Федеральное государственное бюджетное образовательное учреждение высшего образования

«Сибирский государственный университет

телекоммуникаций и информатики»

Кафедра ПМиК

Современные технологии программирования 1

Лабораторная работа №2.

Модульное тестирование библиотеки классов на языке С# средствами Visual Studio

Вариант №2

Выполнил: студент 4 курса

Ф. ИВТ, группа: ИП-711

Мартасов И. О.

Проверил: доцент кафедры ПМиК

Зайцев Михаил Георгиевич

Новосибирск, 2020

**Содержание**

1. Задание.
2. УГП и тестовые наборы данных для тестирования функций класса.
3. Исходные тексты программ на языке C#
4. Результаты выполнения модульных тестов.
5. Результаты покрытия разработанного кода тестами.
6. Выводы по выполненной работе.

**Задание**

Разработайте на языке С# класс, содержащий функции в соответствии с вариантом задания.

Разработайте тестовые наборы данных для тестирования функций класса, по критерию С1.

Протестируйте созданный класс с помощью средств автоматизации модульного тестирования Visual Studio.

Проанализируйте результаты выполненных тестов по объёму покрытия тестируемого кода.

Напишите отчёт о результатах проделанной работы.

**Вариант № 2**

1. Поиск максимума из двух чисел
2. Функция получает двумерный массив вещественных переменных A. Отыскивает и возвращает сумму значений компонентов массива, у 74 которых сумма значений индексов равна максимальному значению второго индекса (индекса столбца).
3. Функция получает двумерный массив вещественных переменных A. Отыскивает и возвращает минимальное значение компонентов массива, лежащих на и выше побочной диагонали

**УГП и тестовые наборы данных для тестирования функций класса**

**Функция 1 (Max):**

**(x, y) = {(x = 3, y = 5, m = 5), (x = 5, y = 3, m = 5)}**

**Функция 2 (Count\_max\_index):**

**(mas, n, m) = {(1.0, 2.0, 3.0, 4.0 ,5.0), {(2.0, 3.0, 4.0, 5.0 ,6.0) , {(3.0, 4.0, 5.0, 6.0 ,7.0) , {(4.0, 5.0, 6.0, 7.0 ,8.0) , {(5.0, 6.0, 7.0, 8.0 ,9.0) вх. массив}, n= 5, m=5, sum = 25)}**

**Функция 3 (Count\_up\_diag):**

**(mas, n, m) = {(1.0, 2.0, 3.0, 4.0 ,5.0), {(2.0, 3.0, 4.0, 5.0 ,6.0) , {(3.0, 4.0, 5.0, 6.0 ,7.0) , {(4.0, 5.0, 6.0, 7.0 ,8.0) , {(5.0, 6.0, 7.0, 8.0 ,9.0) вх. массив}, n= 5, m=5, sum = 25)}**

**Исходные тексты программ на языке C#**

**MaxOfTwoElements.cs**

using System;

namespace ClassLibrary1

{

public class MaxOfTwoElements

{

static public int Max(int x, int y)

{

if (x > y)

{

return x;

}

else

{

return y;

}

}

}

}

**CountMaxIndex.cs**

using System;

using System.Collections.Generic;

using System.Text;

namespace ClassLibrary1

{

public class CountMaxIndex

{

public static double Count\_max\_index(double [,]a, int n, int m)

{

double sum = 0;

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

{

for (int j = 0; j < m; j++)

{

if ((i + j) == m - 1)

{

sum += a[i, j];

}

}

}

return sum;

}

}

}

**LowerDiag.cs**

using System;

using System.Collections.Generic;

using System.Text;

namespace ClassLibrary1

{

public class LowerDiag

{

public static double Count\_up\_diag(double[,] a, int n, int m)

{

double sum = 0;

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

{

for (int j = 0; j < m; j++)

{

if ((i + j) <= n - 1)

{

sum += a[i, j];

}

}

}

return sum;

}

}

}

**TestMyClass.cs**

using Microsoft.VisualStudio.TestTools.UnitTesting;

using ClassLibrary1;

namespace UnitTestLibrary1

{

[TestClass]

public class TestMyClass

{

[TestMethod]

public void Max\_3\_5expected\_5()

{

int expected = 5;

int actual = MaxOfTwoElements.Max(3, 5);

