

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования «Тульский государственный университет»

КАФЕДРА ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

РЕШЕНИЕ ЗАДАЧ ПО ПРОЙДЕННЫМ ТЕМАМ

отчет о лабораторной работе №14

по дисциплине
ТЕХНОЛОГИИ И МЕТОДЫ ПРОГРАММИРОВАНИЯ

ВАРИАНТ 13

Выполнила:	студент гр. 230711	Павлова В.С.
Проверил:	ассистент каф. ИБ	Курбаков М.Ю.

Тула, 2023 г.

ЦЕЛЬ РАБОТЫ

Цель: повторить ранее изученные принципы и применить их на практике при решении задач на различные темы.

ЗАДАНИЕ НА РАБОТУ

В данной работе требуется выполнить следующие задачи в соответствии с вариантом:

Задание №1. *Создание функций в окне форм* (вариант 1)

Написать функцию Func_Z, которая принимает три целых числа в качестве параметров и возвращает целое число равное удвоенной сумме переданных параметров. Запустить функцию Func_Z из процедуры Proba_2. Результат должен быть выведен при помощи окна MsgBox.

Задание №2. *Создание функций на рабочем листе* (вариант 1)

Написать функцию, вычисляющую утроенную сумму двух целых чисел, которые передаются в функцию качестве параметров. Используя стандартный вызов функции с рабочего листа рабочей книги MS Excel, обеспечить вычисление утроенных сумм чисел, помещенных в ячейки «A13» и «B17» рабочего листа «Лист1». Результат вычислений должен помещаться в ячейку «C22».

Задание №3. *Работа с файлами* (вариант 6)

Написать программу, считывающую число из ячейки B3 листа Sheet2 в переменную X, число из ячейки B3 того же листа – в переменную Y. Вычислить $Z=X^2+Y*3+15$. Результат записать в файл output.txt

Задание №4. *Работа со списками* (вариант 3)

Открыть текстовый файл, в котором записана одна строка, содержащая два числа и еще одна строка, содержащая одно число. Считать эти три числа, вычислить их утроенную сумму, вывести результат в выходной файл и в окно

сообщений MsgBox. Имя выходного файла должно быть задано с использованием процедуры InputBox.

Задание №5. Работа с листом и формами, циклы (вариант 1)

На заданном рабочем листе заданной рабочей книги создать таблицу заказов, содержащую информацию о пяти товарах: Название товара, количество, цена, единицы измерения, сумма. Написать макрос, который выдает отчет о сделанных покупках с использованием окна MsgBox. Отчет должен формироваться на основании информации, содержащейся в таблице заказов.

Задание №6. Работа с условиями (вариант 3)

Создать процедуру Proba_Z3, которая запускается при открытии рабочей книги и в окне InputBox сначала запрашивает имя студента, а затем его фамилию. В итоге программа должна в окне сообщения MsgBox по имени и фамилии поздравить студента с окончанием летней сессии, если его фамилия отличается от Иванов или Иванова. В противном случае необходимо просто поздороваться.

Задание №7. Работа с массивами (вариант 3)

Написать функцию F_SORT, получающую в качестве параметра по ссылке массив целочисленных элементов и возвращающую значение максимального элемента из данного массива. Результат сортировки остается в том же массиве. Написать процедуру, в которой необходимо задать массив из десяти элементов: 3, 71, 1, 11, 54, 4, 67, 8, 91, 1 и вызвать функцию F_SORT для данного массива.

ТЕКСТ ПРОГРАММЫ ДЛЯ ЗАДАНИЯ №1

Для описания форм в качестве альтернативы VBA был использован язык C# и платформа WinForms. В листинге 1 представлено содержимое главной формы управления. В листинге 2 – описание формы для реализации функции утроения суммы для задания №1.

