Федеральное государственное образовательное бюджетное учреждение высшего образования

**«Финансовый университет при Правительстве Российской Федерации» (Финансовый университет)**

Колледж информатики и программирования

Специальность 09.02.03 Программирование в компьютерных системах

**ОТЧЕТ**

**ПО УЧЕБНОЙ ПРАКТИКЕ**

УП.01.01 Выполнение работ по одной

Профессиональный модуль \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*(наименование профессионального модуля)*

или нескольким профессиям рабочих, должностям служащих

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Выполнил:

420

Обучающаяся учебной группы №\_\_\_\_\_\_\_\_

(номер группы)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_

Ильиных Д.Г.

*(И.О. Фамилия) (подпись)*

Проверил:

Руководитель практики от колледжа:

Е. Л. Альшакова, И. В. Сиберев

*(И.О. Фамилия)*

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_

*(оценка) (подпись)*

Москва – 2022г.

(Первая учебная практика, C#)

**Практическая работа №1**

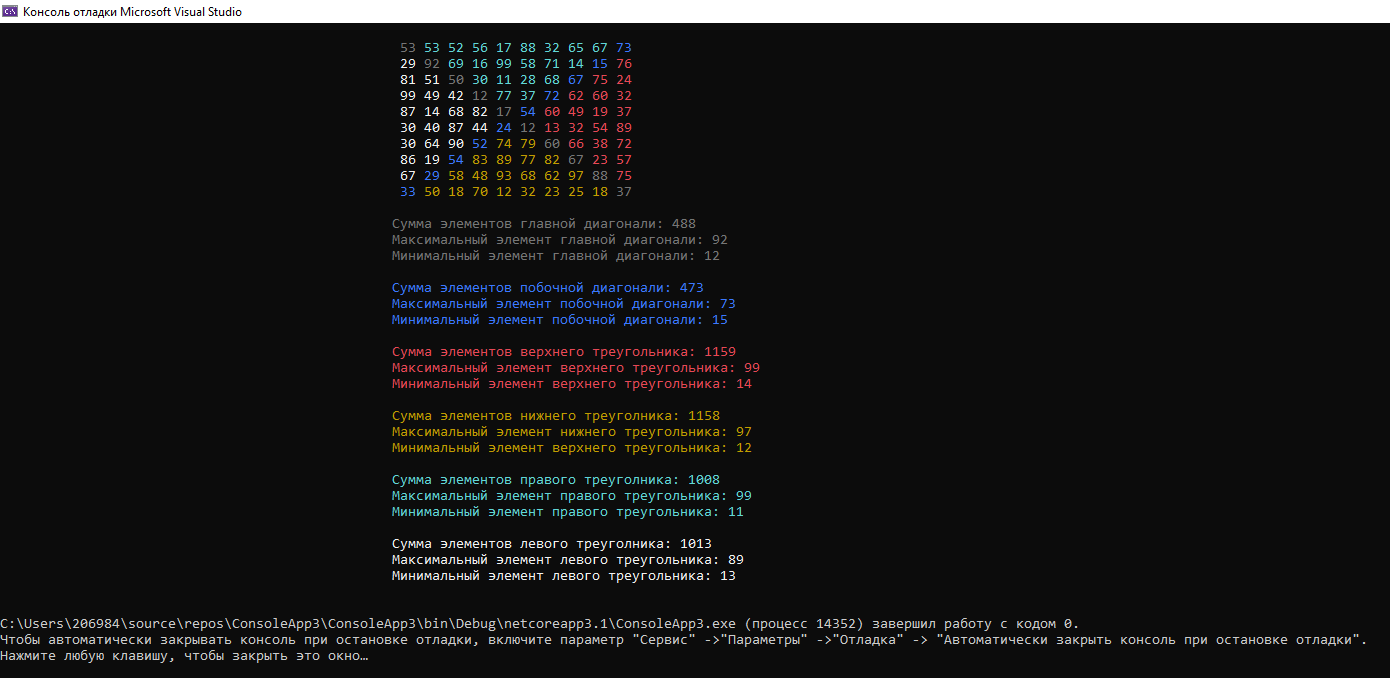
Требуется в консольном проекте создать двумерный числовой массив, заполнить его случайными целыми числами в диапазоне от 10 до 99.

Вывести массив в консоль, установив разный цвет текста для следующих зон массива:

* Зоны главной и побочной диагоналей массива
* Зона выше главной и выше побочной диагоналей (верхний треугольник)
* Зона ниже главной и ниже побочной диагоналей (нижний треугольник)
* Зона выше главной и ниже побочной диагоналей (правый треугольник)
* Зона ниже главной и выше побочной диагоналей (левый треугольник).

Для каждой из зон вычислить следующие значения: sum, min, max, average.

Вывести в консоль эти значения с комментариями, используя для каждого из них такой же цвет текста, как и у соответствующей зоны массива.



using System;

class Program

{

static void Main(string[] args)

{

int n = 10;

int k = 10;

var array = new int[n, k];

var random = new Random();

Console.ForegroundColor = ConsoleColor.Black;

Console.WriteLine();

Colouring(n, k);

Console.WriteLine();

}

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

/// <param name="k">Lenght</param>

static void Colouring(int n, int k)

{

var array = new int[n, k];

var random = new Random();

int sum1 = 0, sum2 = 0, sum3 = 0, sum4 = 0, sum5 = 0, sum6 = 0;

int max1 = 0, max2 = 0, max3 = 0, max4 = 0, max5 = 0, max6 = 0;

int min1 = 100, min2 = 100, min3 = 100, min4 = 100, min5 = 100, min6 = 100;

for (int i = 0; i < 10; i++)

{

Console.CursorLeft = 50;

for (int j = 0; j < 10; j++)

{

array[i, j] = random.Next(10, 100);

if (i == j)

{

if (array[i, j] > max1)

{

max1 = array[i, j];

}

if (array[i, j] < min1)

{

min1 = array[i, j];

}

sum1 += array[i, j];

Console.ForegroundColor = ConsoleColor.DarkGray;

}

else if ((i + j) == (n - 1))

{

if (array[i, j] > max2)

{

max2 = array[i, j];

}

if (array[i, j] < min2)

{

min2 = array[i, j];

}

sum2 += array[i, j];

Console.ForegroundColor = ConsoleColor.Blue;

}

else if (i + j < n - 1 && i < j)

{

if (array[i, j] > max3)

{

max3 = array[i, j];

}

if (array[i, j] < min3)

{

min3 = array[i, j];

}

sum3 += array[i, j];

Console.ForegroundColor = ConsoleColor.Cyan;

}

else if ((i + j) > (n - 1) && i < j)

{

if (array[i, j] > max4)

{

max4 = array[i, j];

}

if (array[i, j] < min4)

{

min4 = array[i, j];

}

sum4 += array[i, j];

Console.ForegroundColor = ConsoleColor.Red;

}

else if ((i + j) > (n - 1) && i > j)

{

if (array[i, j] > max5)

{

max5 = array[i, j];

}

if (array[i, j] < min5)

{

min5 = array[i, j];

}

sum5 += array[i, j];

Console.ForegroundColor = ConsoleColor.DarkYellow;

}

else if ((i + j) < (n - 1) && i > j)

{

if (array[i, j] > max6)

{

max6 = array[i, j];

}

if (array[i, j] < min6)

{

min6 = array[i, j];

}

sum6 += array[i, j];

Console.ForegroundColor = ConsoleColor.White;

}

Console.Write(array[i, j] + " ");

}

Console.WriteLine();

}

Console.ForegroundColor = ConsoleColor.DarkGray;

Console.WriteLine("\n\t\t\t\t\t\t Сумма элементов главной диагонали: " + sum1);

Console.WriteLine("\t\t\t\t\t\t Максимальный элемент главной диагонали: " + max1);

Console.WriteLine("\t\t\t\t\t\t Минимальный элемент главной диагонали: " + min1);

Console.ForegroundColor = ConsoleColor.Blue;

Console.WriteLine("\n\t\t\t\t\t\t Сумма элементов побочной диагонали: " + sum2);

Console.WriteLine("\t\t\t\t\t\t Максимальный элемент побочной диагонали: " + max2);

Console.WriteLine("\t\t\t\t\t\t Минимальный элемент побочной диагонали: " + min2);

Console.ForegroundColor = ConsoleColor.Red;

Console.WriteLine("\n\t\t\t\t\t\t Сумма элементов верхнего треугольника: " + sum6);

Console.WriteLine("\t\t\t\t\t\t Максимальный элемент верхнего треугольника: " + max6);

Console.WriteLine("\t\t\t\t\t\t Минимальный элемент верхнего треугольника: " + min6);

Console.ForegroundColor = ConsoleColor.DarkYellow;

Console.WriteLine("\n\t\t\t\t\t\t Сумма элементов нижнего треуголника: " + sum5);

Console.WriteLine("\t\t\t\t\t\t Максимальный элемент нижнего треугольника: " + max5);

Console.WriteLine("\t\t\t\t\t\t Минимальный элемент верхнего треугольника: " + min5);

Console.ForegroundColor = ConsoleColor.Cyan;

Console.WriteLine("\n\t\t\t\t\t\t Сумма элементов правого треуголника: " + sum3);

Console.WriteLine("\t\t\t\t\t\t Максимальный элемент правого треугольника: " + max3);

Console.WriteLine("\t\t\t\t\t\t Минимальный элемент правого треугольника: " + min3);

Console.ForegroundColor = ConsoleColor.White;

Console.WriteLine("\n\t\t\t\t\t\t Сумма элементов левого треуголника: " + sum4);

Console.WriteLine("\t\t\t\t\t\t Максимальный элемент левого треугольника: " + max4);

Console.WriteLine("\t\t\t\t\t\t Минимальный элемент левого треугольника: " + min4);

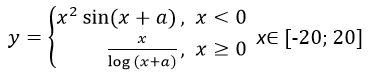
Console.ResetColor();

}

}

**Практическая работа №2**

Создать форму вычислить функцию, под ней поле с функцией:



Значение, а вводится пользователем, при нажатии на кнопку вычислить – вычислить функцию. Сетка грид под этим, в которой столбцы x | y со значениями. Также добавить кнопки «выход» - выйти из файла, «сохранить» - сохранить в файл и «вывести из файла» - вынести значения в таблицу из сохраненного ранее файла.

<Window x:Class="WpfApp1.MainWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:WpfApp1"

mc:Ignorable="d"

Title="MainWindow" Height="450" Width="800">

<Grid Background="#FFF59B9B">

<Button Name="Exit\_click" Content="Выход" HorizontalAlignment="Left" Margin="342,292,0,0" VerticalAlignment="Top" Width="184" Height="50" FontSize="13" Click="Exit\_Click" BorderBrush="#FF0A0101" >

<Button.Background>

<LinearGradientBrush EndPoint="0.5,1" StartPoint="0.5,0">

<GradientStop Offset="0"/>

<GradientStop Offset="1" Color="#FFE017B3"/>

</LinearGradientBrush>

</Button.Background>

</Button>

<Button Name="Calculate" Content="Вычислить" HorizontalAlignment="Left" Margin="343,176,0,0" VerticalAlignment="Top" Width="184" RenderTransformOrigin="0.5,0.5" Height="50" FontSize="13" Click="Calculate\_click" BorderBrush="#FF060101">

<Button.Background>

<LinearGradientBrush EndPoint="0.5,1" StartPoint="0.5,0">

<GradientStop Offset="0"/>

<GradientStop Offset="0.487"/>

<GradientStop Offset="1" Color="#FFE01313"/>

</LinearGradientBrush>

</Button.Background>

<Button.RenderTransform>

<TransformGroup>

<ScaleTransform/>

<SkewTransform/>

<RotateTransform Angle="-0.053"/>

<TranslateTransform/>

</TransformGroup>

</Button.RenderTransform>

</Button>

<Button Name="ClearGridData" Content="Очистить&#xA; таблицу" HorizontalAlignment="Left" Margin="342,347,0,0" VerticalAlignment="Top" Width="184" Height="50" FontSize="13" Click="ClearGridData\_Click" BorderBrush="Black" >

<Button.Background>

<LinearGradientBrush EndPoint="0.5,1" StartPoint="0.5,0">

<GradientStop Offset="0"/>

<GradientStop Offset="1" Color="#FFDCE613"/>

</LinearGradientBrush>

</Button.Background>

</Button>

<Button Name="Write" Content="Записать в файл" HorizontalAlignment="Left" Margin="343,235,0,0" VerticalAlignment="Top" Width="184" Height="50" FontSize="13" Click="Write\_Click" BorderBrush="#FF040000" >

<Button.Background>

<LinearGradientBrush EndPoint="0.5,1" StartPoint="0.5,0">

<GradientStop Offset="0"/>

<GradientStop Offset="0.487"/>

<GradientStop Offset="0.957" Color="#FF1D1A61"/>

</LinearGradientBrush>

</Button.Background>

<Button.RenderTransform>

<TransformGroup>

<ScaleTransform/>

<SkewTransform/>

<RotateTransform Angle="-0.053"/>

<TranslateTransform/>

</TransformGroup>

</Button.RenderTransform>

</Button>

<TextBox Name="input" HorizontalAlignment="Left" Height="20" Margin="117,105,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="220"/>

<TextBlock Margin="82,101,665,283" Text="a=" FontSize="22">

</TextBlock>

<DataGrid Name="Data" HorizontalAlignment="Left" Height="254" Margin="117,143,0,0" VerticalAlignment="Top" Width="220" BorderBrush="#FF000204">

<DataGrid.Background>

<LinearGradientBrush EndPoint="0.5,1" StartPoint="0.5,0">

<GradientStop Offset="0"/>

<GradientStop Offset="0.487"/>

<GradientStop Offset="0.957" Color="#FFD1C9C9"/>

</LinearGradientBrush>

</DataGrid.Background>

</DataGrid>

<Label Content="Введите a(целое число):" HorizontalAlignment="Left" VerticalAlignment="Top" Margin="117,54,0,0" FontSize="16" FontWeight="Bold" FontStyle="Italic" BorderBrush="#FFEEDBDB"/>

