Федеральное агентство связи

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Сибирский государственный университет

телекоммуникаций и информатики»

**Лабораторная работа №1**

**«Модульное тестирование программ на языке С# средствами Visual Studio»**

**Вариант №4**

Выполнил: студент IV курса

ИВТ, гр. ИП-713

Михеев Н.А.

Проверил: ассистент кафедры

ПМиК

Агалаков А.А.

Новосибирск, 2020 г.

**Цель**

Сформировать практические навыки разработки модульных тестов для тестирования функций классов и выполнения модульного тестирования на языке С# с помощью средств автоматизации Visual Studio.

**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.

1. Функция получает целое число b – основание системы счисления и строку s, содержащую представление целой части числа в системы счисления с основанием b. Функция формирует и возвращает из строки s целое число.
2. Функция получает одномерный массива целых переменных. Вычисляет и возвращает максимальный по значению элемент этого массива и номер его индекса.
3. Функция получает одномерный массив целых переменных. Вычисляет и возвращает максимальное значение среди нечётных элементов массива с нечётными значениями индекса и значение индекса (через параметр)

**Реализация**

В ходе выполнения задания был реализован класс с функциями в соответствии с заданием. О каждой функции подробнее:

public static int ToDeci(int b, string str) – функция получает на вход число, являющееся основанием системы счисления и строку с числом в этой системе. В функции происходит преобразование в десятичную систему счисления по известному алгоритму. Чтобы получить числовое значение, имеется вспомогательная функция:

static int GetValFromStr(char c) – получает на вход символ и переводит его в числовое значение – 0..9 в числа 0..9 и A,B,C,D,E,F… в 10,11,12,13,14,15.. и так далее.

public static Tuple<int, int> ArrIndMax(int[] arr) – функция получает на вход массив целых чисел и возвращает кортеж из максимального элемента и его индекса.

public static Tuple<int, int> ArrOddIndMax(int[] arr) – функция получает на вход массив целых чисел и возвращает кортеж из максимального элемента среди нечетных элементов с нечетным значением индекса и его индекс.

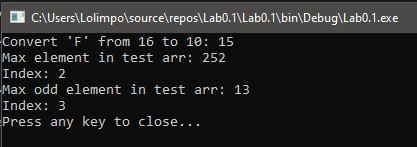


Рис. 1 – Результат проверки работоспособности программы.

Так же были реализованы тесты всех методов по критерию С0 - набор тестов в совокупности должен обеспечить прохождение каждой команды не менее одного раза.

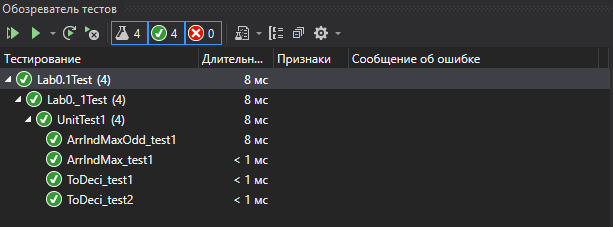
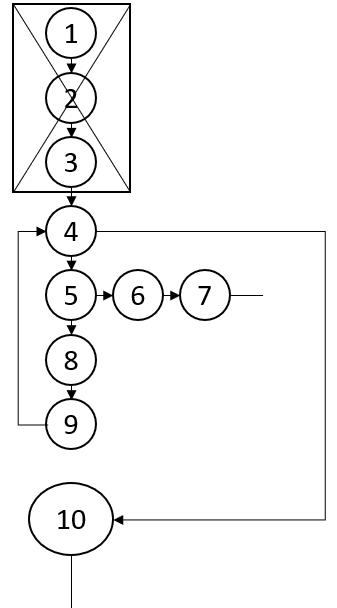


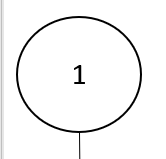
Рис.2 – демонстрация результатов тестирования по критерию С0.

По готовым функциям, были построены управляющие графы программы:

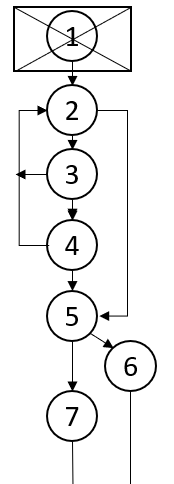
Для public static int ToDeci(int b, string str)**:**



УГП для public static Tuple<int, int> ArrIndMax(int[] arr):



УГП для public static Tuple<int, int> ArrOddIndMax(int[] arr):



**Вывод**

В результате работы над лабораторной работой были сформированы практические навыки разработки функций классов на языке С#, разработка модульных тестов для тестирования функций классов и выполнения модульного тестирования на языке С# с помощью средств автоматизации Visual Studio.

**Листинг программы:**

**Program.cs:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab0.\_1

{

public class Program

{

static int GetValFromStr(char c)

{

if (c >= '0' && c <= '9')

return (int)c - '0';

else

return (int)c - 'A' + 10;

}

public static int ToDeci(int b, string str)

{

int len = str.Length;

int power = 1;

int num = 0;

for (int i = len - 1; i >= 0; i--)

{

if (GetValFromStr(str[i]) >= b)

{

Console.WriteLine("Invalid number");

return -1;

}

num += GetValFromStr(str[i]) \* power;

power \*= b;

}

return num;

}

public static Tuple<int, int> ArrIndMax(int[] arr)

{

return Tuple.Create(arr.Max(), arr.ToList().IndexOf(arr.Max()));

}

public static Tuple<int, int> ArrOddIndMax(int[] arr)

{

int max\_elem = int.MinValue;

for (int i = 1; i < arr.Length; i += 2)

{

if (arr[i] > max\_elem && arr[i] %2 != 0)

{

max\_elem = arr[i];

}

}

if (max\_elem == int.MinValue)

return Tuple.Create(-1, -1);

return Tuple.Create(max\_elem, arr.ToList().IndexOf(max\_elem));

}

static void Main()

{

int[] arr = new int[] { 1, 4, 252, 13, 5 };

Console.WriteLine("Convert 'F' from 16 to 10: " + ToDeci(16, "F"));

var result = ArrIndMax(arr);

Console.WriteLine("Max element in test arr: " + result.Item1 + "\nIndex: " + result.Item2);

result = ArrOddIndMax(arr);

Console.WriteLine("Max odd element in test arr: " + result.Item1 + "\nIndex: " + result.Item2);

#if DEBUG

Console.WriteLine("Press any key to close...");

Console.ReadLine();

#endif

}

}

}

**UnitTest1.cs:**

using System;

using Microsoft.VisualStudio.TestTools.UnitTesting;

using Lab0.\_1;

namespace Lab0.\_1Test

{

[TestClass]

public class UnitTest1

{

[TestMethod]

public void ToDeci\_test1()

{

int expected = 15;

int result = Program.ToDeci(16, "F");

Assert.AreEqual(expected, result);

}

[TestMethod]

public void ToDeci\_test2()

{

int expected = -1;

int result = Program.ToDeci(2, "FFF");

Assert.AreEqual(expected, result);

}

[TestMethod]

public void ArrIndMax\_test1()

{

Tuple<int, int> expected = Tuple.Create(252, 2);

var result = Program.ArrIndMax(new int[] { 1, 4, 252, 14, 5 });

Assert.AreEqual(expected, result);

}

[TestMethod]

public void ArrIndMaxOdd\_test1()

{

Tuple<int, int> expected = Tuple.Create(13, 3);

var result = Program.ArrOddIndMax(new int[] { 1, 4, 252, 13, 5 });

Assert.AreEqual(expected, result);

}

}

}