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»

Выполнила:
студентка группы ИУ5-31Б
Саркисян С. З.

Проверил:
Гапанюк Е.Ю.

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).
    - Создание Mock-объектов (необязательное дополнительное задание).

Текст программы

Файлы:

```
tdd.py
features/
    test.feature
    steps/
        bdd.py
```

tdd.py

```python
import sys
#changed cwd to testible dir
sys.path.append("../lab_python_oop")
from lab_python_oop.Rectangle import Rectangle
from lab_python_oop.Circle import Circle
from lab_python_oop.Square import Square
import unittest

class MyTesting(unittest.TestCase):
    def setUp(self):
        self.a = Rectangle(6, 6, 'blue', 'pryamougolnik')
        self.b = Circle(6, 'green', 'krug')
        self.c = Square(3, 'red', 'kvadrat')

    def test_area(self):
        import math
        self.assertEqual(self.a.area(), 36)
        self.assertEqual(self.b.area(), math.pi*6**2)
        self.assertEqual(self.c.area(), 9)

    def test_color(self):
        self.assertEqual(self.a.color.value, 'blue')
        self.assertEqual(self.b.color.value, 'green')
        self.assertEqual(self.c.color.value, 'red')

    def test_get_name(self):
        self.assertEqual(self.a.get_name(), 'pryamougolnik')
        self.assertEqual(self.b.get_name(), 'krug')
        self.assertEqual(self.c.get_name(), 'kvadrat')

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

test.feature

```
Feature: Testing lab_python_oop

    Scenario Outline: Testing properties of rectangle
        Given rectangle with sides of "<first>" and "<second>", color is "<color>"
and name is "<name>"
        When we try to get properties
        Then we get area of "<area>", color is "<color>" and name is "<name>"

        Examples: Rectangle
            | first | second | name            | color | area   |
            | 9     | 4      | pryamougolnik   | blue  | 45     |
            | 5     | 1      | xXx_rect_xXx    | red   | 5      |
            | 99    | 98     | rEcTaNgLe       | yellow| 9702   |

    Scenario Outline: Testing properties of circle
        Given circle with radius of "<radius>", color is "<color>" and name is
"<name>"
        When we try to get properties
        Then we get area of "<area>", color is "<color>" and name is "<name>"

        Examples: Circle
            | radius | name             | color | area                |
            | 1      | krug             | cyan  | 3.141592653589793   |
            | 12     | mega_krug        | gray  | 37.69911184307752   |
            | 200    | super ultra krug | black | 348.71678454846705  |

    Scenario Outline: Testing properties of circle
        Given square with side of "<side>", color is "<color>" and name is "<name>"
        When we try to get properties
        Then we get area of "<area>", color is "<color>" and name is "<name>"

        Examples: Square
            | side | name          | color | area   |
            | 1    | kvadrat       | lime  | 1      |
            | 12   | kvadratik     | pink  | 144    |
            | 200  | square shape  | white | 40000  |
```

bdd.py

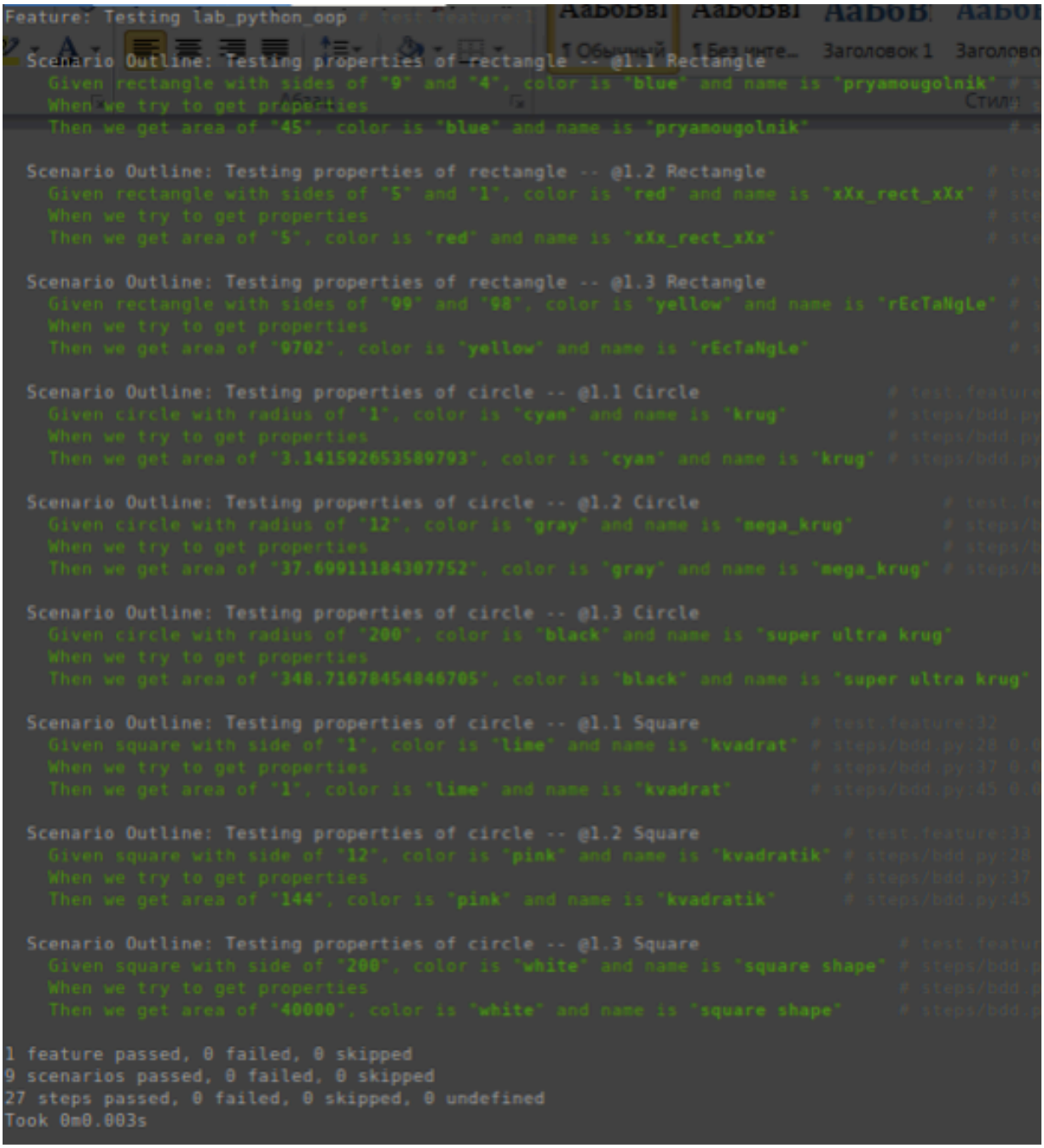
```
from behave import *
import sys