<Image Margin="487,33,24,323" Source="Image1.png" Stretch="Fill"/>

<Label Content="Ильиных Д.Г." HorizontalAlignment="Left" Margin="61,10,0,0" VerticalAlignment="Top"/>

</Grid>

</Window>

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace WpfApp1

{

public class list

{

public int X { get; set; }

public string Y { get; set; }

public list(int x, string y)

{

X = x;

Y = y;

}

}

public partial class MainWindow : Window

{

public

string path = "note.exe";

public MainWindow()

{

InitializeComponent();

}

private void Exit\_Click(object sender, RoutedEventArgs e)

{

this.Close();

}

public void TextBox\_TextChanged(object sender, RoutedEventArgs e)

{

this.Show();

}

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

{

input.TabIndex = 0;

input.Focus();

}

double y;

List<list> result = new List<list>();

private void Calculate\_click(object sender, RoutedEventArgs e)

{

try

{

double.Parse(input.Text);

}

catch (FormatException)

{

MessageBox.Show("Введите число");

return;

}

result.Clear();

Data.ItemsSource = null;

for (int i = -20; i <= 20; i++)

{

if (i >= 0)

{

if (i + double.Parse(input.Text) <= 0 || Math.Log10(i + double.Parse(input.Text)) <= 0)

{

result.Add(new list(i, "Не определён"));

continue;

}

else

{

double var = Math.Round(i / Math.Log10(i + double.Parse(input.Text)), 2);

result.Add(new list(i, var.ToString()));

}

}

else

{

double var = Math.Round(Math.Pow(i, 2) \* Math.Sin(i + double.Parse(input.Text)), 2);

result.Add(new list(i, var.ToString()));

}

}

input.Text = "";

Data.ColumnWidth = 97;

Data.ItemsSource = result;

}

private void Write\_Click(object sender, RoutedEventArgs e)

{

if (Data.ItemsSource == null) return;

StreamWriter sw = new StreamWriter("C:/Users/Public/Documents/file.txt");

for (int i = 0; i <= 40; i++)

{

sw.Write(result[i].Y + "\n");

}

sw.Close();

MessageBox.Show("Запись успешно произведена");

}

private void ClearGridData\_Click(object sender, RoutedEventArgs e)

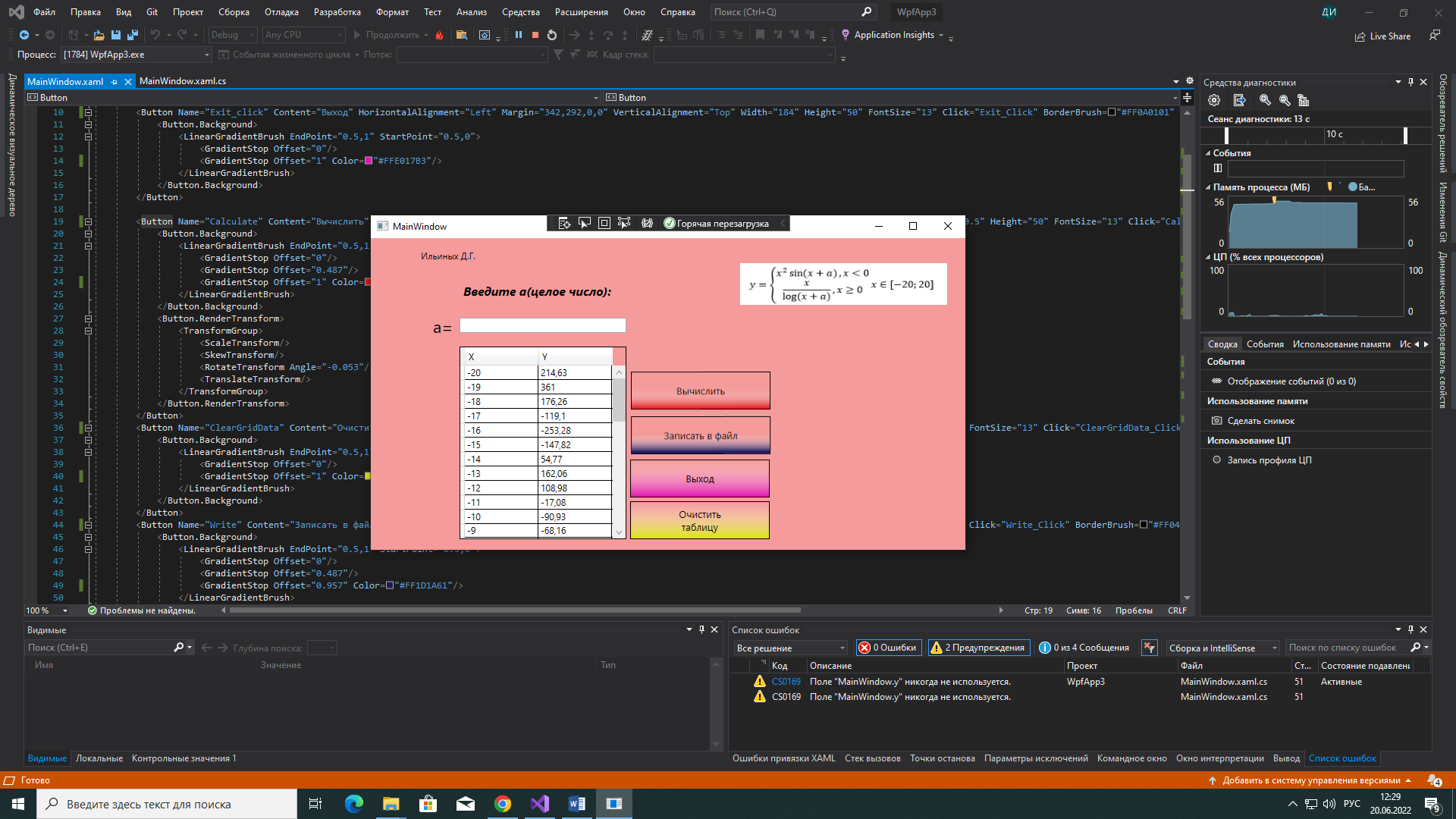
{

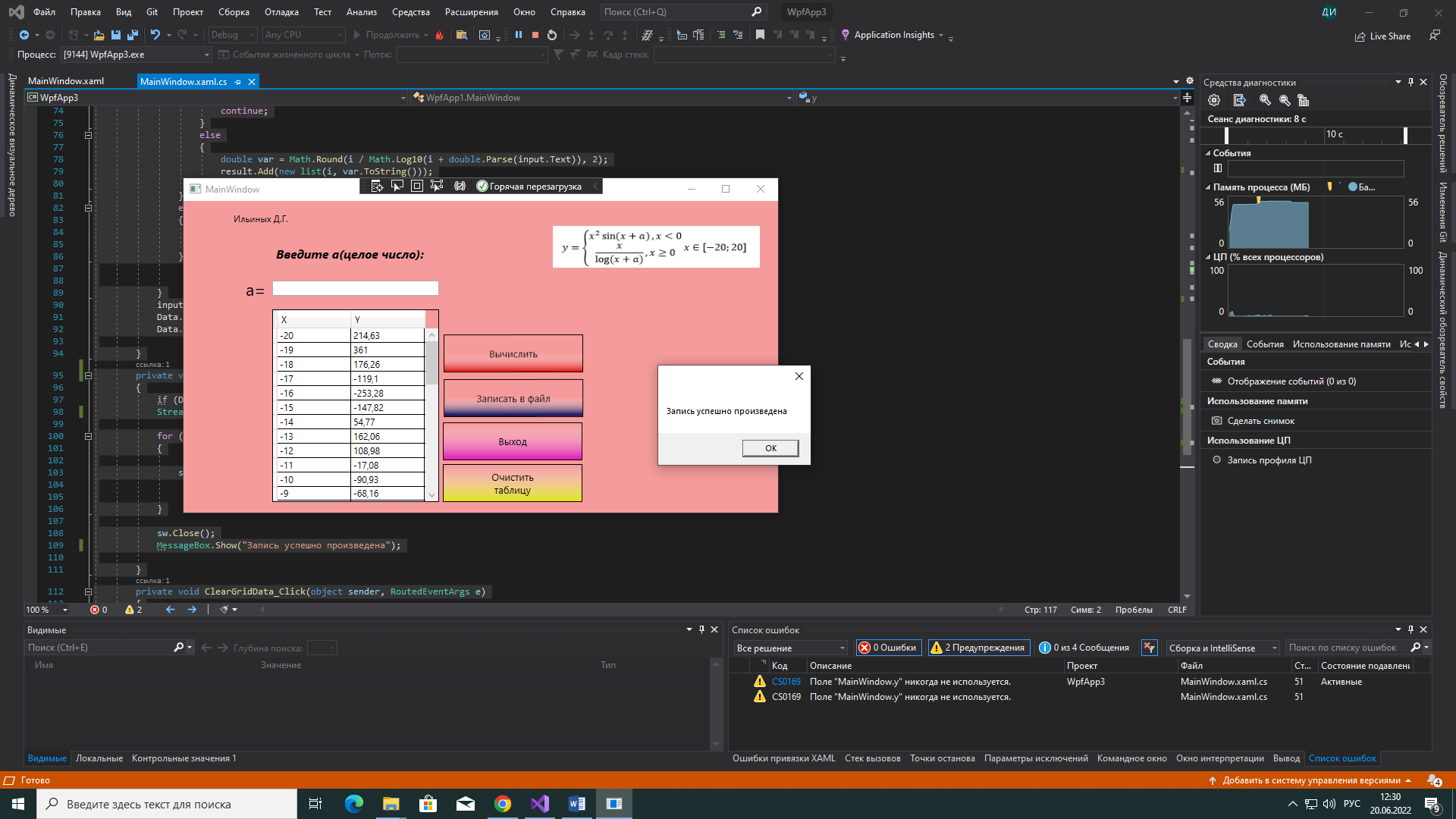
Data.ItemsSource = null;

