

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

Кафедра комп’ютерних систем та мереж

Лабораторна робота № 3

“Разработка и исследование программы обработки массивов.”

По дисциплине “Технологии программирования”

XAI.503.525B.123. 1705067.180

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

4-03-19
(підпис, дата)

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

(підпис, дата) (П.І.Б.)

2018

Задание: разработать на Visual C# проект, использующий библиотеку Windows Forms и реализующий генерацию массива, ввод (вывод) массива из файла, обработку массива и сохранение результатов обработки массива в файле в соответствии с вариантом из таблицы 1. Разработать модульные тесты (unit-тесты) для методов класса операций над одномерным массивом. UML диаграмма вариантов использования проекта приведена на рисунке 1.

Вариант 20

Задание

20.	[0, 3000]	4, 5, 15, 18	xml
-----	-----------	--------------	-----

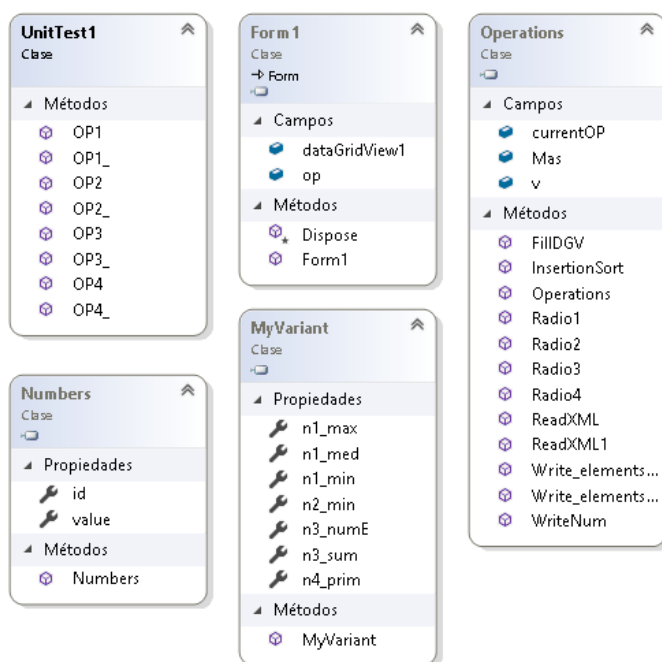
4. Вычислить минимальный, максимальный элементы массива и медиану массива (вычисления проводить без использования встроенных функций C#).

5. Отсортировать элементы массива по возрастанию.

15. Определить сумму и количество элементов массива a_i , у которых число, составленное из двух первых цифр числа a_i , четное.

18. Определить количество простых чисел массива методом пробных делителей.

Диаграмму классов



Form 1
 Clase
 → Form

Campos

- button1
- button2
- button3
- button4
- button5
- button6
- components
- dataGridView1
- groupBox1
- index
- label1
- label2
- label3
- label4
- numericUpDown...
- numericUpDown...
- numericUpDown...
- op
- panel1
- radioButton1
- radioButton2
- radioButton3
- radioButton4
- textBox1
- value

Métodos

- button1_Click
- button2_Click
- button3_Click
- button4_Click
- button5_Click
- button6_Click
- Dispose
- Form1
- Form1_Load
- InitializeCompo...
- label4_Click
- numericUpDown...
- numericUpDown...
- radioButton1_C...
- radioButton1_C...
- textBox1_TextC...

Numbers
 Clase

Propiedades

- id
- value

MyVariant
 Clase

Propiedades

- n1_max
- n1_med
- n1_min
- n2_min
- n3_numE
- n3_sum
- n4_prim

Operations
 Clase

Campos

- currentOP
- Mas
- v

Métodos

- FillDGV
- InsertionSort
- Radio1
- Radio2
- Radio3
- Radio4
- ReadXML
- ReadXML1
- Write_elements...
- Write_elements...
- WriteNum

Текст программы

Form.cs

```

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

namespace Lab2._2._3
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();

            public Operations op = new Operations();

            // Set correct interval
            private void numericUpDown2_ValueChanged(object sender, EventArgs e)
            {
                if
(numericUpDown2.Value>numericUpDown3.Value&&numericUpDown2.Value!
=0&&numericUpDown3.Value!=0)
                {
                    MessageBox.Show("This number can be less than B!");
                    numericUpDown2.Value -= 1;
                }
                else
                {
                    if (numericUpDown3.Value==0||
numericUpDown3.Value<=numericUpDown2.Value)
                    {
                        numericUpDown3.Value = numericUpDown2.Value + 1;
                    }
                }
            }
        }
    }
}