Листинг 1. Описание формы MainForm.cs

```
public partial class MainForm : Form
{
    public MainForm()
    {
        InitializeComponent();
    }

    private void buttonTask7_Click(object sender, EventArgs e)
    {
        var workForm = new WorkForm7();
        workForm.ShowDialog();
    }

    private void buttonTask6_Click(object sender, EventArgs e)
    {
        var workForm = new WorkForm6();
        workForm.ShowDialog();
    }

    private void buttonTask5_Click(object sender, EventArgs e)
    {
        var workForm = new WorkForm5();
        workForm.ShowDialog();
    }

    private void buttonTask4_Click(object sender, EventArgs e)
    {
        var workForm = new WorkForm4();
        workForm.ShowDialog();
    }

    private void buttonTask3_Click(object sender, EventArgs e)
    {
        var workForm = new WorkForm3();
        workForm.ShowDialog();
    }

    private void buttonTask2_Click(object sender, EventArgs e)
    {
        var workForm = new WorkForm2();
        workForm.ShowDialog();
    }

    private void buttonTask1_Click(object sender, EventArgs e)
    {
        var workForm = new WorkForm1();
        workForm.ShowDialog();
    }
}
```

Листинг 2. Описание формы WorkForm1.cs

```
public partial class WorkForm1 : Form
{
    public WorkForm1()
    {
        InitializeComponent();
    }

    private void buttonCalc_Click(object sender, EventArgs e)
    {
        if (textBoxA.Text.Length > 0
            && textBoxB.Text.Length > 0
            && textBoxC.Text.Length > 0)
        {
            int A = int.Parse(textBoxA.Text);
            int B = int.Parse(textBoxB.Text);
            int C = int.Parse(textBoxC.Text);

            textBoxResult.Text = ((A + B + C) * 2).ToString();
        }
    }

    private void textBoxA_KeyPress(object sender, KeyPressEventArgs e)
    {
        char ch = e.KeyChar;

        if (!Char.IsDigit(ch) && ch != 8)
        {
            e.Handled = true;
        }
    }

    private void textBoxB_KeyPress(object sender, KeyPressEventArgs e)
    {
        char ch = e.KeyChar;

        if (!Char.IsDigit(ch) && ch != 8)
        {
            e.Handled = true;
        }
    }

    private void textBoxC_KeyPress(object sender, KeyPressEventArgs e)
    {
        char ch = e.KeyChar;

        if (!Char.IsDigit(ch) && ch != 8)
        {
            e.Handled = true;
        }
    }

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

ДЕМОНСТРАЦИОННЫЙ ПРИМЕР ДЛЯ ЗАДАНИЯ №1

Для решения задач по варианту было разработано одно общее приложение, разделённое по функциональности в соответствии с требованиями каждого задания. Для переключения между ними используются кнопки на главной форме. При запуске приложения отображается сама главная форма (рисунок 1):

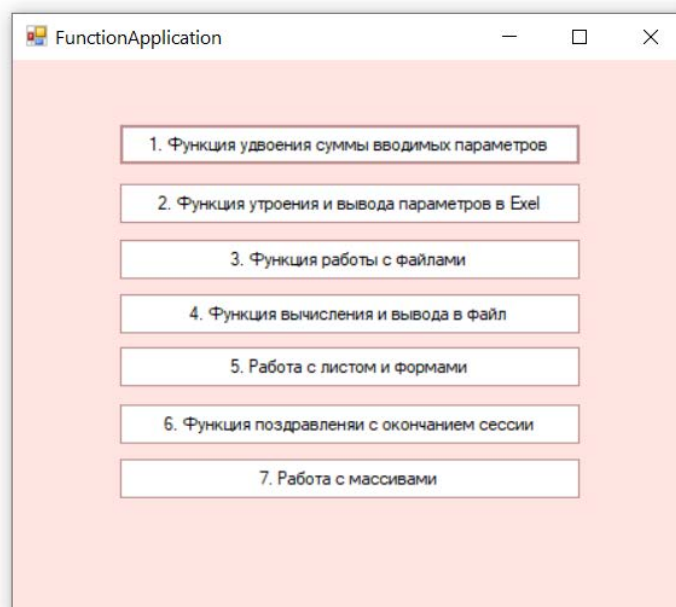


Рисунок 1 – Главная форма для управления процедурами

При нажатии на первую кнопку становится активна рабочая форма для первого задания. Форма позволяет ввести три параметра и вывести их утроенную сумму по нажатии кнопки «Calculate». Пример работы с данной формой приведён на рисунке 2.

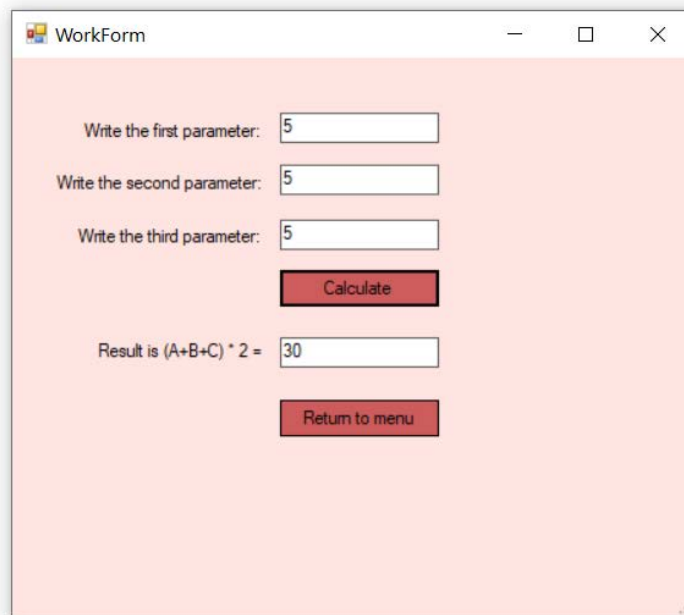


Рисунок 2 – Пример работы первой функции

ТЕКСТ ПРОГРАММЫ ДЛЯ ЗАДАНИЯ №2

В листинге 3 представлено описание формы для реализации функции удвоения суммы и вывода в Excel для задания №2.

Листинг 3. Описание формы WorkForm2.cs