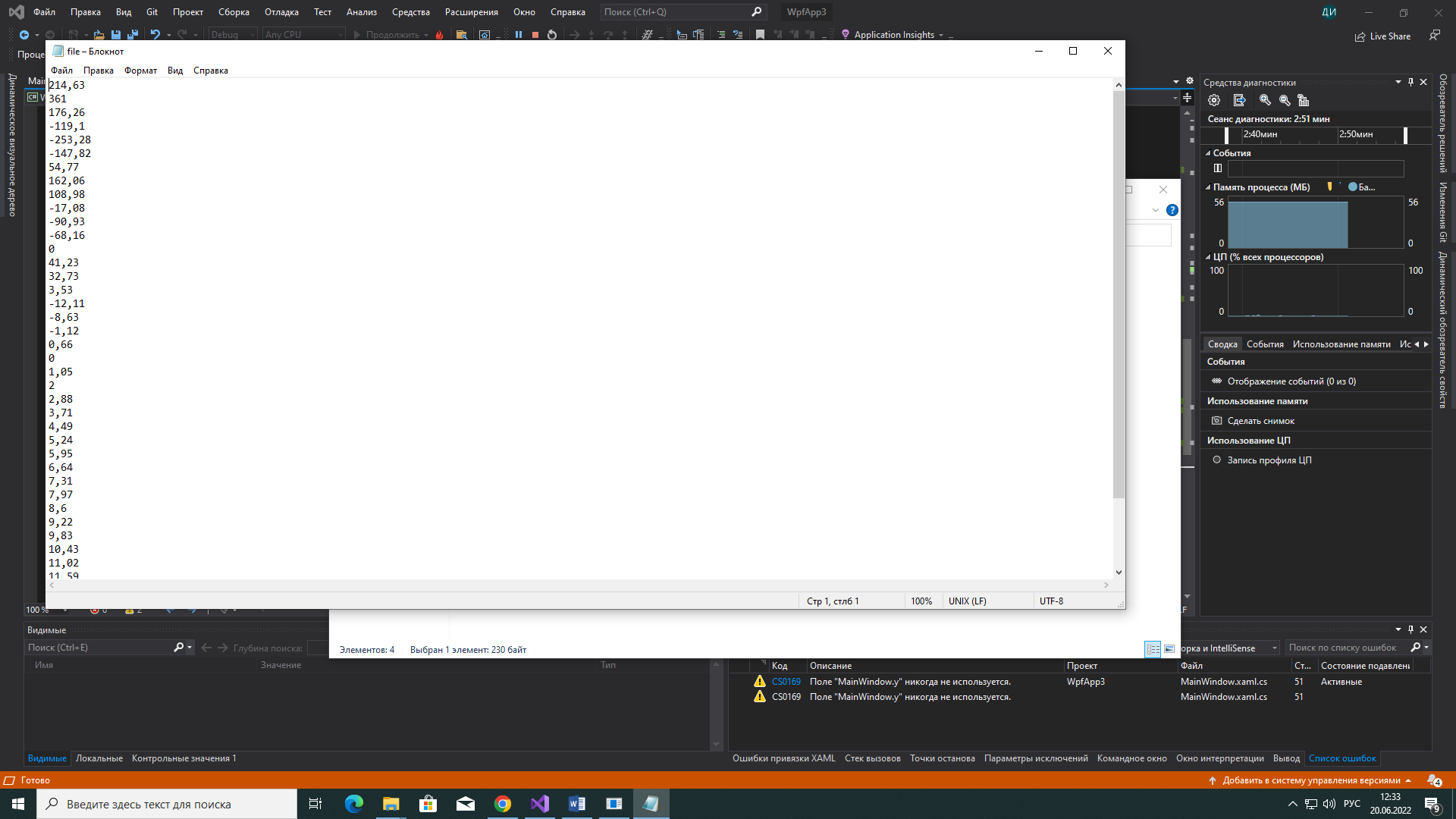
}

}

}

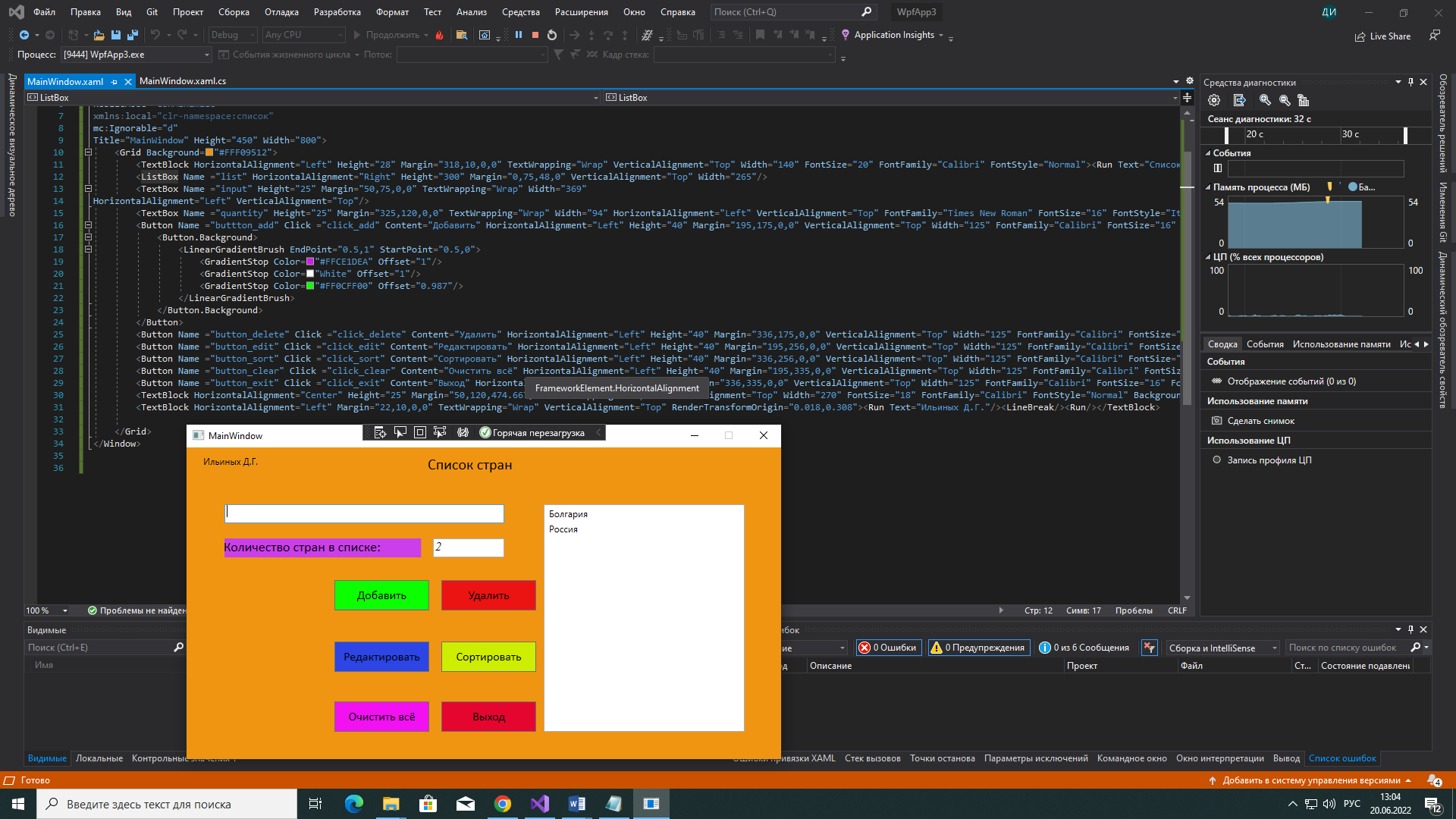






**Практическая работа №3**

Создать программу с использованием листбокса. Добавить окно ввода списка стран, обработать исключения (пробелы и тд). Добавить кнопки:  
- «Добавить» - добавляет в листбокс наименование страны  
- «Редактировать» - Выбрать в списке позицию мышкой и изменить введенный текст.  
- «Удалить» - Выбрать в списке позицию мышкой и удалить, высветится окно с подтверждением операции.  
- «Очистить всё» - очищает весь список, с предупреждением  
- «Сортировать» - сортировка списка по алфавиту  
- «Выход» - выйти из программы.  
Также отображается счетчик записей в списке под окном ввода позиций в список.



<Window x:Class="список.MainWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

ResizeMode="CanMinimize"

xmlns:local="clr-namespace:список"

mc:Ignorable="d"

Title="MainWindow" Height="450" Width="800">

<Grid Background="#FFF09512">

<TextBlock HorizontalAlignment="Left" Height="28" Margin="318,10,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="140" FontSize="20" FontFamily="Calibri" FontStyle="Normal"><Run Text="Список "/><Run Language="ru-ru" Text="стран"/></TextBlock>

<ListBox Name ="list" HorizontalAlignment="Right" Height="300" Margin="0,75,48,0" VerticalAlignment="Top" Width="265"/>

<TextBox Name ="input" Height="25" Margin="50,75,0,0" TextWrapping="Wrap" Width="369"

HorizontalAlignment="Left" VerticalAlignment="Top"/>

<TextBox Name ="quantity" Height="25" Margin="325,120,0,0" TextWrapping="Wrap" Width="94" HorizontalAlignment="Left" VerticalAlignment="Top" FontFamily="Times New Roman" FontSize="16" FontStyle="Italic" TextDecorations="{x:Null}"/>

<Button Name ="buttton\_add" Click ="click\_add" Content="Добавить" HorizontalAlignment="Left" Height="40" Margin="195,175,0,0" VerticalAlignment="Top" Width="125" FontFamily="Calibri" FontSize="16" FontStyle="Normal">

<Button.Background>

<LinearGradientBrush EndPoint="0.5,1" StartPoint="0.5,0">

<GradientStop Color="#FFCE1DEA" Offset="1"/>

<GradientStop Color="White" Offset="1"/>

<GradientStop Color="#FF0CFF00" Offset="0.987"/>

</LinearGradientBrush>

</Button.Background>

</Button>

<Button Name ="button\_delete" Click ="click\_delete" Content="Удалить" HorizontalAlignment="Left" Height="40" Margin="336,175,0,0" VerticalAlignment="Top" Width="125" FontFamily="Calibri" FontSize="16" FontStyle="Normal" Background="#FFEC1313"/>

<Button Name ="button\_edit" Click ="click\_edit" Content="Редактировать" HorizontalAlignment="Left" Height="40" Margin="195,256,0,0" VerticalAlignment="Top" Width="125" FontFamily="Calibri" FontSize="16" FontStyle="Normal" Background="#FF2C45E4"/>

<Button Name ="button\_sort" Click ="click\_sort" Content="Сортировать" HorizontalAlignment="Left" Height="40" Margin="336,256,0,0" VerticalAlignment="Top" Width="125" FontFamily="Calibri" FontSize="16" FontStyle="Normal" Background="#FFCDEE00"/>

<Button Name ="button\_clear" Click ="click\_clear" Content="Очистить всё" HorizontalAlignment="Left" Height="40" Margin="195,335,0,0" VerticalAlignment="Top" Width="125" FontFamily="Calibri" FontSize="16" FontStyle="Normal" RenderTransformOrigin="0.498,0.256" Background="#FFF111F1"/>

<Button Name ="button\_exit" Click ="click\_exit" Content="Выход" HorizontalAlignment="Left" Height="40" Margin="336,335,0,0" VerticalAlignment="Top" Width="125" FontFamily="Calibri" FontSize="16" FontStyle="Normal" Background="#FFE4062E"/>

<TextBlock HorizontalAlignment="Center" Height="25" Margin="50,120,474.667,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="270" FontSize="18" FontFamily="Calibri" FontStyle="Normal" Background="#FFCB3FEA"><Run Text=" Количество "/><Run Language="ru-ru" Text="стран"/><Run Text=" в списке: "/></TextBlock>

<TextBlock HorizontalAlignment="Left" Margin="22,10,0,0" TextWrapping="Wrap" VerticalAlignment="Top" RenderTransformOrigin="0.018,0.308"><Run Text="Ильиных Д.Г."/><LineBreak/><Run/></TextBlock>

</Grid>

</Window>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace список

{

public partial class MainWindow : Window

{

int quantity\_products = 0;

public MainWindow()

{

InitializeComponent();

input.Focus();

quantity.Text = quantity\_products.ToString();

}

private void click\_exit(object sender, RoutedEventArgs e)

{

this.Close();

}

private void click\_add(object sender, RoutedEventArgs e)

{

if (input.Text.Trim() == "")

{

MessageBox.Show("Вы ввели пустую строку!");

input.Focus();

}

else

{

quantity\_products += 1;

quantity.Text = quantity\_products.ToString();

list.Items.Add(input.Text.Trim());

input.Clear();

input.Focus();

}

}

private void click\_delete(object sender, RoutedEventArgs e)

{

if (list.SelectedValue != null)

{

MessageBoxResult result = MessageBox.Show("Вы желаете удалить продукт из списка?", "Удалить", MessageBoxButton.YesNo, MessageBoxImage.Information);

if (result == MessageBoxResult.Yes)

{

list.Items.Remove(list.SelectedValue);

quantity\_products -= 1;

quantity.Text = quantity\_products.ToString();

if (quantity\_products < 0)

{

MessageBox.Show("Списк пуст, невозможно удаление продукта.");

quantity\_products = 0;

quantity.Text = quantity\_products.ToString();

}

input.Focus();

}

input.Focus();

}

else if (quantity\_products >= 0)

{

MessageBox.Show("Для удаления продукта из списка необходимо выделить его мышкой в окне списка.");

input.Focus();

}

else

{

MessageBox.Show("Списк пуст, невозможно удаление продукта.");

quantity\_products = 0;

quantity.Text = quantity\_products.ToString();

input.Focus();

}

}

private void click\_clear(object sender, RoutedEventArgs e)

{

MessageBoxResult result = MessageBox.Show("Вы желаете очистить весь список продуктов?", "Очистить всё", MessageBoxButton.YesNo, MessageBoxImage.Warning);

if (result == MessageBoxResult.Yes)

{

if (quantity\_products <= 0)

{

MessageBox.Show("Очищение пустого списка невозможно.");

input.Focus();

}

else

{

list.Items.Clear();

quantity\_products = 0;

quantity.Text = quantity\_products.ToString();

input.Focus();

}

}

}

private void click\_sort(object sender, RoutedEventArgs e)

{

list.Items.SortDescriptions.Add(new System.ComponentModel.SortDescription("", System.ComponentModel.ListSortDirection.Ascending));

input.Focus();

}

private void click\_edit(object sender, RoutedEventArgs e)

{

if (list.SelectedValue != null)

{

try

{

input.Text = list.SelectedValue.ToString();

list.Items.Remove(list.SelectedValue);

quantity\_products -= 1;

quantity.Text = quantity\_products.ToString();

}

catch { }

}

else

{

MessageBox.Show("Для редактирования списка продуктов необходимо выделить его мышкой в окне списка.");

input.Focus();

}

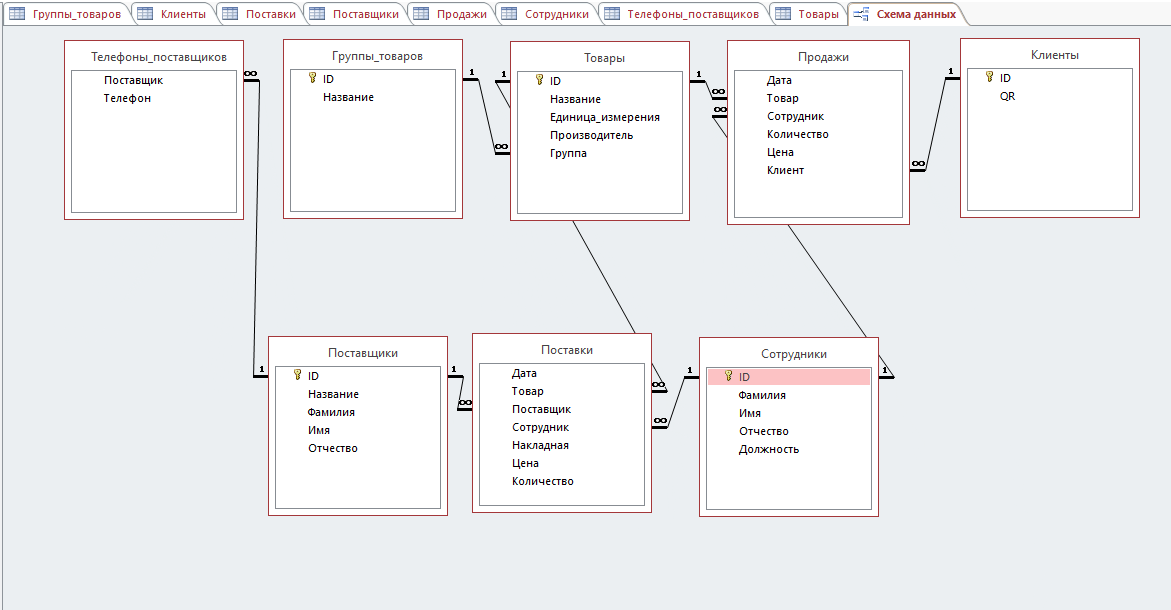
}

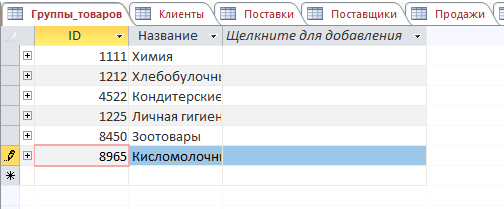
}

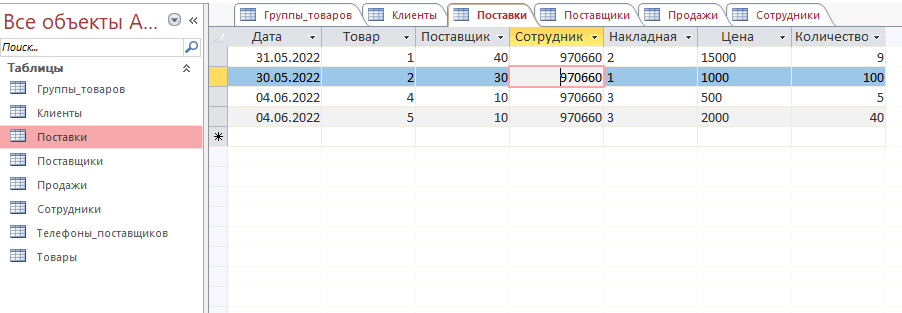
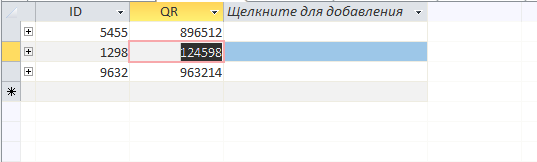
}

