Федеральное государственное бюджетное образовательное учреждение высшего образования

«Сибирский государственный университет телекоммуникаций и информатики»

(СибГУТИ)

Кафедра прикладной математики и кибернетики

Лабораторная работа № 3

по дисциплине «Современные технологии программирования»

Выполнил:

студент группы ИП-712

Алексеев Степан Владимирович

ФИО студента

Работу проверил:

ассистент кафедры Агалаков А.А.

ФИО преподавателя

Новосибирск 2020 г.

Оглавление

[ЗАДАНИЕ 2](#_Toc53401317)

[НАБОРЫ ТЕСТИРОВАНИЯ 3](#_Toc53401318)

[УГП 4](#_Toc53401319)

[ДЕМОНСТРАЦИЯ РАБОТЫ ПРОГРАММЫ 6](#_Toc53401320)

[ВЫВОД 6](#_Toc53401321)

[ПРИЛОЖЕНИЕ 8](#_Toc53401322)

[Листинг 1. Program.cs 8](#_Toc53401323)

[Листинг 2. UnitTest1.cs 10](#_Toc53401324)

# ЗАДАНИЕ

Функция упорядочивает значения переменных x,y,z в порядке убывания их значений, так чтобы x>=y>=z.

Функция получает два положительных целых числа a и b Вычисляет и возвращает их наибольший общий делитель.

Функция получает целое числа a. Формирует и возвращает целое число b из значений чётных разрядов целого числа a. Например: a = 12345, b = 24.

Функция получает двумерный массив вещественных переменных A. Отыскивает и возвращает сумму нечётных значений компонентов массива, лежащих выше главной диагонали

# НАБОРЫ ТЕСТИРОВАНИЯ

Критерий тестирования C2(критерий путей): все условия if должны хотя бы раз выдать как true, так и false. Т.е. каждый возможный путь должен быть пройден хотябы раз.

В методе arrangeXYZ\_inDescendingOrder у массива int[] expected = { 2, 5, 1 }; первый элемент будет в первый раз меньше, во второй раз больше. Т.е. все пути 123456789 и 12345789 будут пройдены.

В методе calculateGCD\_Euclid все пути будут пройдены при вводе 27, 6.

В методе createANumberOfEvenDigitsOf\_a цикл for не будет ни разу пройден при длине строки меньше 2. В остальных случаях всегда будет пройден. Сначала ввожу 123456, потом 7. Это (7) приводит к ошибке, поэтому просто добавил обработку такого исключения.

В методе getSumOfOddDoublesAboveMainDiagonal если длина строк массива будет меньше 2, то всё закончится путём 1,2,3. Ввожу arr = {{2}, {3}, {4}};. В остальных случаях например ветвь 1,2,4,5,12 недостижима. Ветви 1,2,4,5,6,7,8,9,10,11,12 и 1,2,4,5,6,7,8,10,11,12 достижимы при вводе double[,] arr = new double[,] { { 23.7, 10000, 10.4, 56, 56 },