```

```

    }
}
// Set correct interval
private void numericUpDown3_ValueChanged(object sender, EventArgs e)
{
    if (numericUpDown3.Value <= numericUpDown2.Value ||
numericUpDown3.Value == 0)
    {
        MessageBox.Show("This number can be higher than A!");
        numericUpDown3.Value = numericUpDown2.Value + 1;
    }
}
//Close App
private void button6_Click(object sender, EventArgs e)
{
    this.Close();
}
//Generate array random
private void button1_Click(object sender, EventArgs e)
{
    radioButton1.Checked = false;
    radioButton2.Checked = false;
    radioButton3.Checked = false;
    radioButton4.Checked = false;
    textBox1.Text = "";
    if
(numericUpDown1.Value>0&&numericUpDown2.Value>=0&&numericUpDown
3.Value!=0) {
        dataGridView1.RowCount = (int)numericUpDown1.Value;

        dataGridView1 =
op.FillDGV(numericUpDown1,numericUpDown2,numericUpDown3,dataGridVie
w1);

        button2.Enabled = true;
    }
    else
    {
        MessageBox.Show("Parameters to generate the array are incorrect!");
    }
}
//Write array
private void button2_Click(object sender, EventArgs e)
{
    Stream myStream;

```

```

SaveFileDialog saveFileDialog1 = new SaveFileDialog();
saveFileDialog1.Filter = "Xml files (*.xml)|*.xml|All files (*.*)|*.*";
saveFileDialog1.FilterIndex = 1;
saveFileDialog1.RestoreDirectory = true;
if (saveFileDialog1.ShowDialog() == DialogResult.OK)
{
    if ((myStream = saveFileDialog1.OpenFile()) != null)
    {
        myStream.Close();
        op.Write_elements_f1(saveFileDialog1,op.Mas);
    }
}
//Read array
private void button4_Click(object sender, EventArgs e)
{
    dataGridView1=op.ReadXML(dataGridView1);
    numericUpDown1.Value = dataGridView1.RowCount;
}
//Set actions to radiobuttons of the variant
private void radioButton1_Click(object sender, EventArgs e)
{
    RadioButton rd = (RadioButton)sender;
    if (dataGridView1.RowCount > 1)
        button3.Enabled = true;
    else
    {
        rd.Checked = false;
        MessageBox.Show("There are not elements to operate!");
        return;
    }
    if (rd == radioButton1)
    {
        textBox1.Text=op.Radio1(textBox1.Text);
    }
    else if (rd == radioButton2)
    {
        textBox1.Text = op.Radio2(textBox1.Text);
        return;
    }
    else if (rd == radioButton3)
    {
        textBox1.Text = op.Radio3(textBox1.Text);
        return;
    }
}

```

```

        else
        {
            textBox1.Text = op.Radio4(textBox1.Text);
            return;
        }
    }

    private void radioButton1_CheckedChanged(object sender, EventArgs e)
    {

    }

    //Read file 2 with opetations
    private void button5_Click(object sender, EventArgs e)
    {
        textBox1.Text = op.ReadXML1();
    }

    //Write XML file 2 with operations
    private void button3_Click(object sender, EventArgs e)
    {
        if (op.v.n1_med != 0 && op.v.n2_min != "" && op.v.n3_sum != 0 &&
op.v.n4_prim!=null)
            op.Write_elements_f2(op.Mas);
        else
            MessageBox.Show("Do all the operations to save!");
    }

    private void Form1_Load(object sender, EventArgs e)
    {
        if (!File.Exists("Operations.xml")) {
            XmlTextWriter writer = new XmlTextWriter("Operations.xml",
System.Text.Encoding.UTF8);
            writer.WriteStartDocument();
            writer.WriteStartElement("MyVariant");
            writer.WriteEndElement();
            writer.Close();
        }
    }

    private void textBox1_TextChanged(object sender, EventArgs e)
    {

    }

    private void label4_Click(object sender, EventArgs e)
    {

```

```

    }
}
//Class to use in the array Mas
public class Numbers
{
    public int id { get; set; }
    public int value { get; set; }
}
//values to be calculated of my variant
public class MyVariant
{
    public int n1_min { get; set; }
    public int n1_max { get; set; }
    public double n1_med { get; set; }
    public string n2_min { get; set; }
    public int n3_sum { get; set; }
    public int n3_numE { get; set; }
    public string n4_prim { get; set; }
}
//Class with the operations to Read, Write and generate array Mas
public class Operations
{
    //Algorith to sort the Mas
    public List<Numbers> InsertionSort(List<Numbers> inputArray)
    {
        for (int i = 0; i < inputArray.Count - 1; i++)
        {
            for (int j = i + 1; j > 0; j--)
            {
                if (inputArray[j - 1].value > inputArray[j].value)
                {
                    int temp = inputArray[j - 1].value;
                    inputArray[j - 1].value = inputArray[j].value;
                    inputArray[j].value = temp;
                }
            }
        }
        return inputArray;
    }
}
//Current operation of radiobutton
public int currentOP;
//Instance of my variant class
public MyVariant v = new MyVariant();
//Operation 1

