Министерство науки и высшего образования

Российской Федерации

Федеральное государственное бюджетное образовательное   
учреждение высшего образования

«Новосибирский государственный технический университет»



Кафедра теоретической и прикладной информатики

Лабораторная работа № 4  
по дисциплине «Метрология, качество и тестирование программного обеспечения»

**Разработка и модульное тестирование класса Матрица на C# в среде Visual Studio**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Факультет: | ПМИ |  |  |
| Группа: | ПМИМ-01 |  |  |
| Студент: | Ничипиенко В.А. |  |  |
| Вариант: | 6 |  |  |
| Преподаватель: | Зайцев М.Г. |  |  |

Новосибирск

2021

1. **Цель работы**

Сформировать практические навыки разработки и модульного тестирования классов.

1. **Задание**

* Разработать класс Матрица (Matrix) для операций матричной алгебры в соответствии с предложенной ниже спецификацией требований.
* Разработать тестовые наборы для тестирования методов класса на основе заданной спецификации.
* Выполнить модульное тестирование класса средствами модульного тестирования Visual Studio.
* Выполнить анализ покрытия кода методов тестами.

1. **Тестовые наборы данных для тестирования класса**

|  |  |  |  |
| --- | --- | --- | --- |
| **Тестовый набор** | | | |
| **Тест** | **Исходные данные** | | **Ожидаемый результат** |
| 1) Матрица из 0 строк | Создание матрицы с 0 строк и 2 столбцами | | Пользовательское исключение |
| 2) Матрица с отрицательным значением столбцов | Создание матрицы с 2 строками и -2 столбцами | | Пользовательское исключение |
| 3) Изменение несуществующего элемента матрицы | Матрица 2x2 | Изменить элемент (1, 3) | Пользовательское исключение |
| 4) Получение несуществующего элемента матрицы | Матрица 2x2 | Получить элемент (1, 3) | Пользовательское исключение |
| 5) Эквивалентность матриц | Матрица A =  (1 1,  1 1) | Матрица B =  (1 1,  1 1) | Матрицы эквивалентны |
| 6) Сумма матриц | Матрица A =  (1 1,  1 1) | Матрица B =  (2 2,  2 2) | Матрица C =  (3 3,  3 3) |

1. **Исходные тексты программ**

*Текст программы Program.cs*

using System;

namespace ConsoleAppMatrix

{

class Program

{

static void Main(string[] args)

{

try

{

Matrix a = new Matrix(3, 3);

Matrix b = new Matrix(3, 3);

Matrix c;

for (int i = 0; i < a.I; i++)

{

for (int j = 0; j < a.J; j++)

{

a[i, j] = a.J \* i + j;

}

}

a.Show();

for (int i = 0; i < a.I; i++)

{

for (int j = 0; j < a.J; j++)

{

b[i, j] = a.J \* i + j + 1;

}

}

b.Show();

c = a + b;

c.Show();

}

catch (MyException e)

{

Console.WriteLine(e.Message);

}

}

}

}

*Текст программы Matrix.cs*

using System;

namespace ConsoleAppMatrix

{

public class MyException : Exception

{

public MyException(string s) : base(s) { }

}

public class Matrix

{

private int[,] m;

public int I { get; set; }

public int J { get; set; }

public Matrix(int i, int j)

{

if (i <= 0)

throw new MyException(string.Format("Недопустимое значение i = {0}", i));

if (j <= 0)

throw new MyException(string.Format("Недопустимое значение j = {0}", j));

I = i;

J = j;

m = new int[i, j];

}

public int this[int i, int j]

{

get

{

if (i < 0 | i > I - 1)

throw new MyException(string.Format("Неверное значение i = {0}", i));

if (j < 0 | j > J - 1)

throw new MyException(string.Format("Неверное значение j = {0}", j));

return m[i, j];

}

set

{

if (i < 0 | i > I - 1)

throw new MyException(string.Format("Неверное значение i = {0}", i));

if (j < 0 | j > J - 1)

throw new MyException(string.Format("Неверное значение j = {0}", j)); m[i, j] = value;

