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 дисциплине ПМ.01.01 Учебная практика

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

Выполнил:

обучающийся учебной группы № 3ПКС-420

Кокарев Н.А.

*(подпись) (И.О. Фамилия)*

Проверил:

Руководитель практики от колледжа:

Е. Л. Альшакова, И. В. Сиберев

*(подпис ) (И.О. Фамилия)*

*(оценка) (подпись)*

**Москва – 2022**

Специальность Программирование компьютерных систем

*(наименование специальности)*

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

|  |  |  |
| --- | --- | --- |
| **№п/п** | **Темы учебной практики** | **Выполненные задания/работы** |
| **1** | Локальные данные процедур | Дано 50 чисел, среди которых есть по крайней мере одно отрицательное. Найти наибольшее среди отрицательных чисел |
| **2** | Процедуры | Ввести одномерный массив m = {-1, 0, 10, -3, -5, 6, -2, 3, 4}. Вычислить сумму элементов с нечетными номерами. Сформировать и вывести массив n, элементами которого являются индексы положительных элементов массива m. 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 обнулить элементы каждой строки матрицы, большие 5. Исходную матрицу и полученные результаты вывести на экран |
| **5** | Работа с сегментами | Количество чисел, меньших C1 |
| **6** | Работа со стеками |  |

**Практическая работа № 1**

Дано 50 чисел, среди которых есть по крайней мере одно отрицательное. Найти наибольшее среди отрицательных чисел

format PE console

include 'win32a.inc'

start:

proc func

mov ECX,[N]

lea ESI, [X]

mov EBX,0

mov DL,0

SEARCH:

lodsd

or EAX,EAX

jns NEXT

or DL,DL

jnz COMPARE

mov EBX,EAX

mov DL,1

COMPARE:

cmp EAX,EBX

jle NEXT

mov BX,AX

NEXT: loop SEARCH

cinvoke printf, text1,ebx

invoke sleep, 9000

invoke exit, 0

mov sp,bp

pop bp

ret

endp

call func

invoke sleep, 9000

invoke exit, 0

text1 db 'Max negative= %d',0

fpp db '%d',9, 0

N dd 50

X dd -3, 0, 2, 5, 10, -5, 0, 77, 69, -13, -14, 9, 1, 4, 1, 8, 9, 0, -3, 10, 19, -42, 5, 90,71

dd 9, 5, -9, 45, -90, -10, -3, 6, 14, 456, 122, 34, 87, 99, 12, 11, 321, 67, 81, 44, 92, 55, 2, -89, 100

data import

library msvcrt,'MSVCRT.DLL',\

kernel32,'KERNEL32.DLL'

import kernel32,\

sleep,'Sleep'