```



```

public string Radio1(string current)
{
    string rTextBox = current;
    List<Numbers> sMas = InsertionSort(Mas);
    v.n1_max = sMas[sMas.Count-1].value;
    v.n1_min = sMas[0].value;
    if (sMas.Count%2==0)
    {
        v.n1_med = (sMas[sMas.Count / 2].value + sMas[(sMas.Count / 2) -
1].value) / 2.0;
    }
    else
    {
        v.n1_med = sMas[sMas.Count / 2].value;
    }
    rTextBox += "Max:" + v.n1_max + " Min:" + v.n1_min + " Med:" +
v.n1_med+"\r\n";
    currentOP = 1;
    return rTextBox;
}
//Operation 2
public string Radio2(string current)
{
    v.n2_min = "";
    string rTextBox = current;
    List<Numbers> sMas = InsertionSort(Mas);
    for (int c=0;c<sMas.Count;c++)
    {
        v.n2_min += sMas[c].value + " ";
    }
    rTextBox += v.n2_min;
    currentOP = 2;
    return rTextBox+"\r\n";
}
//Operation 3
public string Radio3(string current)
{
    v.n3_numE = 0;
    v.n3_sum = 0;
    string rTextBox = current;
    for (int c=0;c<Mas.Count;c++)
    {
        string num;
        if (Mas[c].value>99)
        {

```

```

        num = Mas[c].value.ToString().Substring(0, 2);
    }
    else
    {
        num = Mas[c].value.ToString();
    }
    if (Int32.Parse(num)%2==0)
    {
        v.n3_numE = v.n3_numE + 1;
        v.n3_sum += Mas[c].value;
    }
}
rTextBox += "Sum:" + v.n3_sum + " Elements:" + v.n3_numE;
currentOP = 3;
return rTextBox+"\r\n";
}
//Operation 4
public string Radio4(string current)
{
    int nPri = 0;
    bool cmp;
    string rTextBox = current;
    for (int c=0;c<Mas.Count;c++)
    {
        cmp = true;
        for (int i=2;i<Mas[c].value;i++)
        {
            if (Mas[c].value%i==0)
            {
                cmp = false;
                break;
            }
        }
        if (cmp&&Mas[c].value>1)
        {
            nPri = nPri + 1;
        }
    }
    rTextBox += "Prim Numbers: " + nPri;
    v.n4_prim = nPri.ToString();
    currentOP = 4;
    return rTextBox+"\r\n";
}
//Funtions to write elements in f1 chosen
public void Write_elements_f1(SaveFileDialog sv, List<Numbers>Mas)

```

```

{
    XmlTextWriter writer = new XmlTextWriter(sv.FileName,
System.Text.Encoding.UTF8);
    writer.WriteStartDocument();
    writer.WriteStartElement("Массив_Mas");
    writer.WriteEndElement();
    writer.Close();
    for (int c=0;c<Mas.Count;c++)
    {
        Numbers n = new Numbers { id=Mas[c].id,value=Mas[c].value };
        WriteNum(n.id.ToString(),n.value.ToString(),sv);
    }
}
//Funtions to write elements in f2 chosen
public void Write_elements_f2(List<Numbers> Mas)
{
    XmlDocument doc = new XmlDocument();
    doc.Load("Operations.xml");

    XmlElement root = doc.DocumentElement;

    XmlElement numberOP1 = doc.CreateElement("op1");
    XmlElement numberOP2 = doc.CreateElement("op2");
    XmlElement numberOP3 = doc.CreateElement("op3");
    XmlElement numberOP4 = doc.CreateElement("op4");

    //XmlAttribute idV1 = doc.CreateAttribute("id");
    //XmlAttribute idV2 = doc.CreateAttribute("id");
    //XmlAttribute idV3 = doc.CreateAttribute("id");
    //XmlAttribute idV4 = doc.CreateAttribute("id");

    XmlElement valueO1_1 = doc.CreateElement("max");
    XmlElement valueO1_2 = doc.CreateElement("min");
    XmlElement valueO1_3 = doc.CreateElement("med");

    XmlElement valueO2_1 = doc.CreateElement("sort_array");

    XmlElement valueO3_1 = doc.CreateElement("Sum");
    XmlElement valueO3_2 = doc.CreateElement("Element_numbers");

    XmlElement valueO4_1 = doc.CreateElement("Prim_numbers");

    //XmlText idText1 = doc.CreateTextNode("0");
    //XmlText idText2 = doc.CreateTextNode("1");
    //XmlText idText3 = doc.CreateTextNode("2");