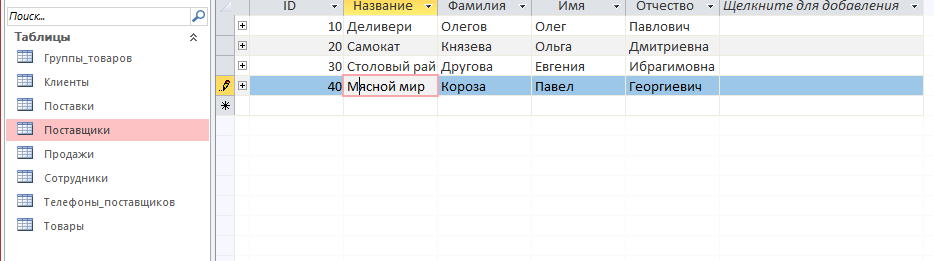
**Практическая работа №4**

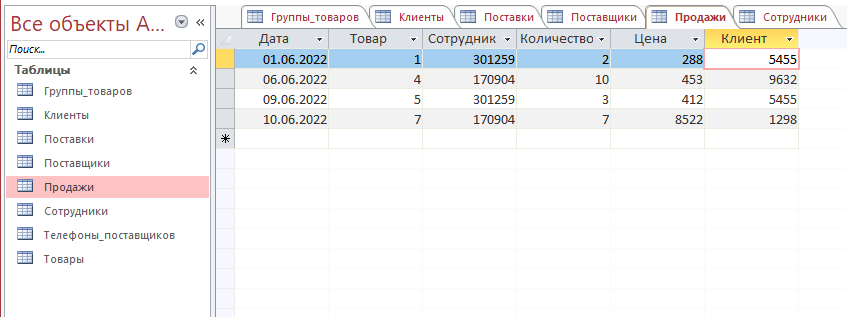
Создать базу данных торгового предприятия. Создать таблицы: «Товар», «Группа товаров», «Сотрудники», «Поставщики», «Клиенты», «Продажи», «Телефоны поставщиков». И создать связи между ними.

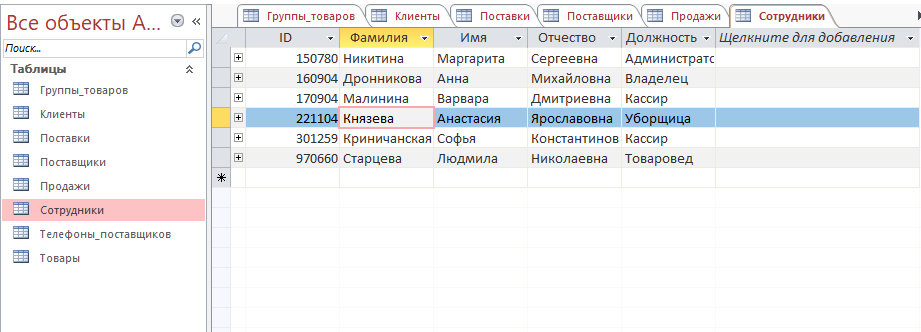


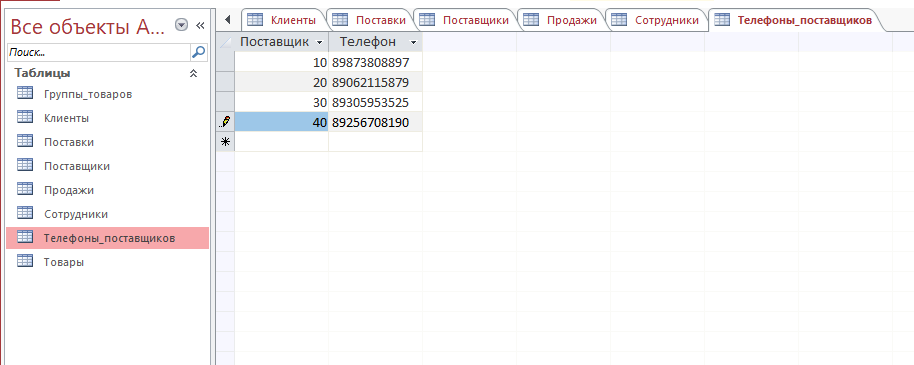


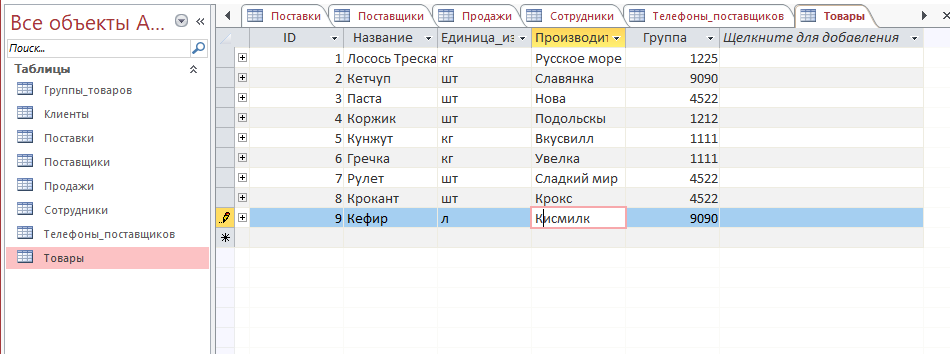












(Первая учебная практика, Ассемблер)

Специальность Программирование компьютерных систем

*(наименование специальности)*

Профессиональный модуль ПМ.01. Разработка модулей программного обеспечения для компьютерных систем

*(наименование профессионального модуля)*

|  |  |  |
| --- | --- | --- |
| **№п/п** | **Темы учебной практики** | **Выполненные задания/работы** |
| **1** | Локальные данные процедур | Записать в регистр EAX число нулевых элементов массива X. |
| **2** | Процедуры | Ввести одномерный массив l = {7, 6, 15, 17, 12, -12, 4, 0, -10, -22}. Заменить в массиве нулями те элементы, модуль которых при делении на 5 дает в остатке 2. В полученном массиве найти максимальный элемент и его номер. Программу разработать на основе пользовательских функций. |
| **3** | Рекурсивные процедуры | 1. Рекурсивно описать процедуру RevPrint(N), которая   печатает в обратном порядке цифры десятичной записи целого  неотрицательного числа N. Например, RevPrint(12345) должна вывести текст 54321.   1. Разработать рекурсивную процедуру вычисления факториала N! (N≥0). |
| **4** | Обработка двумерных массивов | Разместить в памяти компьютера матрицу 4\*3 целых беззнаковых чисел. Написать программу, позволяющую найти сумму всех строк массива. Исходную матрицу и полученные суммы вывести на экран. |
| **5** | Работа с сегментами | Количество чисел, равных C1 |
| **6** | Работа со стеками | Дана целочисленная прямоугольная матрица. 1) Найти среднее арифметическое элементов для каждой строки матрицы. 2) Найти количество строк, среднее арифметическое элементов которых меньше заданной величины. |

**Практическая работа №1**

**Задание:**

X DD 100 DUP(?) ; числа со знаком. Записать в регистр EAX число нулевых элементов массива X.

**Решение:**

format PE console

include 'win32a.inc'

start:

proc func

mov ecx, 15

lea esi, [X]

mov ebx, 0

mov eax,0

search1:

cmp [esi], ebx

jl next

cmp [esi],ebx

jg next

compare:

inc eax

next:

add esi, 4

loop search1

cinvoke printf, text1, eax

invoke sleep, 9000

invoke exit,0

ret

endp

call func

invoke sleep, 9000

invoke exit, 0

X dd 0, 2, 5, 7, 5, 3, 7, 9, 6, 1,4,0,0,5,0

;X dd 100 dup (?)

text db '', 10, 13, 0

text1 db 'Number of zero values= %d',0

fmat db '%d',9,0

; import data in the same section

data import

library msvcrt,'MSVCRT.DLL',\

kernel32,'KERNEL32.DLL'

import kernel32,\

sleep,'Sleep'

import msvcrt,\

printf,'printf',\

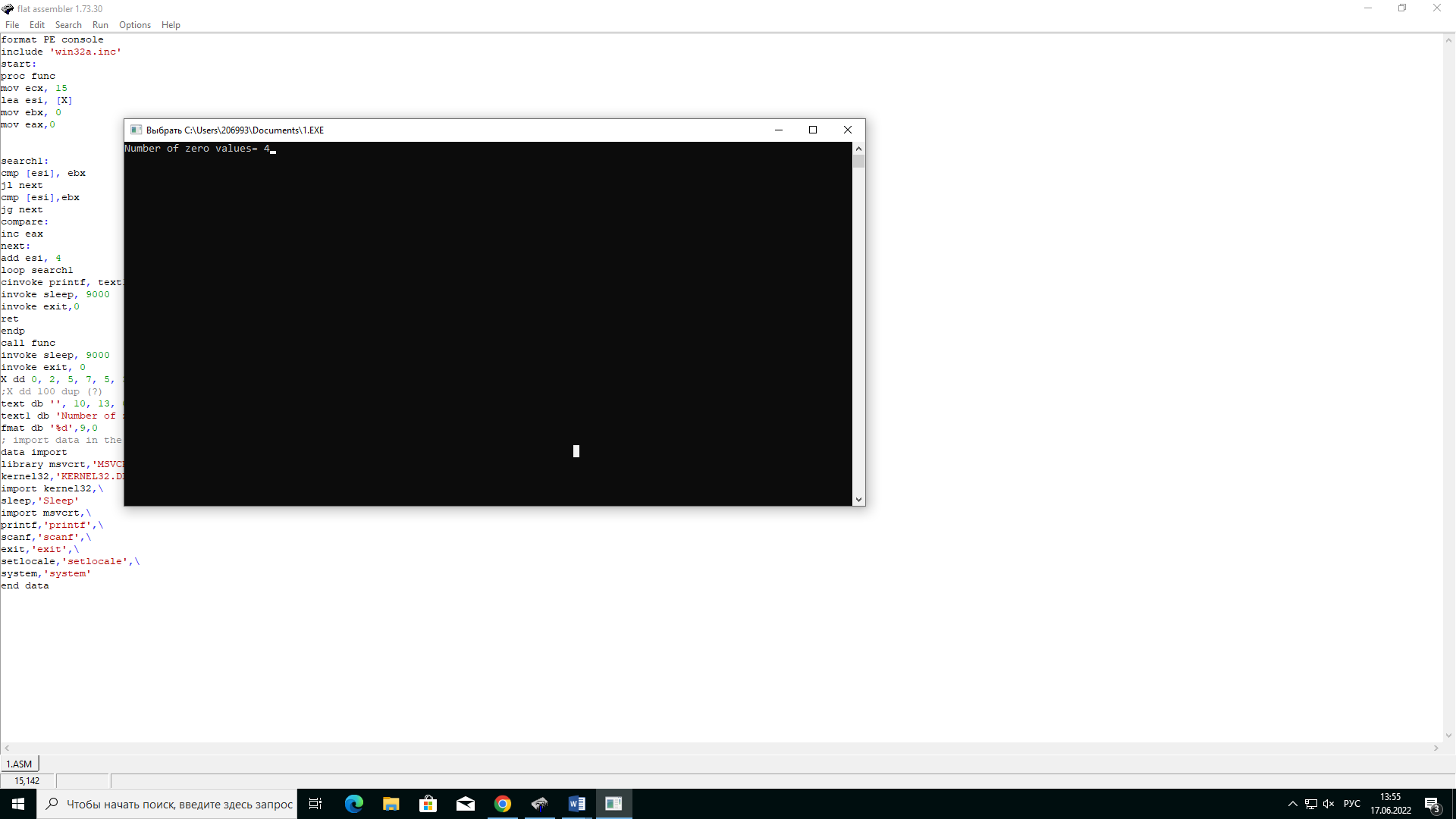
scanf,'scanf',\

exit,'exit',\

setlocale,'setlocale',\

system,'system'

end data



**Практическая работа №2**

**Задание:**

Ввести одномерный массив l = {7, 6, 15, 17, 12, -12, 4, 0, -10, -22}. Заменить в массиве нулями те элементы, модуль которых при делении на 5 дает в остатке 2. В полученном массиве найти максимальный элемент и его номер. Программу разработать на основе пользовательских функций.

**Решение:**

format PE console

include 'win32a.inc'

start:

macro prnt2d op1,op2,op3

{

local met1,met2

mov esi,op1

mov ecx,op2

met2: push ecx

mov ecx,op3

met1: push ecx

mov eax,[esi]

push ecx

cinvoke printf,fmat,eax

pop ecx

add esi,4

pop ecx

loop met1

cinvoke printf, text3

pop ecx

loop met2

}

macro prntn op1,op2

{

local metka

mov esi,op1

mov ecx,op2

mov eax,0

metka:

mov eax,[esi]

push ecx

cinvoke printf,fmat,eax

pop ecx

add esi,4

loop metka

}

prnt2d X,[M],[N]

lea ebx,[Y]

lea esi,[X]

mov ecx,[M]

met2: push ecx

mov ecx,[N]

xor eax,eax

met1: add eax,[esi]

add esi,4

loop met1

mov [ebx],eax

add ebx,4

pop ecx

loop met2

xor eax,eax

lea ebx,[Y]

mov ecx,[M]

met3:

add eax,[ebx]

add ebx,4

loop met3

cinvoke printf, spp,eax

cinvoke printf, text4

prnt2d Y,1,[M]

invoke sleep,60000

invoke exit,0

X dd -1, -2, -3, 2, -1

dd 0, -1, -18, 15 -10

dd -1, -2, -3, 2, -1

dd 1, -1, 18, 10, -10

Y dd 100 dup(?)