{ 38.5, 5, 89, 190, 343},

{ 13, 564, 100, 767, 23 },

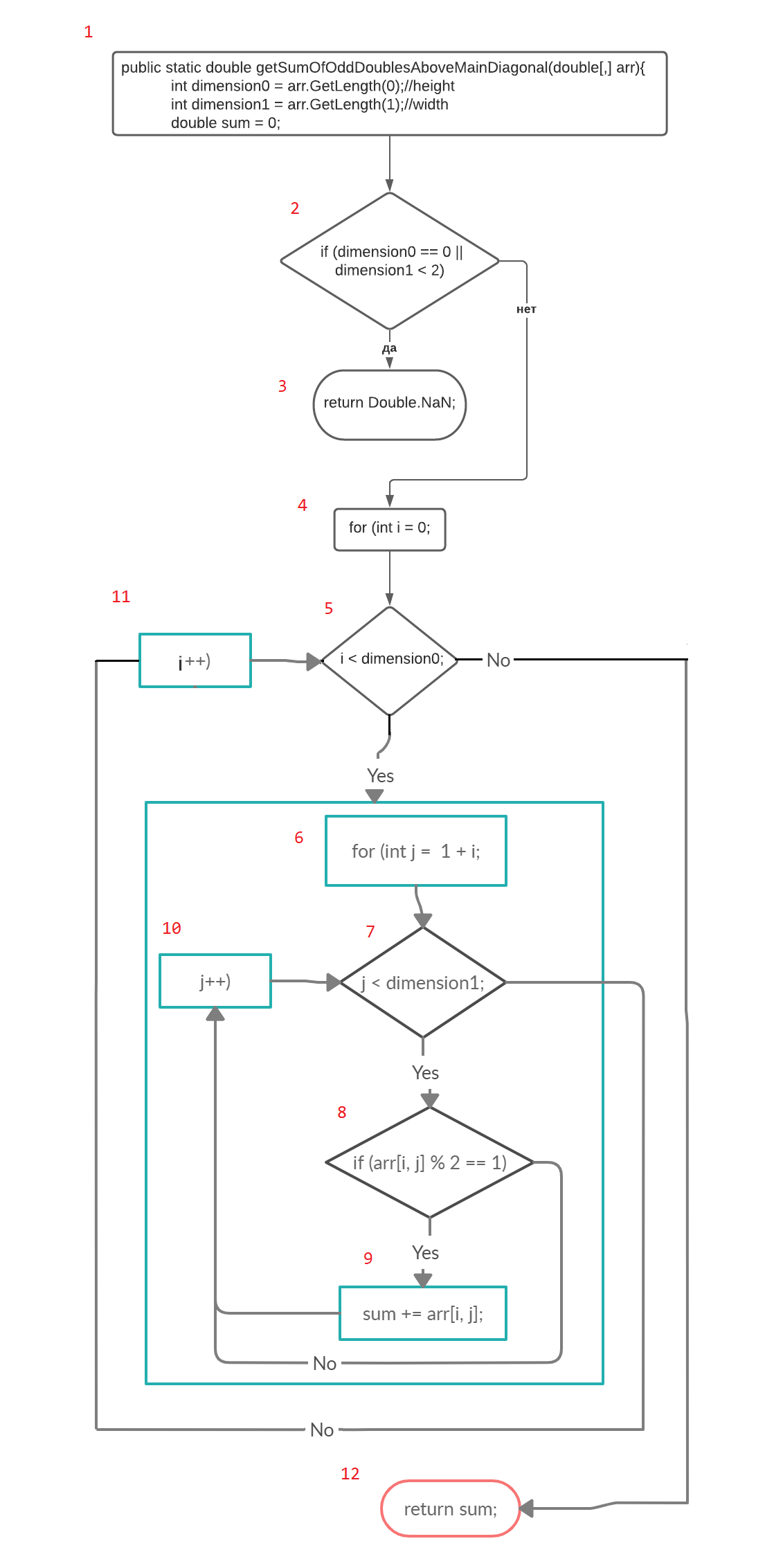
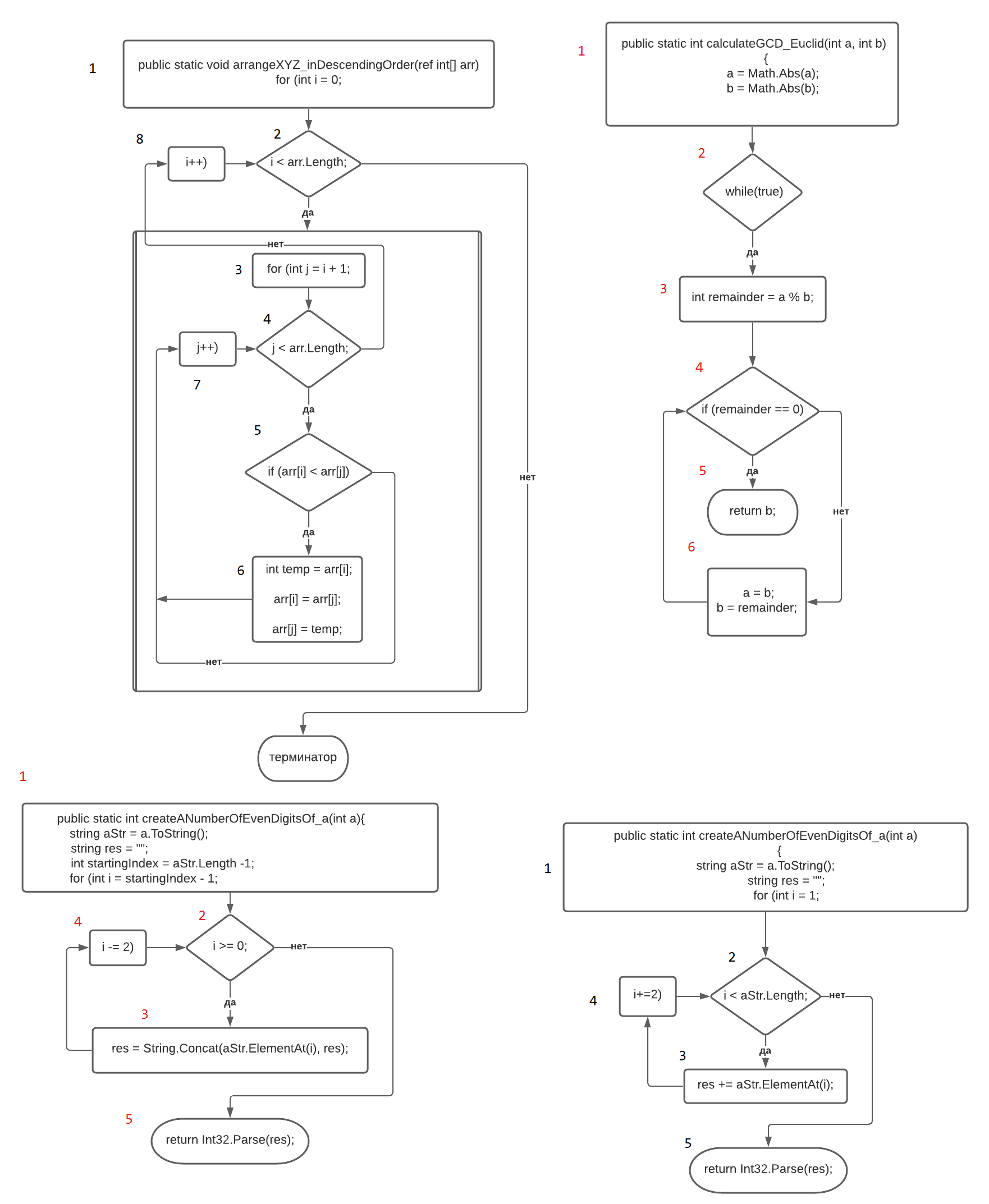
{ 2000, 10, 374, 56, 5 },