}

}

public static Matrix operator +(Matrix a, Matrix b)

{

Matrix c = new Matrix(a.I, a.J);

for (int i = 0; i < a.I; i++)

{

for (int j = 0; j < a.J; j++)

{

c[i, j] = a.m[i, j] + b.m[i, j];

}

}

return c;

}

public static bool operator ==(Matrix a, Matrix b)

{

bool q = true;

for (int i = 0; i < a.I; i++)

{

for (int j = 0; j < a.J; j++)

{

if (a[i, j] != b[i, j])

{

q = false; break;

}

}

}

return q;

}

public static bool operator !=(Matrix a, Matrix b)

{

return !(a == b);

}

public void Show()

{

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

{

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

{

Console.Write("\t" + this[i, j]);

}

Console.WriteLine();

}

Console.WriteLine();

}

public override bool Equals(object obj)

{

return (this as Matrix) == (obj as Matrix);

}

}

}

*Текст программы MatrixTests.cs*

using Microsoft.VisualStudio.TestTools.UnitTesting;

using ConsoleAppMatrix;

namespace MatrixTests

{

[TestClass]

public class MatrixTests

{

[TestMethod]

[ExpectedException(typeof(MyException))]

public void Matrix\_Expected\_MyException\_i()

{

Matrix a = new Matrix(0, 2);

}

[TestMethod]

[ExpectedException(typeof(MyException))]

public void Matrix\_Expected\_MyException\_j()

{

Matrix a = new Matrix(2, -1);

}

[TestMethod]

[ExpectedException(typeof(MyException))]

public void This\_Expected\_MyException\_set\_j()

{

Matrix a = new Matrix(2, 2);

a[1, 3] = 2;

}

[TestMethod]

[ExpectedException(typeof(MyException))]

public void This\_Expected\_MyException\_get\_i()

{

Matrix a = new Matrix(2, 2);

int r = a[3, 1];

}

[TestMethod]

public void Equel()

{

Matrix a = new Matrix(2, 2);

a[0, 0] = 1; a[0, 1] = 1;

a[1, 0] = 1; a[1, 1] = 1;

Matrix b = new Matrix(2, 2);

b[0, 0] = 1;

b[0, 1] = 1;

b[1, 0] = 1;

b[1, 1] = 1;

bool r = a == b;

Assert.IsTrue(r);

}

[TestMethod]

public void Summa()

{

Matrix a = new Matrix(2, 2);

a[0, 0] = 1;

a[0, 1] = 1;

a[1, 0] = 1;

a[1, 1] = 1;

Matrix b = new Matrix(2, 2);

b[0, 0] = 2;

b[0, 1] = 2;

b[1, 0] = 2;

b[1, 1] = 2;

Matrix expected = new Matrix(2, 2);

expected[0, 0] = 3;

expected[0, 1] = 3;

expected[1, 0] = 3;

expected[1, 1] = 3;

Matrix actual = new Matrix(2, 2);

actual = a + b;

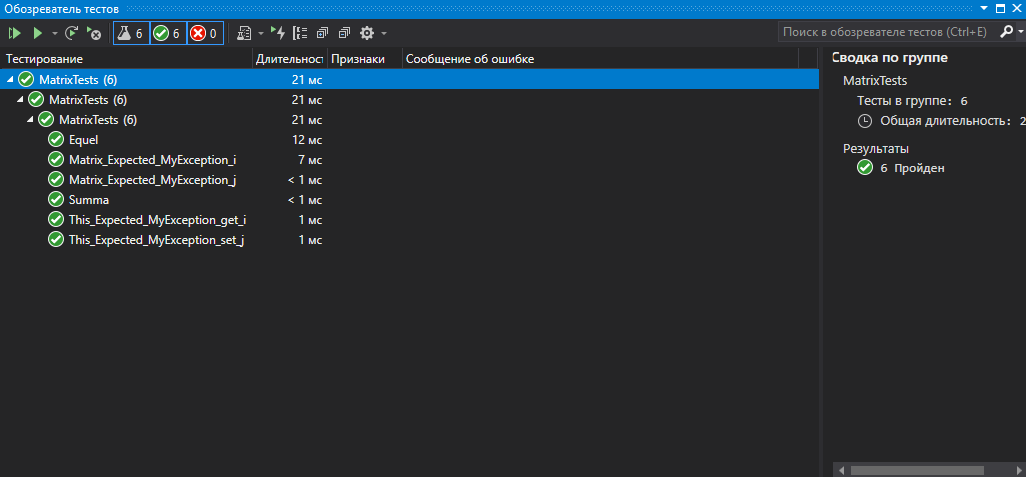
Assert.IsTrue(actual == expected);