N dd 5

M dd 4

text db 10,13, '', 10, 13, 0

text2 db 'Enter N: ',0

text3 db '',10,13,0

text4 db 10,13, '',10,13,0

fmat db '%d',9,0

spp db 10,13,'%d',0

data import

library msvcrt, 'MSVCRT.DLL',\

kernel132, 'KERNEL32.DLL'

import kernel132,\

sleep, 'Sleep'

import msvcrt,\

printf, 'printf',\

scanf,'scanf',\

exit, 'exit',\

setlocal, 'setlocal',\

system, 'system'

end data

**Практическая работа №3**

**Задание 1:**

Рекурсивно описать процедуру RevPrint(N), которая печатает в обратном порядке цифры десятичной записи целого неотрицательного числа N. Например, RevPrint(12345) должна вывести текст 54321.

**Решение:**

format PE console

include 'win32a.inc'

start:

cinvoke printf,text1

cinvoke scanf, fmat, n

mov edx, 0

call RevPrint

invoke sleep, 9000

proc RevPrint

mov eax, [n]

mov ecx, 10

div ecx

mov [n], eax

push eax

cinvoke printf, text2, edx

pop eax

cmp eax, 0

ja f1

ret

f1:

call RevPrint

ret

endp

text1 db 10,13,'Enter n: ',0

text2 db '%d',0

fmat db ' %d',0

n dd ?

data import

library msvcrt,'MSVCRT.DLL',\

kernel32,'KERNEL32.DLL'

import kernel32,\

sleep,'Sleep'

import msvcrt,\

printf,'printf',\

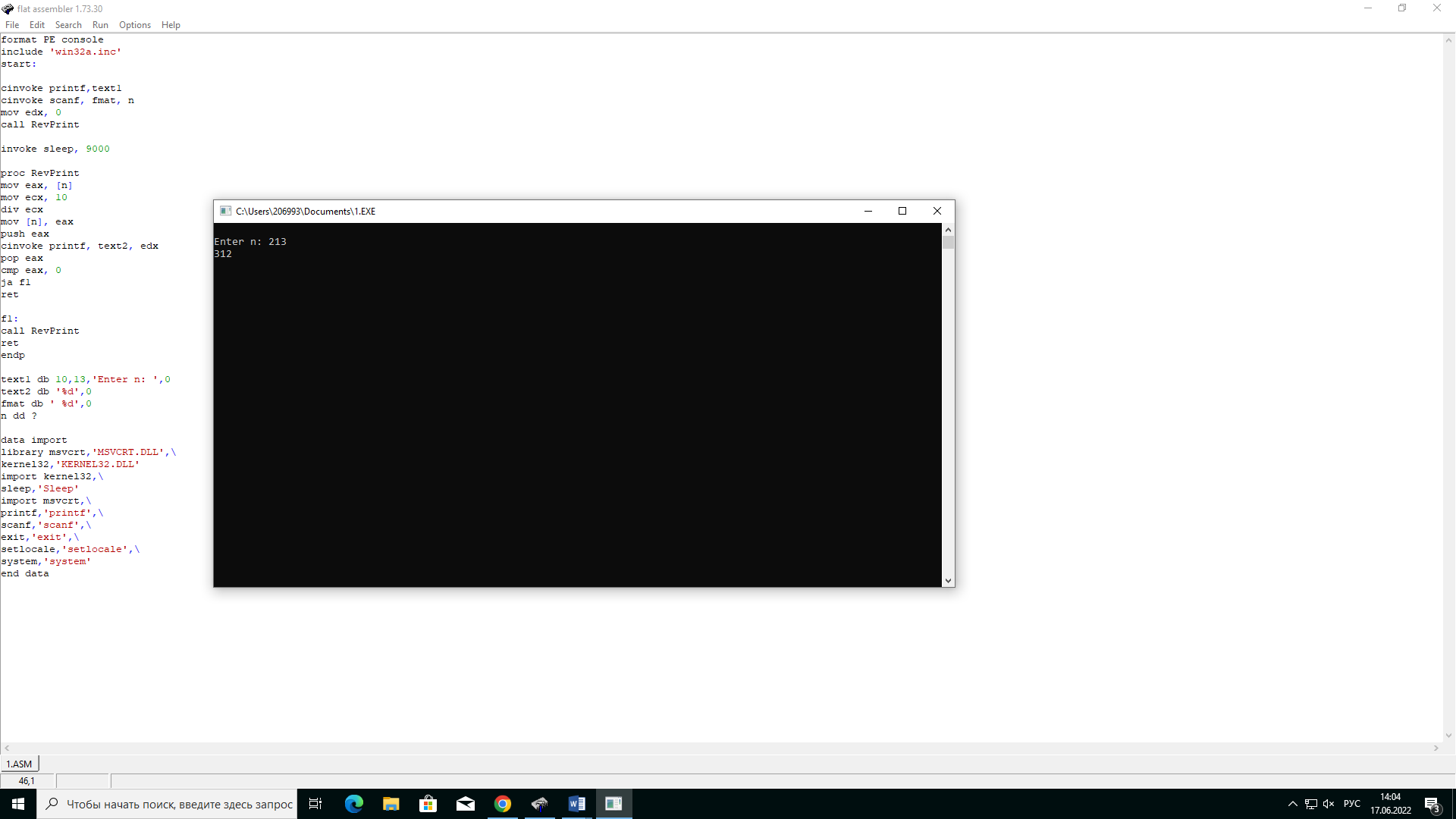
scanf,'scanf',\

exit,'exit',\

setlocale,'setlocale',\

system,'system'

end data



**Задание 2:**

Разработать рекурсивную процедуру вычисления факториала N! (N≥0).

**Решение:**

format PE console

include 'win32a.inc'

start:

cinvoke printf, text1

cinvoke scanf, spp, n

mov eax, [n]

call fn

cinvoke printf,text2, ebx

invoke sleep, 100000000

invoke exit, 0

proc fn

cmp eax,1

ja f1

mov ebx,1

ret

f1: push eax

dec eax

call fn

inc eax

imul ebx, eax

pop eax

ret

endp

n dd ?

text1 db 10,13,'Enter n = ',0

text2 db 10,13,'Result = %d ',0

fmat db '%d',9,0

spp db '%d',0

data import

library msvcrt,'MSVCRT.DLL',\

kernel32,'KERNEL32.DLL'

import kernel32,\

sleep,'Sleep'

import msvcrt,\

printf,'printf',\

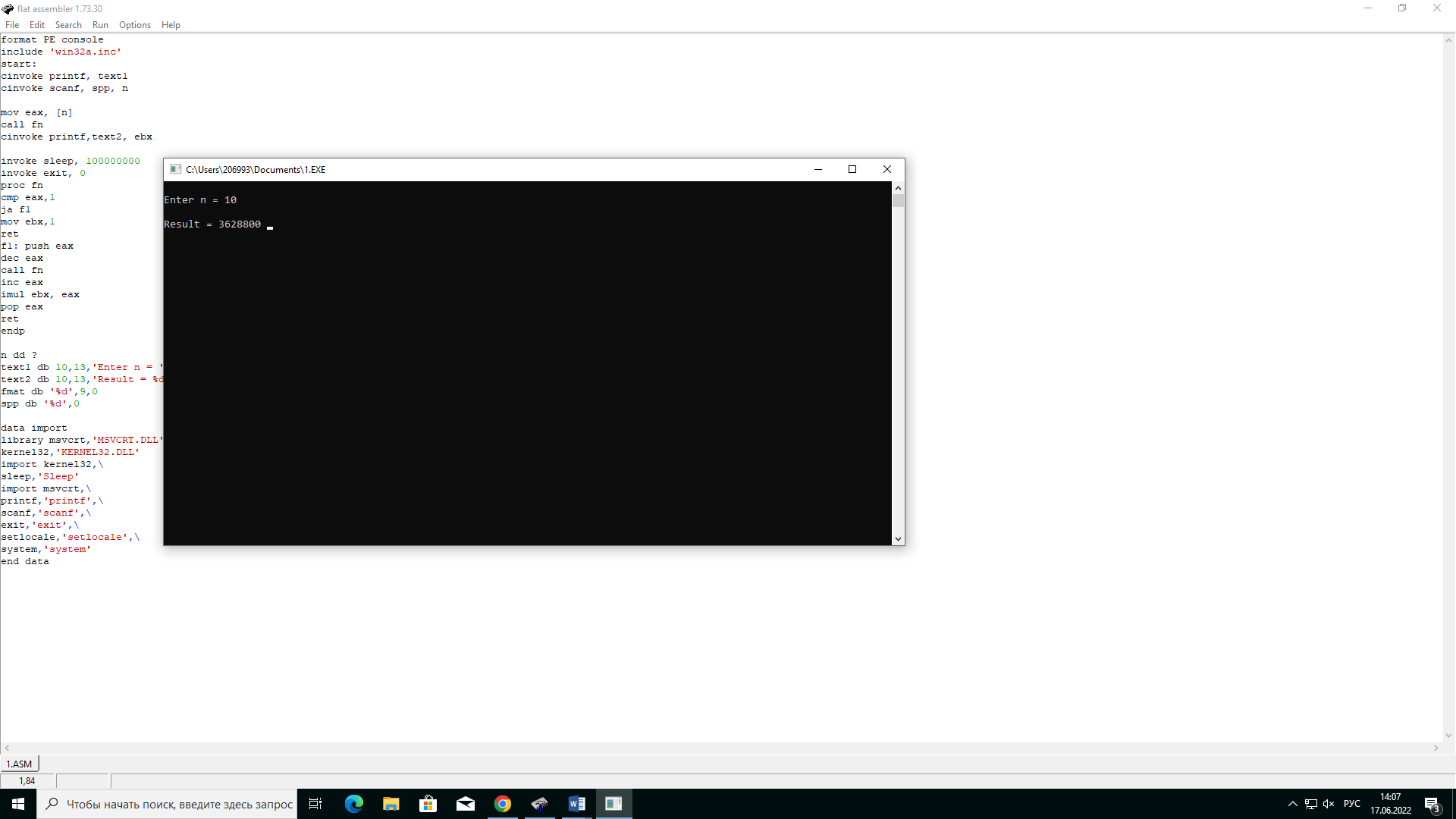
scanf,'scanf',\

exit,'exit',\

setlocale,'setlocale',\

system,'system'

end data



**Практическая работа №4**

**Задание:**

Разместить в памяти компьютера матрицу 4\*3 целых беззнаковых чисел. Написать программу, позволяющую найти сумму всех строк массива. Исходную матрицу и полученные суммы вывести на экран.

**Решение:**

format PE console

include 'win32a.inc'

start:

macro prnt2d op1,op2,op3

{

local met1,met2

mov esi,op1

mov ecx,op2

met2: push ecx

mov ecx,op3

met1: push ecx

mov eax,[esi]

push ecx

cinvoke printf,fmat,eax

pop ecx

add esi,4

pop ecx

loop met1

cinvoke printf, text3

pop ecx

loop met2

}

macro prntn op1,op2

{

local metka

mov esi,op1

mov ecx,op2

mov eax,0

metka:

mov eax,[esi]

push ecx

cinvoke printf,fmat,eax

pop ecx

add esi,4

loop metka

}

prnt2d X,[M],[N]

lea ebx,[Y]

lea esi,[X]

mov ecx,[M]

met2: push ecx

mov ecx,[N]

xor eax,eax

met1: add eax,[esi]

add esi,4

loop met1

mov [ebx],eax

add ebx,4

pop ecx

loop met2

xor eax,eax

lea ebx,[Y]

mov ecx,[M]

met3:

add eax,[ebx]

add ebx,4

loop met3

cinvoke printf, spp,eax

cinvoke printf, text4

prnt2d Y,1,[M]

invoke sleep,60000

invoke exit,0

X dd -1, -2, -3, 2, -1

dd 0, -1, -18, 15 -10

dd -1, -2, -3, 2, -1

dd 1, -1, 18, 10, -10

Y dd 100 dup(?)