```

```

//XmlText idText4 = doc.CreateTextNode("3");

XmlText valueTextO1_1 = doc.CreateTextNode(v.n1_max.ToString());
XmlText valueTextO1_2 = doc.CreateTextNode(v.n1_min.ToString());
XmlText valueTextO1_3 = doc.CreateTextNode(v.n1_med.ToString());

XmlText valueTextO2_1 = doc.CreateTextNode(v.n2_min.ToString());

XmlText valueTextO3_1 = doc.CreateTextNode(v.n3_sum.ToString());
XmlText valueTextO3_2 = doc.CreateTextNode(v.n3_numE.ToString());

XmlText valueTextO4_1 = doc.CreateTextNode(v.n4_prim.ToString());

//idV1.AppendChild(idText1);
//idV2.AppendChild(idText2);
//idV3.AppendChild(idText3);
//idV4.AppendChild(idText4);

valueO1_1.AppendChild(valueTextO1_1);
valueO1_2.AppendChild(valueTextO1_2);
valueO1_3.AppendChild(valueTextO1_3);

valueO2_1.AppendChild(valueTextO2_1);

valueO3_1.AppendChild(valueTextO3_1);
valueO3_2.AppendChild(valueTextO3_2);

valueO4_1.AppendChild(valueTextO4_1);

//numberOP1.Attributes.Append(idV1);
//numberOP2.Attributes.Append(idV2);
//numberOP3.Attributes.Append(idV3);
//numberOP4.Attributes.Append(idV4);

numberOP1.AppendChild(valueO1_1);
numberOP1.AppendChild(valueO1_2);
numberOP1.AppendChild(valueO1_3);

numberOP2.AppendChild(valueO2_1);

numberOP3.AppendChild(valueO3_1);
numberOP3.AppendChild(valueO3_2);

numberOP4.AppendChild(valueO4_1);

```

```

        root.AppendChild(numberOP1);
        root.AppendChild(numberOP2);
        root.AppendChild(numberOP3);
        root.AppendChild(numberOP4);
        doc.Save("Operations.xml");
        v = new MyVariant();
    }
    //Funcion aux to write f1 file
    public void WriteNum(string pID, string value, SaveFileDialog sv)
    {
        XmlDocument doc = new XmlDocument();
        doc.Load(sv.FileName);
        XmlElement root = doc.DocumentElement;
        XmlElement number = doc.CreateElement("number");
        XmlAttribute idV = doc.CreateAttribute("id");
        XmlElement valueV = doc.CreateElement("value");
        XmlText idText = doc.CreateTextNode(pID);
        XmlText valueText = doc.CreateTextNode(value);
        idV.AppendChild(idText);
        valueV.AppendChild(valueText);
        number.Attributes.Append(idV);
        number.AppendChild(valueV);
        root.AppendChild(number);
        doc.Save(sv.FileName);
    }
    //List of values Numbers
    public List<Numbers> Mas=new List<Numbers>();
    //Funtion to fill DataGridView
    public DataGridView FillDGV(NumericUpDown n1, NumericUpDown n2,
    NumericUpDown n3,DataGridView dgv)
    {
        Mas = new List<Numbers>((int)(n1.Value));
        Random rn = new Random();
        for (int c = 0; c < (int)n1.Value; c++)
        {
            Numbers n = new Numbers { id = c, value = rn.Next((int)(n2.Value),
(int)(n3.Value)) };
            Mas.Add(n);
            dgv[0, c].Value = Mas[c].id;
            dgv[1, c].Value = Mas[c].value;
        }
        return dgv;
    }
    //Read XML file of f2 file

```

```

public string ReadXML1()
{
    string result = "";
    try
    {
        using (OpenFileDialog openFileDialog = new OpenFileDialog())
        {
            openFileDialog.InitialDirectory = "c:\\";
            openFileDialog.Filter = "Xml files (*.xml)|*.xml|All files (*.*)|*.*";
            openFileDialog.FilterIndex = 2;
            openFileDialog.RestoreDirectory = true;
            if (openFileDialog.ShowDialog() == DialogResult.OK)
            {
                XmlDocument xDoc = new XmlDocument();
                xDoc.Load(openFileDialog.FileName);
                XmlElement xRoot = xDoc.DocumentElement;
                foreach (XmlNode xnode in xRoot)
                {
                    switch (xnode.Name)
                    {
                        case "op1":
                            v.n1_max = int.Parse(xnode["max"].InnerText);
                            v.n1_min = int.Parse(xnode["min"].InnerText);
                            v.n1_med = double.Parse(xnode["med"].InnerText);
                            result += "Max:" + v.n1_max + " Min:" + v.n1_min + "
Med:" + v.n1_med+"\r\n";
                            break;
                        case "op2":
                            v.n2_min = xnode["sort_array"].InnerText;
                            result += v.n2_min+"\r\n";
                            break;
                        case "op3":
                            v.n3_sum = int.Parse(xnode["Sum"].InnerText);
                            v.n3_numE =
int.Parse(xnode["Element_numbers"].InnerText);
                            result += "Sum:" + v.n3_sum + " Elements:" +
v.n3_numE+"\r\n";
                            break;
                        case "op4":
                            v.n4_prim = (xnode["Prim_numbers"].InnerText);
                            result += "Prim Numbers: " + v.n4_prim+"\r\n";
                            break;
                    }
                }
            }
        }
    }
}

