# Задание

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

Разработайте тестовые наборы данных по критерию С0 для тестирования функций класса.

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

Напишите отчёт о результатах проделанной работы.

## Вариант

1. Функция получает два одномерных массива. В массиве int[] ind хранятся индексы компонентов массива v. Найдите произведение ненулевых компонентов массива v, индексы которых хранятся в массиве ind.
2. Функция получает одномерный массив целых переменных. Вычисляет и возвращает минимальный по значению элемент этого массива и номер его индекса.
3. Функция получает одномерный массив вещественных переменных. В полученном массиве функция переставляет значения компонентов массива в обратном порядке.

# УГП и тестовые наборы данных для тестирования функций класса

|  |  |  |  |
| --- | --- | --- | --- |
| **Номер функции** | **УГП** | **Тестовый набор** | |
|  | **Y** |
| 1 |  |  | 1 |
| 2 |  |  |  |
| 3 |  |  |  |

# Исходные тексты программ на языке C#

Тестируемые функции:

public class ArrayClass

{

// 1

1 public static double calcMultElemsByInd(double[] v, int[] ind)

2 {

3 double mult = 1.0;

4 for (int i = 0;

5 i < ind.Length;

6 i++)

7 {

8 int iv = ind[i];

9 if (v[iv] != 0)

10 {

11 mult \*= v[iv];

12 }

13 }

14 return mult;

15 }

// 2

1 public static int[] findMinElemWithIndex(int[] arr)

2 {

3 int min = arr[0];

4 int indexOfMin = 0;

5 for (int i = 1;

6 i < arr.Length;

7 i++)

8 {

9 if (arr[i] < min)

10 {

11 min = arr[i];

12 indexOfMin = i;

13 }

14 }

15 int[] result = { min, indexOfMin };

16 return result;

17 }

// 3

1 public static void reverseArray(ref double[] arr)

2 {

3 double[] revArr = new double[arr.Length];

4 for (int i = 0;

5 i < arr.Length;

6 i++)

7 {

8 revArr[i] = arr[arr.Length - i - 1];

9 }

10 arr = revArr;

11 }

}

UnitTest1.css:

using Microsoft.VisualStudio.TestTools.UnitTesting;

using System;

using Program1;

namespace UnitTestProject\_1

{

[TestClass]

public class ArrayTest

{

[TestMethod]

public void calcMultElemsByInd\_Test\_C0()

{

// Исходные массивы.

double[] v = new double[] { 1 };

int[] ind = new int[] { 0 };

int expected = 1; // Ожидаемое значение.

var result = ArrayClass.calcMultElemsByInd(v, ind);

Assert.AreEqual(expected, result);

}

[TestMethod]

public void findMinElemWithIndex\_Test\_C0()

{

// Исходные массивы.

int[] x = new int[] { 1, 0 };

int[] expected = new int[] { 0, 1 }; // Ожидаемое значение.

var result = ArrayClass.findMinElemWithIndex(x);

CollectionAssert.AreEqual(expected, result);

}

[TestMethod]

public void reverseArray\_Test\_C0()

{

double[] result = new double[] { 1.0, 2.0 };//Исходный массив.

double[] expected = new double[] { 2.0, 1.0 };//Ожидаемое значение.

ArrayClass.reverseArray(ref result);

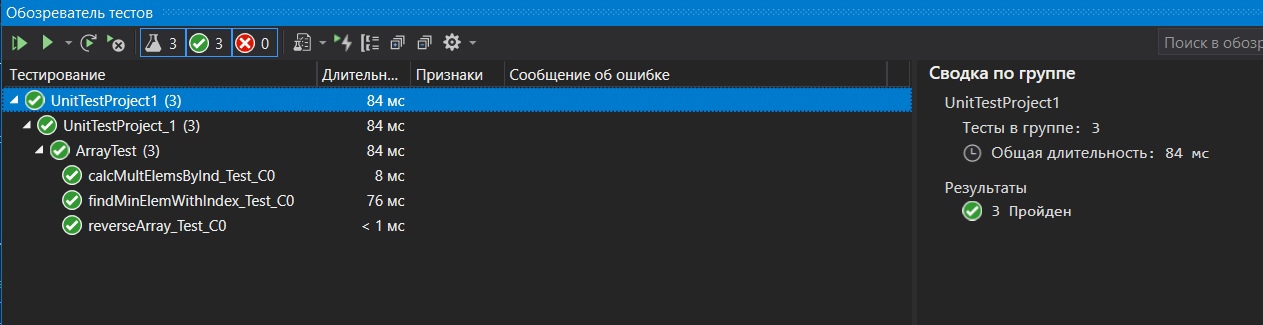
CollectionAssert.AreEqual(expected, result);

}

}

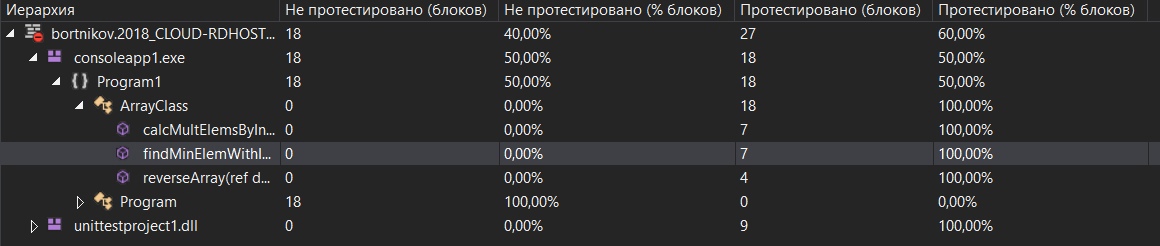
}

# Результаты выполнения модульных тестов



Все тесты выполнились успешно.

# Результаты покрытия разработанного кода тестами



Таким образом, по рисунку выше видно, что написанные тесты по критерию С0 покрыли весь код (100%).

# Вывод

В результате выполнения практической работы был изучен структурный критерий С0. Для его практического применения были разработаны 3 функции на языке C# и написано 3 теста.

Тестирование выполнилось успешно и покрыло весь код.