

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний аерокосмічний університет ім. М.Є. Жуковського
“Харківський авіаційний інститут”

Кафедра комп’ютерних систем та мереж

Лабораторна робота № 4

“Исследование особенностей использования прикладной
объектноориентированной программы.”

По дисциплине “Технологии программирования”

XAI.503.525B.123. 1705067.180

Виконав студент гр. 525B Пеналоза Г.
(№ групи) (П.І.Б.)

15-03-19
(підпис, дата)

Перевірив ст. викладач каф. 503
(науковий ступінь, вчене звання, посада)

(підпис, дата) (П.І.Б.)

2019

Задание:

1. Создать класс Tmas, основными членами которого должны являться:

- поле FA – двумерный квадратный массив целых чисел,
- методы:
 - * проверки корректности используемого индекса;
 - * генерации элементов массива;
 - * выполнения операций обработки массива в соответствии с вариантом;
 - * чтения массива из файла;
 - * записи массива в файл.

2. Разработать модульные тесты (unit-тесты) для методов класса Tmas.

Вариант 20

20	13	9	21	[-100, 500]	txt
----	----	---	----	-------------	-----

- * Отсортировать элементы массива по возрастанию в строках, представить их в квадратном массиве.
- * Определить дисперсию элементов массива.
- * Элементы массива с суммой индексов кратной 3 умножить на (-g).

Текст программы**Form.cs**

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Lab2._2._4
{
    public partial class Form1 : Form
```

```

{
    public Form1()
    {
        InitializeComponent();

        //Instance of windows form with the results
        public Results results=new Results();

        //Instance of information about the program
        public Information information;

        //Button to show the information about the program
        private void button6_Click(object sender, EventArgs e)
        {
            Information info = new Information();
            info.ShowDialog();
        }

        //Button to show the windows with the result of the each operations
        private void button5_Click(object sender, EventArgs e)
        {
            Tmas._g = (int)numericUpDown1.Value;
            results.Refresh();
            results.Show();
            results.Focus();
        }

        //Method to verify if the user select the correct minimum number of the
        interval
        private void numericUpDown5_ValueChanged(object sender, EventArgs e)
        {
            if (numericUpDown5.Value > numericUpDown6.Value)
            {
                MessageBox.Show("This number can be higher than Max!");
                //numericUpDown5.Value -= 1;
                numericUpDown5.Value = numericUpDown6.Value - 1;
            }
            else
            {
                if (numericUpDown6.Value == 0 || numericUpDown6.Value <=
                numericUpDown5.Value)
                {
                    numericUpDown6.Value = numericUpDown5.Value + 1;
                }
            }
        }
    }
}

```

```

    }
}

```

//Method to verify if the user select the correct maximum number of the interval

```

private void numericUpDown6_ValueChanged(object sender, EventArgs e)
{
    if (numericUpDown6.Value <= numericUpDown5.Value)
    {
        MessageBox.Show("This number can be less than Min!");
        numericUpDown6.Value = numericUpDown5.Value + 1;
    }
}

```

//Button to generate the array with random numbers

```

private void button1_Click(object sender, EventArgs e)
{
    if (numericUpDown4.Value!=0&&numericUpDown6.Value!=
numericUpDown5.Value)
    {
        Tmas.GenerarElementos((int)numericUpDown4.Value,
(int)numericUpDown5.Value,(int)numericUpDown6.Value);
        results.Show();
        results.Focus();
        button5.Enabled = true;
        button3.Enabled = true;
    }
    else
    {
        MessageBox.Show("Incorrect parameters!");
    }
}

```

//Button to write the array in the text file

```

private void button3_Click(object sender, EventArgs e)
{
    SaveFileDialog file_input_array = new SaveFileDialog();
    file_input_array.Filter = "txt files (*.txt)|*.txt|All files (*.*)|*.*";
    file_input_array.FilterIndex = 1;
    file_input_array.RestoreDirectory = true;
    file_input_array.Title = "File to save input array";
    if (file_input_array.ShowDialog()==DialogResult.OK)
    {
        Tmas.Write_Input_Array(file_input_array.FileName);
    }
}

```

