

Министерство цифрового развития, связи и массовых коммуникаций Российской
Федерации
Федеральное государственное бюджетное образовательное учреждение высшего
образования
«Сибирский государственный университет телекоммуникаций и информатики»
(СибГУТИ)

Лабораторная работа №2
«Модульное тестирование библиотеки классов
на C# средствами Visual Studio»
Вариант №4

Выполнил: студент 4 курса

ИВТ, гр. ИП-113

Шпилев Д. И.

Проверил: старший преподаватель кафедры ПМиК

Агалаков А.А.

Новосибирск, 2024 г.

Цель

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

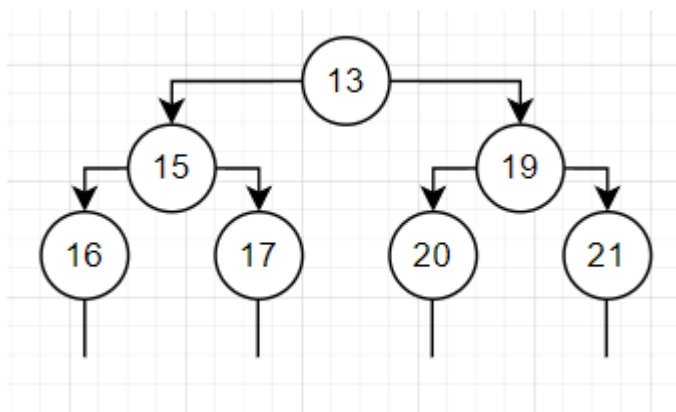
Задание

Разработайте на языке C# класс, содержащий функции в соответствии с вариантом задания. Разработайте тестовые наборы данных для тестирования функций класса, по критерию C1. 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.

1. Поиск максимума из трёх чисел.
2. Функция получает двумерный массив вещественных переменных A. Отыскивает и возвращает произведение значений компонентов массива, у которых сумма значений индексов – чётная.
3. Функция получает двумерный массив вещественных переменных A. Отыскивает и возвращает минимальное значение компонентов массива, лежащих на и ниже главной диагонали

УГП и тестовые наборы данных для тестирования функций класса

```
11  ✓ public static double GetMax(double a, double b, double c)
12  ✓ {
13  ✓     if (a >= b)
14  ✓     {
15  ✓         if (a >= c)
16  ✓             return a;
17  ✓         return c;
18  ✓     }
19  ✓     if (b >= c)
20  ✓         return b;
21  ✓     return c;
22  ✓ }
```



Пути:

(13-15-16) (a = 133.5; b = 133.5; c = 133.5;)

(13-15-17) (a = 133.5; b = 51.2; c = 183.3;)

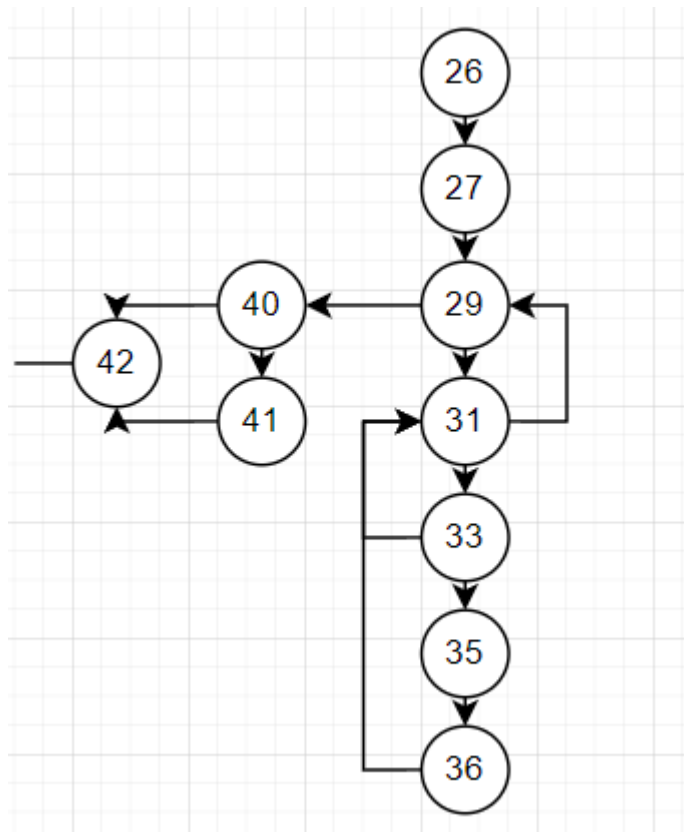
(13-19-20) (a = 3.5; b = 51.2; c = 12.3;)

