**18-КБ-ПР2 Буниф Мажда**

**Отчёт по лабораторной работе №5**

1. **Наименование и цель работы**

**Наименование:** Тестирование методом серого ящика

**Цель работы:** Изучить подход к тестированию методом серого ящика.

1. **Задание на лабораторную работу**

1) Создать класс (в соответствии с вариантом задания из п.5), реализующий преобразование строки из одной системы счисления в другую.

2) Протестировать класс на основе метода серого ящика с использованием средств автоматизации.

3) Составить отчет о результатах проведенного тестирования.

Реализовать класс, преобразующий строку с числом из одной системы счисления в другую. Варианты реализуемых преобразований представлены ниже (в ячейках таблицы указан номер варианта).

По варианту 2: из 3-чной системы в 8-чную.

**3. Текст программы на алгоритмическом языке**

**Class**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab5

{

public class TerToOct

{

public string Convert3To8(string ternary)

{

if (ternary == "" || ternary == "0")

return "0";

else

{

int decimal = 0;

string octal = "";

int decimal = 0;

var chars = ternary.ToCharArray();

int pow = ternary.Length - 1;

for (int i = 0; i < chars.Length; i++)

{

if (Convert.ToInt32(chars[i]) >= 48 && Convert.ToInt32(chars[i]) <= 50)

{

int N;

N = int.Parse(chars[i].ToString());

decimal += N \* (int)Math.Pow(3, pow);

pow--;

}

else throw new ArgumentException("Error! Wrong symbols in strings!");

}

var chars2 = Convert.ToString(decimal).ToCharArray();

while (decimal != 0)

{

int c = decimal % 8;

decimal /= 8;

octal += c;

}

char[] CA = octal.ToCharArray();

Array.Reverse(CA);

return new string(CA);

}

}

}

}

**Решение 2, TestClass**

sing System.Linq;

using System.Text;

using System.Threading.Tasks;

using lab5;

using NUnit.Framework;

namespace TerOctTests

{

[TestFixture]

public class Class1

{

[Test()]

public void NullString()

{

TerToOct c = new TerToOct();

string ex = "0";

string actual = c.Convert3To8(ex);

string expected = "0";

Assert.AreEqual(expected, actual);

}

[Test()]

public void EmptyString()

{

TerToOct c = new TerToOct();

string ex = "";

string actual = c.Convert3To8(ex);

string expected = "0";

Assert.AreEqual(expected, actual);

}

[Test()]

public void DefaultValue()

{

TerToOct c = new TerToOct();

string ex = "22";

string actual = c.Convert3To8(ex);

string expected = "10";

Assert.AreEqual(expected, actual);

}

[Test()]

public void WrongString1()

{

TerToOct c = new TerToOct();

//string ex = "39";

string actual;

Assert.Throws<ArgumentException>(

() => { actual = c.Convert3To8("39"); });

}

[Test()]

public void WrongString2()

{

TerToOct c = new TerToOct();

string actual;

Assert.Throws<ArgumentException>(

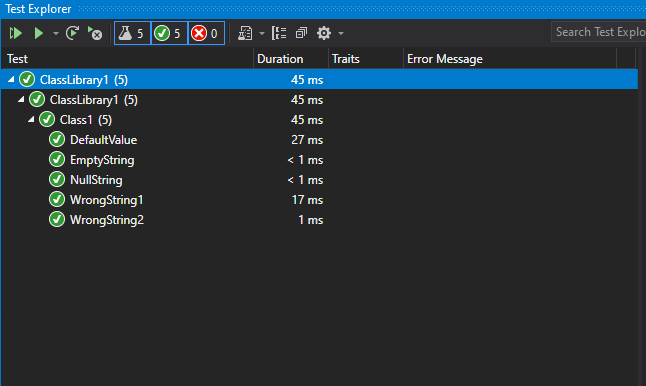
() => { actual = c.Convert3To8("hjeglrjrrjgr"); });

}

}

}

**4. Результат тестов**

****