**Лабораторная работа №11**

**«Разработка динамической библиотеки по работе с массивами с использованием перегрузки операций»**

**Вариант 4**

1. **Цель работы:** научиться создаватьсобственную динамическую библиотеку и использовать ее в проектах
2. **Задание:**

**Задание 1**

В библиотечный модуль необходимо включить следующие подпрограммы:

1. Заполнение массива произвольного размера случайными числами;
2. Заполнение массива значениями из текстового файла;
3. Вывод массива произвольного размера на консоль;
4. Вывод массива произвольного размера в компонент таблица;
5. Вычисление суммы всех элементов массива;
6. Вычисление количества элементов массива, кратных заданному числу;
7. Перегрузка унарных операций ++ и - - для массива;
8. Перегрузка операций сложения и вычитания для двух массивов.

Проверить работу модуля в консольном приложении на примере вызова методов класса.

**Задание 2**

1. Для выполнения второго задания в библиотечный модуль нужно включить дополнительные методы, позволяющие выполнить поставленную задачу по вашему варианту (один или два метода).
2. Создайте тест-кейсы для проверки выполнения 2-х любых операций и метода по вашему варианту.
3. Создайте новый проект Windows Forms, подключите к нему собственную библиотеку. Подготовьте пользовательский интерфейс, добавив на форму нужные компоненты с их последующей настройкой.
4. Создайте методы обработки событий формы.
5. Настройте пользовательский интерфейс таким образом, чтобы некоторые поля ввода и кнопки блокировались, если режим работы не предусматривает их активность.

*Примечание:*

Программа должна выполнять задание с первым или вторым массивом по выбору пользователя. Если в самом массиве происходят изменения (например, перестановка элементов), результат вывести в ту же таблицу. Если же необходимо из двух массивов получить новый массив, то результат выводите в таблицу 3.

**вариант 4:** Разработать метод, который в массиве из *n* целых чисел подсчитывает количество отрицательных и положительных элементов, и если отрицательных элементов окажется больше, меняет знаки всех элементов на противоположные.

1. **Код собственной библиотеки:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.IO;

using System.Windows.Forms;

namespace Лаб\_11

{

public class MyException : ArgumentException

{

public const string ArgumentError = "Массив не определен";

public const string SizeError = "Выход за границы массива!";

public int Value { get; }

public MyException(string message) :base(message)

{

}

public MyException(string message, int value) : base(message)

{

Value = value;

}

}

public class Arr

{

private int[] a;

private int size;

private Random rnd = new Random();

/// <summary>

/// конструктор класса по умолчанию

/// </summary>

public Arr()

{

size = 0;

a = null;

}

/// <summary>

/// конструктор класса с одним параметром размер массива

/// </summary>

/// <param name="n">размер массива</param>

public Arr(int n)

{

a = new int[n];

size = n;

}

/// <summary>

/// ещё один конструктор с инициализацией обычным массивом

/// </summary>

/// <param name="x">массив целого типа</param>

public Arr(int[] x)

{

if (x == null)

throw new MyException(MyException.ArgumentError);

else

{

size = x.Length;

a = new int[size];

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

{

a[i] = x[i];

}

}

}

/// <summary>

/// ещё один конструктор с инициализацией объектом класса Arr

/// </summary>

/// <param name="A">объект класса Arr</param>

public Arr(Arr A)

{

if (A.a == null)

{

throw new MyException(MyException.ArgumentError);

}

else

{

size = A.size;

a = new int[size];

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

{ this.a[i] = A.a[i]; }

}

}

/// <summary>

/// метод заполнения массива случайными числами от -100 до 100

/// </summary>

public void RndInput()

{

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

a[i] = rnd.Next(-100, 101);

}

/// <summary>

/// метод заполнения массива случайными числами от 0 до n1

/// </summary>

/// <param name="n1"></param>

public void RndInput(int n1)

{

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

a[i] = rnd.Next(0, n1);

}

/// <summary>

/// метод заполнения массива случайными числами от n1 до n2

/// </summary>

/// <param name="n1">нижняя граница интервала</param>

/// <param name="n2">верхняя граница интервала</param>

public void RndInput(int n1, int n2)

{

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

a[i] = rnd.Next(n1, n2);

}

/// <summary>

/// перегрузка метода преобразования объекта Arr в троку

/// </summary>

/// <returns>элементы массива через пробел в одну строку</returns>

public override string ToString()

{

string s = "";

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

{

s += String.Format("{0,5}", a[i]);

}

return s;

}

/// <summary>

/// метод вывода массива класса Arr на консоль

/// </summary>

public void Print()

{

Console.WriteLine(this.ToString());

Console.WriteLine();

}

/// <summary>

/// метод распечатки массива в поле надписи

/// </summary>

/// <param name="handel">handel-строка заголовки</param>

/// <param name="lbl">lbl-поле надписи</param>

public void Print(string handel, Label lbl)

{

Console.WriteLine(handel);

Console.WriteLine(lbl);

}

/// <summary>

/// метод распечатки массива в таблицу

/// </summary>

/// <param name="handel">handel-строка заголовки</param>

/// <param name="lbl">lbl-поле надписи</param>

/// <param name="dgv">dgv-таблицаб куда выводится массив</param>

public void Print(string handel, Label lbl, DataGridView dgv)

