

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 №1**  
**«Модульное тестирование программ на языке C# средствами**  
**VisualStudio»**  
**Вариант №4**

Выполнил: студент 4 курса

ИВТ, гр. ИП-013

Копытина Т.А.

Проверил: ассистент кафедры

ПМиК Агалаков А.А.

Новосибирск, 2023 г.

## **Цель**

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

## **Задание**

Разработайте на языке C# класс, содержащий функции в соответствии с вариантом задания. Разработайте тестовые наборы данных по критерию C0 для тестирования функций класса. Протестируйте созданный класс с помощью средств автоматизации модульного тестирования VisualStudio. 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

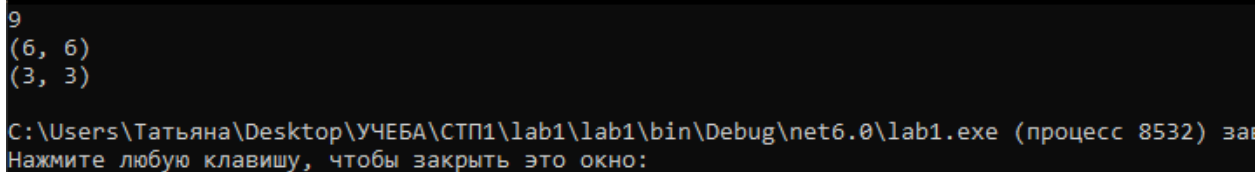
В ходе выполнения задания был реализован класс с функциями в соответствии с заданием. О каждой функции подробнее:

`public static int ToDecimal(int b, strings)` – 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 Sym ToInt(char sym)` – 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>MaxElementId(int[] mas)` – функция получает на вход массив целых чисел и возвращает кортеж из максимального элемента и его индекса.

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



```
9
(6, 6)
(3, 3)

C:\Users\Татьяна\Desktop\УЧЕБА\СТП1\lab1\lab1\bin\Debug\net6.0\lab1.exe (процесс 8532) завершен.
Нажмите любую клавишу, чтобы закрыть это окно:
```

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.

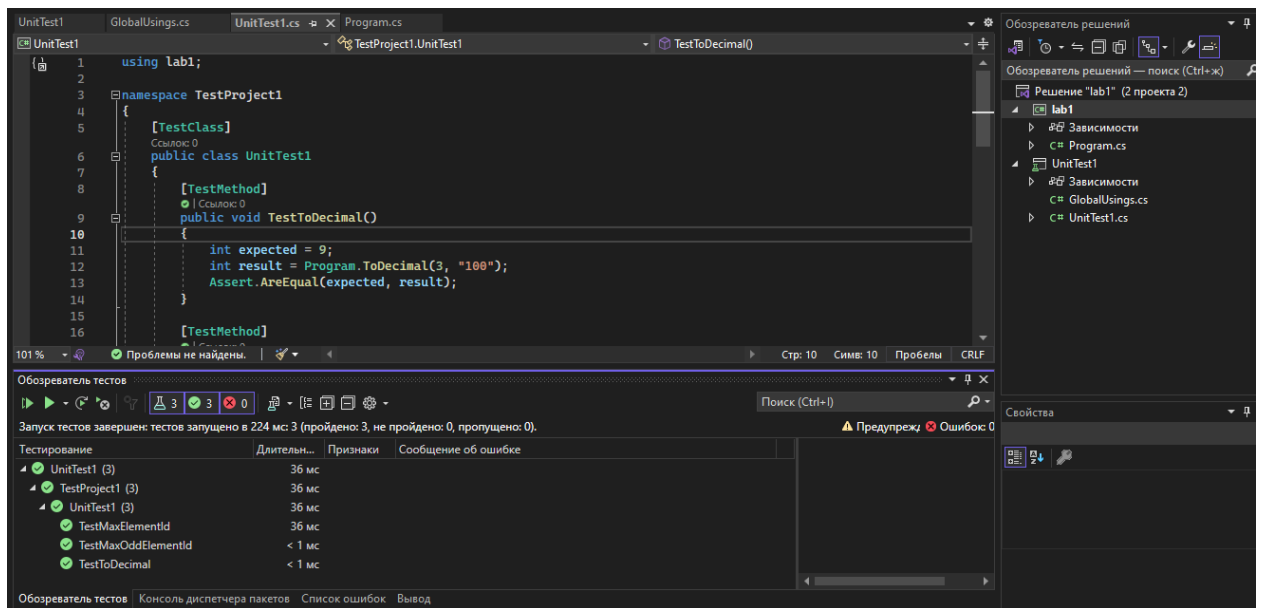
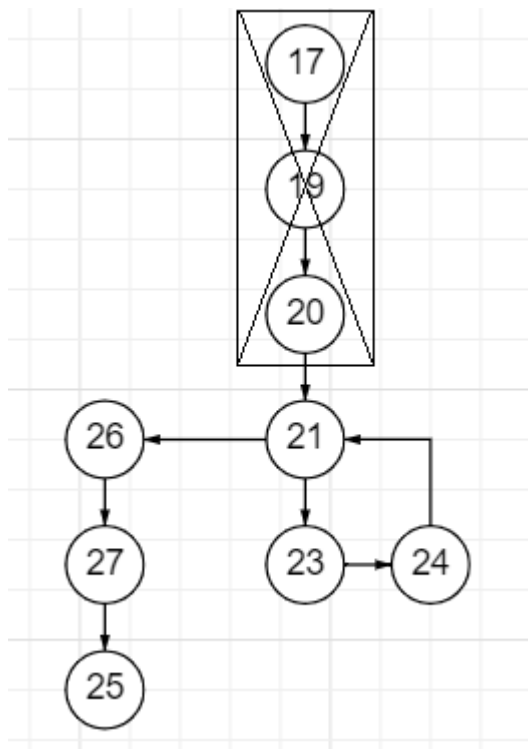


