

**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»**

**Домашнее задание.**

Выполнил:  
студент группы ИУ5-34Б:  
Белозеров Дмитрий Сергеевич  
Подпись и дата:

Проверил:  
преподаватель каф. ИУ5  
Гапанюк Юрий Евгеньевич  
Подпись и дата:

## Условия домашнего задания по курсу БКИТ

1. Модифицируйте код лабораторной работы №5 или №6 таким образом, чтобы он был пригоден для модульного тестирования.
2. Используя материалы лабораторной работы №4 создайте модульные тесты с применением TDD - фреймворка (2 теста) и BDD - фреймворка (2 теста).

### Код программы с использованием TDD фреймворка

#### main\_TDD.py

```
import unittest
import PIL
from PIL import Image
import numpy as np

class Testfield(unittest.TestCase):
    def setUp(self):
        self.list_im = ["photos_for_tests\image_1.jpg",
"photos_for_tests\image_2.jpg"]
    def test_with_size(self):
        res = [865, 1081, 1024, 1280]
        self.assertEqual(res, first_func(self.list_im))

    def test_with_united_photo(self):
        res = Image.open("photos_for_tests\Trifecta.jpg")
        second_func(self.list_im)
        p = Image.open("photos\Trifecta.jpg")
        self.assertEqual(res, p)

def first_func(list_im):
    img1 = PIL.Image.open(list_im[0])
    img2 = PIL.Image.open(list_im[1])
    wid1, hgt1 = img1.size
    wid2, hgt2 = img2.size
    return [wid1, hgt1, wid2, hgt2]

def second_func(list_im):
    imgs = [PIL.Image.open(i) for i in list_im]
    min_shape = sorted([(np.sum(i.size), i.size) for i in imgs])[0][1]
    imgs_comb = np.hstack((np.asarray(i.resize(min_shape)) for i in imgs))
    imgs_comb = PIL.Image.fromarray(imgs_comb)
    imgs_comb.save("photos\Trifecta.jpg")

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

## Результат выполнения

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

main_TDD.py::Testfield::test_with_size
main_TDD.py::Testfield::test_with_united_photo

===== 2 passed, 1 warning in 0.65s =====

Process finished with exit code 0
PASSED [ 50%]PASSED [100%]
```

## Код программы с использованием TDD фреймворка

### main\_BDD.py

```
from pytest_bdd import scenario, given, when, then
import PIL
from PIL import Image
import numpy as np

@scenario('main_BDD_1.feature', 'Getting picture sizes')
def testing_main_1():
    pass

@scenario('main_BDD_2.feature', 'Connecting pictures')
def testing_main_2():
    pass

@given('Some dict', target_fixture = "data")
def given_dict():
    return ["photos_for_tests\image_1.jpg", "photos_for_tests\image_2.jpg"]

@when('Getting sizes of pictures', target_fixture = "result_1")
def result_with_size(data):
    return first_func(data)

@when('Getting 1 picture from 2', target_fixture = "result_2")
def result_with_pictures(data):
    second_func(data)
    p = Image.open("photos\Trifecta.jpg")
    return p

@then('Get picture sizes')
def res_dict_1(result_1):
    assert result_1 == [865, 1081, 1024, 1280]

@then('Get connected picture')
def res_dict_1(result_2):
    assert result_2 == Image.open("photos_for_tests\Trifecta.jpg")

def first_func(list_im):
    img1 = PIL.Image.open(list_im[0])
```

```

img2 = PIL.Image.open(list_im[1])
wid1, hgt1 = img1.size
wid2, hgt2 = img2.size
return [wid1, hgt1, wid2, hgt2]

def second_func(list_im):
    imgs = [PIL.Image.open(i) for i in list_im]
    min_shape = sorted([(np.sum(i.size), i.size) for i in imgs])[0][1]
    imgs_comb = np.hstack((np.asarray(i.resize(min_shape)) for i in imgs))
    imgs_comb = PIL.Image.fromarray(imgs_comb)
    imgs_comb.save("photos\Trifecta.jpg")

```

## main\_BDD\_1.feature

```

Feature: Telegram bot

    Scenario: Getting picture sizes
        Given Some dict
        When Getting sizes of pictures
        Then Get picture sizes

```

## main\_BDD\_2.feature

```

Feature: Telegram bot

    Scenario: Connecting pictures
        Given Some dict
        When Getting 1 picture from 2
        Then Get connected picture

```

## Результат выполнения

```

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

main_BDD.py::testing_main_1 <- venv\lib\site-packages\pytest_bdd\scenario.py PASSED [ 50%]
main_BDD.py::testing_main_2 <- venv\lib\site-packages\pytest_bdd\scenario.py PASSED [100%]

===== 2 passed, 6 warnings in 0.23s =====

Process finished with exit code 0

```