{

lbl.Text = handel;

dgv.RowCount = 1;

dgv.ColumnCount = this.size;

for (int i = 0; i < this.size; i++)

{

dgv.Rows[0].Cells[i].Value = a[i];

}

}

/// <summary>

/// метод индексации элемента массива

/// </summary>

/// <param name="i">номер элемента</param>

/// <returns></returns>

public int this[int i]

{

get

{

if (i>= 0 && i < size)

{

return a[i];

}

else

{

return 0;

}

}

set

{

if (i >= 0 && i < size)

a[i] = value;

else

throw new MyException(MyException.SizeError);

}

}

/// <summary>

/// метод вычисления суммы всех элементов массива

/// </summary>

/// <returns>возвращает сумму элементов массива</returns>

public int Summ()

{

int s = 0;

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

{

s += a[i];

}

return s;

}

/// <summary>

/// Метод нахождения суммы эл-тов, кратных заданным

/// </summary>

/// <param name="n"></param>

/// <returns>сумма эл-тов, кратных заданному числу</returns>

public int Summ(int n)

{

int s = 0;

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

{

if (a[i] % n == 0)

s += a[i];

}

return s;

}

/// <summary>

/// перегрузка метода инкременции

/// </summary>

/// <param name="A"></param>

/// <returns></returns>

public static Arr operator ++(Arr A)

{

Arr temp = new Arr(A.size);

for (int i = 0; i < temp.size; i++)

{

temp[i] = ++A[i];

}

return temp;

}

/// <summary>

/// перегрузка метода декременции

/// </summary>

/// <param name="A"></param>

/// <returns></returns>

public static Arr operator --(Arr A)

{

Arr temp = new Arr(A.size);

for (int i = 0; i < temp.size; i++)

{

temp[i] = --A[i];

}

return temp;

}

/// <summary>

/// перегрузка оператора +: сложение массивов

/// </summary>

/// <param name="A"></param>

/// <param name="B"></param>

/// <returns></returns>

public static Arr operator +(Arr A, Arr B)

{

int tempsize;

if (A.size > B.size)

{

tempsize = A.size;

}

else

{

tempsize = B.size;

}

Arr temp = new Arr(tempsize);

for (int i = 0; i < temp.size; i++)

{

temp[i] = A[i] + B[i];

}

return temp;

}

/// <summary>

/// перегрузка оператора +: сложение массива и целого числа

/// </summary>

/// <param name="A"></param>

/// <param name="B"></param>

/// <returns></returns>

public static Arr operator +(Arr A, int x)

{

Arr temp = new Arr(A.size);

for (int i = 0; i < temp.size; i++)

{

temp[i] = A[i] + x;

}

return temp;

}

/// <summary>

/// перегрузка оператора +: сложение массива и целого числа

/// </summary>

/// <param name="A"></param>

/// <param name="B"></param>

/// <returns></returns>

public static Arr operator +(int x, Arr A)

{

Arr temp = new Arr(A.size);

for (int i = 0; i < temp.size; i++)

{

temp[i] = A[i] + x;

}

return temp;

}

/// <summary>

/// перегрузка оператора -: вычитание массивов

/// </summary>

/// <param name="A"></param>

/// <param name="B"></param>

/// <returns></returns>

public static Arr operator -(Arr A, Arr B)

{

int tempsize;

if (A.size > B.size)

{

tempsize = A.size;

}

else

{

tempsize = B.size;

}

Arr temp = new Arr(tempsize);

for (int i = 0; i < temp.size; i++)

{

temp[i] = A[i] - B[i];

}

return temp;

}

/// <summary>

/// перегрузка оператора -: вычитание целого числа из массива

/// </summary>

/// <param name="A"></param>

/// <param name="x"></param>

/// <returns></returns>

public static Arr operator -(int x, Arr A)

{

Arr temp = new Arr(A.size);

for (int i = 0; i < temp.size; i++)

{

temp[i] = A[i] - x;

}

return temp;

}

/// <summary>

/// перегрузка оператора -: вычитание целого числа из массива

/// </summary>

/// <param name="A"></param>

/// <param name="x"></param>

/// <returns></returns>

public static Arr operator -(Arr A, int x)

{

Arr temp = new Arr(A.size);

for (int i = 0; i < temp.size; i++)

{

temp[i] = A[i] - x;

}

return temp;

}

/// <summary>

/// Метод сравнения (перегруж.)

/// </summary>

/// <param name="obj">объект, с которым сравнивают</param>

/// <returns></returns>

public override bool Equals(object obj)

{

if (obj == null || obj as Arr == null)

return false;

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

if (this[i] != ((Arr)obj)[i])

return false;

return true;

}

/// <summary>

/// Метод вычисления кол-ва элементов, кратных заданному числу

/// </summary>

/// <param name="num">Заданное число</param>

/// <returns>Кол-во эл-тов</returns>

public int count\_krat(int num)

{

int s = 0;

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

{

if (a[i] % num == 0 && a[i] != 0)

s++;

}

return s;

}

/// <summary>

/// Метод чтения эл-тов массива из файла

/// </summary>

/// <param name="fn">Адрес файла</param>

public void file\_read(string fn)