```
public partial class WorkForm2 : Form
{
    public WorkForm2()
    {
        InitializeComponent();
    }

    private void buttonLoadFromExcel_Click(object sender, EventArgs e)
    {
        Excel.Application xlApp = new Excel.Application();
        Excel.Workbook xlWorkbook =
        xlApp.Workbooks.Open(@"C:\Users\Вика\Documents\WorkList.xlsx");
        Excel.Worksheet xlWorksheet = xlWorkbook.Sheets["Лист1"];

        double a13Value = xlWorksheet.Range["A13"].Value;
        double b17Value = xlWorksheet.Range["B17"].Value;

        textBoxA.Text = a13Value.ToString();
        textBoxB.Text = b17Value.ToString();

        xlWorksheet.Range["C22"].Value = (a13Value + b17Value)*3;

        textBoxResult.Text = ((a13Value + b17Value) * 3).ToString();

        Excel.Workbook newWorkbook = xlApp.Workbooks.Add();
        Excel.Worksheet newWorksheet = newWorkbook.Sheets[1];
    }
}
```

Листинг 3. Описание формы WorkForm2.cs (продолжение)

```
newWorksheet.Range["A13"].Value = a13Value;  
newWorksheet.Range["B17"].Value = b17Value;  
newWorksheet.Range["C22"].Value = (a13Value + b17Value) * 3;  
  
