Федеральное государственное бюджетное образовательное учреждение высшего образования

«Сибирский государственный университет телекоммуникаций и информатики»

(СибГУТИ)

Кафедра прикладной математики и кибернетики

Лабораторная работа № 2

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](#_Toc53325184)

[НАБОРЫ ТЕСТИРОВАНИЯ 3](#_Toc53325185)

[УГП 4](#_Toc53325186)

[ДЕМОНСТРАЦИЯ РАБОТЫ ПРОГРАММЫ 5](#_Toc53325187)

[ВЫВОД 5](#_Toc53325188)

[ПРИЛОЖЕНИЕ 6](#_Toc53325189)

[Листинг 1. Program.cs 6](#_Toc53325190)

[Листинг 2. UnitTest1.cs 8](#_Toc53325191)

# ЗАДАНИЕ

Поиск минимума из двух чисел

Функция получает двумерный массив вещественных переменных A. Отыскивает и возвращает максимальное значение компонентов массива.

Функция получает двумерный массив вещественных переменных A. Отыскивает и возвращает максимальное значение компонентов массива, лежащих на и выше побочной диагонали

# НАБОРЫ ТЕСТИРОВАНИЯ

Критерий тестирования C1(критерий ветвей) для maximumOf2DarrayOnAndAboveSecondaryDiagonal достигается набором когда ~~dimension0 < 1, j<0,~~ max при инициализации больше всех остальных элементов массива, т.е. ~~1) arr[,] = {}, 2) arr[,] = {},~~ 3) arr[,] = {{10, 8, 9}, {1, 2, 3}}; и 10 на выходе и double[,] arr = { { 23.7, 100, 374, 56, 56 },