import msvcrt,\

printf,'printf',\

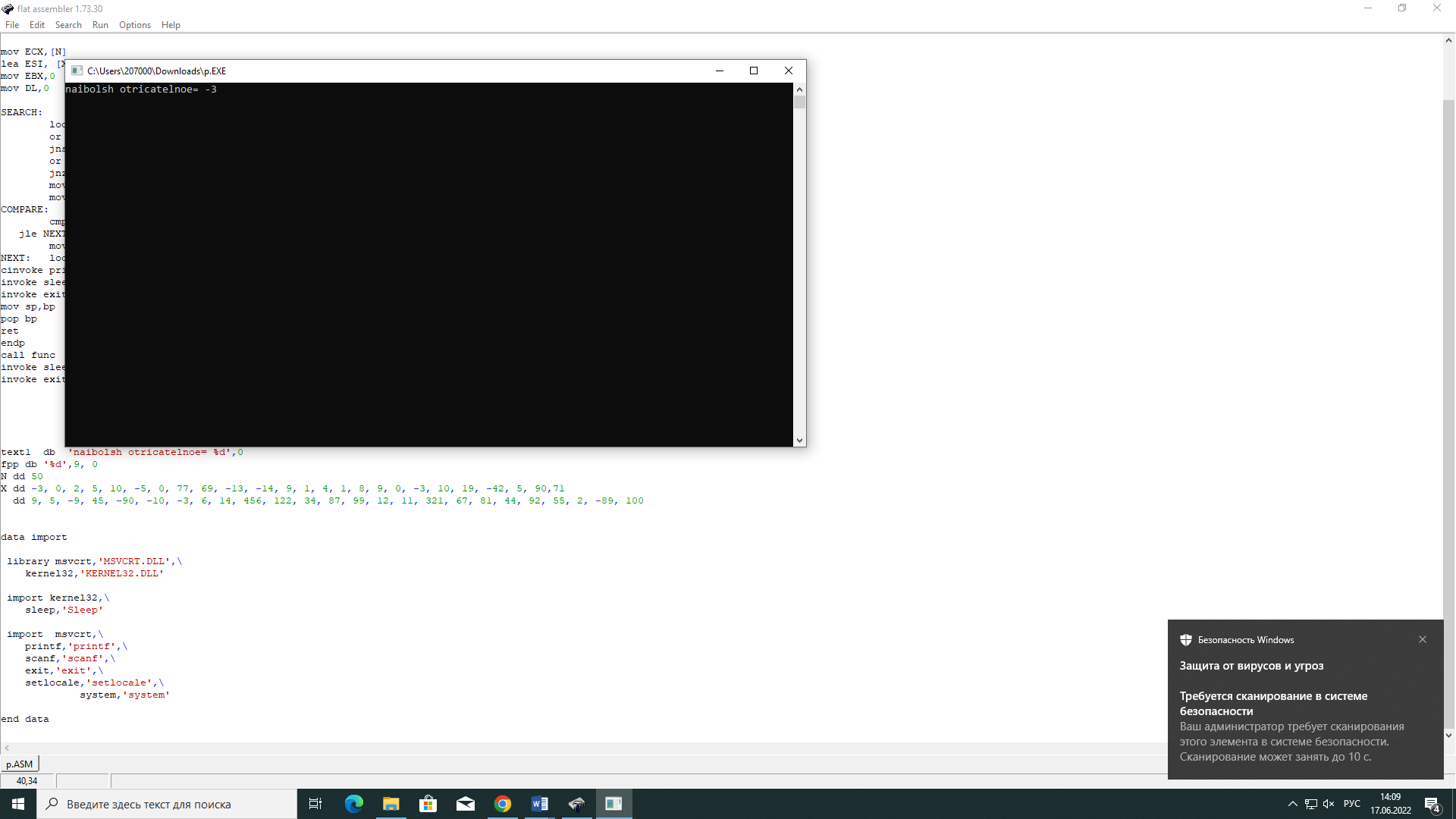
scanf,'scanf',\

exit,'exit',\

setlocale,'setlocale',\

system,'system'

end data



**Практическая работа № 2**

Ввести одномерный массив m = {-1, 0, 10, -3, -5, 6, -2, 3, 4}. Вычислить сумму элементов с нечетными номерами. Сформировать и вывести массив n, элементами которого являются индексы положительных элементов массива m. Программу разработать на основе пользовательских функций.

macro prnt op1, op2 {

local metka

mov ebx, op1

mov ecx,op2

mov eax, 0

metka :

mov eax,[ebx]

push ecx

cinvoke printf, fpp, eax

pop ecx

add ebx, 4

loop metka

}

prnt m,9

lea ebx, [m]

mov ecx,5

mov eax,0

cycl:

add eax, [ebx]

add ebx,8

loop cycl

cinvoke printf, text1, eax

lea ebx, [m]

mov ecx,9

mov eax,0

mov edx,0

mov esi,n

cycl2:

mov eax,[ebx]

cmp eax,0

jg met

add ebx,4

inc edx

loop cycl2

met:

inc [amount]

mov [esi], edx

add esi, 4

add ebx,4

inc edx

loop cycl2

prnt n,[amount]

**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.

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',\

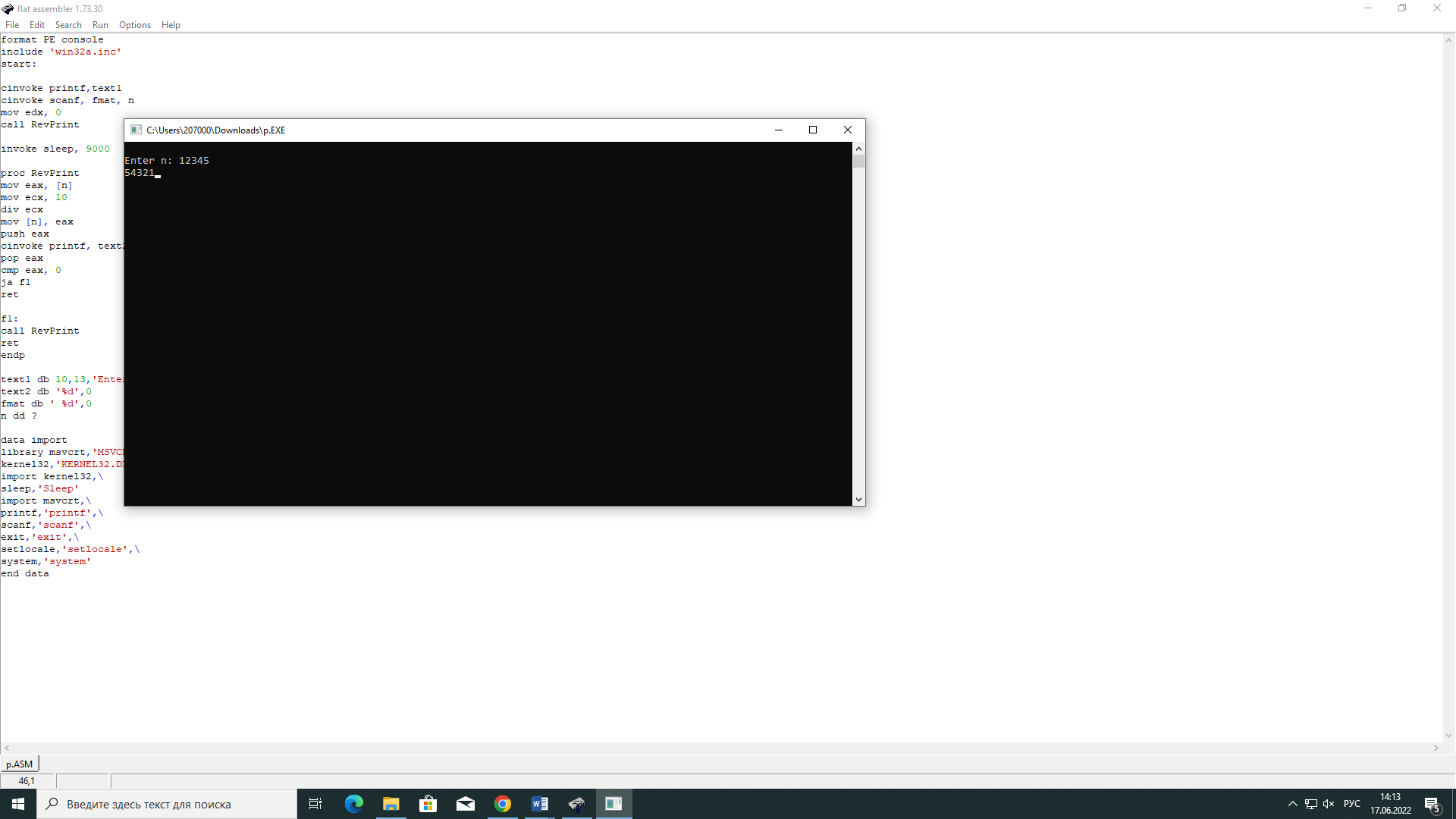
scanf,'scanf',\

exit,'exit',\

setlocale,'setlocale',\

system,'system'