{ 38.5, 5, 89, 190, 343},

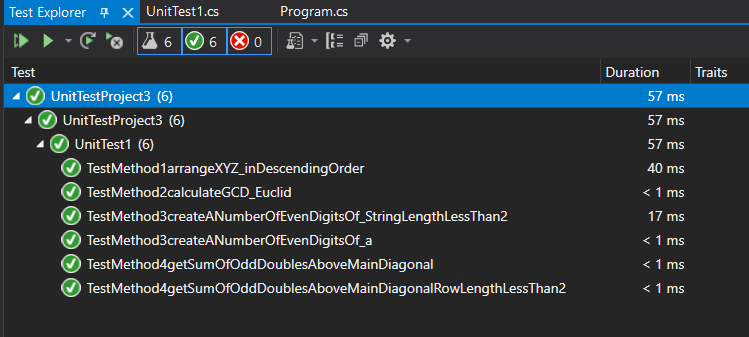
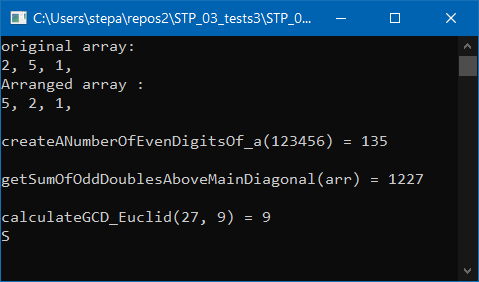
{ 1300, 564, 100, 767, 23000 }

};

# УГП



# ДЕМОНСТРАЦИЯ РАБОТЫ ПРОГРАММЫ



# ВЫВОД

Продвинулся в создании тестов в среде Visual Studio. Разобрался в тестировании по критерию C2. Изучил способы создания UML таблиц с апомощью онлайн сервисов. Изучил возможность связывания программы и тестов для неё в среде Visual Studio.

# 

# ПРИЛОЖЕНИЕ

## Листинг 1. Program.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace STP\_03\_tests3

{

public class Program

{

static void Main(string[] args)

{

/\* int[] expected = { 2, 5, 1 };

Console.WriteLine("original array: ");

for (int i = 0; i < expected.Length; i++)

{

Console.Write(expected[i] + ", ");

}

Console.WriteLine("\nArranged array : ");

arrangeXYZ\_inDescendingOrder(ref expected);

for (int i = 0; i < expected.Length; i++)

{

Console.Write(expected[i] + ", ");

}

Console.WriteLine("\n\ncreateANumberOfEvenDigitsOf\_a(123456) = " + createANumberOfEvenDigitsOf\_a(123456));

double[,] arr = new double[,] { { 23.7, 10000, 10.4, 56, 56 },

{ 38.5, 5, 89, 190, 343},

{ 13, 564, 100, 767, 23 },

{ 2000, 10, 374, 56, 5 },

{ 38.5, 5, 89, 190, 343},

{ 1300, 564, 100, 767, 23000 }

};

Console.WriteLine("\ngetSumOfOddDoublesAboveMainDiagonal(arr) = " + getSumOfOddDoublesAboveMainDiagonal(arr));\*/