```
}
```

```
//Button to open a text file with an array
```

```
private void button2_Click(object sender, EventArgs e)
{
    try
    {
        OpenFileDialog file_with_array = new OpenFileDialog();
        file_with_array.Filter = "txt files (*.txt)|*.txt|All files (*.*)|*.*";
        file_with_array.FilterIndex = 1;
        file_with_array.Title = "File with the array";
        if (file_with_array.ShowDialog() == DialogResult.OK)
        {
            Tmas.Read_File_Array(file_with_array.FileName);
            //MessageBox.Show("-"+file_with_array.FileName+"-");
            results.Refresh();
            results.Show();
            results.Focus();
        }
    }
    catch
    {
        MessageBox.Show("Incorrect format of the file!");
    }
}
```

```
//General method to select and do the operations of my variant
```

```
private void radioButton1_Click(object sender, EventArgs e)
{
    RadioButton rd = (RadioButton)sender;
    if (Tmas.FA!=null)
    {
        if (rd==radioButton1)
        {
            Tmas.op = 1;
        }
        if (rd==radioButton2)
        {
            Tmas.op = 2;
        }
        if (rd==radioButton3)
        {
            Tmas.op = 3;
        }
    }
}
```

```

else
{
    rd.Checked = false;
    MessageBox.Show("Array is not entered!");
}
}

```

//Write a text file with an array entered and the results of the operations until had done all the operations

```

private void button4_Click(object sender, EventArgs e)
{
    SaveFileDialog file_input_array = new SaveFileDialog();
    file_input_array.Filter = "txt files (*.txt)|*.txt|All files (*.*)|*.*";
    file_input_array.FilterIndex = 1;
    file_input_array.RestoreDirectory = true;
    file_input_array.Title = "File to save results!";
    if (file_input_array.ShowDialog() == DialogResult.OK)
    {
        Tmas.SaveResults(file_input_array.FileName);
    }
}

```

//Enable the button to save the array and results when all the operations had done.

```

private void Form1_Activated(object sender, EventArgs e)
{
    if (Tmas.op1 && Tmas.op2 && Tmas.op3 == true)
        button4.Enabled = true;
}
}

```

Information.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

```

```

namespace Lab2._2._4

```

```

{
    public partial class Information : Form
    {
        public Information()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            this.Close();
        }
    }
}

```

Results.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

```

```

namespace Lab2._2._4
{
    public partial class Results : Form
    {
        public Results()
        {
            InitializeComponent();
        }
    }
}

```

//In this method the program is going to put the array entered into the datagridview and the second part is going to execute the selected operation and show into the datagridview and/or textbox

```

private void Results_Activated(object sender, EventArgs e)
{
    textBox1.Text = "";
    dataGridView1.Rows.Clear();
    dataGridView2.Rows.Clear();
    dataGridView1.ColumnCount = Tmas.FA.GetLength(0);
}

```

```

dataGridView2.ColumnCount = Tmas.FA.GetLength(0);
dataGridView1.AutoSizeColumnsMode =
DataGridViewAutoSizeColumnsMode.Fill;
dataGridView2.AutoSizeColumnsMode =
DataGridViewAutoSizeColumnsMode.Fill;
for (int c = 0; c < Tmas.FA.GetLength(0); c++)
{
    string[] row = new string[Tmas.FA.GetLength(1)];
    for (int i = 0; i < Tmas.FA.GetLength(1); i++)
    {
        row[i] = Tmas.FA[c, i].ToString();
    }
    dataGridView1.Rows.Add(row);
    dataGridView2.Rows.Add(row);
}
label1.Text = "N =
"+Tmas.FA.GetLength(0)+"x"+Tmas.FA.GetLength(1)+"; Min = "+Tmas._min+";
Max = "+Tmas._max;
switch (Tmas.op)
{
    case 1:
        Tmas.op1 = true;
        Tmas.OP1();
        dataGridView2.Rows.Clear();
        for (int c = 0; c < Tmas.FA.GetLength(0); c++)
        {
            string[] row = new string[Tmas.FA.GetLength(1)];
            for (int i = 0; i < Tmas.FA.GetLength(1); i++)
            {
                row[i] = Tmas.FA_sort[c, i].ToString();
            }
            dataGridView2.Rows.Add(row);
            textBox1.Text = "Array is sorted ==>";
        }
        break;
    case 2:
        Tmas.op2 = true;
        Tmas.OP2();
        textBox1.Text = "Дисперсию элементов массива:" +
Tmas.var.ToString();
        break;
    case 3:
        Tmas.op3 = true;
        Tmas.OP3();
        dataGridView2.Rows.Clear();

```