{

StreamReader stream = null;

try

{

stream = new StreamReader(fn);

char[] ch = { ' ', '\n' };

string[] elements = (stream.ReadToEnd()).Split(ch, StringSplitOptions.RemoveEmptyEntries);

size = elements.Length;

a = new int[size];

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

{

a[i] = Convert.ToInt32(elements[i]);

}

}

catch

{

throw new Exception("Не удалось открыть файл");

}

finally

{

stream.Close();

}

}

/// <summary>

/// Метод, меняющий знаки эл-тов массива на противоположные,

/// если большинство эл-тов отрицательные

/// </summary>

/// <returns></returns>

public Arr the\_variant\_method()

{

if (size > 0)

{

//подсчет

int sum = 0;

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

{

if (a[i] < 0) sum++;

}

//замена знаков, если отр. > полож.

if (size - sum < sum)

{

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

{

a[i] = a[i] \* (-1);

}

}

}

return this;

}

}

}

1. **Тест-кейсы:**

Код:

using System;

using Microsoft.VisualStudio.TestTools.UnitTesting;

using Лаб\_11;

namespace UnitTestProject1

{

[TestClass]

public class UnitTest1

{

//тест операнда + (прибавление целого числа)

[TestMethod]

public void TestMethod1()

{

Arr Actual = new Arr(new int[] { 10, -5, 4, 19 });

Actual = Actual + 3;

Arr Expected = new Arr(new int[] { 13, -2, 7, 22 });

Assert.AreEqual<Arr>(Expected, Actual, "Результат не соответствует ожидаемому.");

}

//тест операнда - (разница двух массивов)

[TestMethod]

public void TestMethod2()

{

Arr A = new Arr(new int[] { 10, -5, 4, 19 });

Arr B = new Arr(new int[] { -8, 11 });

Arr Actual;

Actual = A - B;

Arr Expected = new Arr(new int[] { 18, -16, 4, 19 });

Assert.AreEqual<Arr>(Expected, Actual, "Результат не соответствует ожидаемому.");

}

//тест метода, заданного по варианту

[TestMethod]

public void TestMethod3()

{

Arr A = new Arr(new int[] { 33, -1, -9, -10, 0 });

A.the\_variant\_method();

Arr Actual = A;

Arr Expected = new Arr(new int[] { -33, 1, 9, 10, 0 });

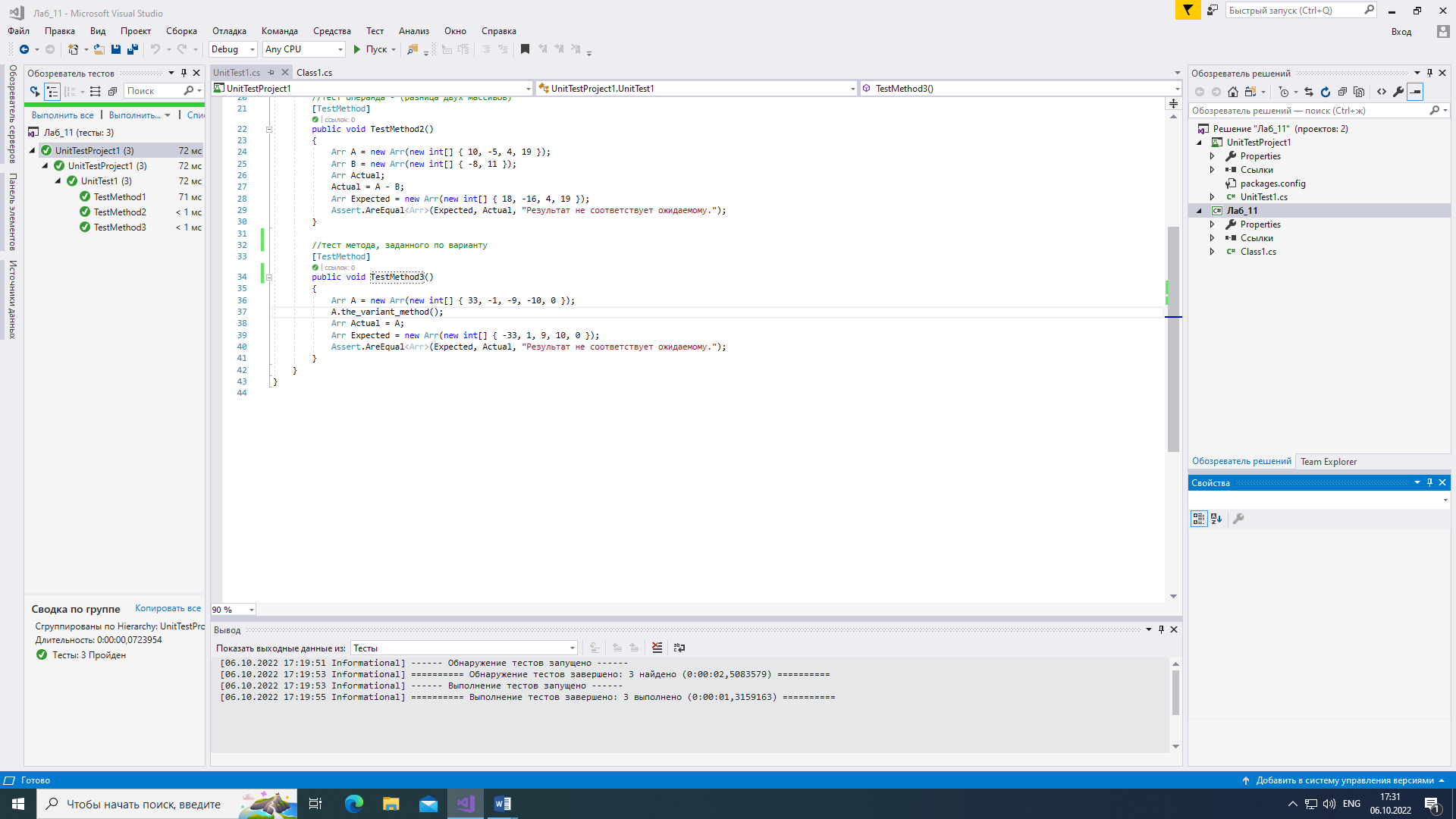
Assert.AreEqual<Arr>(Expected, Actual, "Результат не соответствует ожидаемому.");