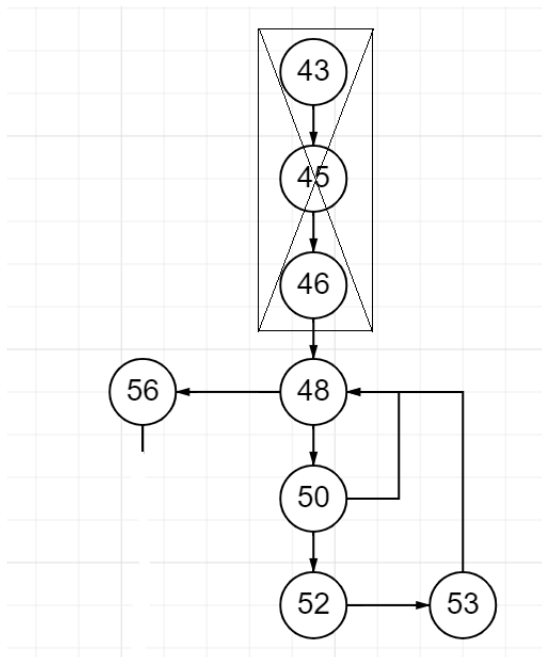
Рис.2 – демонстрация результатов тестирования по критерию C0.

По готовым функциям, были построены управляющие графы программы:

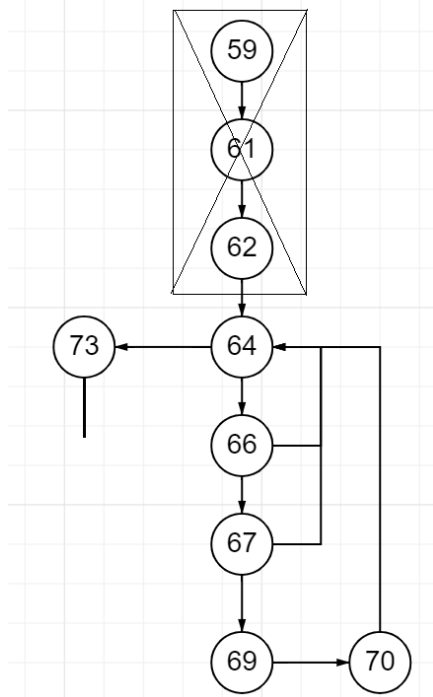
Для public static int ToDecimal (int b, strings):



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



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



## **Вывод**

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

## Листинг программы:

### Program.cs:

```
using System;

namespace lab0._1
{
    public class Program
    {
        static void Main(string[] args)
        {
            int b = 3;
            string s = "100";
            int[] mas = new int[] { 0, 1, -2, 3, 4, -5, 6, -7 };
            Console.WriteLine(ToDecimal(b, s));
            Console.WriteLine(MaxElementId(mas));
            Console.WriteLine(MaxOddElementId(mas));
        }

        public static intToDecimal(int b, string s)
        {
            intbuf = 0;
            intConvertedSymbol = 0;
            for (int i = 0; i <s.Length-1; i++)
            {
                ConvertedSymbol = SymToInt(s[i]);
                buf = (buf + ConvertedSymbol) * b;
            }
            ConvertedSymbol = SymToInt(s[s.Length-1]);
            buf = buf + ConvertedSymbol;
            return buf;
        }

        static intSymToInt(char sym)
        {
            if (sym<= '9' &&sym>= '0')
            {
                return (int)sym - '0';
            }
            else
            {
                return (int)sym - 'A' + 10;
            }
        }

        public static Tuple<int,int>MaxElementId(int[] mas)
        {
            int index = 0;
            intmax_elem = int.MinValue;

            for(int i = 0; i <mas.Length; i++)
            {
                if(mas[i] >max_elem)
                {
                    index = i;
                    max_elem = mas[i];
                }
            }
            return Tuple.Create(max_elem,index);
        }

        public static Tuple<int, int>MaxOddElementId(int[] mas)
```

```

    {
int index = -1;
intmax_elem = int.MinValue;

        for (int i = 1; i <mas.Length; i+=2)
        {
            if (mas[i] >max_elem
                && mas[i] % 2 == 1)
            {
                index = i;
max_elem = mas[i];
            }
        }
        return Tuple.Create(max_elem, index);
    }
}

```



## UnitTest1.cs:

```
using Microsoft.VisualStudio.TestTools.UnitTesting;
using lab0._1;
using System;

namespace TestProject1
{
    [TestClass]
    public class UnitTest1
    {
        [TestMethod]
        public void TestToDecimal()
        {
            int expected = 9;
            int result = Program.ToDecimal(3, "100");
            Assert.AreEqual(expected, result);
        }

        [TestMethod]
        public void TestMaxElementId()
        {
            int[] mas = new int[] { 0, 1, -2, 3, 4, -5, 6, -7 };
            Tuple<int, int> expected = Tuple.Create(6, 6);
            Tuple<int, int> result = Program.MaxElementId(mas);
            Assert.AreEqual(expected, result);
        }

        [TestMethod]
        public void TestMaxOddElementId()
        {
            int[] mas = new int[] { 0, 1, -2, 3, 4, -5, 6, -7 };
            Tuple<int, int> expected = Tuple.Create(3, 3);
            Tuple<int, int> result = Program.MaxOddElementId(mas);
            Assert.AreEqual(expected, result);
        }
    }
}
```