

**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
Жданова Яна
Подпись и дата:

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

Описание задания:

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:

Field.py

```
goods = [
    {'title': 'Cover', 'price': 2000, 'color': 'green'},
    {'title': 'Divan dlya otidixa', 'price': 5300, 'color': 'black'}
]
def field(items, *args):
    res = []
    for i in range(0, len(items)):
        for j in range(0, len(args)):
            if(items[i].get(args[j]) != None):
                res.append(items[i].get(args[j]))
    return res
```

Gen_random.py

```
import random
def gen_random(num_count, begin, end):
    for i in range(0, num_count):
        yield random.randint(begin, end)
```

sort.py

```
def sorting_lambda(data):
    result_with_lambda = sorted(data, key = lambda x: abs(x), reverse=True)
    return result_with_lambda

def sorting_no_lambda(data):
    result = sorted(data, key = abs, reverse=True)
    return result
```

unique.py

```
import Gen_random
class Unique(object):
    def __init__(self, items, **kwargs):
        self.arr = []
        for i in items:
            if len(kwargs) > 0 and kwargs["ignore_case"]:
                if i.lower() not in self.arr:
                    self.arr.append(i.lower())
```

```

        else:
            if i not in self.arr:
                self.arr.append(i)

    def __next__(self):
        item = self.arr[0]
        del self.arr[0]
        return item

    def __iter__(self):
        return self

```

TDD-тестирование

TDDtests.py

```

import unittest
from Field import field
from sort import sorting_lambda, sorting_no_lambda
from unique import Unique

class TestsTDD (unittest.TestCase):
    def test_field(self):
        goods = [
            {'title': 'Cover', 'price': 2000, 'color': 'green'},
            {'title': 'Divan dlya otidixa', 'price': 5300, 'color': 'black'}
        ]
        self.assertEqual(field(goods, 'title', 'price'), ['Cover', 2000, 'Divan
dlya otidixa', 5300])
        #/////////////////////////////////////////////////////////////////
        ///////////////////////////////////////////////////////////////////
    def test_sort_lambda(self):
        data = [4, -30, 100, -100, 123, 1, 0, -1, -4]
        self.assertEqual(sorting_lambda(data), [123, 100, -100, -30, 4, -4, 1, -
1, 0])

    def test_sort_no_lambda(self):
        data = [4, -30, 100, -100, 123, 1, 0, -1, -4]
        self.assertEqual(sorting_no_lambda(data), [123, 100, -100, -30, 4, -4, 1,
-1, 0])

    def test_compare_sorts(self):
        data = [4, -30, 100, -100, 123, 1, 0, -1, -4]
        self.assertEqual(sorting_lambda(data), sorting_no_lambda(data))
        #/////////////////////////////////////////////////////////////////
        ///////////////////////////////////////////////////////////////////
    def test_unique(self):
        D = ['A', 'a', 'B', 'b']
        obj = Unique(D, ignore_case=True)
        self.assertEqual(obj.__next__(), 'a')
        self.assertEqual(obj.__next__(), 'b')

if __name__ == '__main__':

```

```
unittest.main()
```

BDD-тестирование

bdd-test-field_2.py

```
import pytest
from pytest_bdd import scenario, given, when, then
from Field import field

@scenario('bdd_field_2.feature', 'Selection fields')
def test_field():
    pass

@given("i have the data", target_fixture="data")
def data():
    return [
        {'title': 'Cover', 'price': 2000, 'color': 'green'},
        {'title': 'Divan dlya otidixa', 'price': 5300, 'color': 'black'}
    ]

@when("the function takes 2 arguments", target_fixture="my_result")
def function(data):
    return field(data, 'title')

@then("i selected the fields with 2 arguments")
def result(my_result):
    assert my_result == ['Cover', 'Divan dlya otidixa']
```

bdd-test-field_3.py

```
import pytest
from pytest_bdd import scenario, given, when, then
from Field import field

@scenario('bdd_field_3.feature', 'Selection fields')
def test_field():
    pass

@given("i have the data", target_fixture="data")
def data():
    return [
        {'title': 'Cover', 'price': 2000, 'color': 'green'},
        {'title': 'Divan dlya otidixa', 'price': 5300, 'color': 'black'}
    ]

@when("the function takes 3 arguments", target_fixture="my_result")
def function(data):
    return field(data, 'title', 'price')

@then("i selected the fields with 3 arguments")
def result(my_result):
    assert my_result == ['Cover', 2000, 'Divan dlya otidixa', 5300]
```