N dd 5

M dd 4

text db 10,13, '', 10, 13, 0

text2 db 'Enter N: ',0

text3 db '',10,13,0

text4 db 10,13, '',10,13,0

fmat db '%d',9,0

spp db 10,13,'%d',0

data import

library msvcrt, 'MSVCRT.DLL',\

kernel132, 'KERNEL32.DLL'

import kernel132,\

sleep, 'Sleep'

import msvcrt,\

printf, 'printf',\

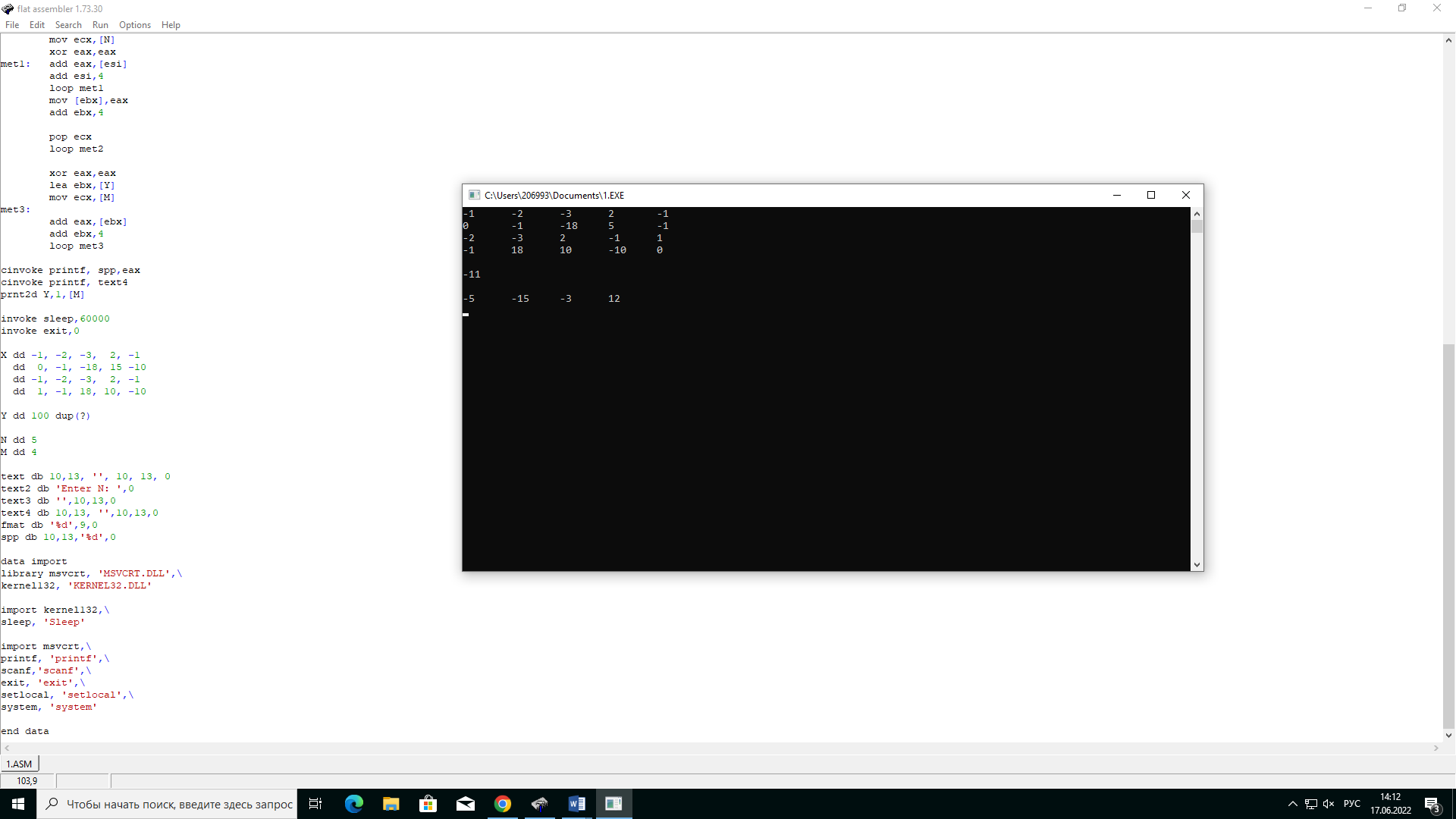
scanf,'scanf',\

exit, 'exit',\

setlocal, 'setlocal',\

system, 'system'

end data



**Практическая работа №5**

**Задание:**

Количество чисел, равных C1

**Решение:**

format PE console

include 'win32a.inc'

entry start

proc Koll

xor eax, eax

cycl:

inc eax

sub ecx, 1

add ebx, 8

loop cycl

ret

endp

start:

macro prntn op1,op2

{

local met

mov ebx,op1

mov ecx,op2

mov eax,0

met:

mov eax,[ebx]

push ecx

cinvoke printf,fpp,eax

pop ecx

add ebx, 4

loop met

}

cinvoke printf, text1

cinvoke scanf,spp, N

prntn A,[N]

lea ebx,[A]

mov ecx,[N] ;записываем в регистр

call Koll ;вызов процедуры

cinvoke printf, text3, eax

cinvoke system,p

invoke exit,0

text1 db 'Enter the number of elements N: ',0

text3 db 10,13, 'Koll-vo = %d',10,13,0

spp db '%d',0

fpp db '%d',9,0

p db 'pause',0

N dd ?

A dd 20,34,45,-9,-100,34,22,0,5, 6, 0, 34, 23, -65, 76, 4, 2, 12, 7

; import data in the same section

data import

library msvcrt, 'MSVCRT.DLL' ,\

kernel32, 'KERNRL32.DLL'

import kernel32,\

sleep, 'Sleep'

import msvcrt,\

printf, 'printf',\

scanf, 'scanf' ,\

exit, 'exit' ,\

system, 'system'

end data



**Практическая работа №6**

**Задание:**

Дана целочисленная прямоугольная матрица. 1) Найти среднее арифметическое элементов для каждой строки матрицы. 2) Найти количество строк, среднее арифметическое элементов которых меньше заданной величины.

**Решение:**

format PE console

include 'win32a.inc'

start:

macro prnt2d op1,op2,op3

{

local met1,met2

mov esi,op1

mov ecx,op2

met2: push ecx

mov ecx,op3

met1: push ecx

mov eax,[esi]

push ecx

cinvoke printf,fmat,eax

pop ecx

add esi,4

pop ecx

loop met1

cinvoke printf, text3

pop ecx

loop met2

}

prnt2d X,[M],[N]

lea esi,[X]

mov ecx,0

mov ebx,0

met1: cmp [esi],dword 0

jne met2

add esi,4

loop met1

met2: inc ecx

cmp ecx,dword 5

je met3

add esi,4

loop met1

met3: inc ebx

mov ecx,0

add esi,4

cinvoke printf, text4

cinvoke printf, text5,ebx

invoke sleep,60000

invoke exit,0

X dd -1, -2, -3, 2, -1

dd 0, -1, -18, 15 -10

dd -1, -2, -3, 2, -1

dd 1, -1, 18, 10, -10

Y dd 100 dup(?)

N dd 5

M dd 4

text db 10,13, '', 10, 13, 0

text2 db 'Enter N: ',0

text3 db '',10,13,0

text4 db 10,13, '',10,13,0

text5 db 'Kolvo !=0 strok: %d'

fmat db '%d',9,0

spp db 10,13,'%d',0

data import

library msvcrt, 'MSVCRT.DLL',\

kernel132, 'KERNEL32.DLL'

import kernel132,\

sleep, 'Sleep'

import msvcrt,\

printf, 'printf',\

scanf,'scanf',\

exit, 'exit',\

setlocal, 'setlocal',\

system, 'system'

end data

**ЗАДАНИЕ № 7**

**Тема: “События C#”**

**Цель работы**: разработать калькулятор на языке C#. В проекте использовать приём создания одного обработчика для нескольких событий.

**Ниже на рисунке представлен фрагмент работающего проекта**

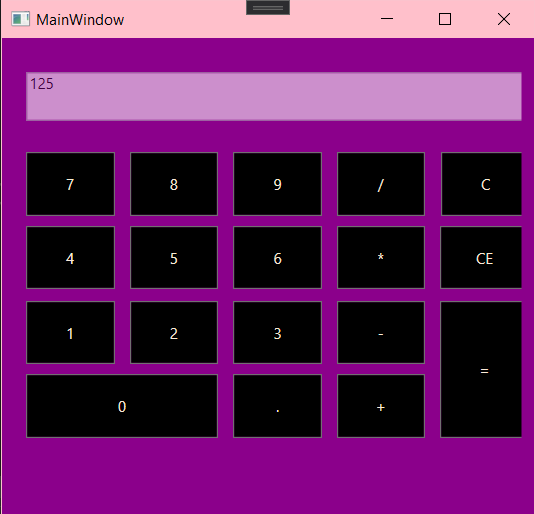


Рисунок 1. Калькулятор

**Код программы:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace WpfApp1

{

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

}

private void CE(object sender, RoutedEventArgs e)

{

TextBox.Clear();

}

private void Symb\_1\_Click(object sender, RoutedEventArgs e)

{

TextBox.Text += 1;

}

private void Symb\_2\_Click(object sender, RoutedEventArgs e)

{

TextBox.Text += 2;

}

private void Symb\_3\_Click(object sender, RoutedEventArgs e)

{

TextBox.Text += 3;

}

private void Symb\_4\_Click(object sender, RoutedEventArgs e)

{

TextBox.Text += 4;

}

private void Symb\_5\_Click(object sender, RoutedEventArgs e)

{

TextBox.Text += 5;

}

private void Symb\_6\_Click(object sender, RoutedEventArgs e)

{

TextBox.Text += 6;

}

private void Symb\_7\_Click(object sender, RoutedEventArgs e)

{

TextBox.Text += 7;

}

private void Symb\_8\_Click(object sender, RoutedEventArgs e)

{

TextBox.Text += 8;

}

private void Symb\_9\_Click(object sender, RoutedEventArgs e)

{

TextBox.Text += 9;

}

private void Symb\_0\_Click(object sender, RoutedEventArgs e)

{

TextBox.Text += 0;

}

private void Button\_Click(object sender, RoutedEventArgs e)

{

TextBox.Text += "√";

}

private void Button\_Click\_1(object sender, RoutedEventArgs e)

{

TextBox.Text += "/";

}

private void Button\_Click\_2(object sender, RoutedEventArgs e)

{

TextBox.Text += "\*";

}

private void Button\_Click\_3(object sender, RoutedEventArgs e)

{

TextBox.Text += "-";

}

private void Button\_Click\_4(object sender, RoutedEventArgs e)

{

TextBox.Text += "+";

}

private void ravno(object sender, RoutedEventArgs e)

{

TextBox.Text = Convert.ToString(Eval(TextBox.Text)).Replace(",", ".");

}

static Double Eval(String expression)

{

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

return Convert.ToDouble(table.Compute(expression, String.Empty));

}

private void Button\_Click\_5(object sender, RoutedEventArgs e)

{

TextBox.Text += ".";

}

private void Button\_Click\_6(object sender, RoutedEventArgs e)

{

TextBox.Text = TextBox.Text.Remove(TextBox.Text.Length - 1);

}

}

}

**XAML:**

<Window x:Class="WpfApp1.MainWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:WpfApp1"

mc:Ignorable="d"

Title="MainWindow" Background="DarkMagenta" Height="419" Width="440">

<Grid Margin="10,10,10,-6">

<Grid.RowDefinitions>

<RowDefinition Height="419\*"/>

<RowDefinition Height="11\*"/>

</Grid.RowDefinitions>

<TextBox Name="TextBox" HorizontalAlignment="Left" IsEnabled="False" Height="39" Margin="9,17,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="403" RenderTransformOrigin="0.496,0.433"/>

<Button Content="CE" Background="Black" Foreground="AntiqueWhite" HorizontalAlignment="Left" Margin="340,140,0,0" VerticalAlignment="Top" Width="72" Height="51" Click="CE" />

<Button Name ="Symb\_1" Content="1" Background="Black" Foreground="AntiqueWhite" HorizontalAlignment="Left" Margin="9,200,0,0" VerticalAlignment="Top" Width="71" Height="51" Click="Symb\_1\_Click"/>

<Button Name ="Symb\_2" Content="2" Background="Black" Foreground="AntiqueWhite" HorizontalAlignment="Left" Margin="92,200,0,0" VerticalAlignment="Top" Width="71" Height="51" Click="Symb\_2\_Click"/>

<Button Name ="Symb\_3" Content="3" Background="Black" Foreground="AntiqueWhite" HorizontalAlignment="Left" Margin="175,200,0,0" VerticalAlignment="Top" Width="71" Height="51" Click="Symb\_3\_Click"/>

<Button Name ="Symb\_4" Content="4" Background="Black" Foreground="AntiqueWhite" HorizontalAlignment="Left" Margin="9,140,0,0" VerticalAlignment="Top" Width="71" Height="51" Click="Symb\_4\_Click"/>

<Button Name ="Symb\_5" Content="5" Background="Black" Foreground="AntiqueWhite" HorizontalAlignment="Left" Margin="92,140,0,0" VerticalAlignment="Top" Width="71" Height="51" Click="Symb\_5\_Click"/>

<Button Name ="Symb\_6" Content="6" Background="Black" Foreground="AntiqueWhite" HorizontalAlignment="Left" Margin="175,140,0,0" VerticalAlignment="Top" Width="71" Height="51" Click="Symb\_6\_Click"/>

<Button Name ="Symb\_7" Content="7" Background="Black" Foreground="AntiqueWhite" HorizontalAlignment="Left" Margin="9,81,0,0" VerticalAlignment="Top" Width="71" Height="51" Click="Symb\_7\_Click"/>

<Button Name ="Symb\_8" Content="8" Background="Black" Foreground="AntiqueWhite" HorizontalAlignment="Left" Margin="92,81,0,0" VerticalAlignment="Top" Width="71" Height="51" Click="Symb\_8\_Click"/>

<Button Name ="Symb\_9" Content="9" Background="Black" Foreground="AntiqueWhite" HorizontalAlignment="Left" Margin="175,81,0,0" VerticalAlignment="Top" Width="71" Height="51" Click="Symb\_9\_Click"/>

<Button Name ="Symb\_0" Content="0" Background="Black" Foreground="AntiqueWhite" HorizontalAlignment="Left" Margin="9,259,0,0" VerticalAlignment="Top" Width="154" Height="51" Click="Symb\_0\_Click"/>

<Button Content="." Background="Black" Foreground="AntiqueWhite" HorizontalAlignment="Left" Margin="175,259,0,0" VerticalAlignment="Top" Width="71" Height="51" Click="Button\_Click\_5"/>

<Button Content="/" Background="Black" Foreground="AntiqueWhite" HorizontalAlignment="Left" Margin="258,81,0,0" VerticalAlignment="Top" Width="70" Height="51" Click="Button\_Click\_1"/>

<Button Content="\*" Background="Black" Foreground="AntiqueWhite" HorizontalAlignment="Left" Margin="258,140,0,0" VerticalAlignment="Top" Width="70" Height="51" Click="Button\_Click\_2"/>

<Button Content="-" Background="Black" Foreground="AntiqueWhite" HorizontalAlignment="Left" Margin="258,200,0,0" VerticalAlignment="Top" Width="70" Height="51" Click="Button\_Click\_3"/>

<Button Content="+" Background="Black" Foreground="AntiqueWhite" HorizontalAlignment="Left" Margin="258,259,0,0" VerticalAlignment="Top" Width="70" Height="51" Click="Button\_Click\_4"/>

<Button Content="=" Background="Black" Foreground="AntiqueWhite" HorizontalAlignment="Left" Margin="340,200,0,0" VerticalAlignment="Top" Width="72" Height="110" Click="ravno"/>

<Button Content="С" Background="Black" Foreground="AntiqueWhite" HorizontalAlignment="Left" Margin="341,81,0,0" VerticalAlignment="Top" Width="72" Height="51" Click="Button\_Click\_6"/>

</Grid>

</Window>

**Вывод:** Я разработал калькулятор на C#, используя один обработчик для нескольких событий.

**ЗАДАНИЕ № 8**

**Тема: “Checkspelling”**

**Цель работы**: вычислить ветвящуюся функцию, представленную на рисунке в проекте. Функция имеет особые точки, связанные с наличием log (x+a) в знаменателе.

