

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

Факультет «Информатика и системы управления»

Кафедра ИУ5. Курс «Базовые компоненты интернет-технологий»
Отчёт по лабораторной работе № 5.

Выполнил:

студент группы ИУ5-516
Афанасьев Д. М.

Проверил:

преподаватель каф. ИУ5
Гапанюк Ю.Е.

Подпись и дата:

Подпись и дата:

Москва, 2022 г.

Задание

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

main.py

-

tests.py

```
from field import field
from json import loads as load_jstr
from behave import *

@given('a goods list')
def given_a_list(context):
    context.list = load_jstr(context.text)

@given("fields #1 {f1} and #2 {f2}")
def given_a_fields(context, f1: {str}, f2: {str}):
    context.F1 = f1
    context.F2 = f2

@given("fields #1 {f1}")
def given_a_field(context, f1: {str}):
    context.F1 = f1
    context.F2 = None

@when('we call function with that data we get result')
def func_call(context):
    if context.F2 is not None:
        context.data = tuple(i for i in field(context.list, context.F1, context.F2))
        return
    context.data = tuple(i for i in field(context.list, context.F1))

@Then("we can assert this data with the tuple")
```

```
def check_data(context):
    assert context.data == tuple(load_jstr(context.text))
```

field.py

```
from __future__ import annotations

def field(items: tuple[dict] | list[dict], *args: str):

    if type(items) not in (list, tuple):
        raise TypeError("Wrong goods format! Try correct types.")

    if not (len(args) > 0):
        raise RuntimeError('No keys specified to print!')

    if tuple(arg for arg in args if arg in items[0].keys()) != args:
        raise KeyError('Selected keys are not in goods list!')

    if len(args) == 1:
        for product in items:
            yield product[args[0]]
    else:
        for product in items:
            yield {arg: product[arg] for arg in args}

if name == " main ":
    goods = [
        {'title': 'Ковер', 'price': 2000, 'color': 'green'},
        {'title': 'Диван для отдыха', 'price': 5300, 'color': 'black'}
    ]

    print("Testing...")

    # Testing
    assert tuple(i for i in field(goods, 'title')) == ('Ковер', 'Диван для
отдыха'), \
        "\n[TestError][#1] -> -Failed-"
    assert tuple(i for i in field(goods, 'title', 'price')) == ({'title':
'Ковер', 'price': 2000},
                                                                {'title':
'Диван для отдыха', 'price': 5300}), \
        "\n[TestError][#2] -> -Failed-"

    print("All tests passed!\n")

    print("Run with args ('title, color'):")
    # Run
    for callback in field(goods, 'title', "color"):
        print(callback)
```

unittests.py

```
import unittest
from field import field

goods = [
    {'title': 'Ковер', 'price': 2000, 'color': 'green'},
    {'title': 'Диван для отдыха', 'price': 5300, 'color': 'black'}
]
```

```

class TestEquation(unittest.TestCase):

    def test_calculate(self):
        self.assertEqual(tuple(i for i in field(goods, 'title')), ('Ковер',
'Диван для отдыха'))
        self.assertEqual(tuple(i for i in field(goods, 'title', 'price')),
({'title': 'Ковер', 'price': 2000},
{'title': 'Диван для отдыха',
'price': 5300}))

    def test_type(self):
        with self.assertRaises(TypeError) as e:
            tuple(i for i in field('', 'title'))

    def test_runtime(self):
        with self.assertRaises(RuntimeError) as e:
            tuple(i for i in field(goods))

    def test_key(self):
        with (self.assertRaises(KeyError)) as e:
            tuple(i for i in field(goods, 'title1', 'price'))

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

```

Анализ результатов

field.py

```

C:\Users\Дани\AppData\Local\Programs\Python\Python311\python.exe C:\Users\Дани\PycharmProjects\LR5\field.py
Testing...
All tests passed!

Run with args ('title, color'):
{'title': 'Ковер', 'color': 'green'}
{'title': 'Диван для отдыха', 'color': 'black'}

Process finished with exit code 0

```

tests.py

```

C:\Users\Дани\AppData\Local\Programs\Python\Python311\python.exe C:\Users\Дани\PycharmProjects\LR5\BDD\steps\tests.py

Process finished with exit code 0

```

unittests.py

```

C:\Users\Дани\AppData\Local\Programs\Python\Python311\python.exe
"C:/Microsoft/Python/PyCharm Community Edition 2022.2.3/plugins/python-
ce/helpers/pycharm/_jb_pytest_runner.py" --path

```

```

C:\Users\Дани\PycharmProjects\LR5\TDD\unittests.py

```

```

Testing started at 15:35 ...

```

```

Launching pytest with arguments C:\Users\Дани\PycharmProjects\LR5\TDD\unittests.py --no-
header --no-summary -q in C:\Users\Дани\PycharmProjects\LR5\TDD

```

```

===== test session starts
=====

```

collecting ... collected 4 items

unittests.py::TestEquation::test_calculate	PASSED	[25%]
unittests.py::TestEquation::test_key	PASSED	[50%]
unittests.py::TestEquation::test_runtime	PASSED	[75%]
unittests.py::TestEquation::test_type	PASSED	[100%]

===== 4 passed in 0.01s
=====

Process finished with exit code 0

main.py

-