end data



Задание 2

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

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',\

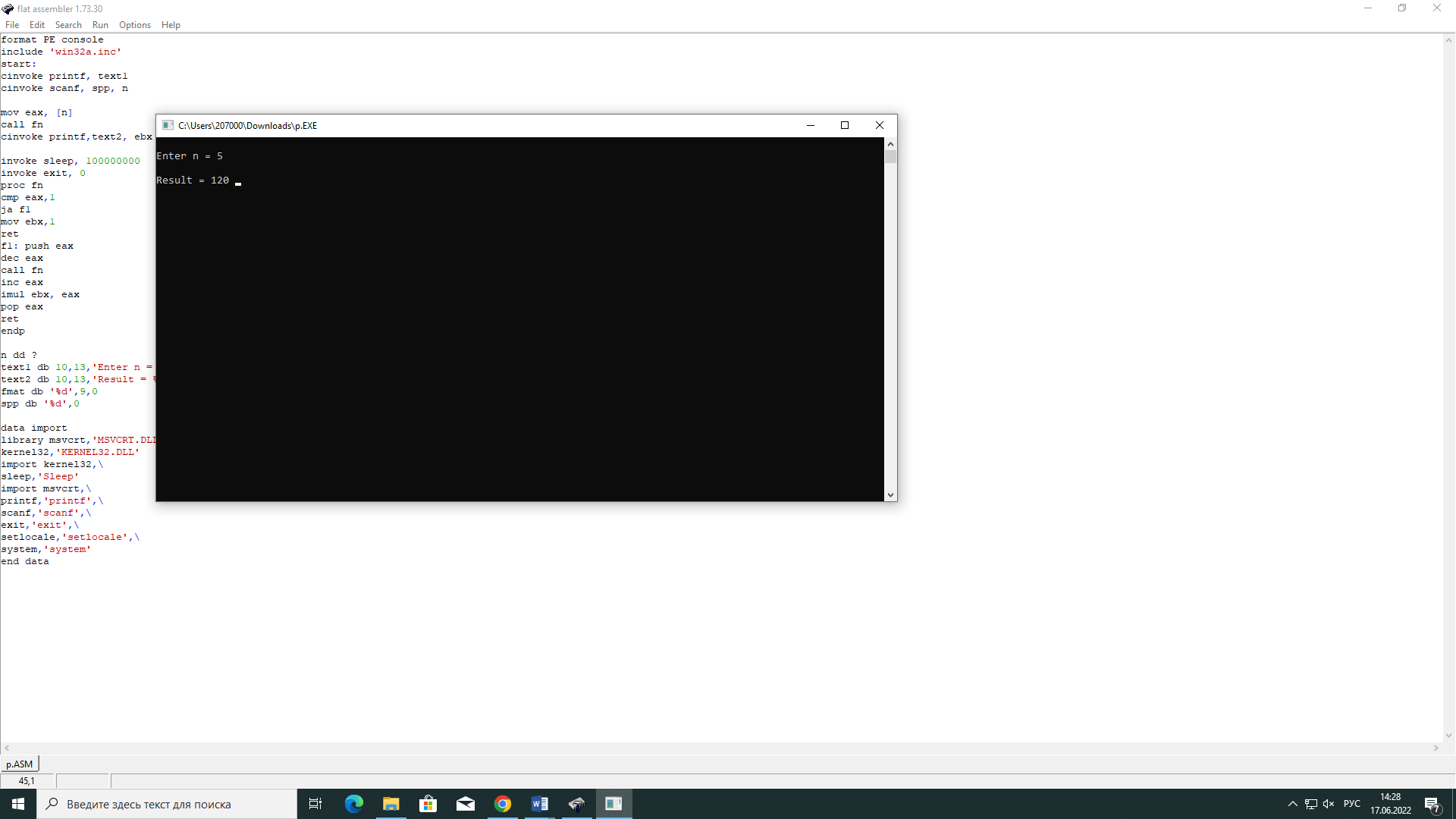
scanf,'scanf',\

exit,'exit',\

setlocale,'setlocale',\

system,'system'