//Console.WriteLine("\ncalculateGCD\_Euclid(27, 6) = " + calculateGCD\_Euclid(27, 6));

createANumberOfEvenDigitsOf\_a(7);

Console.ReadLine();

}

public static void arrangeXYZ\_inDescendingOrder(ref int[] arr)

{

for (int i = 0; i < arr.Length; i++)

{

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

{

if (arr[i] < arr[j])

{

int temp = arr[i];

arr[i] = arr[j];

arr[j] = temp;

}

}

}

}

public static int calculateGCD\_Euclid(int a, int b)

{

a = Math.Abs(a);

b = Math.Abs(b);

// Pull out remainders

while (true)

{

int remainder = a % b;

if (remainder == 0) return b;

a = b;

b = remainder;

};

}

public static int createANumberOfEvenDigitsOf\_a(int a)

{

string aStr = a.ToString();

string res = "";

int startingIndex = aStr.Length -1;

int result = Int32.MaxValue;

// startingIndex = (startingIndex % 2 == 1) ? startingIndex-=1 : startingIndex-=2; //если последний индекс(порядковый номер крайней справа цифры)

//чётный, то уменьшаем его на 1. Счёт ведётся с 1 а не с 0.

for (int i = startingIndex - 1; i >= 0; i -= 2)// индекс 1 - это цифра №2 входного числа, если считать цифры начиная с единицы, а не с нуля

{

res = String.Concat(aStr.ElementAt(i), res);//a way to concatenate strings from the left

}

try

{

result = Int32.Parse(res);

}

catch (Exception)

{

Console.WriteLine("Wrong number format");

}

return result;

}

public static double getSumOfOddDoublesAboveMainDiagonal(double[,] arr)

{

int dimension0 = arr.GetLength(0);//height

int dimension1 = arr.GetLength(1);//width

double sum = 0;

if (dimension0 == 0 || dimension1 < 2) return Double.NaN;//the second double in the first row has to exist to enable

//extraction of an element above the main diagonal

for (int i = 0; i < dimension0; i++)//firstly first row

{

for (int j = 1 + i; j < dimension1; j++)//сначала для всей ширины. Потом на 1 меньше(на следующем ряду).

{//Т.о. хоть матрица толстая, хоть высокая смотреть буду только выше главной диагонали

if (arr[i, j] % 2 == 1) sum += arr[i, j];

}

}

return sum;

}

}

}

## Листинг 2. UnitTest1.cs

using System;

using Microsoft.VisualStudio.TestTools.UnitTesting;

using STP\_03\_tests3;

namespace UnitTestProject3

{

[TestClass]

public class UnitTest1

{

[TestMethod]

public void TestMethod1arrangeXYZ\_inDescendingOrder()

{

int[] expected = { 2, 5, 1 };

Program.arrangeXYZ\_inDescendingOrder(ref expected);

int[] actual = { 5, 2, 1 };

CollectionAssert.AreEqual(expected, actual);

}

[TestMethod]

public void TestMethod2calculateGCD\_Euclid()

{

int a = 6, b = 27, res = 3;

int gcdProg = Program.calculateGCD\_Euclid(a, b);

Assert.AreEqual(gcdProg, res);

}

[TestMethod]

public void TestMethod3createANumberOfEvenDigitsOf\_a()

{

int a = 123456, x = 135;

int derivedProg = Program.createANumberOfEvenDigitsOf\_a(a);

Assert.AreEqual(derivedProg, x);

}

[TestMethod]

public void TestMethod3createANumberOfEvenDigitsOf\_StringLengthLessThan2()

{

int a = 7, x = Int32.MaxValue;

int derivedProg = Program.createANumberOfEvenDigitsOf\_a(a);

Assert.AreEqual(derivedProg, x);

}

[TestMethod]

public void TestMethod4getSumOfOddDoublesAboveMainDiagonal()

{

double[,] arr = new double[,] { { 23.7, 10000, 10.4, 56, 56 },

{ 38.5, 5, 89, 190, 343},

{ 13, 564, 100, 767, 23 },

{ 2000, 10, 374, 56, 5 },

{ 38.5, 5, 89, 190, 343},

{ 1300, 564, 100, 767, 23000 }

};

double expected = 1227;

double actual = Program.getSumOfOddDoublesAboveMainDiagonal(arr);

Assert.AreEqual(expected, actual);

}

[TestMethod]

public void TestMethod4getSumOfOddDoublesAboveMainDiagonalRowLengthLessThan2()

{

double[,] arr = new double[,] {{2},{3},{4}

};

double expected = Double.NaN;

double actual = Program.getSumOfOddDoublesAboveMainDiagonal(arr);

Assert.AreEqual(expected, actual);

}

}

}