**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 С#. 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 StringChanger и ChangerBuilder, второй из которых создает объект первого. Код класса StringChanger, представлен в листинге 4.1.

Листинг 4.1 – Код класса StringChanger

namespace Laba4;

public class StringChanger

{

public string ChangeSymbols(string str)

{

if (str == null) throw new ArgumentNullException(nameof(str));

return str.Replace('?', '&');

}

}

Класс второго класса ChangerBuilder показан в листинге 4.2.

Листинг 4.2 – Код класса ChangerBuilder

namespace Laba4;

public class ChangerBuilder

{

static StringChanger MakeStringChanger()

{

StringChanger stringChanger = new StringChanger();

return stringChanger;

}

}

Тест должен проверять создался ли объект необходимого класса. Листинг кода класса тестового драйвера, выполняющего само тестирование показан в листинге 4.3. Метод Run() вызывается в главном методе проекта.

Листинг 4.3 – Код тестового драйвера

using System;

using System.Collections.Immutable;

using System.Text;

namespace Laba4

{

public class BuilderTester

{

private static StreamWriter \_log = new StreamWriter("C:\\term5\\тпо\\5\\log.log");

private static ChangerBuilder builder;

private void Init()

{

\_log.WriteLine("Инициализация ChangeBuilder");

Console.WriteLine("Инициализация ChangeBuilder");

builder = new ChangerBuilder();

\_log.WriteLine("ChangerBuilder создан");

Console.WriteLine("Инициализация ChangeBuilder");

}

private void ChangerBuilderTest()

{

Init();

if(builder != null)

{

\_log.WriteLine("ChangerBuilder не равен null");

Console.WriteLine("ChangerBuilder не равен null");

}

else

{

\_log.WriteLine("ChangerBuilder равен null");

Console.WriteLine("Тип созданного объекта - " + builder.GetType().Name);

}

if(builder.GetType().Name == "ChangerBuilder")

{

\_log.WriteLine("Тип созданного объекта - ChangerBuilder");

Console.WriteLine("Тип созданного объекта - ChangerBuilder");

}

else

{

\_log.WriteLine("Тип созданного объекта - " + builder.GetType().Name);

Console.WriteLine("Тип созданного объекта - " + builder.GetType().Name);

}

\_log.Close();

}

public void Run()

{

ChangerBuilderTest();

}

}

}

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.

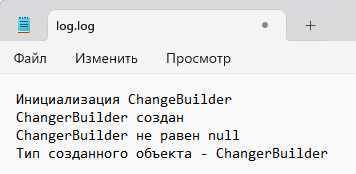


Рисунок 4.1 – Вывод log-файла

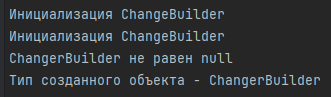


Рисунок 4.1 – Вывод в консоль

**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.