

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

Факультет «Информатика и системы управления»

Кафедра ИУ5. Курс «Парадигмы и конструкции языков программирования»

Отчет по лабораторной работе №5
«Модульное тестирование в Python»

Выполнил:

Каженец Д.Н.
ИУ5-31Б

Подпись и дата:

Проверил:

Гапанюк Е.Ю.

Подпись и дата:

Постановка задачи

Выберите любой фрагмент кода из лабораторных работ 1 или 2 или 3-4.

Модифицируйте код таким образом, чтобы он был пригоден для модульного тестирования.

Разработайте модульные тесты. В модульных тестах необходимо применить следующие технологии:

TDD - фреймворк (не менее 3 тестов).

BDD - фреймворк (не менее 3 тестов).

TDD - фреймворк

```
import unittest
from main import Rectangle, Circle, Square
from math import pi

class TestGeometricShapes(unittest.TestCase):

    def test_rectangle_area(self):
        rect = Rectangle(8, 4, 'синий')
        self.assertEqual(rect.area(), 32)

    def test_rectangle_repr(self):
        rect = Rectangle(8, 4, 'синий')
        expected = "Название: Прямоугольник, Ширина: 8, Высота: 4, Цвет: синий, Площадь: 32"
        self.assertEqual(repr(rect), expected)

    def test_circle_area(self):
        circ = Circle(3, 'зеленый')
        self.assertAlmostEqual(circ.area(), pi * 3**2, places=2)

    def test_square_name(self):
        sqr = Square(5, 'красный')
        self.assertEqual(sqr.name(), "Квадрат")

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

Ran 4 tests in 0.002s

OK

BDD - фреймворк

```
from behave import given, when, then
from main import Rectangle, Circle, Square
import math

@given('a rectangle with width {width:d} and height {height:d}')
def step_given_rectangle(context, width, height):
    context.rectangle = Rectangle(width, height, 'синий')
```

```

@when('I calculate the area')
def step_when_calculate_area(context):
    context.area = context.rectangle.area()

@then('the area should be {expected_area:d}')
def step_then_area_should_be(context, expected_area):
    assert context.area == expected_area, f"Expected {expected_area}, but got {context.area}"

@given('a square with side {side:d}')
def step_given_square(context, side):
    context.square = Square(side, 'красный')

@then('the name should be "{expected_name}"')
def step_then_square_name_should_be(context, expected_name):
    assert context.square.name() == expected_name, f"Expected {expected_name}, but got {context.square.name()}"

@given('I have a circle with radius {radius:d} and color "{color}"')
def step_create_circle(context, radius, color):
    context.circle = Circle(radius, color)

@when('I calculate the area of the circle')
def step_calculate_area(context):
    context.area = context.circle.area()

@then('the area should be approximately {expected:f}')
def step_check_area(context, expected):
    assert math.isclose(context.area, expected, rel_tol=1e-2)

```

Feature: Geometric Shapes # features/shapes.feature:1

```

Scenario: Calculate area of a rectangle      # features/shapes.feature:3
  Given a rectangle with width 8 and height 4 # features/steps/my_steps.py:5
  When I calculate the area                  # features/steps/my_steps.py:9
  Then the area should be 32                 # features/steps/my_steps.py:13

Scenario: Get name of the square             # features/shapes.feature:8
  Given a square with side 5                 # features/steps/my_steps.py:17
  Then the name should be "Квадрат"          # features/steps/my_steps.py:21

Scenario: Calculating circle area            # features/shapes.feature:12
  Given I have a circle with radius 3 and color "blue" # features/steps/my_steps.py:25
  When I calculate the area of the circle          # features/steps/my_steps.py:29
  Then the area should be approximately 28.27      # features/steps/my_steps.py:33

```

```

1 feature passed, 0 failed, 0 skipped
3 scenarios passed, 0 failed, 0 skipped
8 steps passed, 0 failed, 0 skipped, 0 undefined

```