

Отчет

По лабораторной работе – 6

Дисциплина «Парадигмы и конструкции языков программирования»

Студент: Артёмова Дарья

Группа: ИБМЗ – 34Б

Выбран код из лабораторной работы №2, проводятся тест TDD и BDD (не менее трех тестов).

Пример изначального кода geometric_figures.py:

```
import math
from abc import ABC, abstractmethod

class Color:
    def __init__(self, color):
        self.color = color

class Figure(ABC):
    @abstractmethod
    def area(self):
        pass

    def get_name(self):
        return self.__class__.__name__

    def __repr__(self):
        return f"Фигура: {self.get_name()}, Цвет: {self.color.color}, Площадь: {self.area():.2f}"

class Rectangle(Figure):
    def __init__(self, width, height, color):
        if width <= 0 or height <= 0:
            raise ValueError("Ширина и высота должны быть положительными числами.")
        self.width = width
        self.height = height
        self.color = Color(color)

    def area(self):
        return self.width * self.height

class Circle(Figure):
    def __init__(self, radius, color):
        if radius <= 0:
            raise ValueError("Радиус должен быть положительным числом.")
        self.radius = radius
        self.color = Color(color)

    def area(self):
        return math.pi * (self.radius ** 2)

class Square(Rectangle):
    def __init__(self, side_length, color):
        super().__init__(side_length, side_length, color)
```

Пример кода с запуском теста TDD (unittest/5 тестов)

test_geometric_figures_unittest.py:

```
import unittest
from geometric_figures import Rectangle, Circle, Square

class TestFigures(unittest.TestCase):
    def test_rectangle_area(self):
        rect = Rectangle(5, 10, "blue")
        self.assertEqual(rect.area(), 50)

    def test_circle_area(self):
        circ = Circle(5, "green")
        self.assertAlmostEqual(circ.area(), 78.5398, places=4)

    def test_square_area(self):
        square = Square(5, "red")
        self.assertEqual(square.area(), 25)

    def test_rectangle_error(self):
        with self.assertRaises(ValueError):
            Rectangle(-5, 10, "blue")

    def test_circle_error(self):
        with self.assertRaises(ValueError):
            Circle(-5, "green")

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

Тесты, написанные с использованием фреймворка unittest, который является стандартным инструментом тестирования в Python. Они проверяют отдельные аспекты работы классов.

Выводы кода:

```
Testing started at 1:44 ...
Launching pytest with arguments C:\Users\26276\PycharmProjects\pythonProject1\test_geometric_

===== test session starts =====
collecting ... collected 5 items

test_geometric_figures_unittest.py::TestFigures::test_circle_area PASSED [ 20%]
test_geometric_figures_unittest.py::TestFigures::test_circle_error PASSED [ 40%]
test_geometric_figures_unittest.py::TestFigures::test_rectangle_area PASSED [ 60%]
test_geometric_figures_unittest.py::TestFigures::test_rectangle_error PASSED [ 80%]
test_geometric_figures_unittest.py::TestFigures::test_square_area PASSED [100%]

===== 5 passed in 0.02s =====

Process finished with exit code 0
```

```
(venv) PS C:\Users\26276\PycharmProjects\pythonProject1> python -m unittest test_geometric_figures_unittest.py
.....
-----
Ran 5 tests in 0.001s

OK
```

Пример кода с запуском теста BDD (pytest/5 месное) test_geometric_figures_pytest.py:

```
import pytest
import math
from geometric_figures import Rectangle, Circle, Square

@pytest.fixture
def rectangle_fixture():
    return Rectangle(5, 10, "blue")

@pytest.fixture
def circle_fixture():
    return Circle(5, "green")

@pytest.fixture
def square_fixture():
    return Square(5, "red")

def test_rectangle_area_bdd(rectangle_fixture):
    assert rectangle_fixture.area() == 50

def test_circle_area_bdd(circle_fixture):
    assert math.isclose(circle_fixture.area(), 78.5398, abs_tol=0.001)

def test_square_area_bdd(square_fixture):
    assert square_fixture.area() == 25

def test_rectangle_error_bdd():
    with pytest.raises(ValueError):
        Rectangle(-5, 10, "blue")

def test_circle_error_bdd():
    with pytest.raises(ValueError):
        Circle(-5, "green")
```

Тесты, написанные с использованием фреймворка pytest. pytest — это более гибкий и расширяемый фреймворк, чем unittest. @pytest.fixture декоратор, создающий фикстуры (предварительно настроенные объекты), которые затем используются в тестах.

Выводы кода:

```
Testing started at 1:45 ...
Launching pytest with arguments C:\Users\26276\PycharmProjects\pythonProject1\test_geometric_f

===== test session starts =====
collecting ... collected 5 items

test_geometric_figures_pytest.py::test_rectangle_area_bdd PASSED [ 20%]
test_geometric_figures_pytest.py::test_circle_area_bdd PASSED [ 40%]
test_geometric_figures_pytest.py::test_square_area_bdd PASSED [ 60%]
test_geometric_figures_pytest.py::test_rectangle_error_bdd PASSED [ 80%]
test_geometric_figures_pytest.py::test_circle_error_bdd PASSED [100%]

===== 5 passed in 0.02s =====

Process finished with exit code 0
```

```
(venv) PS C:\Users\26276\PycharmProjects\pythonProject1> pytest test_geometric_figures_pytest.py
===== test session starts =====
platform win32 -- Python 3.12.2, pytest-8.3.3, pluggy-1.5.0
rootdir: C:\Users\26276\PycharmProjects\pythonProject1
plugins: bdd-7.3.0, browserstack-0.1
collected 5 items

test_geometric_figures_pytest.py ..... [100%]

===== 5 passed in 0.02s =====
```