test_field_bdd.py:23
    Then I should receive an error "Необходимо передать хотя бы одно поле." # features/steps/test_field_bdd.py:37

1 feature passed, 0 failed, 0 skipped
3 scenarios passed, 0 failed, 0 skipped
9 steps passed, 0 failed, 0 skipped, 0 undefined
Took 0m0.008s
PS D:\3_sem>
```