МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ

«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

ФАКУЛЬТЕТ БИЗНЕСА

Кафедра Экономической информатики

**ОТЧЁТ О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ №7**

по дисциплине

«Разработка программных приложений»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Выполнил:** |  | **Проверил:** |  | |
| Студент | *Лапшин Е.В.* | Преподаватель | *Зайцев А.А.* | |
| Факультет | *Факультет бизнеса* |  |  | |
| Направление (специальность) подготовки | *Бизнес-информатика* | Балл: \_\_\_\_\_\_\_\_\_ | |  |
| Группа | *ФБИ-23* | Оценка *\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_* | | |
| Шифр |  |  |  | |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | |
| Подпись студента | | Подпись преподавателя | | |
| Дата сдачи: | «\_\_»\_\_\_\_\_\_\_\_\_\_20\_\_г. | Дата защиты: | «\_\_»\_\_\_\_\_\_\_\_\_\_20\_\_г. | |

Новосибирск 2025

Оглавление

[ЦЕЛЬ ЛАБОРАТОРНОЙ РАБОТЫ 3](#_Toc198403030)

[Введение 4](#_Toc198403031)

[Раздел I. Реализация юнит тестов при помощи модулей unittest и 5](#_Toc198403032)

[pytest 5](#_Toc198403033)

[Заключение 10](#_Toc198403034)

[Приложение 11](#_Toc198403035)

# **ЦЕЛЬ ЛАБОРАТОРНОЙ РАБОТЫ**

Изучение методов написания юнит тестов для программ на Python.

# Введение

Лабораторная работа №7 посвящена изучению методов модульного тестирования на языке Python. Основное внимание уделено использованию стандартной библиотеки unittest и популярного фреймворка pytest для проверки корректности работы функций и классов. В процессе работы необходимо реализовать функцию определения типа треугольника, разработать соответствующий класс Triangle и обеспечить тестирование как позитивных, так и негативных сценариев.

# Раздел I. Реализация юнит тестов при помощи модулей unittest и

# pytest

1. Написать на Python функцию get\_triangle\_type с тремя аргументами,

которые интерпретируются как длины сторон треугольника. Функция должна

возвращать строку с типом треугольника: "nonequilateral", "isosceles", "equilateral".

Если переданные аргументы некорректны, то функция должна выбрасывать

исключение IncorrectTriangleSides. Функцию get\_triangle\_type сохранить в файле

triangle\_func.py.

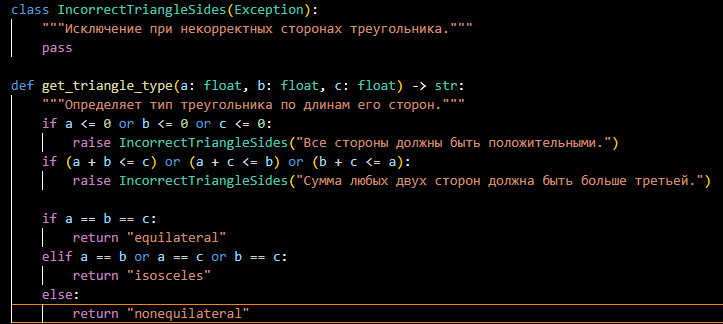


Рисунок 1 - triangle\_func.py

2. Составить в файле check.txt чек-лист с наборами данных для

позитивного и негативного тестирования.

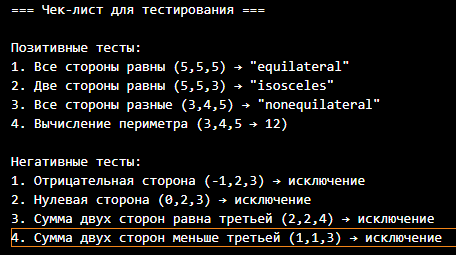


Рисунок 2 - check.txt

3. Организовать тестирование функции get\_triangle\_type с помощью

стандартной библиотеки unittest. Тесты сохранить в файле test\_func.py.