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

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

{ 200, 10, 374, 56, 5 },

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

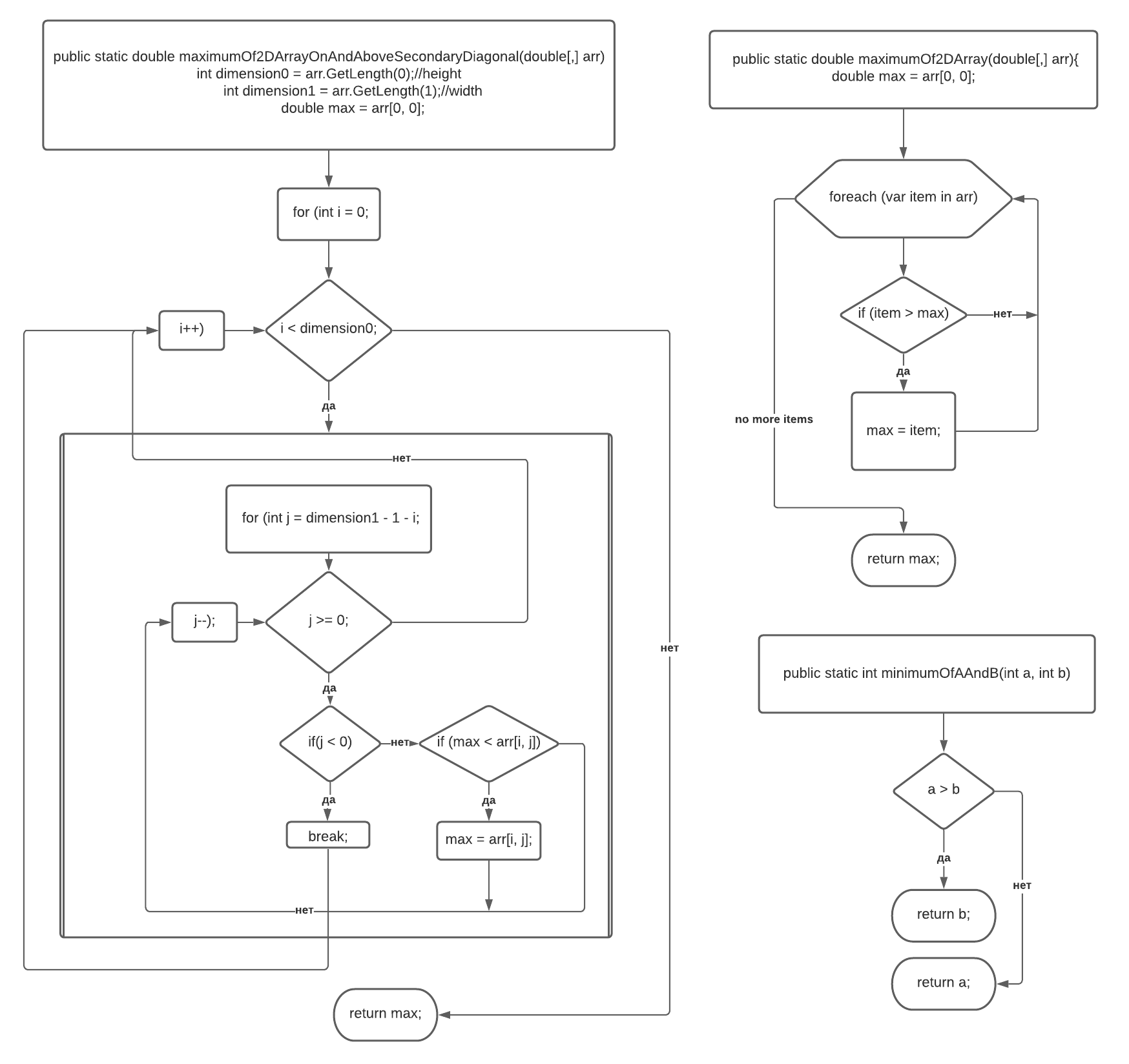
{ 1300, 564, 100, 767, 23 }

}; и 564 на выходе.

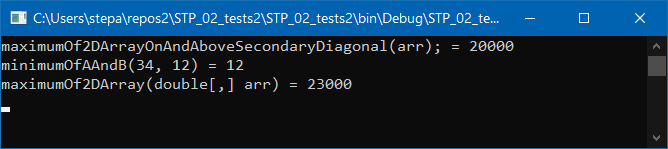
Для метода maximumOf2Darray набор такой же, как для maximumOf2DarrayOnAndAboveSecondaryDiagonal.

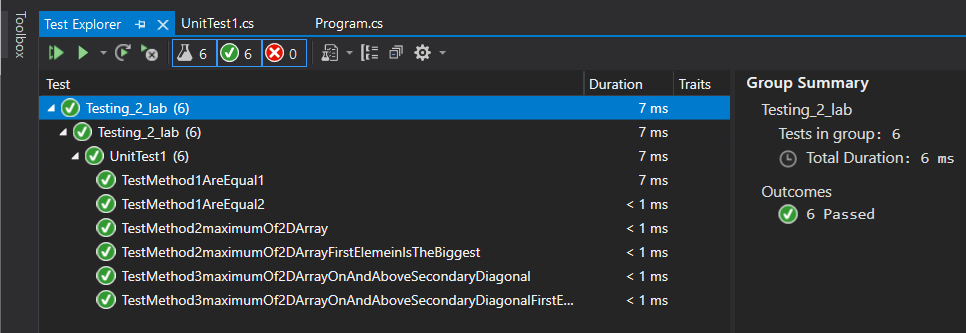
Для метода minimumOfAAndB наборы 2, 1 и 1, 2 приведут соответственно к проходу по ветви a>b -> true, потом по ветви else -> false.

# УГП



# ДЕМОНСТРАЦИЯ РАБОТЫ ПРОГРАММЫ





# ВЫВОД

Научился создавать тесты в среде Visual Studio. Разобрался в различных способах тестирования по критерию C1. Изучил способы создания UML таблиц, в частности foreach loop. Изучил возможность связывания программы и тестов для неё в среде Visual Studio. Разобрался с неоднозначностью определения побочной диагонали матрицы.