```

        for (int c = 0; c < Tmas.FA.GetLength(0); c++)
        {
            string[] row = new string[Tmas.FA.GetLength(1)];
            for (int i = 0; i < Tmas.FA.GetLength(1); i++)
            {
                row[i] = Tmas.new_elements[c, i].ToString();
            }
            dataGridView2.Rows.Add(row);
            textBox1.Text = "Элементы массива с суммой индексов кратной
3 умножить на (-g) ==>";
        }
        break;
    }
}

//Button to hide the form
private void button1_Click(object sender, EventArgs e)
{
    this.Visible = false;
}
}
}

```

Tmas.cs

```

using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Lab2._2._4
{
    public class Tmas
    {
        public static int[,] FA;//Array entered
        public static int _min;//min interval
        public static int _max;//max interval
        public static int op=0;//Operation selected

        public static int[,] FA_sort;//1 Operation => array with sort rows
        public static double var;//2 Operation => Определить дисперсию элементов
        массива
    }
}

```

public static int[,] new_elements;//3 Operation => Элементы массива с суммой индексов кратной 3 умножить на (-g).

public static int _g;//3 Operation => g value of the numericUpDown

//Variables to change to true when they are execute to enable the save button results

public static bool op1 = false;

public static bool op2 = false;

public static bool op3 = false;

//Method to save the array entered and the results of each operations of my variant

public static void SaveResults(string file)

{

FileStream fs = new FileStream(file, FileMode.Create, FileAccess.Write);

if (fs != null)

{

StreamWriter wr = new StreamWriter(fs);

if (FA != null)

{

wr.WriteLine("Input array");

for (int i = 0; i < FA.GetLength(0); i++)

{

for (int c = 0; c < FA.GetLength(1); c++)

{

wr.Write(FA[i, c]+" ");

}

wr.Write("\r\n");

}

wr.Write("\r\n");

}

wr.WriteLine("Operation 1:");

if (FA_sort!=null)

{

wr.WriteLine("Sort array by rows");

for (int i = 0; i < FA.GetLength(0); i++)

{

for (int c = 0; c < FA.GetLength(1); c++)

{

wr.Write(FA_sort[i, c]+" ");

}

wr.Write("\r\n");

}

wr.Write("\r\n");

}

```

        wr.WriteLine("Operation 2:");
        wr.WriteLine("дисперсию элементов массива: "+var);
        wr.Write("\r\n");
        wr.WriteLine("Operation 3:");
        if (new_elements != null)
        {
            wr.WriteLine("Элементы массива с суммой индексов кратной 3
умножить на (-g).");
            for (int i = 0; i < FA.GetLength(0); i++)
            {
                for (int c = 0; c < FA.GetLength(1); c++)
                {
                    wr.Write(new_elements[i, c]+" ");
                }
                wr.Write("\r\n");
            }
            wr.Write("\r\n");
        }
        wr.Flush();
        wr.Close();
        fs.Close();
    }
}

```

```

//Method to generate and save random numbers into array
public static void GenerarElementos(int _l,int min,int max)
{
    _min = min;
    _max = max;
    FA = new int[_l,_l];
    Random rm = new Random();
    for (int c=0;c<_l;c++)
    {
        for (int i=0;i<_l;i++)
        {
            FA[c, i] = rm.Next(min,max);
        }
    }
}

```

```

//Write the array entered into the text file
public static void Write_Input_Array(string file)
{
    FileStream fs = new FileStream(file, FileMode.Create, FileAccess.Write);
    if (fs != null)

```

```

{
    StreamWriter wr = new StreamWriter(fs);
    if (FA != null)
    {
        for (int i = 0; i < FA.GetLength(0); i++)
        {
            for (int c=0;c<FA.GetLength(1);c++)
            {
                wr.WriteLine(FA[i,c]);
            }
        }
    }
    else { MessageBox.Show("Массив не создан!", "Сообщение"); }
    wr.Flush();
    wr.Close();
    fs.Close();
}
}