string filename = @"C:\Users\Вика\Documents\WorkList.xlsx" +  
DateTime.Now.ToString("yyyyMMddHHmmss") + ".xlsx";  
newWorkbook.SaveAs(filename);  
newWorkbook.Close();  
xlWorkbook.Close();  
xlApp.Quit();  
}  
private void returnButton_Click(object sender, EventArgs e)  
{  
    this.Close();  
}  
}
```

ДЕМОНСТРАЦИОННЫЙ ПРИМЕР ДЛЯ ЗАДАНИЯ №2

Форма позволяет считать данные из ячейки «A13» и «B17» рабочего листа «Лист1». Результат вычислений должен помещается в ячейку «C22» по нажатию кнопки «Calculate». Для проверки данные выводятся в окно формы. Пример работы с данной формой приведён на рисунке 3.

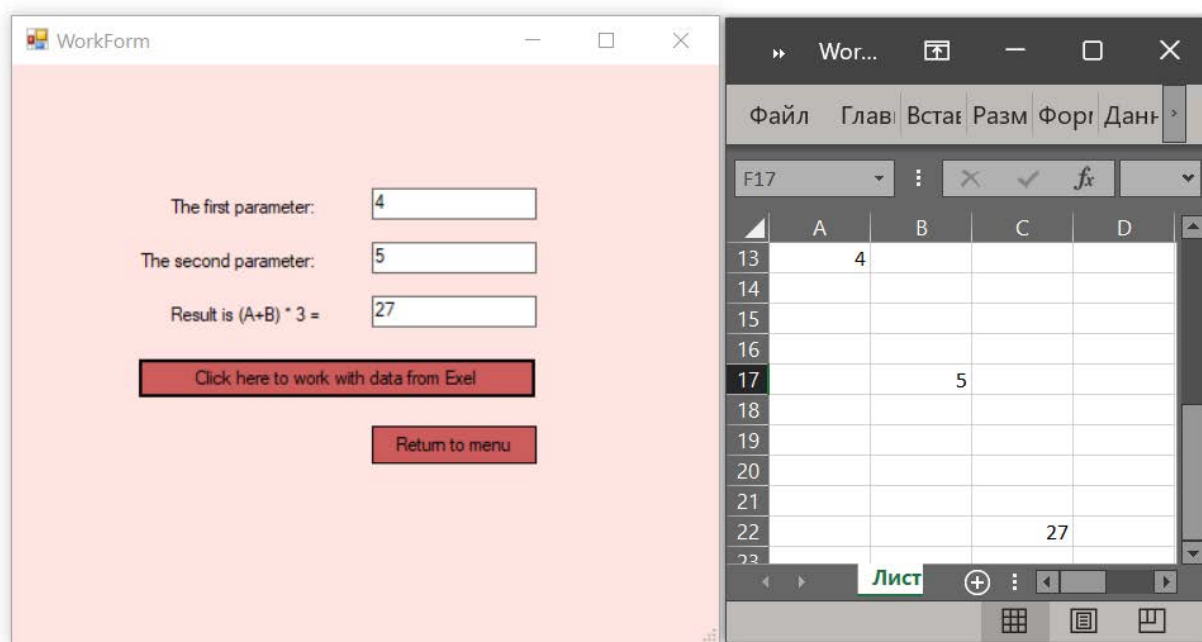


Рисунок 3 – Пример работы второй функции

ТЕКСТ ПРОГРАММЫ ДЛЯ ЗАДАНИЯ №3

В листинге 4 представлено описание формы для реализации функции вычислений и вывода результата в текстовый файл для задания №3.

Листинг 4. Описание формы WorkForm3.cs

```
public partial class WorkForm3 : Form
{
    public WorkForm3()
    {
        InitializeComponent();
    }

    private void buttonLoadFromExel_Click(object sender, EventArgs e)
    {
        Excel.Application xlApp = new Excel.Application();
        Excel.Workbook xlWorkbook =
xlApp.Workbooks.Open(@"C:\Users\Вика\Documents\WorkList2.xlsx");
        Excel.Worksheet xlWorksheet = xlWorkbook.Sheets["Лист1"];

        double X = xlWorksheet.Range["B3"].Value;
        double Y = X;

        textBoxA.Text = X.ToString();

        textBoxResult.Text = (X * X + 3 * Y + 15).ToString();

        Excel.Workbook newWorkbook = xlApp.Workbooks.Add();
        Excel.Worksheet newWorksheet = newWorkbook.Sheets[1];

        newWorksheet.Range["B3"].Value = X;
        newWorksheet.Range["B4"].Value = X * X + 3 * Y + 15;

        string fileExelName = @"C:\Users\Вика\Documents\WorkList2.xlsx" +
DateTime.Now.ToString("yyyyMMddHHmmss") + ".xlsx";
        newWorkbook.SaveAs(fileExelName);

        newWorkbook.Close();
        xlWorkbook.Close();
        xlApp.Quit();

        string fileTXTName = @"C:\Users\Вика\Documents\output.txt";
        using (StreamWriter sw = File.CreateText(fileTXTName))
        {
            sw.WriteLine("Результат: " + (X * X + 3 * Y + 15).ToString());
        }
    }

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

ДЕМОНСТРАЦИОННЫЙ ПРИМЕР ДЛЯ ЗАДАНИЯ №3

Форма позволяет считать данные из ячейки «B3» рабочего листа «Лист1». Результат вычислений помещается в ячейку «B4» по нажатию соответствующей кнопки, а также выводится в текстовый файл. Для проверки данные выводятся в окно формы. Пример работы с данной формой приведён на рисунке 4.