**Ниже на рисунке представлен фрагмент работающего проекта**

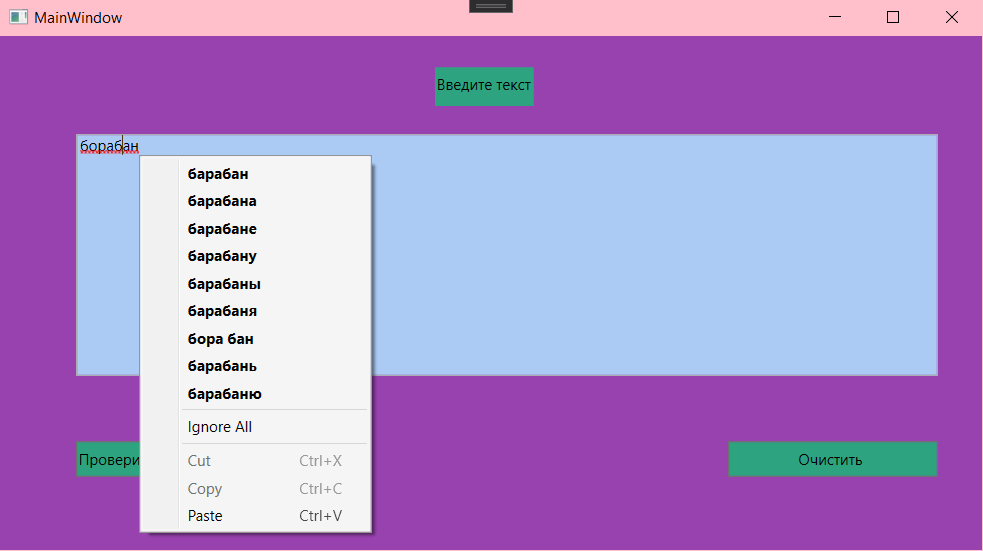


Рисунок 1. Вычисление ветвящийся функции

**Код программы:**

using System;

using System.Collections.Generic;

using System.Windows;

using Word = Microsoft.Office.Interop.Word;

namespace Checkspelling

{

public partial class MainWindow

{

private void Button\_Click(object sender, RoutedEventArgs e)

{

int count = 0;

var Word = new Microsoft.Office.Interop.Word.Application();

var res = Word.CheckSpelling(TextBox.Text);

string[] mas = TextBox.Text.Split(' ');

foreach (string s in mas)

{

if (Word.CheckSpelling(s) == false)

{

count += 1;

MessageBox.Show("В слове допущена ошибка " + s);

}

}

if (count == 0)

{

MessageBox.Show("Ваш текст полностью верен");

}

}

public MainWindow()

{

InitializeComponent();

}

private void Button\_Click\_1(object sender, RoutedEventArgs e)

{

TextBox.Clear();

}

}

}

**XAML**:

<Window x:Class="Checkspelling.MainWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:Checkspelling"

mc:Ignorable="d"

Title="MainWindow" Height="450" Width="800" Background="#9842AF">

<Grid>

<Label Content="Введите текст" HorizontalAlignment="Center" Margin="348,25,358.6,0" Background="#2DA380" VerticalAlignment="Top" Height="31" Width="86"/>

<TextBox SpellCheck.IsEnabled="True" Name ="TextBox" Background="#ACCBF4" HorizontalAlignment="Left" Margin="61,78,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="689" Height="194"/>

<Button Content="Проверить орфографию текста" HorizontalAlignment="Left" Background="#2DA380" Margin="61,324,0,0" VerticalAlignment="Top" Height="29" Click="Button\_Click" />

<Button Content="Очистить " HorizontalAlignment="Left" Margin="582,324,0,0" Background="#2DA380" VerticalAlignment="Top" Height="29" Width="168" Click="Button\_Click\_1"/>

</Grid>

</Window>

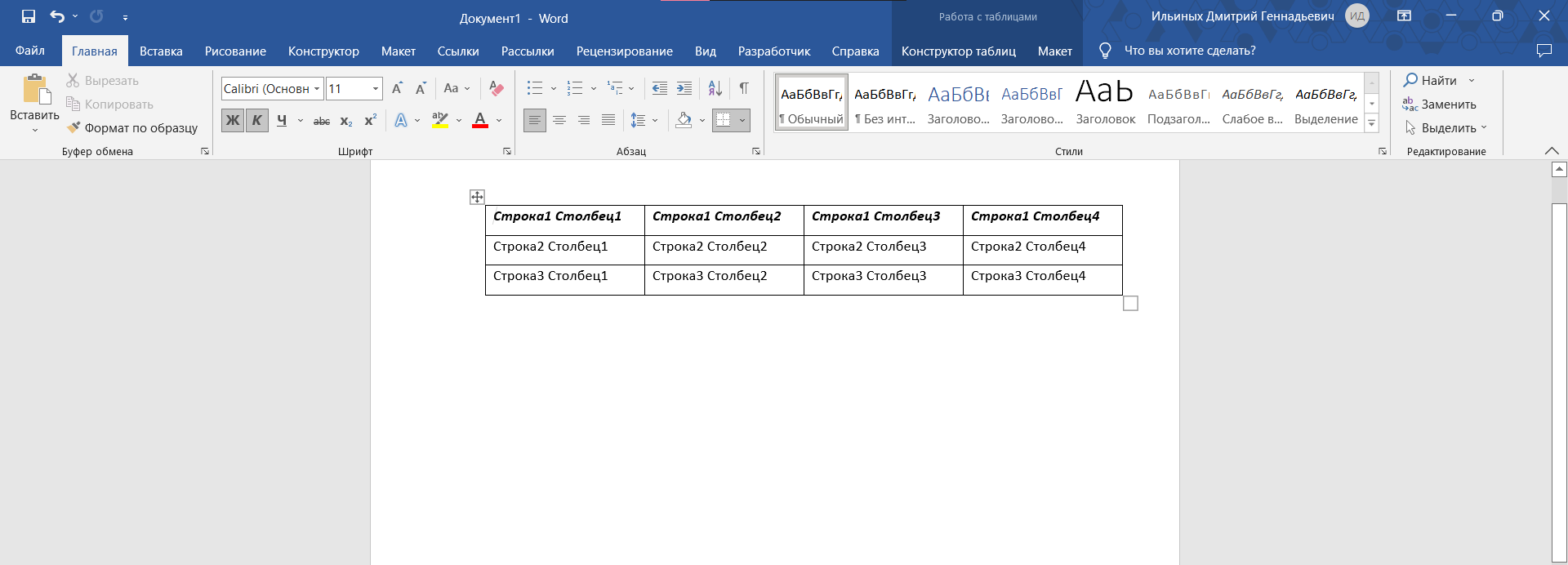
**Вывод:** Я разработал программу для проверки орфографии.

**ЗАДАНИЕ № 9**

**Тема: “Создание таблицы в Word”**

**Цель работы**: необходимо разработать программу, которая при нажатии на кнопку создаёт таблицу в MS Word.

**Ниже на рисунке представлен фрагмент работающего проекта**



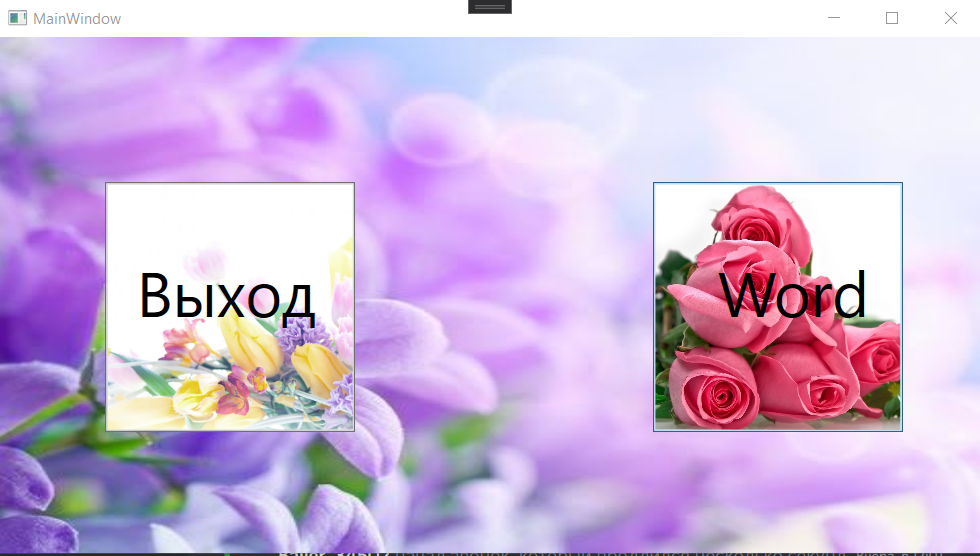


Рисунок 1. Вычисление ветвящийся функции

**Код программы:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using System.Xml.Linq;

using Microsoft.Office.Interop.Word;

using Window = System.Windows.Window;

namespace Word\_Table

{

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

}

private void Exit(object sender, RoutedEventArgs e)

{

System.Windows.Application.Current.Shutdown();

}

private void Vord(object sender, RoutedEventArgs e)

{

createtable();

}

private void createtable()

{

object oMissing = System.Reflection.Missing.Value;

object oEndOfDoc = "\\endofdoc";

Microsoft.Office.Interop.Word.\_Application objWord;

Microsoft.Office.Interop.Word.\_Document objDoc;

objWord = new Microsoft.Office.Interop.Word.Application();

objWord.Visible = true;

objDoc = objWord.Documents.Add(ref oMissing, ref oMissing,

ref oMissing, ref oMissing);

int i = 0;

int j = 0;

Microsoft.Office.Interop.Word.Table objTable;

Microsoft.Office.Interop.Word.Range wrdRng = objDoc.Bookmarks.get\_Item(ref oEndOfDoc).Range;

objTable = objDoc.Tables.Add(wrdRng, 3, 4, ref oMissing, ref oMissing);

objTable.Range.ParagraphFormat.SpaceAfter = 7;

string strText;

for (i = 1; i <= 3; i++)

for (j = 1; j <= 4; j++)

{

strText = "Строка" + i + " Столбец" + j;

objTable.Cell(i, j).Range.Text = strText;

}

objTable.Rows[1].Range.Font.Bold = 1;

objTable.Rows[1].Range.Font.Italic = 1;

objTable.Borders.Shadow = true;

this.Close();

}

}

}

**XAML**:

<Window x:Class="Word\_Table.MainWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:Word\_Table"

mc:Ignorable="d"

Title="MainWindow" Height="450" Width="800">

<Grid>

<Image Source="C:\Users\Дима\Pictures\912e09f7d513f20ae76cf444682e42ba.jpg" Stretch="Fill" RenderTransformOrigin="0.505,0.535"/>

<Button HorizontalAlignment="Left" Margin="523,116,0,0" VerticalAlignment="Top" RenderTransformOrigin="0.609,0.413" Height="200" Width="200" Click="Vord">

<Image Source="C:\Users\Дима\Pictures\Holidays\_\_\_International\_Womens\_Day\_Pink\_roses\_on\_March\_8\_057380\_.jpg" RenderTransformOrigin="0.5,0.5" Height="384" Width="365">

<Image.RenderTransform>

<TransformGroup>

<ScaleTransform/>

<SkewTransform AngleY="0.182"/>

<RotateTransform/>

<TranslateTransform Y="0.397"/>

</TransformGroup>

</Image.RenderTransform>

</Image>

</Button>

<Button HorizontalAlignment="Left" Margin="85,116,0,0" VerticalAlignment="Top" Height="200" Width="200" Click ="Exit">

<Image Source="C:\Users\Дима\Pictures\depositphotos\_9129064-stock-photo-collage-postcard-art-background-mix.jpg" Height="243" Width="295"/>

</Button>

<Label Content="Word" HorizontalAlignment="Left" Margin="570,165,0,0" FontSize ="50" VerticalAlignment="Top" Height="86" Width="153"/>

<Label Content="Выход" HorizontalAlignment="Left" Margin="105,165,0,0" FontSize ="50" VerticalAlignment="Top" Height="86" Width="153" RenderTransformOrigin="0.829,0.614"/>

</Grid>

</Window>

**Вывод:** Я разработал программу для создания таблицы в Microsoft Word.

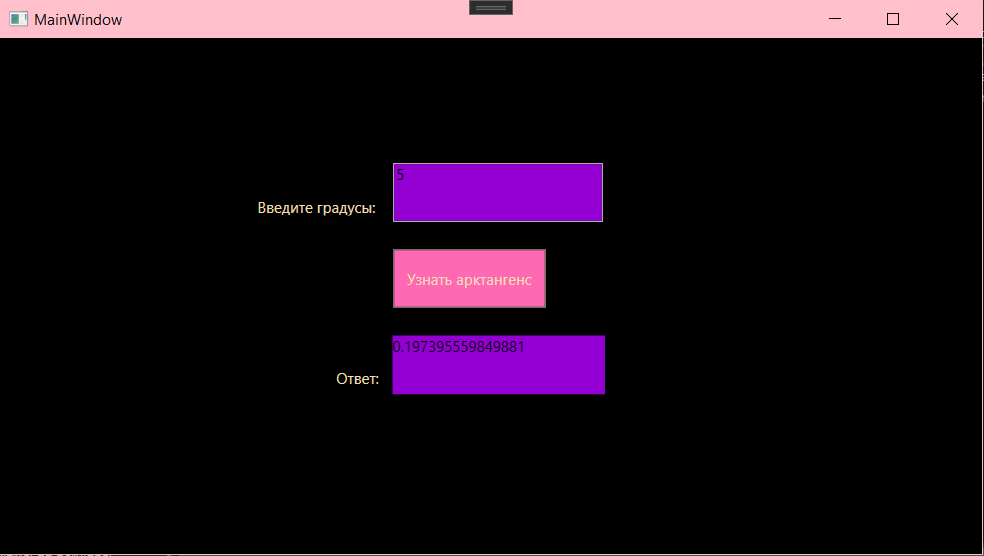
**ЗАДАНИЕ № 10**

**Тема: “Использование функций Excel”**

**Цель работы**: Cоздать собственный проект с использованием функций MS

Excel не используя функцию нахождения числа Pi.