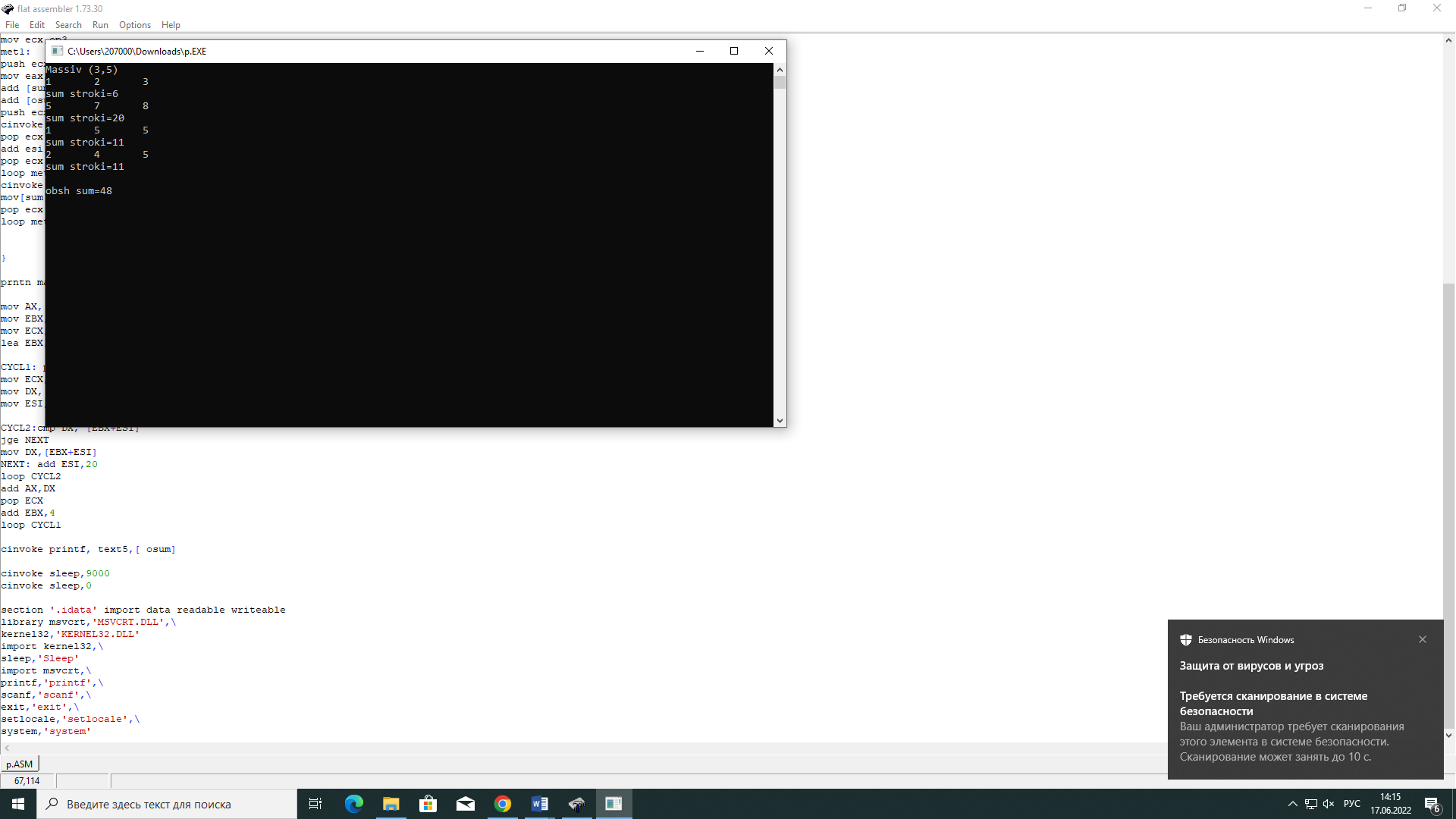
end data



**Практическая работа № 4**

Разместить в памяти компьютера матрицу 4\*3 целых беззнаковых чисел. Написать программу, позволяющую обнулить элементы каждой строки матрицы, большие 5. Исходную матрицу и полученные результаты вывести на экран

format PE console  
entry start  
include 'win32a.inc'  
  
  
section '.data' data readable writeable  
text1 db 'Massiv (3,5)',13,10,0  
text5 db 13,10,'obsh sum=%d',13,10,0  
text3 db 13,10,'sum stroki=%d',10,13,0  
text4 db '',13,10,10, 0  
fmat db '%d',9,0  
l dd 4  
v dd 3  
p db 'pause',0  
sum dd 0  
osum dd 0  
mas dd 1,2,3  
dd 5,7,8  
dd 1,5,5  
dd 2,4,5  
  
  
start:  
  
cinvoke printf, text1  
macro prntn op1, op2,op3 {  
local metka  
mov esi,op1  
mov ecx,op2  
met2:  
push ecx  
mov ecx,op3  
met1:  
push ecx  
mov eax,[esi]  
add [sum],eax  
add [osum],eax  
push ecx  
cinvoke printf, fmat, eax  
pop ecx  
add esi, 4  
pop ecx  
loop met1  
cinvoke printf,text3, [sum]  
mov[sum],0  
pop ecx  
loop met2  
  
  
}  
  
prntn mas, 4,3  
  
mov AX, 0  
mov EBX, 0  
mov ECX,5  
lea EBX, [mas]  
  
CYCL1: push ECX  
mov ECX,2  
mov DX,[EBX]  
mov ESI,20  
  
CYCL2:cmp DX, [EBX+ESI]  
jge NEXT  
mov DX,[EBX+ESI]  
NEXT: add ESI,20  
loop CYCL2  
add AX,DX  
pop ECX  
add EBX,4  
loop CYCL1  
  
cinvoke printf, text5,[ osum]  
  
cinvoke sleep,9000  
cinvoke sleep,0  
  
section '.idata' import data readable writeable  
library msvcrt,'MSVCRT.DLL',\  
kernel32,'KERNEL32.DLL'  
import kernel32,\  
sleep,'Sleep'  
import msvcrt,\  
printf,'printf',\  
scanf,'scanf',\  
exit,'exit',\  
setlocale,'setlocale',\  
system,'system'



