**3 ЛАБОРАТОРНАЯ РАБОТА №3**

**«Исследование способов модульного тестирования программного**

**обеспечения»**

**3.1 Цель работы**

Исследовать основные подходы к модульному тестированию программного обеспечения. Приобрести практические навыки составления модульных тестов для объектно-ориентированных программ.

**3.2 Вариант задания**

Выбрать в качестве тестируемого один из классов, спроектированных в лабораторных работах №№ 1 – 4. Составить спецификацию тестового случая для одного из методов выбранного класса. Реализовать тестируемый класс и необходимое тестовое окружение на языке С#. Выполнить тестирование с выводом результатов на экран и сохранением в log-файл. Проанализировать результаты тестирования, сделать выводы.

**3.3 Ход выполнения работы**

3.3.1 Была составлена спецификация тестового случая для метода, который меняет вопросительные знаки в строке на амперсанты. Код класса, который необходимо протестировать описан в листинге 3.1.

Листинг 3.1 – Код тестового класса

namespace Lab3;

public class StringChanger

{

public string ChangeSymbols(string str)

{

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

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

}

}

Для непосредственного тестирования было необходимо сделать тестовое окружение. Код для абстрактного класса теста показан в листинге 3.2.

Листинг 3.2 – Код абстрактного класса теста

namespace Lab3;

public class StringChanger

{

public string ChangeSymbols(string str)

{

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

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

}

}

Листинг кода класса тестового драйвера, выполняющего само тестирование показан в листинге 3.3.

Листинг 3.3 – Код тестового драйвера

using Lab3;

public class TestDriver: Tester

{

StringChanger example;

public void ChangeStringTester()

{

example = new StringChanger();

Run();

}

private void Run()

{

ChangeStringTest1();

}

private void ChangeStringTest1()

{

string[] tests = {"a", "?", "abcde", "?ab23ef", "1bc?defg", "123bcdefg?", "??ab???c1d???"};

string[] testsExpected = {"a", "&", "abcde", "&ab23ef", "1bc&defg", "123bcdefg&", " &&ab&&&c1d&&&"};

for(int i = 0; i <= 6; i++)

{

Console.WriteLine("Input: " + tests[i] + " Expected: " + testsExpected[i] + " Actual: " + example.ChangeSymbols(tests[i]));

LogMessage("Input: " + tests[i] + " Expected: " + testsExpected[i] + " Actual: " + example.ChangeSymbols(tests[i]));

}

}

static void Main()

{

TestDriver stringTester = new TestDriver();

stringTester.ChangeStringTester();

Log.Close();

}

}

Помимо предыдущих классов был реализован класс для формирования log-файлов, код которого показан в листинге 3.4.

Листинг 3.4 – Код класса для формирования log-файлов

namespace Lab3;

public class Log

{

private static StreamWriter \_log = new StreamWriter("log.log");

public static void Add(string msg)

{

\_log.WriteLine(msg);

}

public static void Close()

{

\_log.Close();

}

}

Сформированный файл содержит выводы всех тестов с их результатами (рисунок 3.1). Можно сделать вывод, что программа работает исправно.

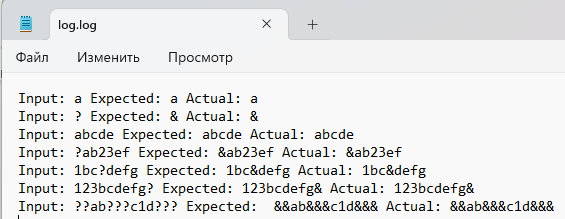


Рисунок 3.1 – Вывод log-файла

**Выводы**

В ходе лабораторной работы были исследованы основные подходы к модульному тестированию программного обеспечения. Также были приобретены практические навыки составления модульных тестов для объектно-ориентированных програм. В конце выполнения лабораторной работы был написан отчет.