```

```

    }
}
catch { MessageBox.Show("Problems to read the XML file format!"); }
return result;
}
//Read XML file of f1 file
public DataGridView ReadXML(DataGridView dgv)
{
    try
    {
        using (OpenFileDialog openFileDialog = new OpenFileDialog())
        {
            openFileDialog.InitialDirectory = "c:\\";
            openFileDialog.Filter = "Xml files (*.xml)|*.xml|All files (*.*)|*.*";
            openFileDialog.FilterIndex = 2;
            openFileDialog.RestoreDirectory = true;
            if (openFileDialog.ShowDialog() == DialogResult.OK)
            {
                Mas.Clear();
                XmlDocument xDoc = new XmlDocument();
                xDoc.Load(openFileDialog.FileName);
                XmlElement xRoot = xDoc.DocumentElement;
                foreach (XmlNode xnode in xRoot)
                {
                    Numbers num = new Numbers();
                    num.id = int.Parse(xnode.Attributes[0].Value);
                    num.value = int.Parse(xnode["value"].InnerText);
                    Mas.Add(num);
                }
            }
        }
        dgv.RowCount = Mas.Count();
        for (int c = 0; c < Mas.Count; c++)
        {
            dgv[0, c].Value = Mas[c].id;
            dgv[1, c].Value = Mas[c].value;
        }
    }
    catch { MessageBox.Show("Problems to read the XML file format!"); }
    return dgv;
}
}

```

UnitTest1

```

using System;
using Microsoft.VisualStudio.TestTools.UnitTesting;
using Lab2._2._3;
using System.Collections.Generic;

namespace UnitTest
{
    [TestClass]
    public class UnitTest1
    {
        [TestMethod]
        public void OP1()
        {
            //5,2,12,5,9,10,1 => 1,2,5,5,9,10,12
            var min_e="1";
            var max_e="12";
            var med_e="5";
            Operations op = new Operations();
            op.Mas = new List<Numbers> { new Numbers() { id = 0, value = 5 }, new
Numbers() { id = 1, value = 2 }, new Numbers() { id = 2, value = 12 }, new
Numbers() { id = 3, value = 5 }, new Numbers() { id = 4, value = 9 }, new
Numbers() { id = 5, value = 10 }, new Numbers() { id = 6, value = 1 } };
            op.Radio1("");

            Assert.AreEqual(min_e,op.v.n1_min.ToString(),max_e,op.v.n1_max.ToString(),m
ed_e,op.v.n1_med.ToString());
        }
        [TestMethod]
        public void OP1_()
        {
            //5,2,12,6,9,10,1,3 => 1,2,3,6,5,9,10,12 => 11/2=5.5
            var min_e = "1";
            var max_e = "12";
            var med_e = "5.5";
            Operations op = new Operations();
            op.Mas = new List<Numbers> { new Numbers() { id = 0, value = 5 }, new
Numbers() { id = 1, value = 2 }, new Numbers() { id = 2, value = 12 }, new
Numbers() { id = 3, value = 6 }, new Numbers() { id = 4, value = 9 }, new
Numbers() { id = 5, value = 10 }, new Numbers() { id = 6, value = 1 }, new
Numbers() { id=7,value=3} };
            op.Radio1("");
            Assert.AreEqual(min_e, op.v.n1_min.ToString(), max_e,
op.v.n1_max.ToString(), med_e, op.v.n1_med.ToString());
        }
    }
}