**Практическая работа № 5**

Количество чисел, меньших C1

format PE console

include 'win32a.inc'

entry start

proc zxc

cycl:

cmp eax,[ebx]

jg met

jmp ex

met:

inc [count]

ex:

add ebx,4

loop cycl

ret

endp

start:

macro prnt op1, op2 {

local metka

mov ebx, op1

mov ecx,op2

mov eax, 0

metka :

mov eax,[ebx]

push ecx

cinvoke printf, fpp, eax

pop ecx

add ebx, 4

loop metka

}

cinvoke printf,text2,0

prnt m,10

lea ebx, [m]

mov ecx,10

mov eax,[ebx]

call zxc

cinvoke printf,text1,[count]

cinvoke system,p

invoke exit,0

text1 db 10, 13, 'Result = %d',10, 0

text2 db 10, 13, 'massive:',10, 0

fpp db '%d ', 0

spp db '%d',0

p db 'pause',0

m dd 5,2,2,2,2,2,2,2,2,10

count dd 0

data import

library msvcrt,'MSVCRT.DLL',\

kernel32,'KERNEL32.DLL'

import kernel32,\

sleep,'Sleep'

import msvcrt,\

printf,'printf',\

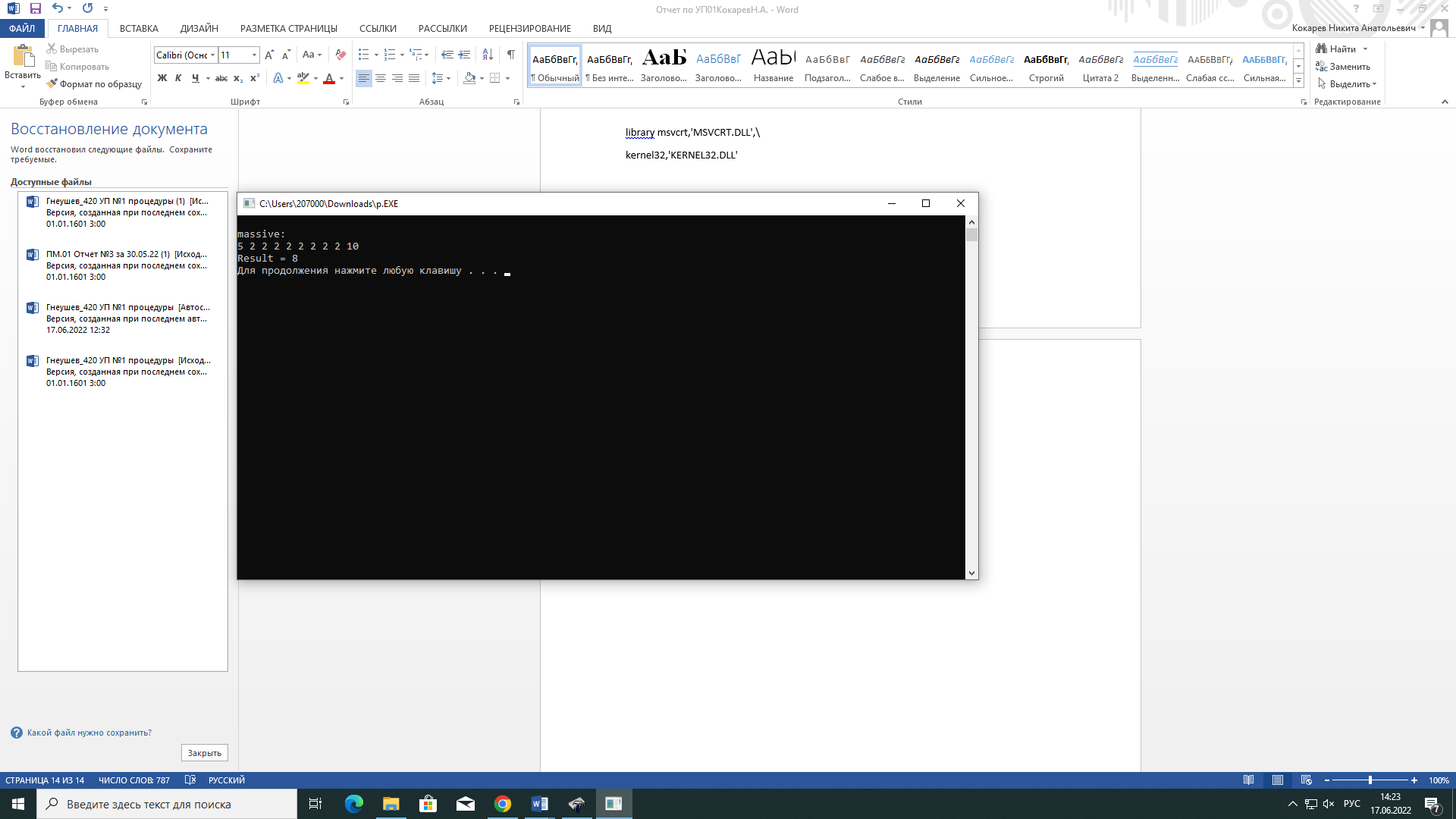
scanf,'scanf',\

exit,'exit',\

setlocale,'setlocale',\

system,'system'

end data



**Задание 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();

raskraska(n, k);

Console.WriteLine();

}

/// <param name="n">Высота</param>

/// <param name="k">Длина</param>

static void raskraska(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.Red;

