**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](https://github.com/ugapanyuk/BKIT_2022/wiki/lab_python_oop)»

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-35Б |  | Преподаватель каф. ИУ5 |
| Коныгина Дарья |  | Нардид А.Н. |
| Подпись и дата: |  | Подпись и дата: |

Москва, 2022 г.

1. Задание:

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

2. Текст программы:

Модульное тестирование разрабатывалось на основе кода для ЛР2.

Функции вычисления площади фигур, которые тестировались. Функция вычисления площади круга была изменена с правильной:

def square(self):  
 return math.pi\*(self.r)\*\*2

на неправильную:

def square(self):  
 return math.pi\*(self.r)

Файл feature.feature

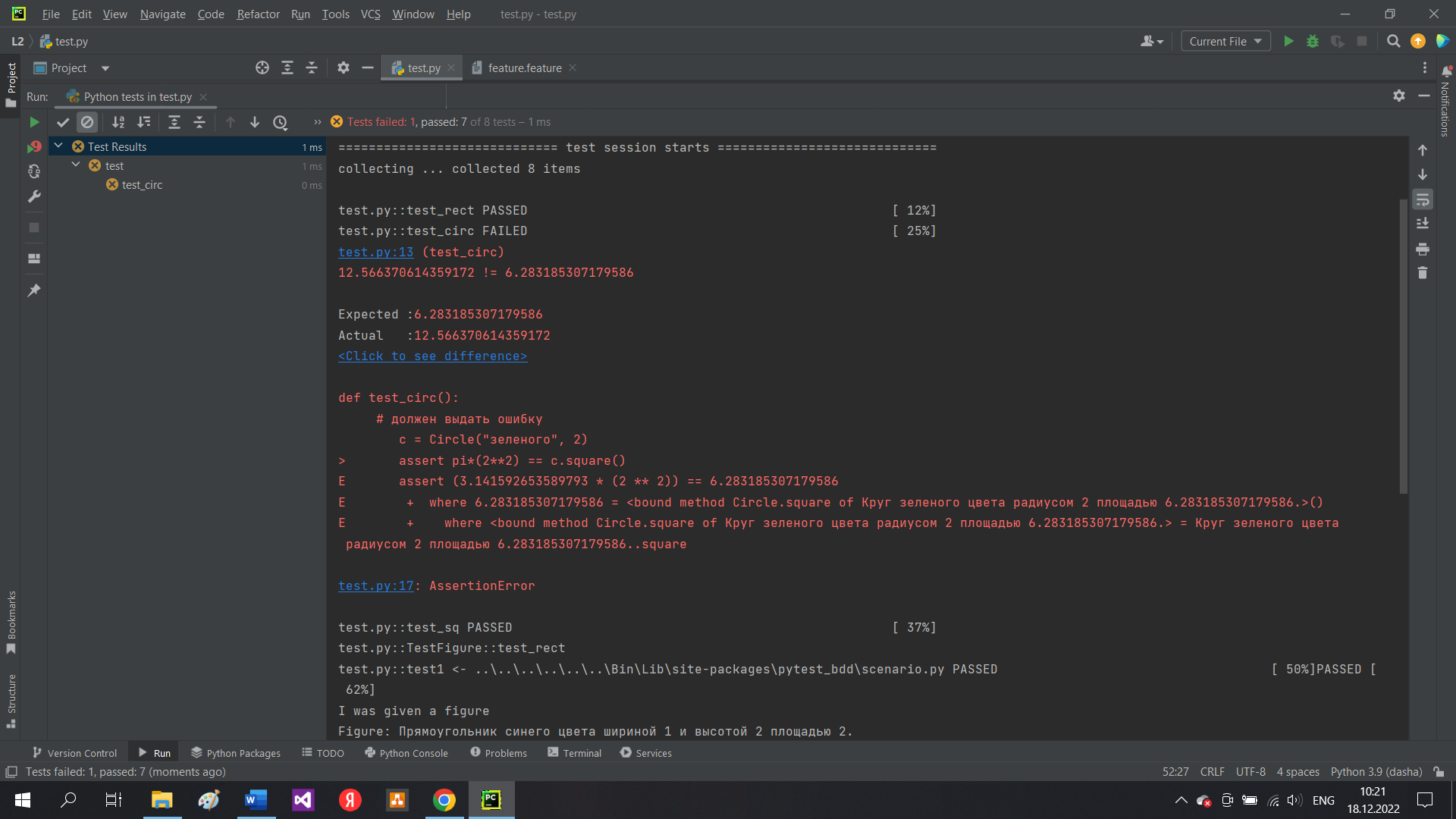
Feature: Printing characterictics of a figure  
 Scenario: Print characterictics of the figure  
 Given I was given a figure  
 When I checked functions  
 Then I print characterictics of the figure

Файл test.py

from lab\_python\_oop.rectangle import Rectangle  
from lab\_python\_oop.circle import Circle  
from lab\_python\_oop.square import Square  
import pytest  
import unittest  
from pytest\_bdd import scenario, given, when, then  
from math import \*  
  
#TDD\_1  
def test\_rect():  
 s = Rectangle("синего", 1, 2)  
 assert 1\*2 == s.square()

# должен выдать ошибку!!!!!!!!!!!!!!!!!!!!!!!!!!!!!  
def test\_circ():  
  
 c = Circle("зеленого", 2)  
 assert pi\*(2\*\*2) == c.square()  
  
def test\_sq():  
 c = Square("красного", 4)  
 assert 4\*4 == c.square()  
  
  
#TDD\_2  
class TestFigure(unittest.TestCase):  
 def test\_rect(self):  
 self.assertEqual(Rectangle("синего", 1, 2).square(), 2)  
  
  
#BDD  
@scenario('feature.feature', 'Print characterictics of the figure')  
def test1():  
 print('Scenarios: Characterictics of the figure')  
  
@given("I was given a figure")  
def test2():  
 print("\nI was given a figure")  
 print(f"Figure: {Rectangle('синего', 1, 2)}\n")  
  
@when('I checked functions')  
def test3():  
 r = Rectangle('синего', 1, 2)  
 if r.square() == 2:  
 print("All correct")  
  
@then('I print characterictics of the figure')  
def test4():  
 print('I print characterictics of the figure')  
 r = Rectangle('синего', 1, 2)  
 print(r)  
  
if \_\_name\_\_ == "\_\_main\_\_":  
   
 unittest.main()

Экранные формы с примерами выполнения программы:



Изображение выглядит как текст, снимок экрана, монитор, электроника

Автоматически созданное описание