(13-19-21) (a = 3.5; b = 5; c = 12.3;)

```

24  public static double ProductOfEvenIndexSum(double[,] A)
25  {
26      double product = 1.0;
27      bool found = false;
28
29      for (int i = 0; i < A.GetLength(0); i++)
30      {
31          for (int j = 0; j < A.GetLength(1); j++)
32          {
33              if ((i + j) % 2 == 0)
34              {
35                  product *= A[i, j];
36                  found = true;
37              }
38          }
39      }
40      if (!found)
41          product = 0;
42      return product;
43  }

```



Пути (26-27-29-40-41-42) (26-27-29-40-42) (26-27-29-31-40-41-42) (26-27-29-31-40-42) (26-27-29-31-33-40-41-42) (26-27-29-31-33-40-42) (26-27-29-31-33-35-36-40-41-42) (26-27-29-31-33-35-36-40-42)

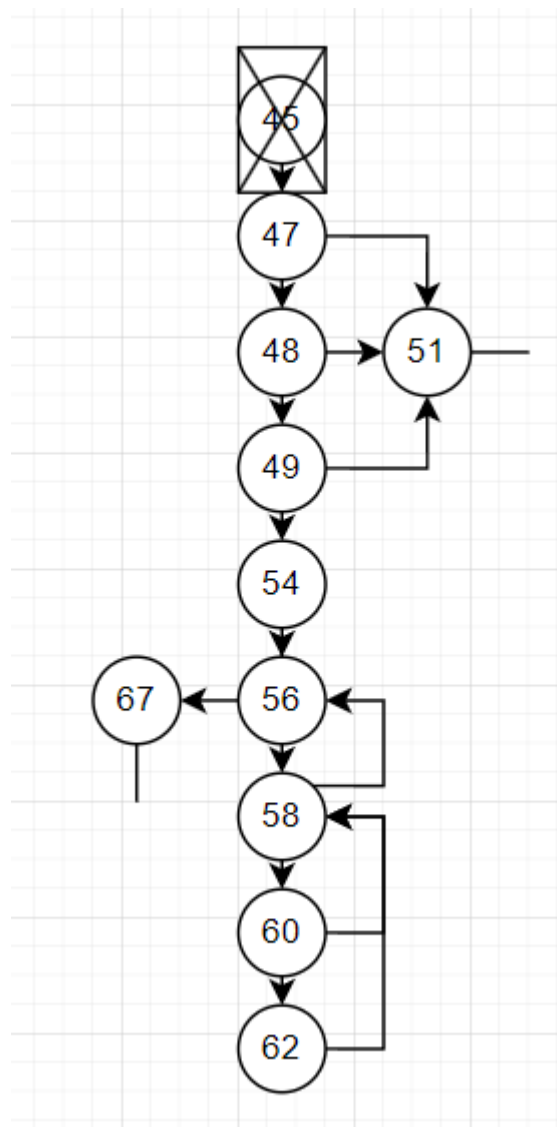
Тесты:

A = { { 5.0 } };

A = { { 2.0, 3.0 }, { 4.0, 5.0 } };

A = { { } };

```
45 public static double FindMinOnAndBelowMainDiagonal(double[,] A)
46 {
47     if (A == null
48         || A.GetLength(0) == 0
49         || A.GetLength(1) == 0)
50     {
51         throw new ArgumentException("Null array or empty");
52     }
53
54     double min = A[0, 0];
55
56     for (int i = 0; i < A.GetLength(0); i++)
57     {
58         for (int j = 0; j <= i; j++)
59         {
60             if (A[i, j] < min)
61             {
62                 min = A[i, j];
63             }
64         }
65     }
66
67     return min;
68 }
```



Для полного тестирования нужно выполнить исключения из пунктов 47, 48, 49 и попасть в пункт 51, пройти путь 56-58-60-62-67.