}

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.Green;

}

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.Magenta;

}

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.Blue;

}

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.Yellow;

}

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.Red;

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

Console.WriteLine("\t\t\t\t\t\t Мин элемент первой диагонали " + max1);

Console.WriteLine("\t\t\t\t\t\t Мин элемент первой диагонали " + min1);

Console.ForegroundColor = ConsoleColor.Green;

Console.WriteLine("\t\t\t\t\t\t Сумма элементов второй диагонали " + sum2);

Console.WriteLine("\t\t\t\t\t\t Мин элемент второй диагонали " + max2);

Console.WriteLine("\t\t\t\t\t\t Мин элемент второй диагонали " + min2);

Console.ForegroundColor = ConsoleColor.Magenta;

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

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

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

Console.ForegroundColor = ConsoleColor.Yellow;

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

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

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

Console.ForegroundColor = ConsoleColor.Blue;

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

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

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

Console.ForegroundColor = ConsoleColor.White;

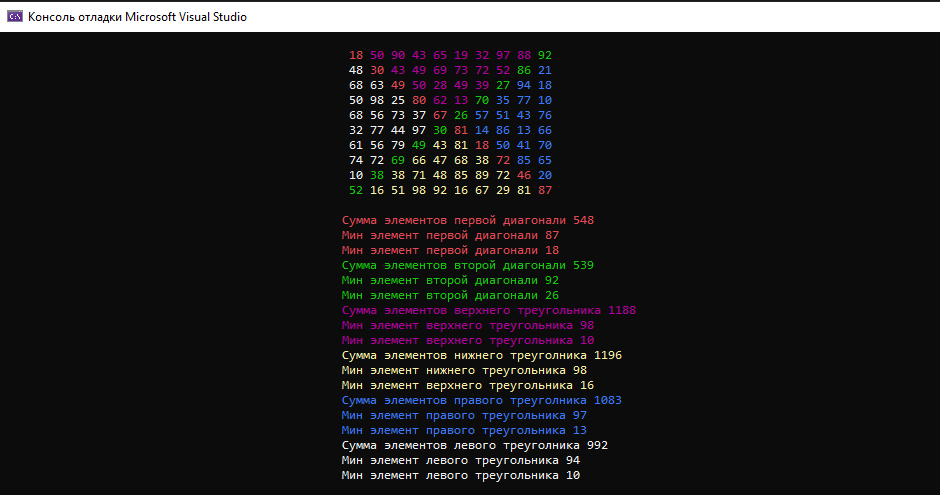
Console.WriteLine("\t\t\t\t\t\t Сумма элементов левого треуголника " + sum4);

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

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

}

}



**Задание 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.

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)//загрузить в файл

{

if (Data.ItemsSource == null) return;

StreamWriter sw = new StreamWriter("D:/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;

}

}

}

**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" Height="450" Width="800">

<Grid>

<Grid.Background>

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

<GradientStop Color="#CCCCFF" Offset="0"/>

<GradientStop Color="#CC99FF" Offset="1"/>

</LinearGradientBrush>

</Grid.Background>

<Button Name="Exit\_click" Content="Выход" HorizontalAlignment="Left" Margin="367,312,0,0" VerticalAlignment="Top" Width="184" Height="63" Click="Exit\_Click" >

<Button.Background>

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

<GradientStop Color="#CCCCCC" Offset="0"/>

<GradientStop Color="#99FFFF" Offset="1"/>

</LinearGradientBrush>

</Button.Background>

</Button>

<Button Name="Calculate" Content="Вычислить" HorizontalAlignment="Left" Margin="371,233,0,0" VerticalAlignment="Top" Width="180" RenderTransformOrigin="0.5,0.5" Height="64" Click="Calculate\_click" BorderBrush="#FFEEDDDD">

<Button.Background>

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

<GradientStop Color="#CCCCCC" Offset="0"/>

<GradientStop Color="#99FFFF" Offset="1"/>

</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="594,312,0,0" VerticalAlignment="Top" Width="176" Height="63" FontSize="16" Click="ClearGridData\_Click" >

<Button.Background>

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

<GradientStop Color="#CCCCCC" Offset="0"/>

<GradientStop Color="#99FFFF" Offset="1"/>

</LinearGradientBrush>

</Button.Background>

</Button>

<Button Name="Write" Content="Записать в файл" HorizontalAlignment="Left" Margin="593,235,0,0" VerticalAlignment="Top" Width="176" Height="62" Click="Write\_Click" >

<Button.Background>

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

<GradientStop Color="#CCCCCC" Offset="0"/>

<GradientStop Color="#99FFFF" Offset="1"/>

</LinearGradientBrush>

</Button.Background>

<Button.RenderTransform>

<TransformGroup>

<ScaleTransform/>

<SkewTransform/>

<RotateTransform Angle="-0.053"/>

<TranslateTransform/>

</TransformGroup>

</Button.RenderTransform>

</Button>

<TextBox Name="input" HorizontalAlignment="Left" Height="34" Margin="117,91,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="220"/>