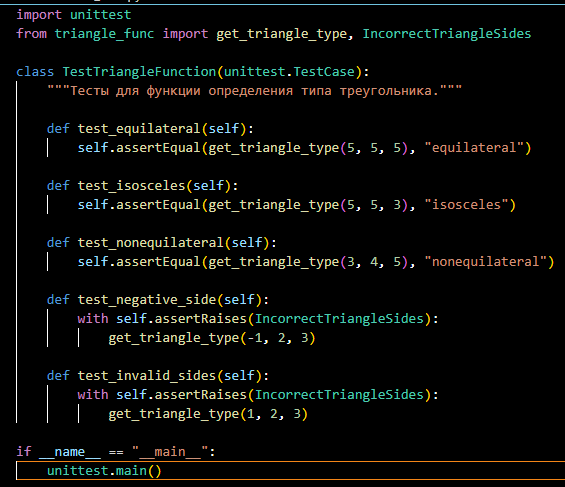


Рисунок 3 - test\_func.py

4. Реализовать на языке Python класс Triangle, описывающий

треугольник, включащий методы:

1. Конструктор \_\_init\_\_- принимает на вход три числа, которые

интерпретируются как длины сторон треугольника. Если входные данные

некорректные, конструктор должен выбрасывать исключение

IncorrectTriangleSides.

2. Метод triangle\_type - возвращает строку с типом треугольника:

"nonequilateral", "isosceles", "equilateral".

3. Метод perimeter - возвращает периметр треугольника.

5. Код с описанием класса поместить в файл triangle\_class.py.

6. Написать позитивные и негативные модульные тесты для проверки

корректности создания объекта-треугольника и работы его методов с помощью

фреймворка pytest. Тесты сохранить в файле test\_class.py.

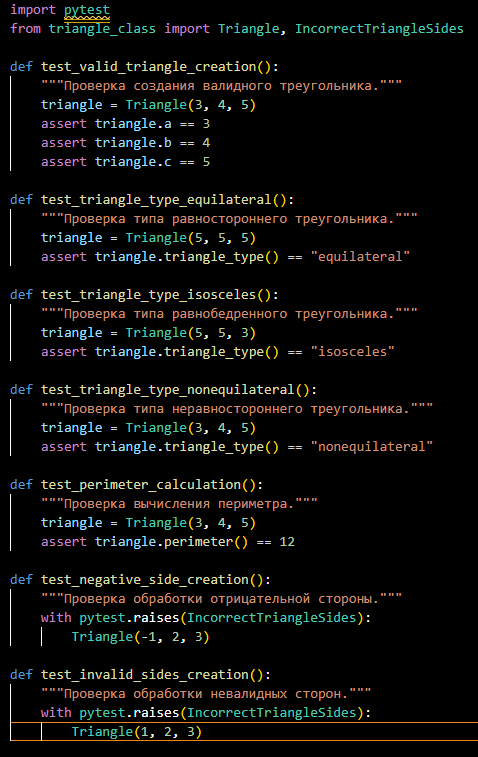


Рисунок 4 - test\_class.py

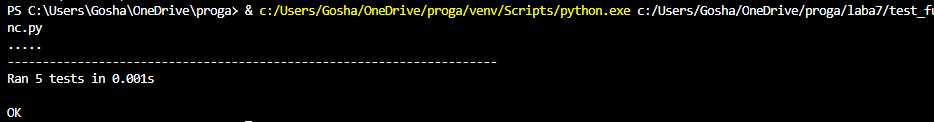


Рисунок 5 - результат

# Заключение

В результате выполнения лабораторной работы были получены практические навыки разработки и тестирования программных компонентов с использованием инструментов unittest и pytest. Реализация функции и класса, а также написание тестов, позволили убедиться в важности модульного тестирования для обеспечения стабильности и надёжности программ. Полученные знания могут быть успешно применены при создании более сложных программных решений.

# Приложение

Ссылка на гитхаб <https://github.com/lupusin/proga>