# 

# ПРИЛОЖЕНИЕ

## Листинг 1. Program.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace STP\_02\_tests2

{

public class Program

{

static void Main(string[] args)

{

double[,] arr = new double[,] { { 23.7, 10000, 374, 56, 56 },

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

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

{ 20000, 10, 374, 56, 5 },

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

{ 1300, 564, 100, 767, 23000 }

};

Console.WriteLine("maximumOf2DArrayOnAndAboveSecondaryDiagonal(arr); = " + maximumOf2DArrayOnAndAboveSecondaryDiagonal(arr));

Console.WriteLine("minimumOfAAndB(34, 12) = " + minimumOfAAndB(34, 12));

Console.WriteLine("maximumOf2DArray(double[,] arr) = " + maximumOf2DArray(arr));

Console.ReadKey();

}

public static int minimumOfAAndB(int a, int b)

{

if (a > b)

return b;

else return a;

}

public static double maximumOf2DArray(double[,] arr)

{

double max = arr[0, 0];

foreach (var item in arr)

{

if (item > max)

max = item;

}

return max;

}

//Побочной диагональю матрицы называется диагональ, проведённая из левого нижнего угла матрицы в правый верхний(верно для квадратной матрицы)

//Однако в прямоугольной матрице диагональ надо проводить из верхнего правого угла

public static double maximumOf2DArrayOnAndAboveSecondaryDiagonal(double[,] arr)

{

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

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

double max = arr[0, 0];

for (int i = 0; i < dimension0; i++)

{

for (int j = dimension1 - 1 - i; j >= 0; j--)//сначала для всей ширины. Потом на 1 меньше(на следующем ряду).

{//Т.о. хоть матрица толстая, хоть высокая смотреть буду только на побочную диагональ и выше

if (j < 0) break;

if (max < arr[i, j]) max = arr[i, j];

}

}

return max;

}

}

}

## Листинг 2. UnitTest1.cs

using System;

using Microsoft.VisualStudio.TestTools.UnitTesting;

using STP\_02\_tests2;

namespace Testing\_2\_lab

{

[TestClass]

public class UnitTest1

{

[TestMethod]

public void TestMethod1AreEqual1()

{

int a = 2;

int b = 1;

int min = 1;

int programsMin = Program.minimumOfAAndB(a, b);

Assert.AreEqual(programsMin, min);

}

[TestMethod]

public void TestMethod1AreEqual2()

{

int a = 1;

int b = 2;

int min = a;

int programsMin = Program.minimumOfAAndB(a, b);

Assert.AreEqual(programsMin, min);

}

[TestMethod]

public void TestMethod2maximumOf2DArray()

{

double[,] arr = new double[,] { { 23.7d, 10 }, { 38.5d, 5 } };

double max = 38.5d;

double programsMax = Program.maximumOf2DArray(arr);

Assert.AreEqual(programsMax, max);

}

[TestMethod]

public void TestMethod3maximumOf2DArrayOnAndAboveSecondaryDiagonal()

{

double[,] arr = { { 23.7, 100, 374, 56, 56 },

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

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

{ 200, 10, 374, 56, 5 },

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

{ 1300, 564, 100, 767, 23 }

};

double max = 564;

double programsMax = Program.maximumOf2DArrayOnAndAboveSecondaryDiagonal(arr);

Assert.AreEqual(programsMax, max);

}

[TestMethod]

public void TestMethod3maximumOf2DArrayOnAndAboveSecondaryDiagonalFirstElemeinIsTheBiggest()

{

double[,] arr = { { 10, 8, 9 }, { 1, 2, 3 } };

double max = 10;

double programsMax = Program.maximumOf2DArrayOnAndAboveSecondaryDiagonal(arr);

Assert.AreEqual(programsMax, max);

}

[TestMethod]

public void TestMethod2maximumOf2DArrayFirstElemeinIsTheBiggest()

{

double[,] arr = { { 10, 8, 9 }, { 1, 2, 3 } };

double max = 10;

double programsMax = Program.maximumOf2DArrayOnAndAboveSecondaryDiagonal(arr);

Assert.AreEqual(programsMax, max);

}

}

}