**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 ИУ5-31Б

Корецкий К.В.

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

**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:**

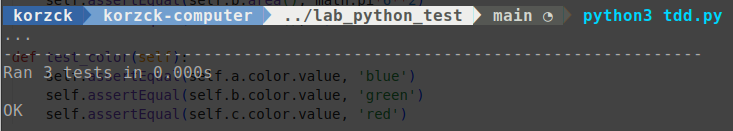
**if shape.color.value == color:**

**return True**

**return False**

**Экранные формы с примерами выполнения программы**

tdd.py



bdd.py