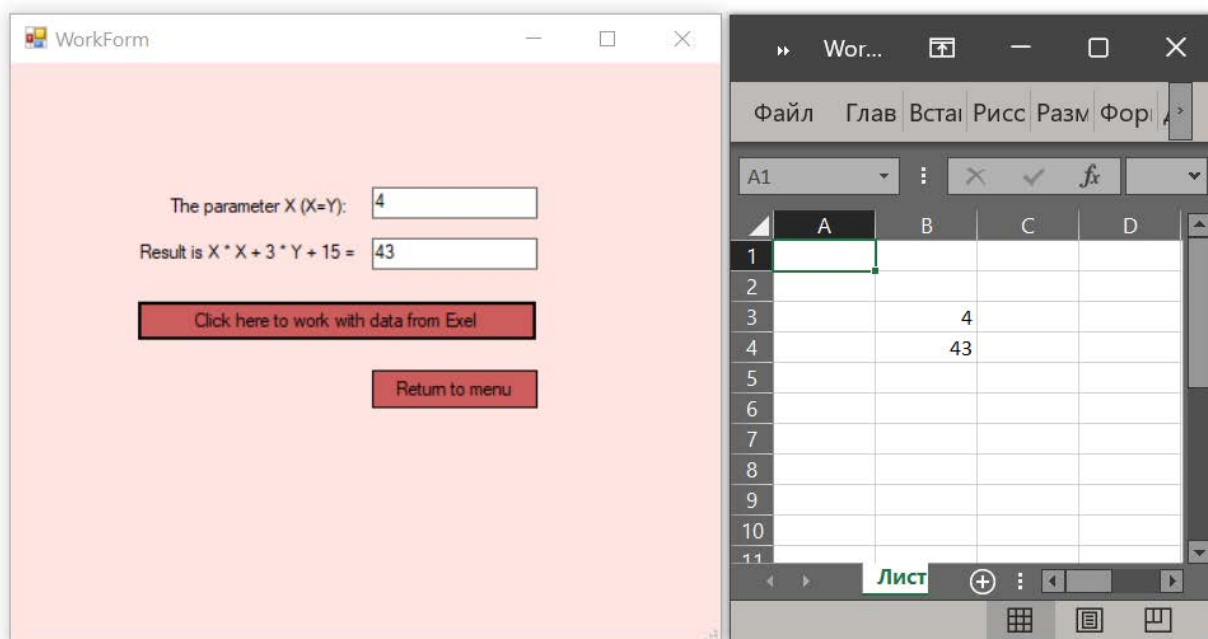


Рисунок 4 – Пример работы третьей функции

Результат вычислений, как и требуется в задании, выводится также в текстовый файл, как показано на рисунке 5.

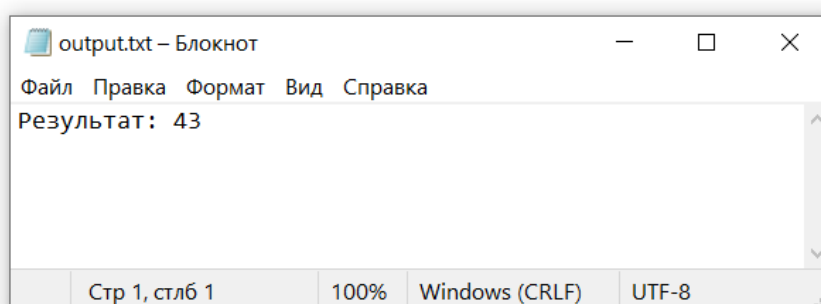


Рисунок 5 – Вывод результата вычислений третьей функции

ТЕКСТ ПРОГРАММЫ ДЛЯ ЗАДАНИЯ №4

В листинге 5 представлено описание формы для реализации функции считывания данных из файла и вывода результата вычислений в текстовый файл для задания №4.

Листинг 5. Описание формы WorkForm4.cs

```
public partial class WorkForm4 : Form
{
    public WorkForm4()
    {
        InitializeComponent();
    }
}
```

Листинг 5. Описание формы WorkForm4.cs (продолжение)

```
private void return_Click(object sender, EventArgs e) => this.Close();

private void buttonLoadFromTxt_Click(object sender, EventArgs e)
{
    var reader = new StreamReader(@"C:\Users\Вика\Documents\input.txt");

    string line = reader.ReadLine();
    string[] nums = line.Split();
    int X = int.Parse(nums[0]);
    int Y = int.Parse(nums[1]);

    line = reader.ReadLine();
    int Z = int.Parse(line);

    if (textBoxPath.Text != string.Empty)
    {
        StreamWriter writer = new StreamWriter(textBoxPath.Text);
        writer.WriteLine((X + Y + Z) * 3);
        textBoxResult.Text = ((X + Y + Z) * 3).ToString();
        writer.Close();
    }
    reader.Close();
}
```

ДЕМОНСТРАЦИОННЫЙ ПРИМЕР ДЛЯ ЗАДАНИЯ №4

Форма позволяет считать данные из текстового файла input.txt, произвести вычисления и вывести данные по вводимому пути (в текстовый файл). Необходимо вводить корректный путь. Для проверки результат выводится и в форму тоже. Пример работы с данной формой приведён на рисунке 6.

