МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ  
им. Н.Э. Баумана

Факультет “Информатика и системы управления”  
Кафедра “Системы обработки информации и управления”

A logo with a gold star and blue ribbon

Description automatically generated

Дисциплина “Парадигмы и конструкции языков программирования”

Отчет по лабораторной работе №4  
“Модульное тестирование в Python”

**Выполнил:**Студент группы ИУ5-24Б  
Изотов Е.А.  
**Преподаватель:**Нардид А.Н.

Москва 2025

Задание

1. Необходимо для произвольной предметной области реализовать от одного до трех шаблонов проектирования: один порождающий, один структурный и один поведенческий. В качестве справочника шаблонов можно использовать следующий каталог. Для сдачи лабораторной работы в минимальном варианте достаточно реализовать один паттерн.
2. В модульных тестах необходимо применить следующие технологии:

* TDD - фреймворк.
* BDD - фреймворк.
* Создание Mock-объектов.

Код(steps.py)

# steps.py  
from behave import given, when, then  
import sys  
import os  
  
# Добавляем путь к исходному коду  
sys.path.append(os.path.dirname(os.path.dirname(os.path.abspath(\_\_file\_\_))))  
  
from main import AreaCalculating  
  
@given('тип фигуры "{shapetype}" и стороны {a}, {b}, {c}')  
def step\_given\_triangle\_sides(context, shapetype, a, b, c):  
 context.shapetype = shapetype  
 context.a = int(a)  
 context.b = int(b)  
 context.c = int(c)  
  
@given('тип фигуры "{shapetype}" и сторона {a}')  
def step\_given\_square\_side(context, shapetype, a):  
 context.shapetype = shapetype  
 context.a = int(a)  
 context.b = 0  
 context.c = 0  
  
@given('тип фигуры "{shapetype}" и стороны {a}, {b}')  
def step\_given\_rectangle\_sides(context, shapetype, a, b):  
 context.shapetype = shapetype  
 context.a = int(a)  
 context.b = int(b)  
 context.c = 0  
  
@when('вычисляется площадь')  
def step\_when\_calculate\_area(context):  
 try:  
 context.result = AreaCalculating(context.shapetype, context.a, context.b, context.c)  
 context.exception = None  
 except Exception as error:  
 context.result = None  
 context.exception = error  
  
@then('результат должен быть {expected\_result}')  
def step\_then\_check\_result(context, expected\_result):  
 assert context.result == int(expected\_result), f"Ожидалось: {expected\_result}, Получено: {context.result}"  
  
@then('должен вернуться ValueError')  
def step\_then\_check\_value\_error(context):  
 assert context.result == ValueError or context.exception is not None, "Ожидался ValueError, но функция завершилась успешно"

Код(tests.py)

# tests.py  
from main import AreaCalculating  
  
def TestInputFigure():  
 try:  
 AreaCalculating(1, 1, 1, 1)  
 except ValueError:  
 pass # Ожидалось, что будет вызвано исключение  
 else:  
 assert False, "Ожидалось ValueError"  
  
def TestInputNumbers():  
 assert AreaCalculating("rectangle", -1, 1, 1) == ValueError  
 assert AreaCalculating("rectangle", -1, -1, 1) == ValueError  
 assert AreaCalculating("square", 1, 1, -1) == ValueError  
  
def TestResult():  
 assert AreaCalculating("triangle", 3, 4, 5) == 6  
 assert AreaCalculating("square", 3, 0, 0) == 9  
 assert AreaCalculating("rectangle", 5, 12, 0) == 60  
  
TestInputFigure()  
TestInputNumbers()  
TestResult()

Код(main.py)

# main.py  
import math  
  
def AreaCalculating(shapetype, a, b, c):  
 if isinstance(shapetype, str) and a > 0 and b >= 0 and c >= 0:  
 if shapetype == "rectangle":  
 return a \* b  
 elif shapetype == "square":  
 return a \* a  
 elif shapetype == "triangle":  
 if a + b > c and a + c > b and b + c > a: # Проверка на треугольное неравенство  
 per = (a + b + c) / 2  
 return int(math.sqrt((per \* (per - a) \* (per - b) \* (per - c))))  
 else:  
 return ValueError("Invalid sides for a triangle")  
 else:  
 return ValueError("Invalid shape type")  
 else:  
 return ValueError("Invalid input values")  
  