sys.path.append("../../lab_python_oop/lab_python_oop")
from Circle import Circle
from Rectangle import Rectangle
from Square import Square

@given('rectangle with sides of "{first}" and "{second}", color is "{color}" and
name is "{name}"')
def step_impl(context, first, second, name, color):
    global shape
    try:
        shape = Rectangle(int(first), int(second), name, color)
        return True
    except:
        return False

@given('circle with radius of "{radius}", color is "{color}" and name is "{name}"')
def step_impl(context, radius, name, color):
    global shape
    try:
        shape = Circle(int(radius), name, color)
        return True
```

```
    except:
        return False

@given('square with side of "{side}", color is "{color}" and name is "{name}"')
def step_impl(context, side, name, color):
    global shape
    try:
        shape = Square(int(side), name, color)
        return True
    except:
        return False

@when('we try to get properties')
def step_impl(context):
    if shape.area():
        if shape.get_name():
            if shape.color.value:
                return True
    return False

@then('we get area of "{area}", color is "{color}" and name is "{name}"')
def step_impl(context, area, color, name):
    if shape.area() == area:
        if shape.get_name() == name:
            if shape.color.value == color:
                return True
    return False
```

Экранные формы с примерами выполнения программы

tdd.py

bdd.py



```
Feature: Testing lab_python_oop # test.feature:1

  Scenario Outline: Testing properties of rectangle -- @1.1 Rectangle
    Given rectangle with sides of "9" and "4", color is "blue" and name is "pryamougolnik"
    When we try to get properties
    Then we get area of "45", color is "blue" and name is "pryamougolnik"

  Scenario Outline: Testing properties of rectangle -- @1.2 Rectangle
    Given rectangle with sides of "5" and "1", color is "red" and name is "xXx_rect_xXx"
    When we try to get properties
    Then we get area of "5", color is "red" and name is "xXx_rect_xXx"

  Scenario Outline: Testing properties of rectangle -- @1.3 Rectangle
    Given rectangle with sides of "99" and "98", color is "yellow" and name is "rEcTaNgLe"
    When we try to get properties
    Then we get area of "9702", color is "yellow" and name is "rEcTaNgLe"

  Scenario Outline: Testing properties of circle -- @1.1 Circle
    Given circle with radius of "1", color is "cyan" and name is "krug"
    When we try to get properties
    Then we get area of "3.141592653589793", color is "cyan" and name is "krug"

  Scenario Outline: Testing properties of circle -- @1.2 Circle
    Given circle with radius of "12", color is "gray" and name is "mega_krug"
    When we try to get properties
    Then we get area of "37.69911184307752", color is "gray" and name is "mega_krug"

  Scenario Outline: Testing properties of circle -- @1.3 Circle
    Given circle with radius of "200", color is "black" and name is "super ultra krug"
    When we try to get properties
    Then we get area of "348.71678454846705", color is "black" and name is "super ultra krug"

  Scenario Outline: Testing properties of circle -- @1.1 Square
    Given square with side of "1", color is "lime" and name is "kvadrat"
    When we try to get properties
    Then we get area of "1", color is "lime" and name is "kvadrat"

  Scenario Outline: Testing properties of circle -- @1.2 Square
    Given square with side of "12", color is "pink" and name is "kvadratik"
    When we try to get properties
    Then we get area of "144", color is "pink" and name is "kvadratik"

  Scenario Outline: Testing properties of circle -- @1.3 Square
    Given square with side of "200", color is "white" and name is "square shape"
    When we try to get properties
    Then we get area of "40000", color is "white" and name is "square shape"

1 feature passed, 0 failed, 0 skipped
9 scenarios passed, 0 failed, 0 skipped
27 steps passed, 0 failed, 0 skipped, 0 undefined
Took 0m0.003s
```