}

}

}

Скриншот успешного выполнения:



1. **Код консольного приложения:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using Лаб\_11;

namespace Проверка\_Лаб\_11

{

class Program

{

static void Main(string[] args)

{

Arr A = new Arr(5);

Arr B = new Arr(10);

A.RndInput(50);

B.RndInput(-30, 30);

A.Print();

B.Print();

Console.WriteLine();

Arr S = A + B;

Arr R = A - B;

S.Print();

R.Print();

Console.WriteLine();

S++;

R--;

S.Print();

R.Print();

Console.WriteLine("\n\nКол-во эл-тов массива, кратных 3: " + S.count\_krat(3));

R.RndInput();

R.Print();

}

}

}

1. **Форма:**

Button13

Button14

button3

numericUpDown1

label4

label3

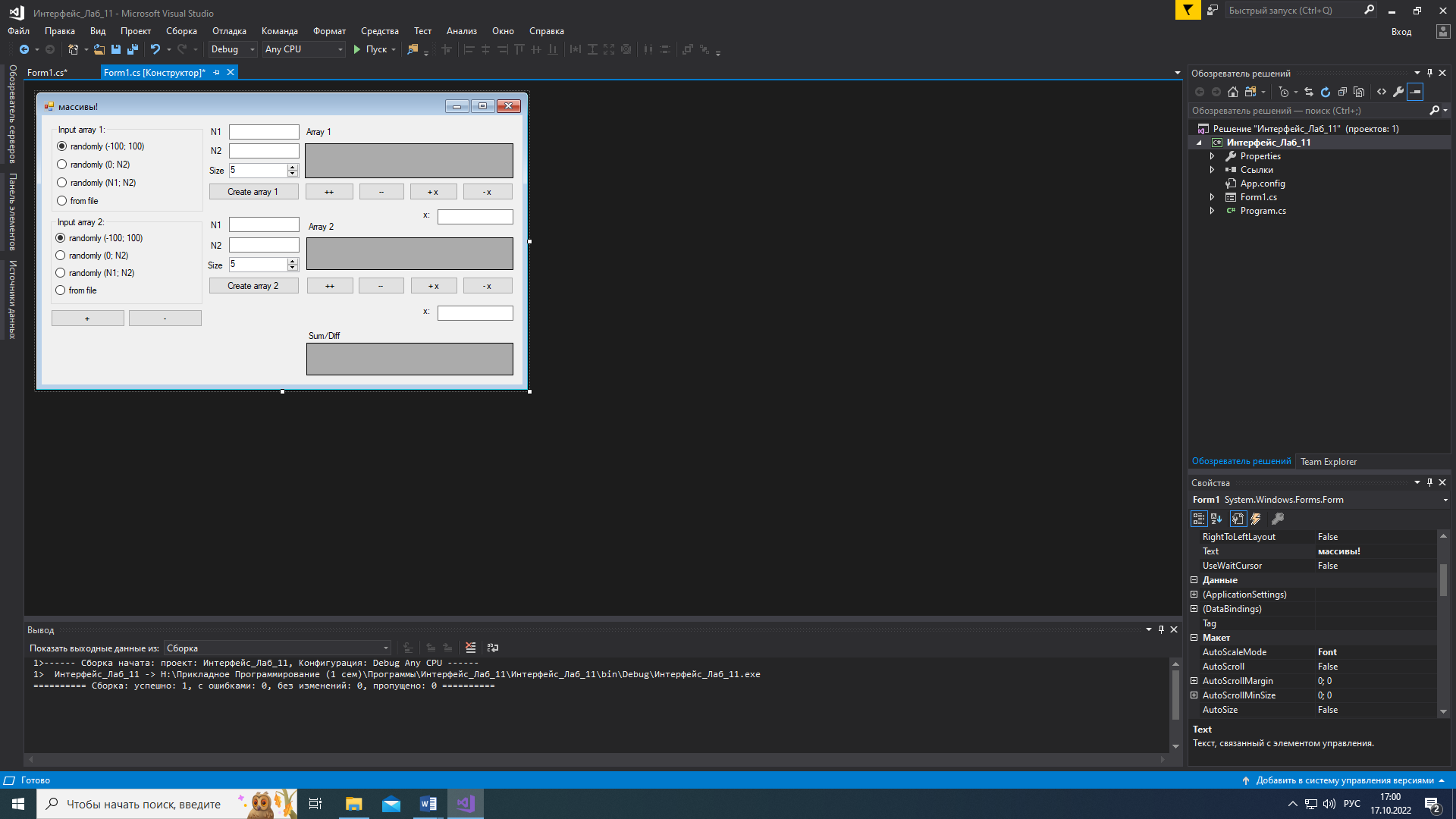
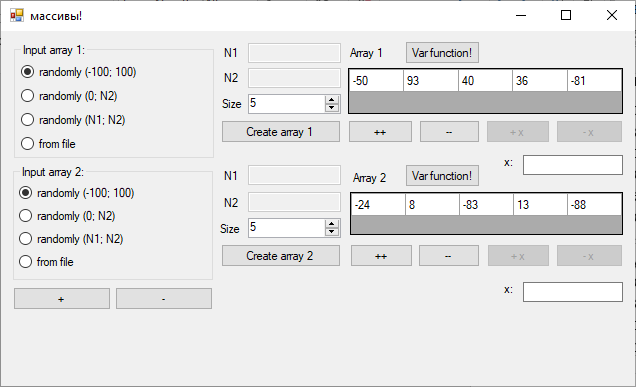
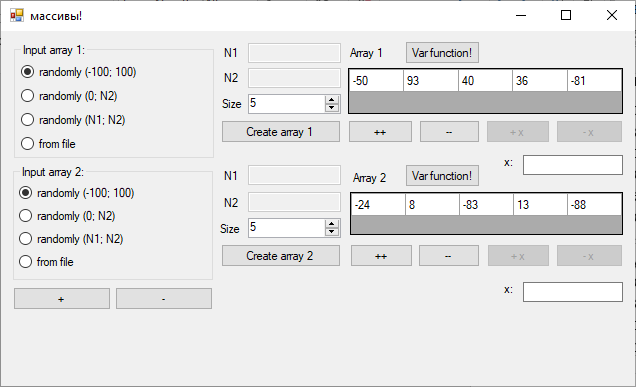
groupBox1