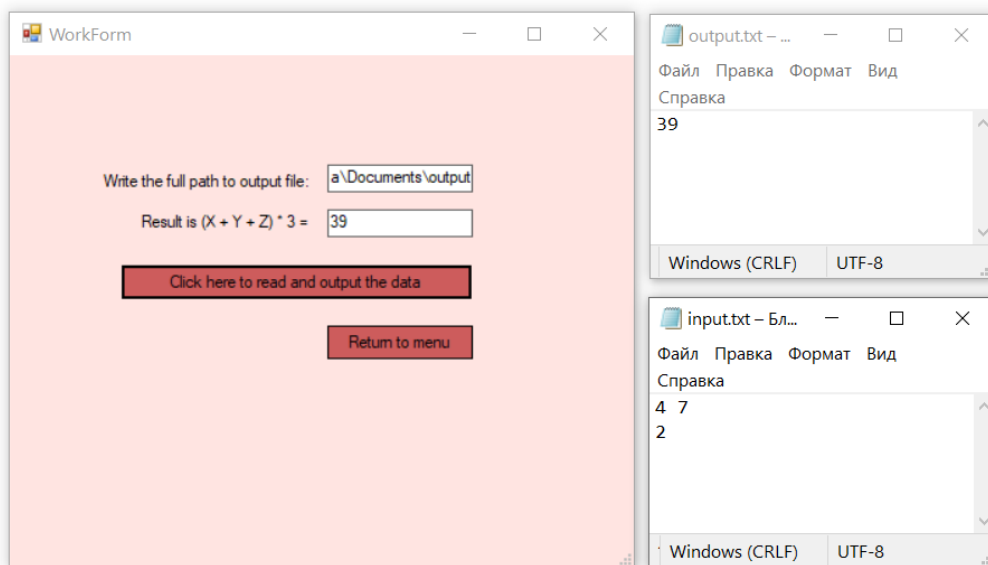


Рисунок 6 – Вывод результата вычислений четвёртой функции

ТЕКСТ ПРОГРАММЫ ДЛЯ ЗАДАНИЯ №5

В листинге 6 представлено описание формы для реализации обработки Excel-таблицы и вывода данных в форму для задания №5.

Листинг 6. Описание формы WorkForm5.cs

```
public partial class WorkForm5 : Form
{
    public WorkForm5()
    {
        InitializeComponent();
    }

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

    private void buttonMakeReport_Click(object sender, EventArgs e)
    {
        double totalSum = 0;
        string filePath = @"C:\Users\Вика\Documents\Orders.xlsx";
        Excel.Application xlApp = new Excel.Application();
        Excel.Workbook xlWorkBook = xlApp.Workbooks.Open(filePath);
        Excel.Worksheet xlWorkSheet;
        Excel.Range range;

        xlWorkSheet = (Excel.Worksheet)xlWorkBook.Worksheets.get_Item(1);

        range = xlWorkSheet.get_Range("A1", "E6");

        var table = new System.Data.DataTable();

        for (int i = 1; i <= range.Columns.Count; i++)
        {
            table.Columns.Add(range.Cells[1, i].Value.ToString());
        }

        for (int i = 2; i <= range.Rows.Count; i++)
        {
            DataRow row = table.Rows.Add();
            for (int j = 1; j <= range.Columns.Count; j++)
            {
                row[j - 1] = range.Cells[i, j].Value.ToString();

                if (j == 5)
                {
                    totalSum += Convert.ToDouble(range.Cells[i, j].Value);
                }
            }
        }

        dataGridViewOrders.DataSource = table;

        textBoxTotalSum.Text = totalSum.ToString();
        xlWorkBook.Close(true, null, null);
        xlApp.Quit();
    }
}
```

ДЕМОНСТРАЦИОННЫЙ ПРИМЕР ДЛЯ ЗАДАНИЯ №5

Форма позволяет считать данные из Excel-таблицы, подсчитать итоговую сумму заказа и вывести данные в форму. Пример работы с данной формой приведён на рисунке 7.