//Read an array from textfile with the correct format
public static void Read_File_Array(string file)
{
    try
    {
        FileStream fs = new FileStream(file, FileMode.Open, FileAccess.Read);
        if (fs != null)
        {
            StreamReader r = new StreamReader(fs);
            string values = r.ReadToEnd();
            int size = 0;
            for (int c = 0; c < values.Length; c++)
            {
                if (values[c] == '\n')
                    size = size + 1;
            }
            size = (int)Math.Sqrt(size);
            FA = new int[size, size];
            fs.Position = 0;
            for (int k = 0; k < size; k++)
            {
                for (int i = 0; i < size; i++)
                {
                    FA[k, i] = Convert.ToInt32(r.ReadLine());
                }
            }
        }
    }
}

```

```

    }
    r.Close();
    fs.Close();
    //ADD RANGE
}
}
catch
{
    MessageBox.Show("The text file does not have the correct format!");
}
}

```

//First operation of my variant: Отсортировать элементы массива по возрастанию в строках, представить их в квадратном массиве.

```

public static void OP1()
{
    FA_sort = (int[,])FA.Clone();
    int t;
    for (int c=0;c<FA.GetLength(0);c++)
    {
        for (int i=1;i<FA.GetLength(1);i++)
        {
            for (int ii=FA.GetLength(1)-1;ii>=i;ii--)
            {
                if (FA_sort[c,ii-1]>FA_sort[c,ii])
                {
                    t = FA_sort[c,ii-1];
                    FA_sort[c, ii - 1] = FA_sort[c,ii];
                    FA_sort[c,ii] = t;
                }
            }
        }
    }
}

```

//Second operation of my variant: Определить дисперсию элементов массива.

```

public static void OP2()
{
    double med, sum = 0, sum1=0;
    for (int c=0;c<FA.GetLength(0);c++)
    {
        for (int i=0;i<FA.GetLength(1);i++)
        {
            sum += FA[c,i];

```

```

        sum1 += Math.Pow(FA[c, i], 2);
    }
}
med = sum / Math.Pow(FA.GetLength(0), 2);
var = (sum1 / (Math.Pow(FA.GetLength(0), 2))) - Math.Pow(med, 2);
}

```

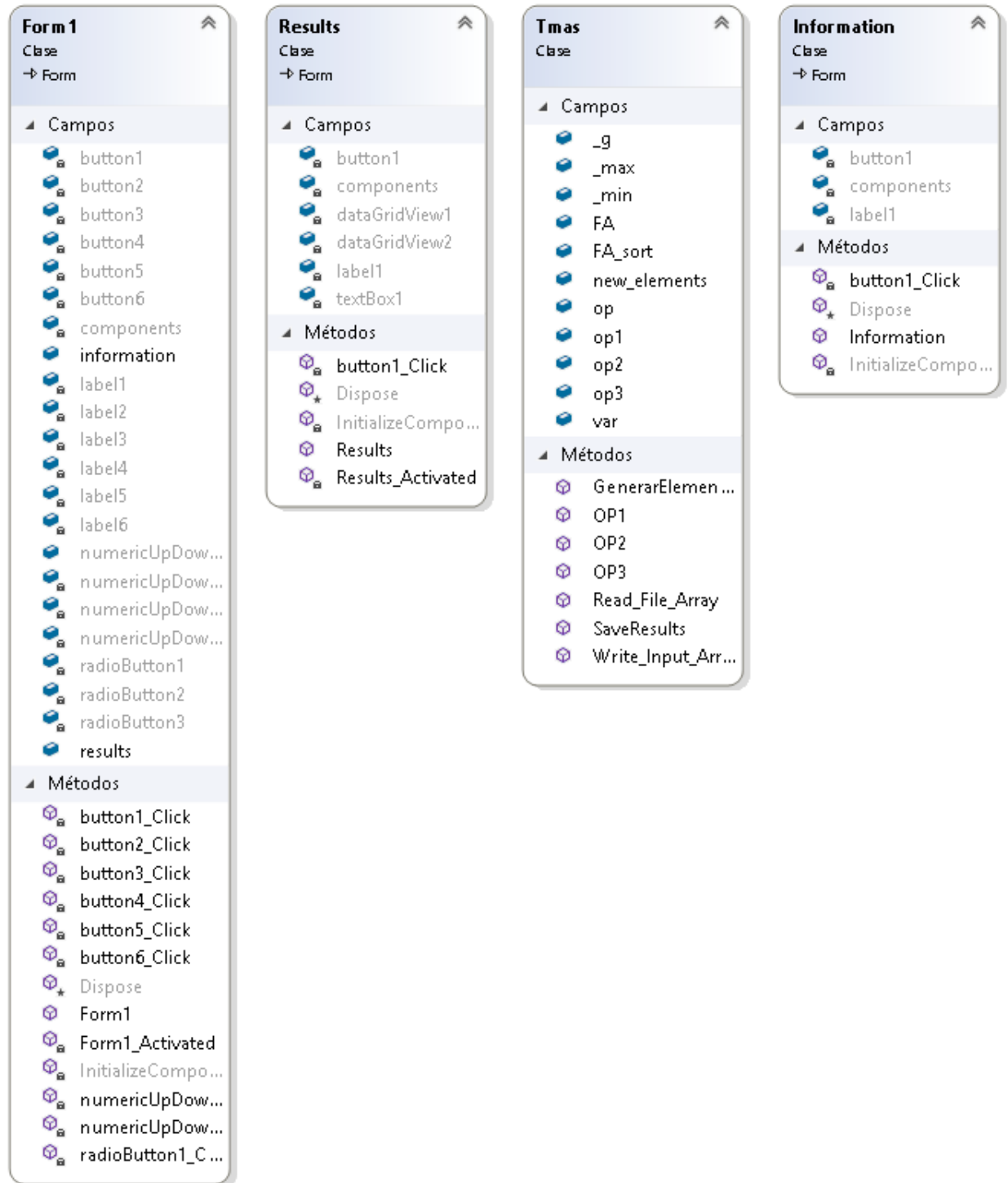
//3 Operation of my variant: Элементы массива с суммой индексов кратной 3 умножить на (-g).

