4 ЛАБОРАТОРНАЯ РАБОТА №4

«Исследование способов интеграционного тестирования программного обеспечения»

4.1 Цель работы

Исследовать основные принципы интеграционного тестирования программного обеспечения. Приобрести практические навыки организации интеграционных тестов для объектно-ориентированных программ.

4.2 Вариант задания

Выбрать в качестве тестируемого взаимодействие двух или более классов, спроектированных в лабораторных работах №1 – 4. Составить спецификацию тестового случая. Реализовать тестируемые классы и необходимое тестовое окружение на языке C#. Выполнить тестирование с выводом результатов на экран и сохранением в log-файл. Проанализировать результаты тестирования, сделать выводы.

4.3 Ход выполнения работы

4.3.1 Для интеграционного тестирования было выбрано взаимодействие класса MatrixCounter и MatrixCounterBuilder, второй из которых создает объект первого. Код классов, представлен в листинге 4.1.

Листинг 4.1 – Код классов

```
public class MatrixCounter
{
    private MatrixCounter() {}

    public int CountNegativeNumbersInRowsWithZeros(List<List<int>> data)
    {
```

```
        int result = 0;
        foreach (var row in data)
        {
            if (row.Contains(0))
            {
                var amount = row.Count(x => x < 0);
                result += amount;
            }
        }

        return result;
    }

    public class MatrixCounterBuilder()
    {
        public MatrixCounter Build()
        {
            return new MatrixCounter();
        }
    }
}
```

Тест должен проверять создался ли объект необходимого класса. Листинг кода класса теста, выполняющего само тестирование показан в листинге 4.3.

Листинг 4.3 – Код теста

```
namespace IntegrationTests
{
    public class MatrixIntegrationTests
    {
        private readonly ITestOutputHelper output;
        private static readonly string LogFilePath =
@"C:\Users\k_dod\source\repos\5Semestr\tpo\lr4\log.log";

        public MatrixIntegrationTests(ITestOutputHelper output)
        {
```

```
            this.output = output;
            Log.Logger = new LoggerConfiguration()
                .MinimumLevel.Verbose()
                .WriteTo.File(LogFilePath)
                .CreateLogger();

            SelfLog.Enable(Console.Error);
        }

        [Fact]
        public void Create_MatrixCounter_Test()
        {
            // Arrange
            var builder = new MatrixCounter.MatrixCounterBuilder();

            // Act
            MatrixCounter? matrixCounter = builder.Build();

            // Assert
matrixCounter.ShouldNotBeNull().ShouldBeOfType<MatrixCounter>();

            string matrixCounterType = matrixCounter == null ?
"undefined" : matrixCounter.GetType().ToString();
            Log.Information($"Creating matrix counter. Expected type:
{typeof(MatrixCounter)}, Result: {matrixCounterType} ");
            output.WriteLine($"Creating matrix counter. Expected type:
{typeof(MatrixCounter)}, Result: {matrixCounterType} ");

            Log.CloseAndFlush();
        }
    }
}
```

На рисунке 4.1 показан вывод в log-файл, где описаны действия во время прохождения теста. На рисунке 4.2 показан тот же вывод, но только в консоль.

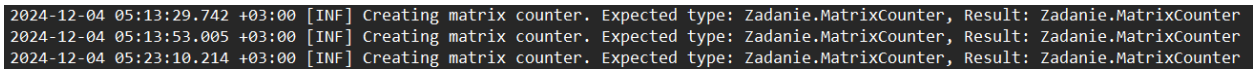

Рисунок 4.1 – Вывод log-файла

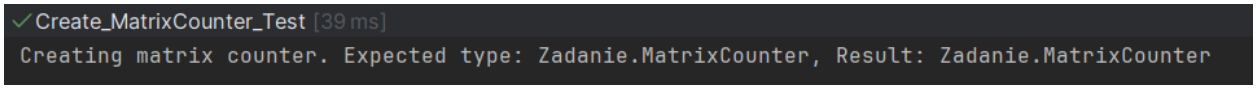

Рисунок 4.2 – Вывод в консоль

Выводы

В ходе лабораторной работы были исследованы основные принципы интеграционного тестирования программного обеспечения. Также были приобретены практические навыки организации интеграционных тестов для объектно-ориентированных программ. В конце выполнения лабораторной работы был написан отчет.