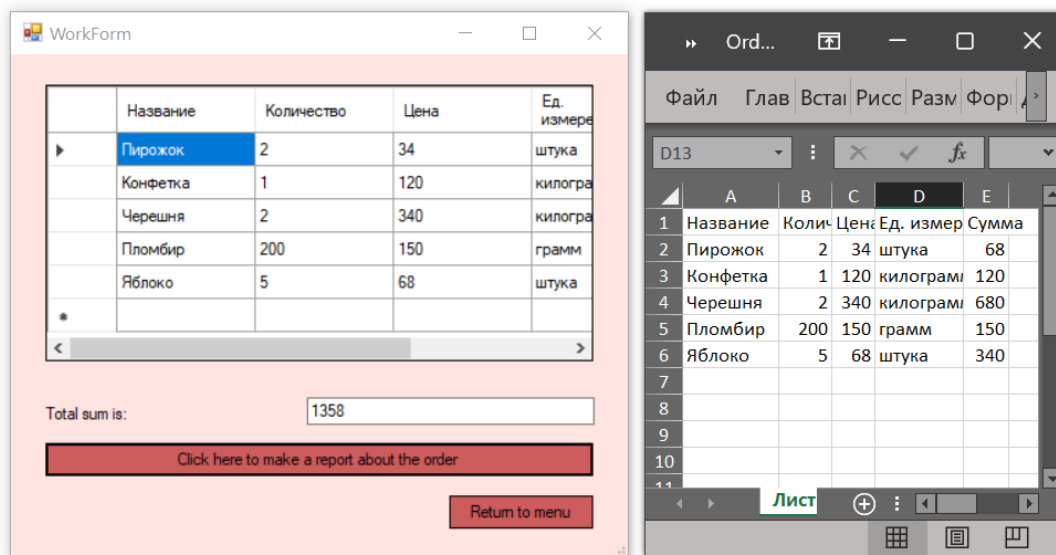


Рисунок 7 – Пример работы пятой функции

ТЕКСТ ПРОГРАММЫ ДЛЯ ЗАДАНИЯ №6

В листинге 7 представлено описание формы для реализации обработки данных из окон ввода для задания №6.

Листинг 7. Описание формы WorkForm6.cs

```
public partial class WorkForm6 : Form
{
    public WorkForm6() => InitializeComponent();

    private void buttonCheckStudent_Click(object sender, EventArgs e)
    {
        if (textBoxSurname.Text == "Иванов"
            || textBoxSurname.Text == "Иванова")
        {
            MessageBox.Show("Здравствуйте! Вы не закрыли сессию :(");
        }
        else
        {
            MessageBox.Show($"Здравствуйте, {textBoxFirstName.Text} {textBoxSurname.Text}! Поздравляем с окончанием летней сессии!");
        }
    }

    private void returnButton_Click(object sender, EventArgs e) =>
        this.Close();
}
```

ДЕМОНСТРАЦИОННЫЙ ПРИМЕР ДЛЯ ЗАДАНИЯ №6

Форма позволяет считать данные из окон ввода имени и фамилии и, в зависимости от данных, поздравляет с окончанием сессии. Или не поздравляет. Пример работы с данной формой приведён на рисунке 8.