```

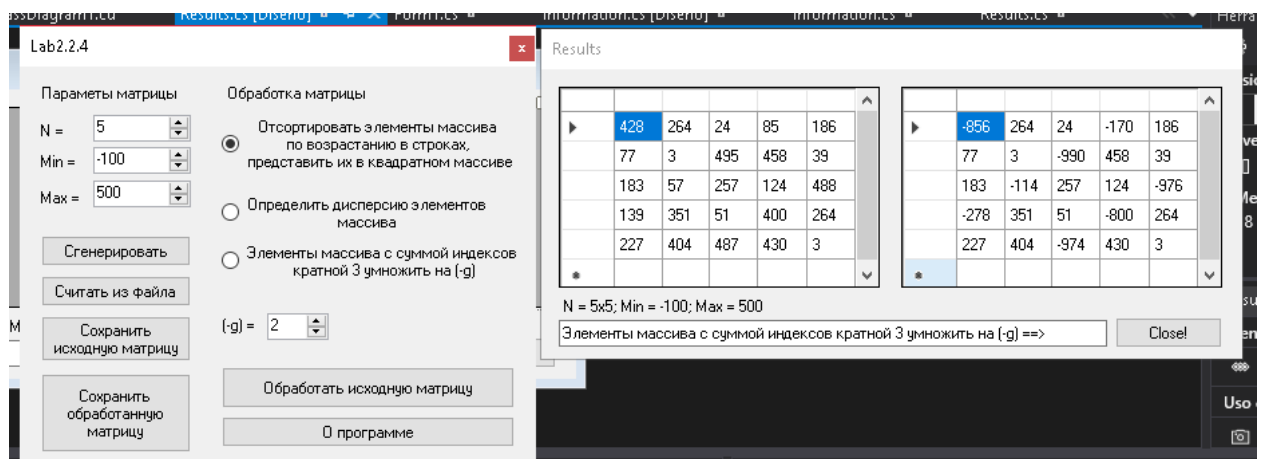
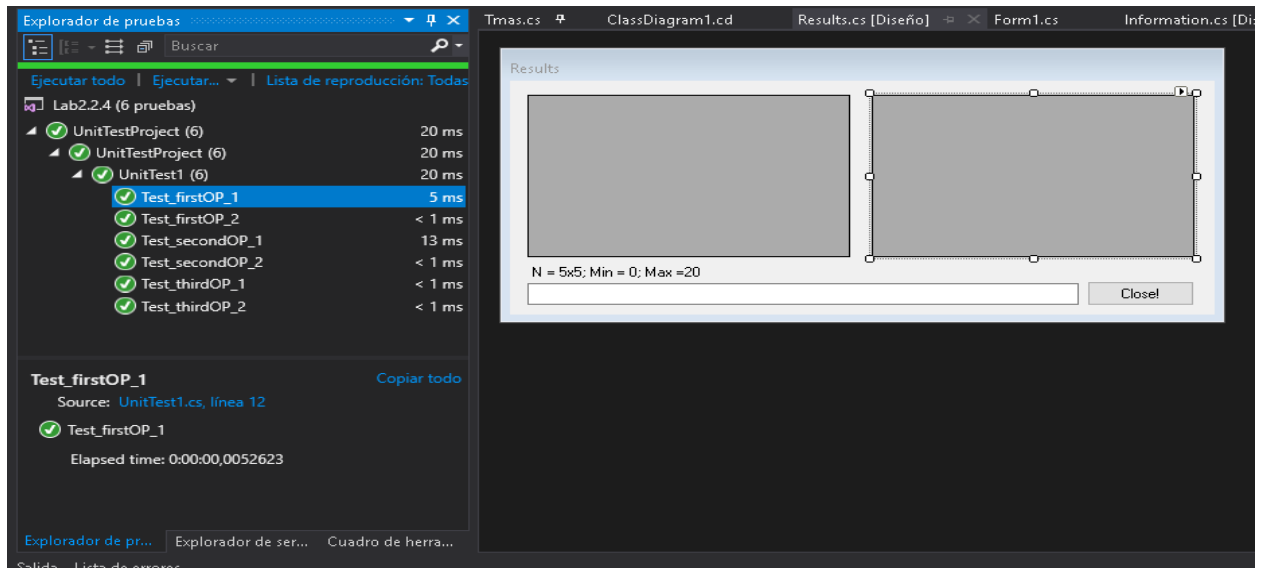
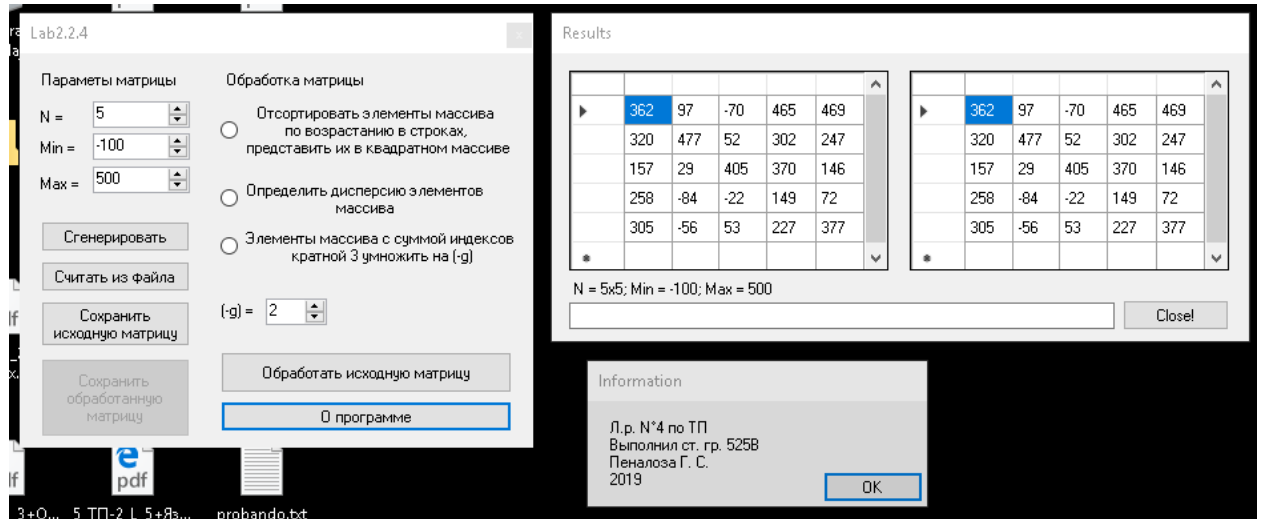
public static void OP3()
{
    new_elements = (int[,])FA.Clone();
    for (int c = 0; c < new_elements.GetLength(0); c++)
    {
        for (int i = 0; i < new_elements.GetLength(1); i++)
        {
            if ((c + i) % 3 == 0)
            {
                new_elements[c, i] = new_elements[c, i] * (-1) * (_g);
            }
        }
    }
}
}
}
}
}
}
}
}

```

Диаграмму классов



Скриншоты



Lab2.2.4

Параметры матрицы

N = 5
Min = -100
Max = 500

Сгенерировать
Считать из файла
Сохранить исходную матрицу
Сохранить обработанную матрицу

Обработка матрицы

☐ Отсортировать элементы массива по возрастанию в строках, представить их в квадратном массиве

☒ Определить дисперсию элементов массива

☐ Элементы массива с суммой индексов кратной 3 умножить на (-g)

(-g) = 2

Обработать исходную матрицу
О программе

Results

	-81	156	45	212	138
	-77	445	-5	116	-68
	269	328	481	432	230
	118	185	116	9	42
	454	218	145	437	229

N = 5x5; Min = -100; Max = 500
Дисперсию элементов массива: 28578.5984

Close!

Lab2.2.4

Параметры матрицы

N = 5
Min = -100
Max = 500

Сгенерировать
Считать из файла
Сохранить исходную матрицу
Сохранить обработанную матрицу

Обработка матрицы

☐ Отсортировать элементы массива по возрастанию в строках, представить их в квадратном массиве

☐ Определить дисперсию элементов массива

☒ Элементы массива с суммой индексов кратной 3 умножить на (-g)

(-g) = 3