**Ниже на рисунке представлен фрагмент работающего проекта**



**Код программы:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace ExcelForm

{

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

}

private void ToRad(object sender, EventArgs e)

{

var XL = new Microsoft.Office.Interop.Excel.Application();

var rad = XL.WorksheetFunction.Acot(Convert.ToDouble(TextBox.Text));

XL.Quit();

}

private void Button\_Click(object sender, RoutedEventArgs e)

{

var XL = new Microsoft.Office.Interop.Excel.Application();

var rad = XL.WorksheetFunction.Acot(Convert.ToDouble(TextBox.Text));

TxtBlck.Text = Convert.ToString(rad);

XL.Quit();

}

}

}

**XAML:**

<Window x:Class="ExcelForm.MainWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:ExcelForm"

mc:Ignorable="d"

Title="MainWindow" Height="450" Width="800" Background="Black">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="49\*"/>

<RowDefinition Height="11\*"/>

</Grid.RowDefinitions>

<TextBox Name="TextBox" Background="DarkViolet" Foreground="Black" HorizontalAlignment="Left" Margin="314,100,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="168" Height="47"/>

<Label Content="Введите градусы:" Foreground="Wheat" HorizontalAlignment="Left" Margin="201,122,0,0" VerticalAlignment="Top" Height="25" Width="108"/>

<Button Content="Узнать арктангенс" HorizontalAlignment="Left" Foreground="Wheat" Background="HotPink" Margin="314,169,0,0" VerticalAlignment="Top" Click="Button\_Click" Height="47" Width="123"/>

<TextBlock Name="TxtBlck" Background="DarkViolet" HorizontalAlignment="Left" Margin="314,238,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Height="47" Width="170"/>

<Label Content="Ответ:" Foreground="Wheat" HorizontalAlignment="Left" Margin="264,259,0,0" VerticalAlignment="Top" Height="26" Width="45"/>

</Grid>

</Window>

**Вывод:** Я создал собственный проект с использованием функций MS

Excel для нахождения арктангенса.

**ЗАДАНИЕ № 7**

**Тема: “Текстовый редактор”**

**Цель работы**: написать программу на языке C#, которая реализует

работу с файлами в различных кодировках.

Программа должна обладать следующим функционалом:

Чтение информации из файла в кодировке Unicode

Чтение информации из файла в кодировке Win1251

Чтение информации из файла с разрешением RTF

Чтение бинарных файлов

Сохранение информации в файл с кодировкой Unicode

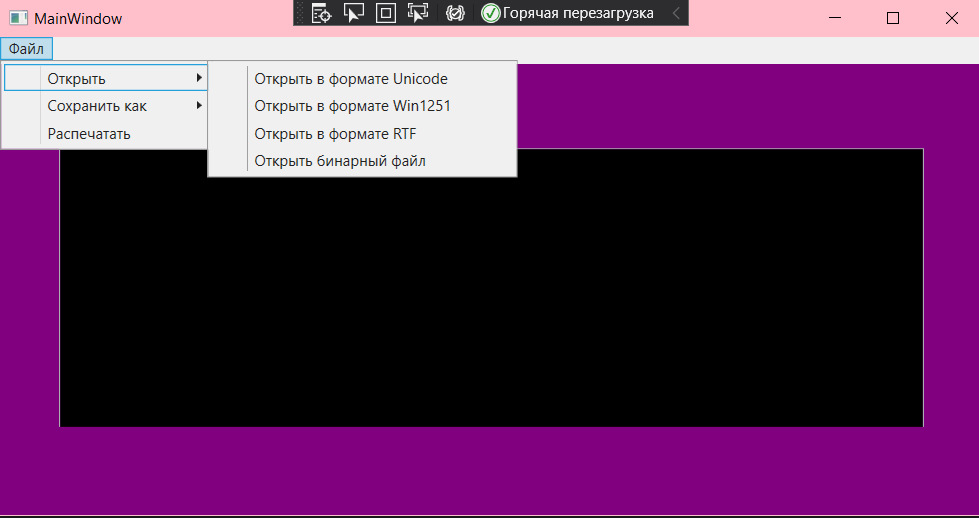
Сохранение информации в файл с кодировкой Win1251

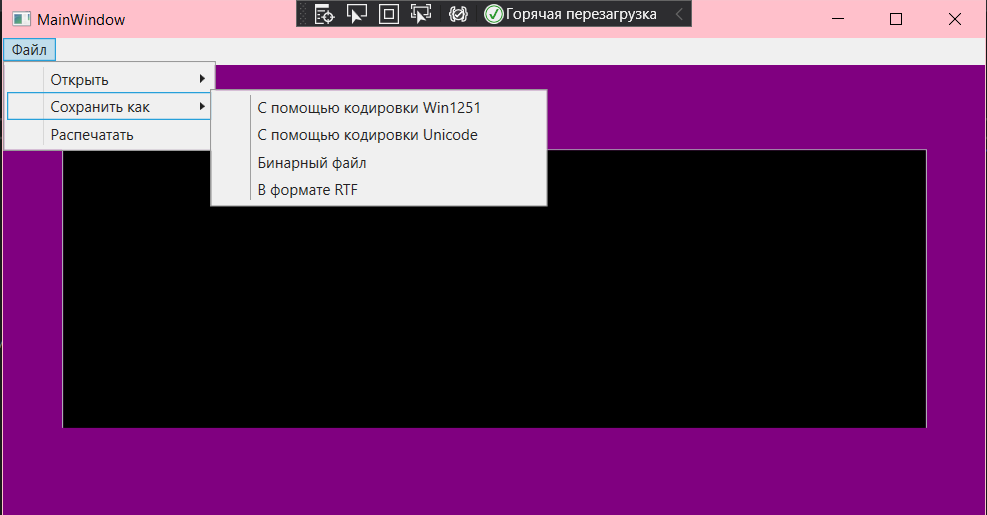
Сохранение информации в файл с разрешением RTF

Сохранение информации в бинарный файл

Печать текстового документа.

**Ниже на рисунке представлен фрагмент работающего проекта**





**Код программы:**

using Microsoft.Win32;

using System;

using System.IO;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using Microsoft.SqlServer.Server;

using System.Runtime.Serialization.Formatters.Binary;

namespace Text\_editor

{

public partial class MainWindow : Window

{

Microsoft.Win32.SaveFileDialog dlg = new Microsoft.Win32.SaveFileDialog();

Microsoft.Win32.OpenFileDialog odlg = new Microsoft.Win32.OpenFileDialog();

public MainWindow()

{

InitializeComponent();

dlg.FileName = "Документ";

odlg.FileName = "Документ";

}

private void Save\_as\_RTF(object sender, RoutedEventArgs e)

{

dlg.DefaultExt = ".rtf";

dlg.Filter = "Текстовый документ (.rtf)|\*.rtf";

string richText = new TextRange(rtxtbx.Document.ContentStart, rtxtbx.Document.ContentEnd).Text;

if (dlg.ShowDialog() == false)

{

return;

}

string filename = dlg.FileName;

File.WriteAllText(filename, richText);

MessageBox.Show("Файл успешно сохранён!");

}

private void Save\_as\_Unicode(object sender, RoutedEventArgs e)

{

dlg.DefaultExt = ".txt";

dlg.Filter = "Текстовый документ (.txt)|\*.txt";

string richText = new TextRange(rtxtbx.Document.ContentStart, rtxtbx.Document.ContentEnd).Text;

if (dlg.ShowDialog() == false)

{

return;

}

string filename = dlg.FileName;

File.WriteAllText(filename, richText, Encoding.Unicode);

MessageBox.Show("Файл успешно сохранён!");

}

private void Save\_as\_Win1251(object sender, RoutedEventArgs e)

{

dlg.DefaultExt = ".txt";

dlg.Filter = "Текстовый документ (.txt)|\*.txt";

string richText = new TextRange(rtxtbx.Document.ContentStart, rtxtbx.Document.ContentEnd).Text;

if (dlg.ShowDialog() == false)

{

return;

}

string filename = dlg.FileName;

File.WriteAllText(filename, richText, Encoding.GetEncoding("windows-1251"));

MessageBox.Show("Файл успешно сохранён!");

}

private void Save\_as\_BinaryFormat(object sender, RoutedEventArgs e)

{

dlg.DefaultExt = ".bin";

dlg.Filter = "Текстовый документ (.bin)|\*.bin";

BinaryFormatter bf = new BinaryFormatter();

if (dlg.ShowDialog() == false)

{

return;

}

string filename = dlg.FileName;

string richText = new TextRange(rtxtbx.Document.ContentStart, rtxtbx.Document.ContentEnd).Text;

using (BinaryWriter writer = new BinaryWriter(File.Open(filename, FileMode.OpenOrCreate)))

{

writer.Write(richText);

MessageBox.Show("Файл успешно сохранён!(на рабочий стол)");

}

}

private void Open\_as\_RTF(object sender, RoutedEventArgs e)

{

rtxtbx.Document.Blocks.Clear();

odlg.DefaultExt = ".rtf";

odlg.Filter = "Текстовый документ (.rtf)|\*.rtf";

bool? dialogResult = odlg.ShowDialog();

if (dialogResult.HasValue && dialogResult.Value)

{

TextRange range = new TextRange(rtxtbx.Document.ContentStart, rtxtbx.Document.ContentEnd);

System.IO.FileStream file = new System.IO.FileStream(odlg.FileName, System.IO.FileMode.Open, System.IO.FileAccess.Read);

range.Load(file, System.Windows.DataFormats.Rtf);

file.Dispose();

MessageBox.Show("Файл открыт!");

}

}

private void Open\_as\_Unicode(object sender, RoutedEventArgs e)

{

string filename = "";

rtxtbx.Document.Blocks.Clear();

bool? result = odlg.ShowDialog();

if (result == true)

{

filename = odlg.FileName;

}

string filetext = File.ReadAllText(filename);

rtxtbx.Document.Blocks.Add(new Paragraph(new Run(filetext)));

MessageBox.Show("Файл открыт!");

}

private void Open\_as\_Win1251(object sender, RoutedEventArgs e)

{

string filename = "";

rtxtbx.Document.Blocks.Clear();

bool? result = odlg.ShowDialog();

if (result == true)

{

filename = odlg.FileName;

}

string filetext = File.ReadAllText(filename, System.Text.Encoding.GetEncoding("windows-1251"));

rtxtbx.Document.Blocks.Add(new Paragraph(new Run(filetext)));

MessageBox.Show("Файл открыт!");

}

private void Open\_as\_Binary(object sender, RoutedEventArgs e)

{

string filename = "";

rtxtbx.Document.Blocks.Clear();

bool? result = odlg.ShowDialog();

//if (result == true)

{

filename = odlg.FileName;

}

string filetext = File.ReadAllText(filename);

rtxtbx.Document.Blocks.Add(new Paragraph(new Run(filetext)));

MessageBox.Show("Файл открыт!");

}

private void PrintWindow(object sender, RoutedEventArgs e)

{

PrintDialog p = new PrintDialog();

if (p.ShowDialog() == true)

{

p.PrintVisual(grid1, "Печать");

}

}

}

}

**XAML:**

<Window x:Class="Text\_editor.MainWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:Text\_editor"

mc:Ignorable="d"

Title="MainWindow" Height="420" Width="800" Background="Purple">

<Grid Name="grid1">

<Grid.RowDefinitions>

<RowDefinition Height="177\*"/>

<RowDefinition Height="40\*"/>

</Grid.RowDefinitions>

<Menu Margin="0,0,0,290">

<MenuItem Header="Файл" Height="18">

<MenuItem Header="Открыть">

<MenuItem Header="Открыть в формате Unicode" Click="Open\_as\_Unicode"/>

<MenuItem Header="Открыть в формате Win1251" Click="Open\_as\_Win1251"/>

<MenuItem Header="Открыть в формате RTF" Click="Open\_as\_RTF" />

<MenuItem Header="Открыть бинарный файл" Click="Open\_as\_Binary"/>

</MenuItem>

<MenuItem Header="Сохранить как">

<MenuItem Header="С помощью кодировки Win1251" Click="Save\_as\_Win1251"/>

<MenuItem Header="С помощью кодировки Unicode" Click="Save\_as\_Unicode"/>

<MenuItem Header="Бинарный файл" Click="Save\_as\_BinaryFormat"/>

<MenuItem Header="В формате RTF" Click="Save\_as\_RTF"/>

</MenuItem>

<MenuItem Header="Распечатать" Click="PrintWindow"/>

</MenuItem>

</Menu>

<RichTextBox Name="rtxtbx" HorizontalAlignment="Center" Background="Black" VerticalAlignment="Top" Width="692" Height="256" Margin="0,89,0,0">

<FlowDocument>

<Paragraph>

<Run/>

</Paragraph>

</FlowDocument>

</RichTextBox>

</Grid>

</Window>

**Вывод:** Я разработал текстовый редактор на c#.