form1

label1

textBox2

label7



radioButton3

radioButton2

label9

textBox5

button8

button7

button6

button5

dataGridView1

label2

dataGridView2

textBox1

button9

button10

button11

button12

radioButton8

radioButton7

radioButton6

radioButton5

groupBox2

radioButton4

radioButton1

label11

button1

textBox6

dataGridView3

textBox3

button2

label10

textBox4

label5

numericUpDown2

button4

label8

label6

1. **Код модуля формы:**

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;

using Лаб\_11;

namespace Интерфейс\_Лаб\_11

{

public partial class Form1 : Form

{

Arr arr1 = null;

Arr arr2 = null;

public Form1()

{

InitializeComponent();

}

/// <summary>

/// load default settings

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void Form1\_Load(object sender, EventArgs e)

{

label11.Visible = false;

dataGridView3.Visible = false;

//size identification

int a1\_s = Convert.ToInt32(numericUpDown1.Value);

int a2\_s = Convert.ToInt32(numericUpDown2.Value);

//creating arrays

arr1 = new Arr(a1\_s);

arr2 = new Arr(a2\_s);

//filling arrays with elements

if(radioButton1.Checked == true)

{

arr1.RndInput();

}

if (radioButton5.Checked == true)

{

arr2.RndInput();

}

//filling tablets

arr1.Print("Array 1", label1, dataGridView1);

arr2.Print("Array 2", label2, dataGridView2);

}

/// <summary>

/// Iput type changing

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void radioButton1\_CheckedChanged(object sender, EventArgs e)

{

if (radioButton1.Checked == true || radioButton4.Checked == true)

{

button3.Enabled = true;

textBox1.Enabled = false;

textBox2.Enabled = false;

}

if (radioButton5.Checked == true || radioButton8.Checked == true)

{

button4.Enabled = true;

textBox3.Enabled = false;

textBox4.Enabled = false;

}

if (radioButton2.Checked == true)

{

textBox1.Enabled = false;

textBox2.Enabled = true;

if (textBox2.Text == "")

button3.Enabled = false;

}

if (radioButton6.Checked == true)

{

textBox3.Enabled = false;

textBox4.Enabled = true;

if (textBox4.Text == "")

button4.Enabled = false;

}

if (radioButton3.Checked == true)

{

textBox1.Enabled = true;

textBox2.Enabled = true;

if (textBox2.Text == "" || textBox1.Text == "")

button3.Enabled = false;

}

if (radioButton7.Checked == true)

{

textBox3.Enabled = true;

textBox4.Enabled = true;

if (textBox3.Text == "" || textBox4.Text == "")

button4.Enabled = false;

}

}

/// <summary>

/// create array 1

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void button3\_Click(object sender, EventArgs e)

{

label11.Visible = false;

dataGridView3.Visible = false;

//creating an array

int a1\_s = Convert.ToInt32(numericUpDown1.Value);

arr1 = new Arr(a1\_s);

//filling an array

try

{

if (radioButton1.Checked)

{

arr1.RndInput();

}

else if (radioButton2.Checked)