Для исключения берем `A = null`; пустой массив `A = { }`; и `A = { { } }`;

`A = {{ 5.0, 3.2 }, { 3.5, 5.2 }}`;

`A = { { 5.0 } }`;

Листинг программы:

Program.cs:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Lab2
{
    public class Program
    {
        public static double GetMax(double a, double b, double c)
        {
            if (a >= b)
            {
                if (a >= c)
                    return a;
                return c;
            }
            if (b >= c)
                return b;
            return c;
        }

        public static double ProductOfEvenIndexSum(double[,] A)
        {
            double product = 1.0;
            bool found = false;

            for (int i = 0; i < A.GetLength(0); i++)
            {
                for (int j = 0; j < A.GetLength(1); j++)
                {
                    if ((i + j) % 2 == 0)
                    {
                        product *= A[i, j];
                        found = true;
                    }
                }
            }
            if (!found)
                product = 0;
            return product;
        }

        public static double FindMinOnAndBelowMainDiagonal(double[,] A)
        {
            if (A == null
                || A.GetLength(0) == 0
                || A.GetLength(1) == 0)
            {
                throw new ArgumentException("Null array or empty");
            }

            double min = A[0, 0];

            for (int i = 0; i < A.GetLength(0); i++)
            {
                for (int j = 0; j <= i; j++)
                {
                    if (A[i, j] < min)
                    {

```

```
        min = A[i, j];
    }
}
return min;
}
```


UnitTest1.cs:

```
using Lab2;
using Microsoft.VisualStudio.TestTools.UnitTesting;
using System;

namespace UnitTestLab2
{
    [TestClass]
    public class GetMax
    {
        [TestMethod]
        public void GetMaxC()
        {
            //arrange
            double a = 3.5;
            double b = 5;
            double c = 12.3;
            //act
            double result = Lab2.Program.GetMax(a, b, c);
            //assert
            Assert.AreEqual(c, result);
        }

        [TestMethod]
        public void GetMaxB()
        {
            //arrange
            double a = 3.5;
            double b = 51.2;
            double c = 12.3;
            //act
            double result = Lab2.Program.GetMax(a, b, c);
            //assert
            Assert.AreEqual(b, result);
        }

        [TestMethod]
        public void GetMaxA()
        {
            //arrange
            double a = 133.5;
            double b = 133.5;
            double c = 133.5;
            //act
            double result = Lab2.Program.GetMax(a, b, c);
            //assert
            Assert.AreEqual(a, result);
        }

        [TestMethod]
        public void GetMaxC2()
        {
            //arrange
            double a = 133.5;
            double b = 51.2;
            double c = 183.3;
            //act
            double result = Lab2.Program.GetMax(a, b, c);
            //assert
            Assert.AreEqual(c, result);
        }
    }

    [TestClass]
    public class ProductOfEvenIndexSum
```

```

{
    [TestMethod]
    public void OneElement()
    {
        //arrange
        double[,] A = { { 5.0 } };
        //act
        double result = Lab2.Program.ProductOfEvenIndexSum(A);
        //assert
        Assert.AreEqual(5.0, result);
    }
    [TestMethod]
    public void TwoSizeLenghts()
    {
        //arrange
        double[,] A = {
            { 2.0, 3.0 },
            { 4.0, 5.0 }
        };
        //act
        double result = Lab2.Program.ProductOfEvenIndexSum(A);
        //assert
        Assert.AreEqual(10.0, result);
    }
    [TestMethod]
    public void ZeroElements()
    {
        //arrange
        double[,] A = { { } };
        //act
        double result = Lab2.Program.ProductOfEvenIndexSum(A);
        //assert
        Assert.AreEqual(0, result);
    }
}