```



```

    }
    [TestMethod]
    public void OP2()
    {
        //5,2,12,5,9,10,1 => 1,2,5,5,9,10,12
        var array_e = "1 2 5 5 9 10 12 ";
        Operations op = new Operations();
        op.Mas = new List<Numbers> { new Numbers() { id = 0, value = 5 }, new
Numbers() { id = 1, value = 2 }, new Numbers() { id = 2, value = 12 }, new
Numbers() { id = 3, value = 5 }, new Numbers() { id = 4, value = 9 }, new
Numbers() { id = 5, value = 10 }, new Numbers() { id = 6, value = 1 } };
        op.Radio2("");
        Assert.AreEqual(array_e, op.v.n2_min);
    }
    [TestMethod]
    public void OP2_()
    {
        //5,2,12,5,9,10,11 => 2,5,5,9,10,11,12
        var array_e = "2 5 5 9 10 11 12 ";
        Operations op = new Operations();
        op.Mas = new List<Numbers> { new Numbers() { id = 0, value = 5 }, new
Numbers() { id = 1, value = 2 }, new Numbers() { id = 2, value = 12 }, new
Numbers() { id = 3, value = 5 }, new Numbers() { id = 4, value = 9 }, new
Numbers() { id = 5, value = 10 }, new Numbers() { id = 6, value = 11 } };
        op.Radio2("");
        Assert.AreEqual(array_e, op.v.n2_min);
    }
    [TestMethod]
    public void OP3()
    {
        //53 427 242 211 126 227 8
        var numE_e = "5";
        var sum_e = "1030";
        Operations op = new Operations();
        op.Mas = new List<Numbers> { new Numbers() { id = 0, value = 53 },
new Numbers() { id = 1, value = 427 }, new Numbers() { id = 2, value = 242 },
new Numbers() { id = 3, value = 211 }, new Numbers() { id = 4, value = 126 },
new Numbers() { id = 5, value = 127 }, new Numbers() { id = 6, value = 8 } };
        op.Radio3("");
        Assert.AreEqual(numE_e, op.v.n3_numE.ToString(), sum_e,
op.v.n3_sum.ToString());
    }
    [TestMethod]
    public void OP3_()
    {

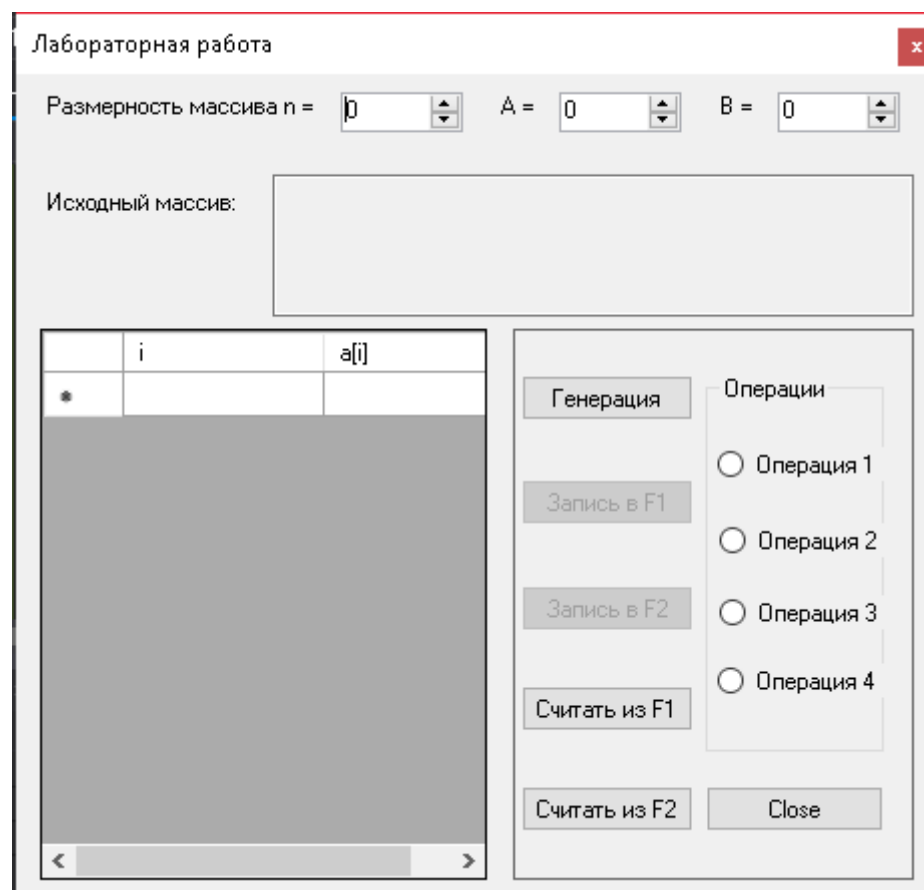
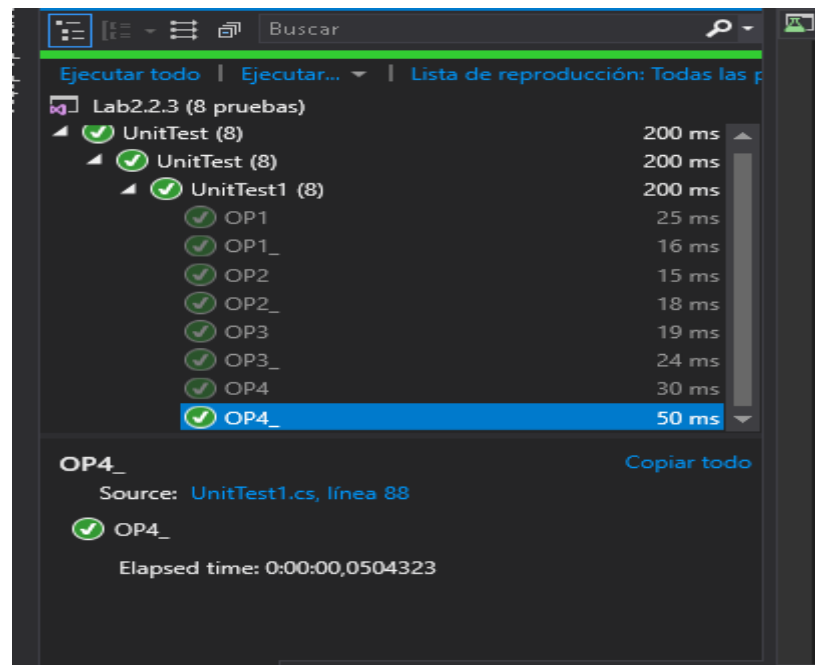
```

```

//296 214 125 502 385 480 245
var sum_e = "1737";
var numE_e = "5";
Operations op = new Operations();
op.Mas = new List<Numbers> { new Numbers() { id = 0, value = 296 },
new Numbers() { id = 1, value = 214 }, new Numbers() { id = 2, value = 125 },
new Numbers() { id = 3, value = 502 }, new Numbers() { id = 4, value = 385 },
new Numbers() { id = 5, value = 480 }, new Numbers() { id = 6, value = 245 } };
op.Radio3("");
Assert.AreEqual(numE_e, op.v.n3_numE.ToString(), sum_e,
op.v.n3_sum.ToString());
}
[TestMethod]
public void OP4()
{
//65 2 119 96 115 84 101
var pri_e = "2";
Operations op = new Operations();
op.Mas = new List<Numbers> { new Numbers() { id = 0, value = 65 },
new Numbers() { id = 1, value = 2 }, new Numbers() { id = 2, value = 119 }, new
Numbers() { id = 3, value = 96 }, new Numbers() { id = 4, value = 115 }, new
Numbers() { id = 5, value = 84 }, new Numbers() { id = 6, value = 101 } };
op.Radio4("");
Assert.AreEqual(pri_e, op.v.n4_prim);
}
[TestMethod]
public void OP4_()
{
//46 57 39 98 47 6 42
var pri_e = "1";
Operations op = new Operations();
op.Mas = new List<Numbers> { new Numbers() { id = 0, value = 46 },
new Numbers() { id = 1, value = 57 }, new Numbers() { id = 2, value = 39 }, new
Numbers() { id = 3, value = 98 }, new Numbers() { id = 4, value = 47 }, new
Numbers() { id = 5, value = 6 }, new Numbers() { id = 6, value = 42 } };
op.Radio4("");
Assert.AreEqual(pri_e, op.v.n4_prim);
}
}
}