{

int N2 = Convert.ToInt32(textBox2.Text);

arr1.RndInput(N2);

}

else if (radioButton3.Checked)

{

int N1 = Convert.ToInt32(textBox1.Text);

int N2 = Convert.ToInt32(textBox2.Text);

arr1.RndInput(N1, N2);

}

else if (radioButton4.Checked)

{

arr1.file\_read("E:\\Колледж\\Прикладное Программирование\\Программы\\Интерфейс\_Лаб\_11\\Интерфейс\_Лаб\_11\\array1.txt");

}

//preparing a tablet

dataGridView1.ColumnCount = a1\_s;

//filling a tablet

arr1.Print("Array 1", label1, dataGridView1);

}

catch (Exception ex)

{

MessageBox.Show("Ошибка: " + ex.Message);

}

}

/// <summary>

/// create array 2

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void button4\_Click(object sender, EventArgs e)

{

label11.Visible = false;

dataGridView3.Visible = false;

//creating an array

int a2\_s = Convert.ToInt32(numericUpDown2.Value);

arr2 = new Arr(a2\_s);

//filling an array

try

{

if (radioButton5.Checked)

{

arr2.RndInput();

}

else if (radioButton6.Checked)

{

int N2 = Convert.ToInt32(textBox4.Text);

arr2.RndInput(N2);

}

else if (radioButton7.Checked)

{

int N1 = Convert.ToInt32(textBox3.Text);

int N2 = Convert.ToInt32(textBox4.Text);

arr2.RndInput(N1, N2);

}

else if (radioButton8.Checked)

{

arr2.file\_read("E:\\Колледж\\Прикладное Программирование\\Программы\\Интерфейс\_Лаб\_11\\Интерфейс\_Лаб\_11\\array2.txt");

}

//preparing a tablet

dataGridView2.ColumnCount = a2\_s;

//filling a tablet

arr2.Print("Array 2", label2, dataGridView2);

}

catch (Exception ex)

{

MessageBox.Show("Ошибка: " + ex.Message);

}

}

/// <summary>

/// check the N1 and N2 boxes if changed

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void textBox1\_TextChanged(object sender, EventArgs e)

{

if (radioButton3.Checked)

{

if (textBox2.Text != "" && textBox1.Text != "")

button3.Enabled = true;

else

button3.Enabled = false;

}

}

private void textBox2\_TextChanged(object sender, EventArgs e)

{

if (radioButton2.Checked)

{

if (textBox2.Text != "")

button3.Enabled = true;

else

button3.Enabled = false;

}

else if (radioButton3.Checked)

{

if (textBox2.Text != "" && textBox1.Text != "")

button3.Enabled = true;

else

button3.Enabled = false;

}

}

private void textBox3\_TextChanged(object sender, EventArgs e)

{

if (radioButton7.Checked)

{

if (textBox4.Text != "" && textBox3.Text != "")

button4.Enabled = true;

else

button4.Enabled = false;

}

}

private void textBox4\_TextChanged(object sender, EventArgs e)

{

if (radioButton6.Checked)

{

if (textBox4.Text != "")

button4.Enabled = true;

else

button4.Enabled = false;

}

else if (radioButton7.Checked)

{

if (textBox4.Text != "" && textBox3.Text != "")

button4.Enabled = true;

else

button4.Enabled = false;

}

}

/// <summary>

/// sum arrays

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void button1\_Click(object sender, EventArgs e)

{

//creating an array

Arr arr3;

arr3 = arr1 + arr2;

//showing it in tablet

dataGridView3.Visible = true;

label11.Visible = true;

arr3.Print("arr1 + arr2 = ", label11, dataGridView3);

}

/// <summary>

/// diff arrays

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void button2\_Click(object sender, EventArgs e)

{

//creating an array

Arr arr3;

arr3 = arr1 - arr2;

//showing it in tablet

dataGridView3.Visible = true;

label11.Visible = true;

arr3.Print("arr1 - arr2 = ", label11, dataGridView3);

}

/// <summary>

/// array1++

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void button5\_Click(object sender, EventArgs e)

{

label11.Visible = false;

dataGridView3.Visible = false;

arr1++;

arr1.Print("Array 1", label1, dataGridView1);

}

/// <summary>

/// array2++

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void button9\_Click(object sender, EventArgs e)

{

label11.Visible = false;

dataGridView3.Visible = false;

arr2++;

arr2.Print("Array 2", label2, dataGridView2);

}

/// <summary>

/// array1--

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void button6\_Click(object sender, EventArgs e)

{

label11.Visible = false;

dataGridView3.Visible = false;

arr1--;

arr1.Print("Array 1", label1, dataGridView1);

}

/// <summary>

/// array2--

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void button10\_Click(object sender, EventArgs e)

{

label11.Visible = false;

dataGridView3.Visible = false;

arr2--;

arr2.Print("Array 2", label2, dataGridView2);

}

/// <summary>

/// array1 + x

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void button7\_Click(object sender, EventArgs e)

{

label11.Visible = false;

dataGridView3.Visible = false;

int x = Convert.ToInt32(textBox5.Text);

arr1 = arr1 + x;

arr1.Print("Array 1", label1, dataGridView1);

}

/// <summary>

/// array2 + x

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void button11\_Click(object sender, EventArgs e)