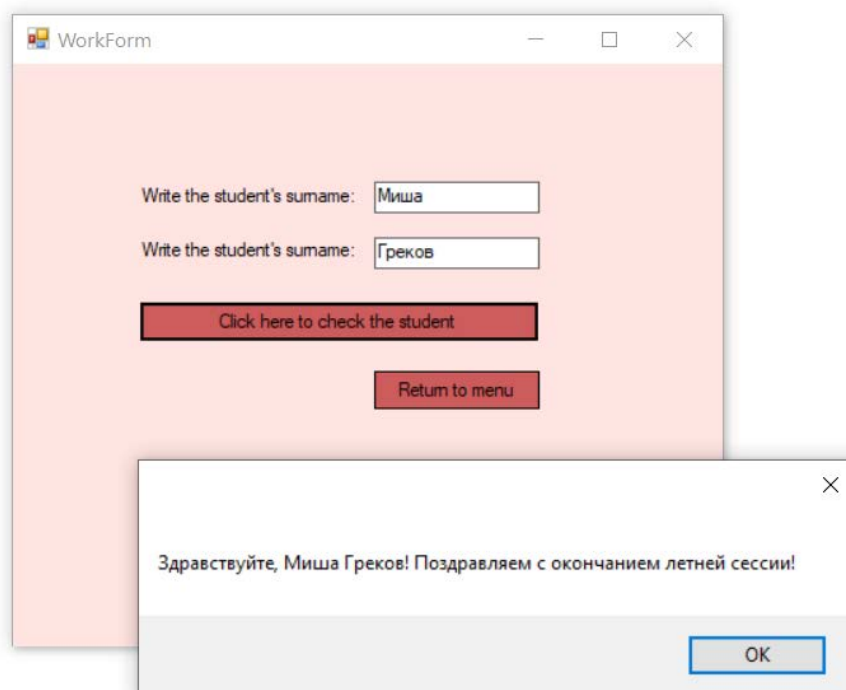


Рисунок 8 – Пример работы шестой функции

ТЕКСТ ПРОГРАММЫ ДЛЯ ЗАДАНИЯ №7

В листинге 8 представлено описание формы для реализации обработки массива для задания №7.

Листинг 8. Описание формы WorkForm7.cs

```
public partial class WorkForm7 : Form
{
    public WorkForm7() => InitializeComponent();

    private void buttonSort_Click(object sender, EventArgs e)
    {
        var array = ReadArray();
        SortingArray(ref array);
        int max = array[array.Length - 1];
        textBoxArray.Text = string.Join(" ", array);
        textBoxMax.Text = max.ToString();
    }
    static void SortingArray(ref int[] arr) => Array.Sort(arr);

    private int[] ReadArray()
    {

```

Листинг 8. Описание формы WorkForm7.cs (продолжение)

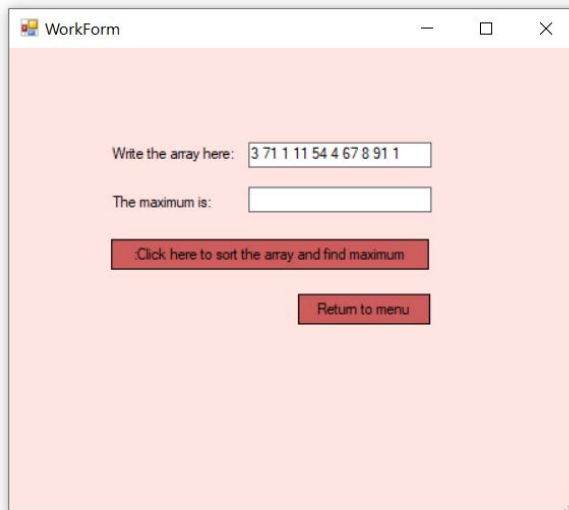
```
string[] input = textBoxArray.Text.Split(' ');
int[] array = new int[input.Length];
for (int i = 0; i < input.Length; i++)
{
    array[i] = int.Parse(input[i]);
}
return array;
}

private void returnButton_Click(object sender, EventArgs e) =>
this.Close();
}
```

ДЕМОНСТРАЦИОННЫЙ ПРИМЕР ДЛЯ ЗАДАНИЯ №7

Форма позволяет считать массив целых чисел, отсортировать его и вывести максимум. Пример работы с данной формой приведён на рисунке 9.

А)



Б)

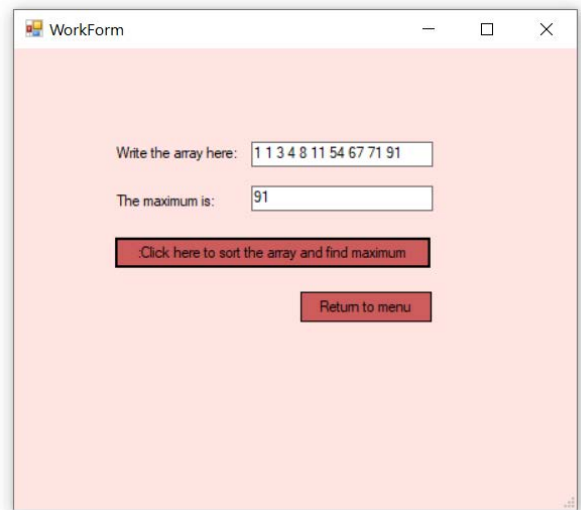


Рисунок 9 – Пример работы шестой функции: а) массив до сортировки; б) массив после сортировки

ВЫВОД

В ходе выполнения данной работы на примере интерфейса программирования приложений Windows Forms я повторила изученные принципы работы с формами и их элементами, а также выполнила предложенные задания.