```

Скриншоты



```

<?xml version="1.0" encoding="ISO-8859-1"?>
- <MyVariant>
  - <op1>
    <max>2937</max>
    <min>415</min>
    <med>1980,5</med>
  </op1>
  - <op2>
    <sort_array>415 786 871 1104 1779 2182 2418 2811 2916 2937 </sort_array>
  </op2>
  - <op3>
    <Sum>6015</Sum>
    <Element_numbers>3</Element_numbers>
  </op3>
  - <op4>
    <Prim_numbers>0</Prim_numbers>
  </op4>
</MyVariant>

```

```

<?xml version="1.0" encoding="ISO-8859-1"?>
- <Массив_Mas>
  - <number id="0">
    <value>415</value>
  </number>
  - <number id="1">
    <value>786</value>
  </number>
  - <number id="2">
    <value>871</value>
  </number>
  - <number id="3">
    <value>1104</value>
  </number>
  - <number id="4">
    <value>1779</value>
  </number>
  - <number id="5">
    <value>2182</value>
  </number>
  - <number id="6">
    <value>2418</value>
  </number>
  - <number id="7">
    <value>2811</value>
  </number>
  - <number id="8">
    <value>2916</value>
  </number>
  - <number id="9">
    <value>2937</value>
  </number>
</Массив_Mas>

```

Лабораторная работа

Размерность массива n = 10 A = 0 B = 3000

Исходный массив:
Max:2937 Min:415 Med:1980,5
415 786 871 1104 1779 2182 2418 2811 2916 2937
Sum:6015 Elements:3
Prim Numbers: 0

	i	a[i]
▶	0	871
	1	2916
	2	2418
	3	415
	4	786
	5	2811
	6	1104
	7	2182
	8	2937
*	9	1779