Обработать исходную матрицу
О программе

Results

	-81	156	45	212	138
	-77	445	-5	116	-68
	269	328	481	432	230
	118	185	116	9	42
	454	218	145	437	229

N = 5x5; Min = -100; Max = 500
Элементы массива с суммой индексов кратной 3 умножить на (-g) ==>

Close!

Lab2.2.4

Параметры матрицы

N = 7
Min = -100
Max = 500

Сгенерировать
Считать из файла
Сохранить исходную матрицу
Сохранить обработанную матрицу

Обработка матрицы

☐ Отсортировать элементы массива по возрастанию в строках, представить их в квадратном массиве

☐ Определить дисперсию элементов массива

☒ Элементы массива с суммой индексов кратной 3 умножить на (-g)

(-g) = 3

Обработать исходную матрицу
О программе

Results

	2...	169	4...	80	1...	48
	3...	456	4...	331	2...	110
	40	181	-50	17	1...	-75
	2...	344	3...	398	-65	-40
	3...	450	3...	275	58	281
	1...	-65	2...	425	2...	464

N = 7x7; Min = -100; Max = 500
Элементы массива с суммой индексов кратной 3 умножить на (-g) ==>

Close!

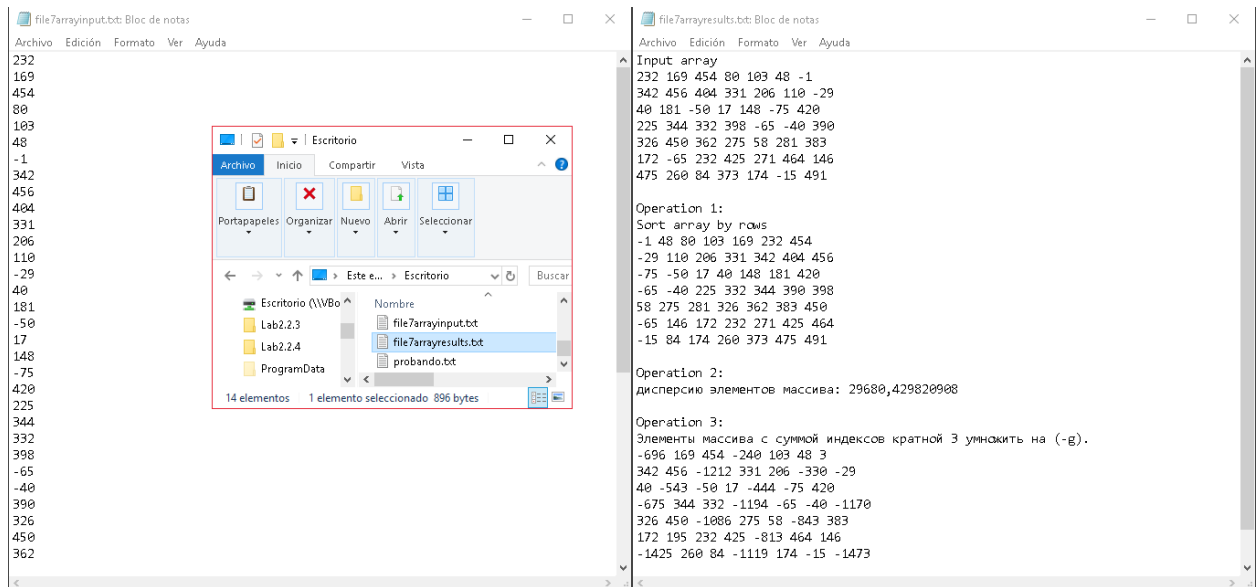


Таблица 1 - Поля и методы класса А и их назначение

Nº	Поле	Назначение
1	button1	Generate array
2	button2	Read array form textfile
3	button3	Write the entered array into file tex
4	button4	Save the results in the text file
5	button5	Execute the operations accordint the radioButtons
6	button6	About the program
7	information	Instance of about the program
8	label1	Titles in the program
9	label2	Titles in the program
10	label3	Titles in the program
11	label4	Titles in the program
12	label5	Titles in the program
13	label6	Titles in the program