{

label11.Visible = false;

dataGridView3.Visible = false;

int x = Convert.ToInt32(textBox6.Text);

arr2 = arr2 + x;

arr2.Print("Array 2", label2, dataGridView2);

}

/// <summary>

/// array1 - x

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void button8\_Click(object sender, EventArgs e)

{

label11.Visible = false;

dataGridView3.Visible = false;

int x = Convert.ToInt32(textBox5.Text);

arr1 = arr1 - x;

arr1.Print("Array 1", label1, dataGridView1);

}

/// <summary>

/// array2 - x

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void button12\_Click(object sender, EventArgs e)

{

label11.Visible = false;

dataGridView3.Visible = false;

int x = Convert.ToInt32(textBox6.Text);

arr2 = arr2 - x;

arr2.Print("Array 2", label2, dataGridView2);

}

/// <summary>

/// enable +- x buttons

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void textBox5\_TextChanged(object sender, EventArgs e)

{

if (textBox5.Text != "" && textBox5.Text != "-")

{

button7.Enabled = true;

button8.Enabled = true;

}

else

{

button7.Enabled = false;

button8.Enabled = false;

}

if (textBox6.Text != "" && textBox6.Text != "-")

{

button11.Enabled = true;

button12.Enabled = true;

}

else

{

button11.Enabled = false;

button12.Enabled = false;

}

}

/// <summary>

/// check textbox (only nums, "-" and backspace)

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void textBox1\_KeyPress(object sender, KeyPressEventArgs e)

{

if (e.KeyChar == (char)Keys.Back)

return;

if (e.KeyChar > '0' && e.KeyChar <= '9')

return;

if (e.KeyChar == '-' && ((TextBox)sender).Text == "")

return;

if (e.KeyChar == '0' && ((TextBox)sender).Text != "" && ((TextBox)sender).Text != "-")

return;

e.KeyChar = '\0';

}

/// <summary>

/// the variant function

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void button13\_Click(object sender, EventArgs e)

{

arr1 = arr1.the\_variant\_method();

arr1.Print("Array 1", label1, dataGridView1);

}

private void button14\_Click(object sender, EventArgs e)

{

arr2 = arr2.the\_variant\_method();

arr2.Print("Array 2", label2, dataGridView2);

}

}

}

1. **Тестовый сценарий:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Проверяемые требования** | **Сообщения программы и вводимые значения** | **Ожидаемые результаты** | **Результаты** |
| Способность программы обеспечить выбор способа ввода исходных данных | Нажать переключатель:  «randomly (-100; 100)» | Случайные числовые значения в диапазоне от -100 до 100 отображаются в соответствующей таблице по нажатию кнопки «Create array» | Соответствует ожиданиям.  (см. Рисунок 1) |
| Нажать переключатель:  «randomly (0; N2)» | При заполненном поле N2 cлучайные числовые значения в диапазоне от 0 до N2 отображаются в соответствующей таблице по нажатию кнопки «Create array» | Соответствует ожиданиям.  (см. Рисунок 2) |
| Нажать переключатель:  «randomly (N1; N2)» | При заполненных полях N1 N2 cлучайные числовые значения в диапазоне от N1 до N2 отображаются в соответствующей таблице по нажатию кнопки «Create array» | Соответствует ожиданиям.  (см. Рисунок 3) |
| Нажать переключатель:  «from file» | По нажатию кнопки «Create array» числовые значения из соответствующего файла отображаются в соответствующей таблице | Соответствует ожиданиям.  (см. Рисунок 4) |
| Способность программы обеспечить контроль вводимых значений | Ввести символы в поля ввода | В полях допустимы только цифры, минус и клавиша BackSpace.  Недопустимые символы клавиатуры блокируются и не отображаются в поле ввода. | Соответствует ожиданиям. |
| Способность программы обеспечить контроль пустых полей ввода | Попробовать оставить поле ввода пустым | Кнопки блокируются при наличии пустых полей ввода, необходимых для выполнения задачи | Соответствует ожиданиям.  (см. Рисунок 5) |
| Способность программы вычислять сумму/разность двух массивов и выводить результат в отдельную таблицу | Массив 1:  (заполнен случайно)  45 -47 59 -91 65  Массив 2:  (из файла)  6 2 -13 2 98 | В таблицу 3 выводится результат сложения двух массивов поэлементно, размер результирующего массива по размеру большего из них.  Массив 3:  51 -45 46 -89 163 | Соответствует ожиданиям.  (См. рисунок 6) |
| Способность программы генерировать массив при нажатии соответствующей кнопки и выводить его в соответствующую таблицу | Выбрать способ ввода массива, заполнить необходимые поля, нажать кнопку «Create array» | В соответствующую таблицу выведется массив согласно способу ввода | Соответствует ожиданиям. |
| Способность программы прибавлять/вычитать число x к/от каждого элемента соответствующего массива при нажатии кнопки | Создать массив, заполнить поле числа x, нажать кнопку –х/+х для соответстующего массива | Число х прибавится/вычтется к/из соответствующего массива | Соответствует ожиданиям. |
| Способность программы выполнять метод по варианту | Сгенерировать массив, в котором будет   1. Больше отрицательных эл-тов 2. Больше положительных эл-тов   Нажать кнопку «var function!» | В случае 1 все знаки у эл-тов массива поменяются на противоположные, в случае 2 - нет | Результаты соответствуют ожиданиям |