Assert.AreEqual(expected, actual);

}

[TestMethod]

public void Max\_3\_5\_expected\_7()

{

int expected = 5;

int actual = MaxOfTwoElements.Max(5, 3);

Assert.AreEqual(expected, actual);

}

[TestMethod]

public void Count\_max\_index\_expected\_20()

{

const int n = 5;

const int m = 5;

double[,] mas = new double[n, m]

{

{1.0, 2.0, 3.0, 4.0, 5.0},

{2.0, 3.0, 4.0, 5.0, 6.0},

{3.0, 4.0, 5.0, 6.0, 7.0},

{4.0, 5.0, 6.0, 7.0, 8.0},

{5.0, 6.0, 7.0, 8.0, 9.0}

};

double expected = 25.0;

double actual = CountMaxIndex.Count\_max\_index(mas, n, m);

Assert.AreEqual(expected, actual);

}

[TestMethod]

public void Count\_lower\_diag\_expected\_10()

{

const int n = 5;

const int m = 5;

double[,] mas = new double[n, m]

{

{1.0, 2.0, 3.0, 4.0, 5.0},

{2.0, 3.0, 4.0, 5.0, 6.0},

{3.0, 4.0, 5.0, 6.0, 7.0},

{4.0, 5.0, 6.0, 7.0, 8.0},

{5.0, 6.0, 7.0, 8.0, 9.0}

};

double expected = 55.0;

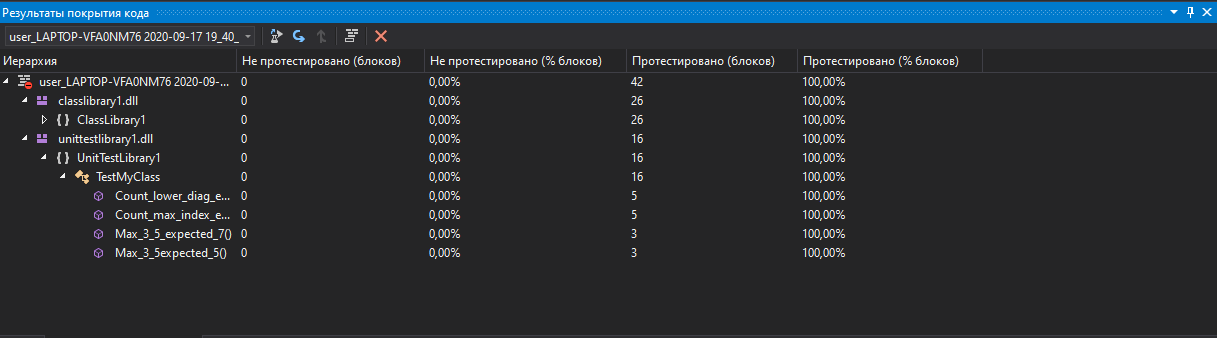
double actual = LowerDiag.Count\_up\_diag(mas, n, m);

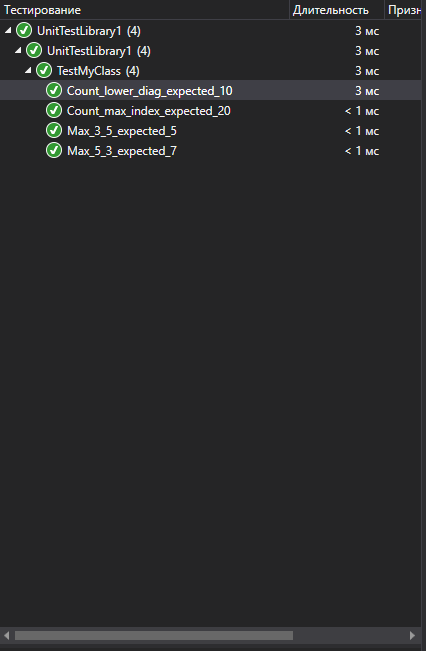
Assert.AreEqual(expected, actual);

}

}

}

**Результаты выполнения модульных тестов и результаты покрытия разработанного кода тестам**



**Выводы по выполненной работе**

Были сформированы практические навыки разработки модульных тестов для библиотек классов C# и выполнения модульного тестирования с помощью средств автоматизации Visual Studio