14	numericUpDown1	-g

15	numericUpDown6	max
16	numericUpDown5	min
17	numericUpDown4	Size array
18	radioButton1	First operation
19	radioButton2	2 operation
20	radioButton3	3 operation
21	results	Instance of form of the results
22	button1	Close the result windows
23	dataGridView1	Array entered
24	dataGridView2	Array according the operation
25	label1	Title with information about the array
26	textBox1	Field to show results
27	_g	Constant of 3 operation
28	_max	Max array
29	_min	Min array
30	FA	Array entered
31	FA_sort	Array sorted
32	new_elements	Array with the result of the 3 operation
33	op	Current option of the variant
34	op1	If the 1 operation was executed
35	op2	If the 2 operation was executed
36	op3	If the 3 operation was executed
37	var	variance
38	button1	Close the information windows
39	label1	Titles in information windows form
40	Метод	Назначение
41	GenerarElementos	Generate array with random numbers
42	OP1	Execute the first operation of my variant
43	OP2	Execute the 2 operation of my variant

44	OP3	Execute the 3 operation of my variant
45	Read_File_Array	Write the array entered into the file tex
46	SaveResults	Save into the text file the results of the operations
47	Write_Input_Array	Write the array entered into the file text

Таблица 2 - Обработчики событий проекта и их назначение

N°	Обработчик события	Назначение
1	button1_Click	Generate array
2	button2_Click	Read array form text file
3	button3_Click	Write the array in the text file