<TextBlock Margin="82,90,665,294" Text="a=" FontSize="22">

</TextBlock>

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

<DataGrid.Background>

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

<GradientStop Color="#CCCCCC" Offset="0"/>

<GradientStop Color="#99FFFF" Offset="1"/>

</LinearGradientBrush>

</DataGrid.Background>

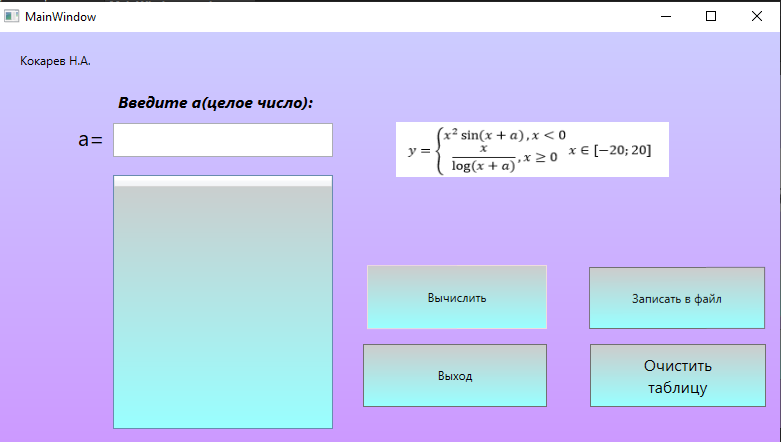
</DataGrid>

<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="400,90,111,266" Source="Image1.png" Stretch="Fill"/>

<Label Content="Кокарев Н.А." HorizontalAlignment="Left" Margin="19,15,0,0" VerticalAlignment="Top"/>

</Grid>

</Window>****

**Задание 3.** Создать программу с использованием листбокса. Добавить окно ввода списка продуктов, обработать исключения (пробелы и тд). Добавить кнопки:

- «Добавить» - добавляет в листбокс наименование товара

- «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.

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; //количество продуктов

public MainWindow()

{

InitializeComponent();

input.Focus();

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

}

private void click\_exit(object sender, RoutedEventArgs e)//выход из организации списка (с закрытием списка)

{

this.Close();

}

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

{

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)//удаление продукта

{

if (list.SelectedValue != null)

{

MessageBoxResult result = MessageBox.Show("Вы желаете удалить продукт?", "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("список 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)//очистка списка

{

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)//сортировка

{

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

input.Focus();

}

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

{

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();

}

}

}

}

**XAML:**

<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>

<Grid.Background>

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

<GradientStop Color="#CCCCFF" Offset="0"/>

<GradientStop Color="#CC99FF" Offset="1"/>

</LinearGradientBrush>

</Grid.Background>

<TextBlock HorizontalAlignment="Center" Height="28" Margin="0,10,0,0" TextWrapping="Wrap" Text="Список продуктов в магазине" VerticalAlignment="Top" Width="270" FontSize="20" FontFamily="Times New Roman" FontStyle="Italic"/>

<ListBox Name ="list" HorizontalAlignment="Right" Height="300" Margin="0,75,50,0" VerticalAlignment="Top" Width="235"/>

<TextBox Name ="input" Height="25" Margin="50,75,0,0" TextWrapping="Wrap" Width="400"

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

<TextBox Name ="quantity" Height="25" Margin="250,120,0,0" TextWrapping="Wrap" Width="200" 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="50,175,0,0" VerticalAlignment="Top" Width="125" FontFamily="Times New Roman" FontSize="16" FontStyle="Italic"/>

<Button Name ="button\_delete" Click ="click\_delete" Content="Удалить" HorizontalAlignment="Left" Height="40" Margin="225,175,0,0" VerticalAlignment="Top" Width="125" FontFamily="Times New Roman" FontSize="16" FontStyle="Italic"/>

<Button Name ="button\_edit" Click ="click\_edit" Content="Редактировать" HorizontalAlignment="Left" Height="40" Margin="50,266,0,0" VerticalAlignment="Top" Width="125" FontFamily="Times New Roman" FontSize="16" FontStyle="Italic"/>

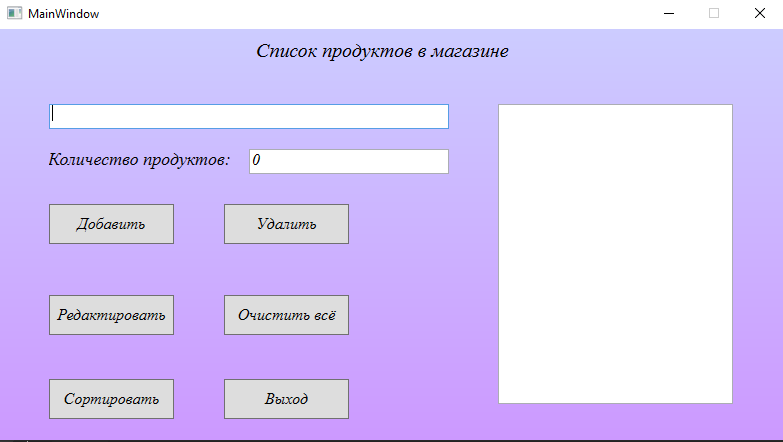
<Button Name ="button\_sort" Click ="click\_sort" Content="Сортировать" HorizontalAlignment="Left" Height="40" Margin="50,350,0,0" VerticalAlignment="Top" Width="125" FontFamily="Times New Roman" FontSize="16" FontStyle="Italic"/>

<Button Name ="button\_clear" Click ="click\_clear" Content="Очистить всё" HorizontalAlignment="Left" Height="40" Margin="225,266,0,0" VerticalAlignment="Top" Width="125" FontFamily="Times New Roman" FontSize="16" FontStyle="Italic"/>