1. **Скриншоты в процессе испытания:**

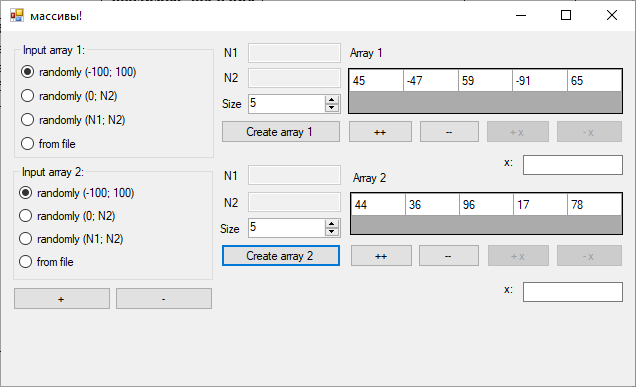


Рисунок 1

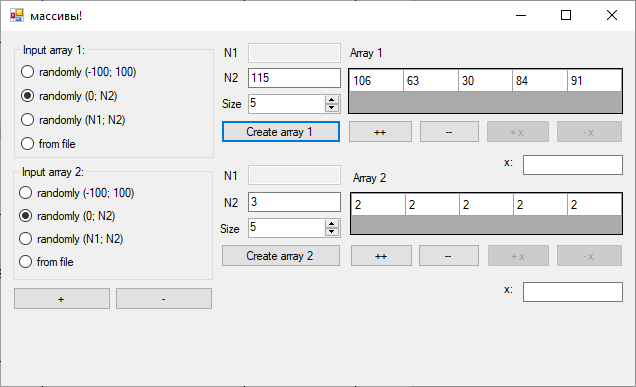


Рисунок 2

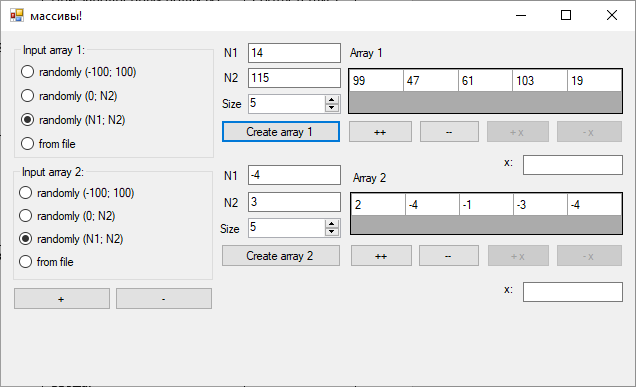


Рисунок 3

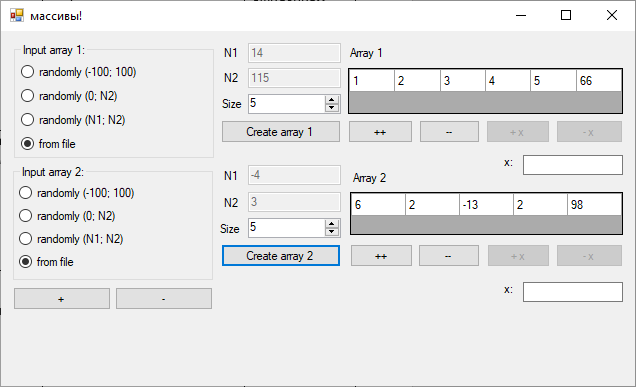


Рисунок 4

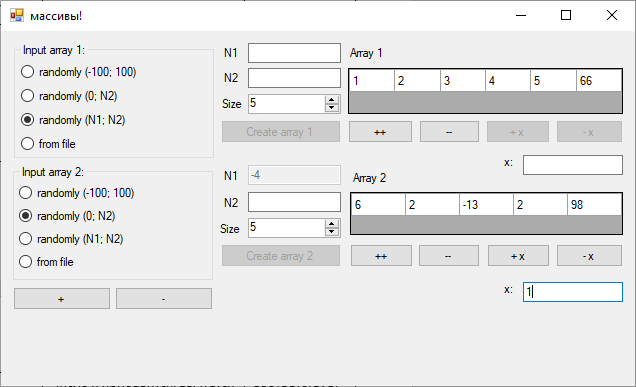


Рисунок 5

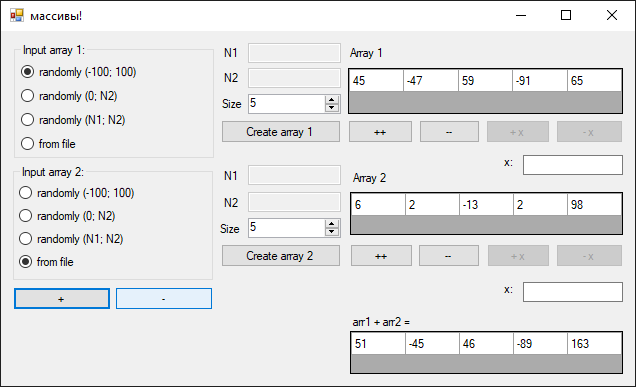


Рисунок 6