# Пример использования  
print(AreaCalculating("triangle", 3, 4, 5)) # Должно вернуть 6  
print(AreaCalculating("square", 3, 0, 0)) # Должно вернуть 9  
print(AreaCalculating("rectangle", 5, 12, 0)) # Должно вернуть 60

Код(steps.py)

# steps.py  
from behave import given, when, then  
import sys  
import os  
  
# Добавляем путь к исходному коду  
sys.path.append(os.path.dirname(os.path.dirname(os.path.abspath(\_\_file\_\_))))  
  
from main import AreaCalculating  
  
@given('тип фигуры "{shapetype}" и стороны {a}, {b}, {c}')  
def step\_given\_triangle\_sides(context, shapetype, a, b, c):  
 context.shapetype = shapetype  
 context.a = int(a)  
 context.b = int(b)  
 context.c = int(c)  
  
@given('тип фигуры "{shapetype}" и сторона {a}')  
def step\_given\_square\_side(context, shapetype, a):  
 context.shapetype = shapetype  
 context.a = int(a)  
 context.b = 0  
 context.c = 0  
  
@given('тип фигуры "{shapetype}" и стороны {a}, {b}')  
def step\_given\_rectangle\_sides(context, shapetype, a, b):  
 context.shapetype = shapetype  
 context.a = int(a)  
 context.b = int(b)  
 context.c = 0  
  
@when('вычисляется площадь')  
def step\_when\_calculate\_area(context):  
 try:  
 context.result = AreaCalculating(context.shapetype, context.a, context.b, context.c)  
 context.exception = None  
 except Exception as error:  
 context.result = None  
 context.exception = error  
  
@then('результат должен быть {expected\_result}')  
def step\_then\_check\_result(context, expected\_result):  
 assert context.result == int(expected\_result), f"Ожидалось: {expected\_result}, Получено: {context.result}"  
  
@then('должен вернуться ValueError')  
def step\_then\_check\_value\_error(context):  
 assert context.result == ValueError or context.exception is not None, "Ожидался ValueError, но функция завершилась успешно"

Код(area.feature)

#language: ru  
Функция: Вычисление площади геометрических фигур  
 Как пользователь системы  
 Я хочу вычислять площадь различных фигур  
 Чтобы получать точные математические результаты  
  
 Сценарий: Вычисление площади треугольника с целыми сторонами  
 Дано тип фигуры "triangle" и стороны 3, 4, 5  
 Когда вычисляется площадь  
 Тогда результат должен быть 6  
  
 Сценарий: Вычисление площади квадрата  
 Дано тип фигуры "square" и сторона 3  
 Когда вычисляется площадь  
 Тогда результат должен быть 9  
  
 Сценарий: Вычисление площади прямоугольника  
 Дано тип фигуры "rectangle" и стороны 5, 12  
 Когда вычисляется площадь  
 Тогда результат должен быть 60  
  
 Сценарий: Вычисление площади треугольника с дробными сторонами  
 Дано тип фигуры "triangle" и стороны 5, 5, 5  
 Когда вычисляется площадь  
 Тогда результат должен быть 10  
  
 Сценарий: Некорректный тип фигуры  
 Дано тип фигуры "circle" и сторона 5  
 Когда вычисляется площадь  
 Тогда должен вернуться ValueError  
  
 Сценарий: Отрицательные значения сторон  
 Дано тип фигуры "triangle" и стороны -1, 4, 5  
 Когда вычисляется площадь  
 Тогда должен вернуться ValueError  
  
 Сценарий: Нулевые значения сторон для треугольника  
 Дано тип фигуры "triangle" и стороны 0, 4, 5  
 Когда вычисляется площадь  
 Тогда должен вернуться ValueError  
  
 Сценарий: Некорректный тип данных для фигуры  
 Дано тип фигуры 123 и сторона 5  
 Когда вычисляется площадь  
 Тогда должен вернуться ValueError  
  
 Сценарий: Вычисление площади квадрата с нулевыми дополнительными параметрами  
 Дано тип фигуры "square" и стороны 3, 0, 0  
 Когда вычисляется площадь  
 Тогда результат должен быть 9  
  
 Сценарий: Вычисление площади прямоугольника с нулевым третьим параметром  
 Дано тип фигуры "rectangle" и стороны 5, 12, 0  
 Когда вычисляется площадь  
 Тогда результат должен быть 60

Работа приложения