}

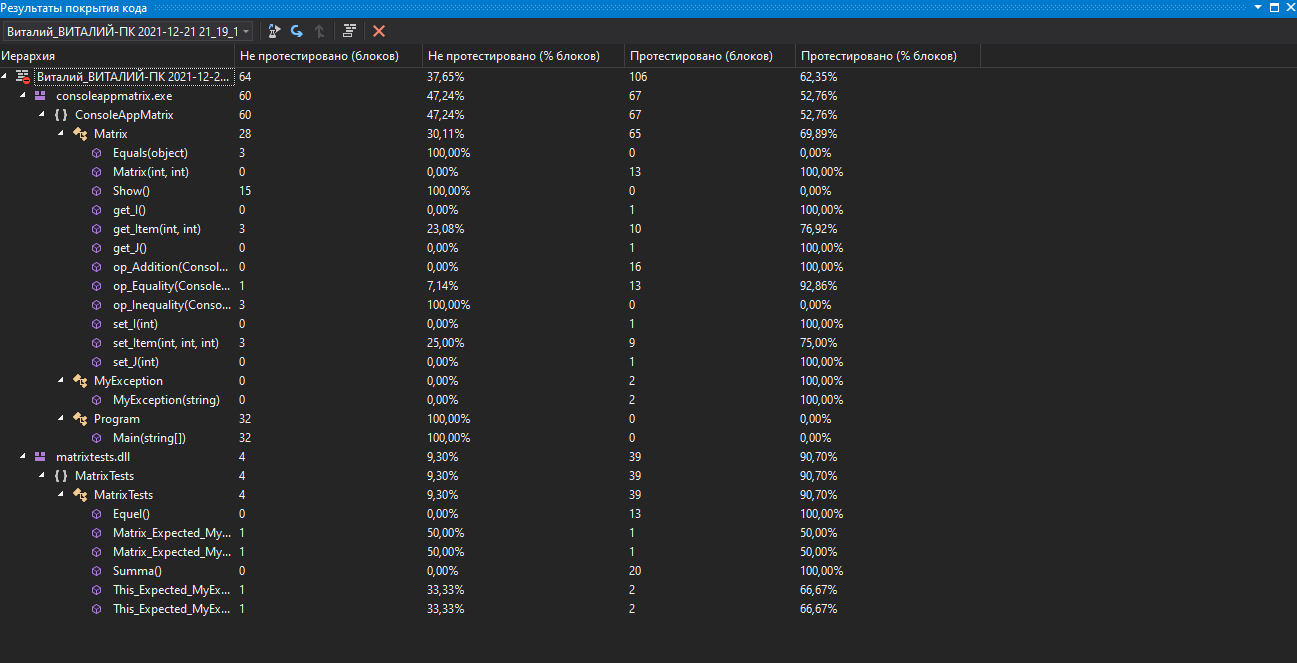
}

}

1. **Результаты выполнения модульных тестов**

****

1. **Результаты покрытия разработанного кода тестами**

****

1. **Выводы по выполненной работе**

В ходе выполнения работы были сформированы практические навыки разработки и модульного тестирования классов.