[TestClass]
public class FindMinOnAndBelowMainDiagonal
{
    [TestMethod]
    public void OneElement()
    {
        //arrange
        double[,] A = { { 5.0 } };
        //act
        double result = Lab2.Program.FindMinOnAndBelowMainDiagonal(A);
        //assert
        Assert.AreEqual(5.0, result);
    }

    [TestMethod]
    public void ManyElement()
    {
        //arrange
        double[,] A = {
            { 5.0, 3.2 },
            { 3.5, 5.2 }
        };
        //act
        double result = Lab2.Program.FindMinOnAndBelowMainDiagonal(A);
        //assert
        Assert.AreEqual(3.5, result);
    }

    [TestMethod]

```

```

    public void NullArray()
    {
        //arrange
        //act
        //assert
        Assert.ThrowsException<ArgumentException>(() =>
Lab2.Program.FindMinOnAndBelowMainDiagonal(null));
    }

    [TestMethod]
    public void ZeroLengthArray()
    {
        //arrange
        double[,] A = { };
        //act
        //assert
        Assert.ThrowsException<ArgumentException>(() =>
Lab2.Program.FindMinOnAndBelowMainDiagonal(A));
    }

    [TestMethod]
    public void ZeroLengthSubArray()
    {
        //arrange
        double[,] A = { { } };
        //act
        //assert
        Assert.ThrowsException<ArgumentException>(() =>
Lab2.Program.FindMinOnAndBelowMainDiagonal(A));
    }
}

```

Результаты выполнения модульных тестов

| Обозреватель тестов | | | |
|---------------------------------|-------------|----------|---------------------|
| Готово | | | |
| Тестирование | Длительн... | Признаки | Сообщение об ошибке |
| ✔ UnitTestLab2 (12) | 14 мс | | |
| ✔ UnitTestLab2 (12) | 14 мс | | |
| ✔ FindMinOnAndBelowMainDiagonal | 1 мс | | |
| ✔ ManyElement | < 1 мс | | |
| ✔ NullArray | < 1 мс | | |
| ✔ OneElement | < 1 мс | | |
| ✔ ZeroLengthArray | < 1 мс | | |
| ✔ ZeroLengthSubArray | 1 мс | | |
| ✔ GetMax (4) | < 1 мс | | |
| ✔ GetMaxA | < 1 мс | | |
| ✔ GetMaxB | < 1 мс | | |
| ✔ GetMaxC | < 1 мс | | |
| ✔ GetMaxC2 | < 1 мс | | |
| ✔ ProductOfEvenIndexSum (3) | 13 мс | | |
| ✔ OneElement | 13 мс | | |
| ✔ TwoSizeLenghts | < 1 мс | | |
| ✔ ZeroElements | < 1 мс | | |

Результаты покрытия разработанного кода тестами.

| | | | | |
|--|-------------------|------------------|-----------------------|----------------------|
| Результаты покрытия кода | | | | |
| d_shp_KASPENIUM_2024-09-11.18_45_31.co | | | | |
| Hierarchy | | | | |
| d_shp_KASPENIUM_2024-09-11.18_45_31.coverage | Covered (%Blocks) | Covered (Blocks) | Not Covered (%Blocks) | Not Covered (Blocks) |
| lab2.dll | 97,03% | 98 | 2,97% | 3 |
| { } Lab2 | 100,00% | 46 | 0,00% | 0 |
| Program | 100,00% | 46 | 0,00% | 0 |
| GetMax(double, double, double) | 100,00% | 8 | 0,00% | 0 |
| ProductOfEvenIndexSum(double[0...,0...]) | 100,00% | 15 | 0,00% | 0 |
| FindMinOnAndBelowMainDiagonal(double[0...,0...]) | 100,00% | 23 | 0,00% | 0 |
| unittestlab2.dll | 94,55% | 52 | 5,45% | 3 |

Вывод

Были сформированы практические навыки разработки модульных тестов для библиотек классов C# и выполнения модульного тестирования с помощью средств автоматизации Visual Studio, разработан класс на языке C#, содержащий функции в соответствии с вариантом задания, разработаны тестовые наборы данных для тестирования функций класса, по критерию C1, не были получены результаты покрытия кода тестами в связи с ограничением версии Visual Studio.