Генерация

Запись в F1

Запись в F2

Считать из F1

Считать из F2

Операции

☐ Операция 1

☐ Операция 2

☐ Операция 3

☒ Операция 4

Close

Лабораторная работа

Размерность массива n = 10 A = 0 B = 3000

Исходный массив:
88 124 180 325 1483 1511 2531 2568 2912 2955
Max:2955 Min:88 Med:1497
Sum:2200 Elements:5
Prim Numbers: 3

	i	a[i]
▶	0	124
	1	325
	2	2568
	3	2531
	4	2955
	5	180
	6	88
	7	1511
	8	1483
*	9	2912

Генерация

Запись в F1

Запись в F2

Считать из F1

Считать из F2

Операции

☐ Операция 1

☐ Операция 2

☐ Операция 3

☒ Операция 4

Close

 Mas.xml	5/3/2019 7:05	Archivo XML	6 KB
 Mas2.xml	5/3/2019 7:08	Archivo XML	12 KB
 Mas3.xml	5/3/2019 7:08	Archivo XML	18 KB
 Mas4.xml	5/3/2019 7:06	Archivo XML	23 KB
 Mas5.xml	5/3/2019 7:06	Archivo XML	29 KB
 Mas6.xml	5/3/2019 7:07	Archivo XML	35 KB
 Mas7.xml	5/3/2019 7:07	Archivo XML	41 KB
 Mas8.xml	5/3/2019 7:09	Archivo XML	46 KB
 Mas9.xml	5/3/2019 7:09	Archivo XML	52 KB
 Mas10.xml	5/3/2019 7:10	Archivo XML	58 KB
 Operations.xml	5/3/2019 7:20	Archivo XML	5 KB

Таблица 1 - Результаты работы проекта

N	100	200	300	400	500	600	700	800	900	1000
Объем памяти файла с массивом	6	12	18	23	29	35	41	46	52	58
Объем памяти файла с результатами обработки массива	1	2	2	3	3	4	4	4	5	6

Таблица 1 - Поля и методы класса A и их назначение (Пример)

N°	Поле	Назначение
1	button1	Generate Mas
2	button2	Write F1 file
3	button3	Write F2 file
4	button4	Read F1 file
5	button5	Read F2 file
6	button6	Close
7	dataGridView1	DataGridView to store the values

8	label1	Titles of elements
9	label2	Titles of elements
10	label3	Titles of elements
11	label4	Titles of elements
12	numericUpDown1	Size of Mas
13	numericUpDown2	Inteval x
14	numericUpDown3	Interval y
15	op	Instance of operations
16	textBox1	Textbox to put results
17	radioButton1	Options of the variant (1)
18	radioButton2	Options of the variant (2)
19	radioButton3	Options of the variant (3)
20	radioButton4	Options of the variant (4)
21	id	Id of number
22	clave	Value of the Mas
23	n1_max	Properties of the results (max)
24	n1_min	Properties of the results
25	n1_med	Properties of the results
26	n2_min	Properties of the results
27	n3_numE	Properties of the results
28	n3_Sum	Properties of the results
29	n4_prim	Properties of the results
30	Mas	Array
31	v	Instance of variant properties
32	currentOP	Current operation
33	Метод	Назначение
34	FillDGV	Fill dataGridView1
35	InsertionSort	Sort method
36	Radio1	Funtion of the 1 variant
37	Radio2	Funtion of the 2 variant
38	Radio3	Funtion of the 3 variant
39	Radio4	Funtion of the 4 variant
40	ReadXML1	Read XML f2

41	ReadXML	Read XML f1
42	WriteNum	Aux method to write xml file f1
43	Write_elements_f2	Funtion to write elements f2
44	Write_elements_f1	Funtion to write elements f1

**Таблица 2 - Обработчики событий проекта и их назначение
(Пример)**

N°	Обработчик события	Назначение
1	button1_Click	Event to execute funtion to generate Mas
2	button2_Click	– funtion to write f1
3	button3_Click	– function to write f2
4	button4_Click	– function to read f1
5	button5_Click	– function to read f2
6	button6_Click	– fuction to close app
7	Form1_load	Verify if xml file f2 exist or create new
8	radioButton1_Click	General event to control radioButtons
9	numericUpDown1_Click	General event to control numericUpDowns

Выводы

В ходе лабораторной практики мы изучили больше компонентов Windows Forms (radiobutton, dataGridView и т. Д.) И управление файлами (txt, json, xml).

В соответствии с моим вариантом мне пришлось использовать XML-файлы для хранения данных. Кроме того, для выполнения некоторых операций с массивом, созданным с использованием OPP.

Было сложно понять, как сохранить полученные данные.

Улучшения для реализации:

* Лучше организовать данные, используя json.

Использованные источники

1. <https://docs.microsoft.com/es-es/dotnet/>
2. https://es.wikipedia.org/wiki/Divisi%C3%B3n_tentativa