

5 Лабораторная работа №5

Исследование способов модульного тестирования программного обеспечения в среде JUnit/ NUnit

5.1. Цель работы

Исследовать эффективность использования методологии TDD при разработке программного обеспечения. Получить практические навыки использования фреймворка JUnit/NUnit для модульного тестирования программного обеспечения.

5.2. Постановка задачи

Был выдан вариант 14.

Реализовать на языке Java/C# один из классов, спроектированных в лабораторной работе № 1. Методы класса при этом не реализовывать. Разработать для созданного класса набор модульных тестов, включающий тесты для каждого метода. Запустить набор тестов, проанализировать и сохранить результаты. Поочередно реализовать методы класса, выполняя тестирование при каждом изменении программного кода. После того, как весь набор тестов будет выполняться успешно, реализацию классов можно считать завершённой.

5.3. Ход выполнения работы

5.3.1. Описание обязанностей тестового класса

Была разработана следующая спецификация:

- Название тестируемого класса: Matrix.

- Описание: проводится тестирование метода `columnPositionWithLongestSeries` — метода, возвращающего номер столбца, содержащего самую длинную серию одинаковых элементов.
- Название тестового случая: `test_longestSeriesInMiddle_1column`.
- Входные данные: матрица $\begin{pmatrix} 1 & 1 & 1 & 1 \\ -1 & 2 & -1 & -1 \\ 10 & 2 & 10 & 10 \\ 20 & 20 & 20 & 20 \end{pmatrix}$.
- Выходные данные: число 1.

5.3.2. Текст теста

На основе описания в предыдущем пункте был создан тестовый класс `MatrixTest`, вызывающий `public`-методы тестируемого класса. Текст класса представлен в листинге 1.

Листинг 1 — Тестовый драйвер `MatrixTest`

```
package org.cory7666.softwaretestingexample.task1;

import org.junit.jupiter.api.Assertions;
import org.junit.jupiter.api.Test;

public class MatrixTest
{
    @Test
    public void test_longestSeriesInMiddle_1column ()
    {
        Assertions
            .assertEquals(
                1,
                new Matrix(new int[][] { { 1, -1, 10, 20 }, { 1, 2, 2, 20 }, { 1, -1, 10, 20 }, { 1, -1, 10, 20 } })
                    .columnPositionWithLongestSeries());
    }
}
```

Таким образом каждый из методов, имеющих аннотацию «`@Test`», в тестовом классе тестирует один метод `Matrix#columnPositionWithLongestSeries`.

5.3.3. Тестовый запуск и анализ полученных данных

После написания кода программы был запущен тест. Рисунок 5.1 содержит результат проведения теста.

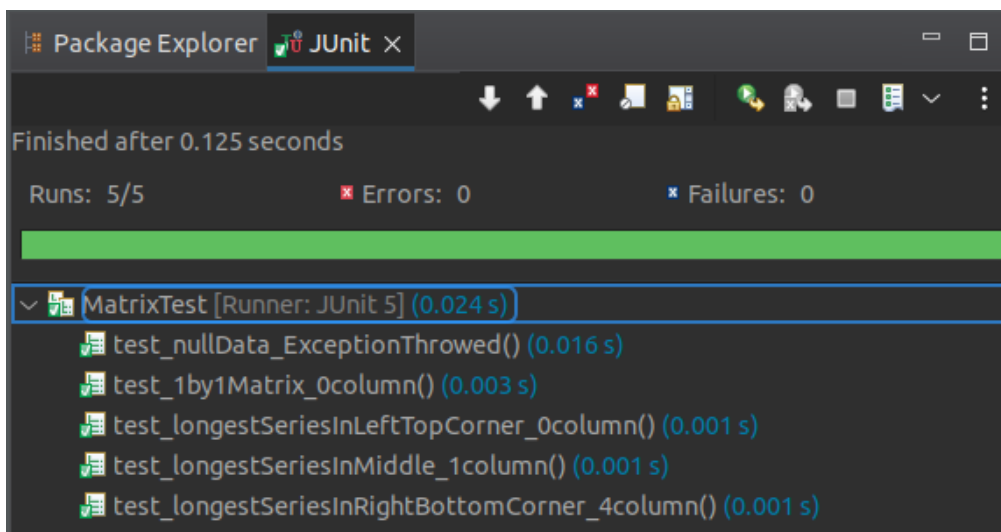


Рисунок 5.1 —

Результат работы JUnit теста

Выводы по результатам работы

При выполнении данной лабораторной работы были получены навыки составления модульных тестов в среде JUnit. Также были получены навыки программирования при работе по методологии TDD. Выявлено, что подобного типа тесты следует применять в случаях разработки программы через тестирование.