4	button4_Click	Write a text file with an array entered and the results of the operations until had done all the operations
5	button5_Click	Button to show the windows with the result of the each operations
6	button6_Click	Button to show the information about the program
7	Form1_Activated	Check the operations
8		General method to select and do the operations of my variant
9	radioButton1_Click	General method to select and do the operations of my variant
10	numericUpDown5_Value Changed	Method to verify if the user select the correct minimum number of the interval
11	numericUpDown6_Value Changed	Method to verify if the user select the correct maximum number of the interval
12	button1_Click	Button to hide the form
13	Results_Activated	In this method the program is going to put the array entered into the

		datagridview and the second part is going to execute the selected operation and show into the datagridview and/or textbox
--	--	---

Выводы

В лабораторной практике изучалось управление файлами, модульные тесты и передача и обмен данными между классами и формами.

В моем варианте используйте текстовые файлы для хранения и чтения данных.

Класс Tmas.cs использовался для обмена свойствами и методами между двумя формами программы: «From1.cs» и «Results.cs».

Никакие данные не были переданы напрямую между участниками. Если не все через класс Tmas.cs

Как улучшение: управление данными с помощью файлов json намного лучше.

Использованные источники

1. <https://docs.microsoft.com/es-es/dotnet/>
2. https://es.wikipedia.org/wiki/Divisi%C3%B3n_por_tentativa