bdd_sorting_lambda.py

```
import pytest
from pytest_bdd import scenario, given, when, then
from sort import sorting_lambda

@scenario('bddsort_lambda.feature', 'Sorting array')
def test_sort():
    pass

@given("i have the array for sorting - data", target_fixture="data")
def data():
    return [4, -30, 100, -100, 123, 1, 0, -1, -4]

@when("sorting is including lambda function", target_fixture="my_result")
def function(data):
    with_labda = data
    return sorting_lambda(with_labda)

@then("i sorted array with lambda fuction")
def result(my_result):
    assert my_result == [123, 100, -100, -30, 4, -4, 1, -1, 0]
```

bdd_sorting_no_lambda.py

```
import pytest
from pytest_bdd import scenario, given, when, then
from sort import sorting_no_lambda

@scenario('bddsort_nolambda.feature', 'Sorting array')
def test_sort():
    pass

@given("i have the array for sorting - data", target_fixture="data")
def data():
    return [4, -30, 100, -100, 123, 1, 0, -1, -4]

@when("sorting is not including lambda function", target_fixture="my_result")
def function(data):
    with_labda = data
    return sorting_no_lambda(with_labda)

@then("i sorted array without lambda fuction")
def result(my_result):
    assert my_result == [123, 100, -100, -30, 4, -4, 1, -1, 0]
```

bdd_field_2.feature

```
Feature: Selection fields
Scenario: Selection fields
    Given i have the data
    When the function takes 2 arguments
```

```
Then i selected the fields with 2 arguments
```

bdd_field_3.feature

Feature: Selection fields

Scenario: Selection fields

Given i have the data

When the function takes 3 arguments

Then i selected the fields with 3 arguments

bddsort_lambda.feature

Feature: Sorting

Scenario: Sorting array

Given i have the array for sorting - data

When sorting is including lambda function

Then i sorted array with lambda fuction

bddsort_nolambda.feature

Feature: Sorting

Scenario: Sorting array

Given i have the array for sorting - data

When sorting is not including lambda function

Then i sorted array without lambda fuction

Пример выполнения программы:

TDD-тестирование

```
ПРОБЛЕМЫ  ВЫХОДНЫЕ ДАННЫЕ  КОНСОЛЬ ОТЛАДКИ  ТЕРМИНАЛ

Windows PowerShell
(C) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

PS C:\Users\user\Desktop\please> & c:/Users/user/Desktop/please/venv/Scripts/Activate.ps1
(venv) PS C:\Users\user\Desktop\please> & 'c:\Users\user\Desktop\please\venv\Scripts\python.
debugpy\launcher' '50468' '--' 'c:\Users\user\Desktop\please\TDDtests.py'
.....
-----
Ran 5 tests in 0.003s

OK
(venv) PS C:\Users\user\Desktop\please> █
```

BDD-тестирование

ПРОБЛЕМЫ ВЫХОДНЫЕ ДАННЫЕ КОНСОЛЬ ОТЛАДКИ ТЕРМИНАЛ

```
PS C:\Users\user\Desktop\please> & c:/Users/user/Desktop/please/venv/Scripts/Activate.ps1
(venv) PS C:\Users\user\Desktop\please> pytest -q "C:/Users/user/Desktop/please/bdd_test_field_2.py"
.
1 passed in 0.05s
(venv) PS C:\Users\user\Desktop\please> pytest -q "C:/Users/user/Desktop/please/bdd_test_field_3.py"
.
1 passed in 0.03s
(venv) PS C:\Users\user\Desktop\please> pytest -q "C:/Users/user/Desktop/please/bdd_sorting_lambda.py"
.
1 passed in 0.03s
(venv) PS C:\Users\user\Desktop\please> pytest -q "C:/Users/user/Desktop/please/bdd_sorting_no_lambda.py"
.
1 passed in 0.03s
(venv) PS C:\Users\user\Desktop\please> 
```