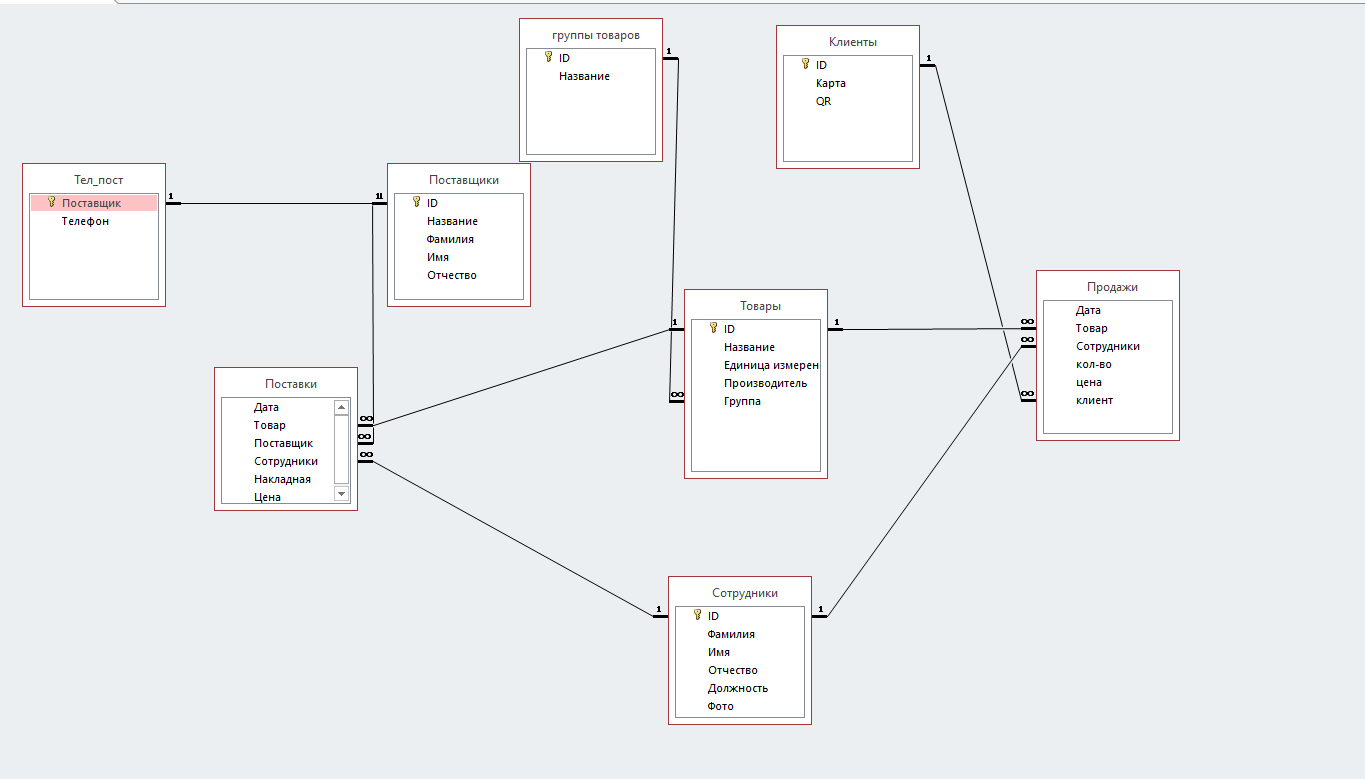
<Button Name ="button\_exit" Click ="click\_exit" Content="Выход" HorizontalAlignment="Left" Height="40" Margin="225,350,0,0" VerticalAlignment="Top" Width="125" FontFamily="Times New Roman" FontSize="16" FontStyle="Italic"/>

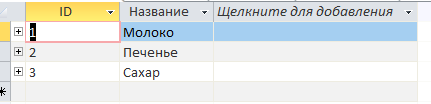
<TextBlock HorizontalAlignment="Center" Height="25" Margin="50,120,474.667,0" TextWrapping="Wrap" Text=" Количество продуктов: " VerticalAlignment="Top" Width="270" FontSize="18" FontFamily="Times New Roman" FontStyle="Italic"/>

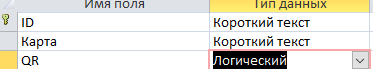
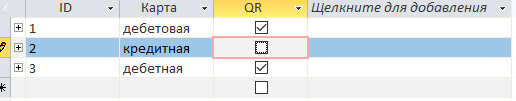
</Grid>

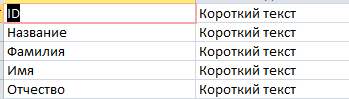
</Window>****

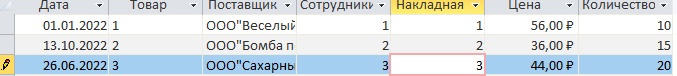
**Задание 4.** Создать базу данных торгового предприятия. Создать таблицы: «Товар», «Группа товаров», «Сотрудники», «Поставщики», «Клиенты», «Продажи», «Телефоны поставщиков». В третьем лице соединить пропорционально цепи логически выяснить цепь соединения базы данных путем изготовления острых линий и попарно соединяющихся прямых

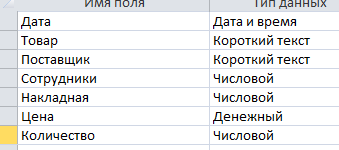
****

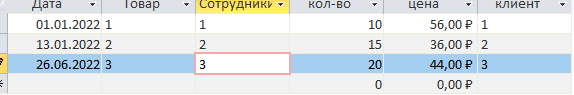


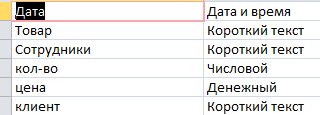
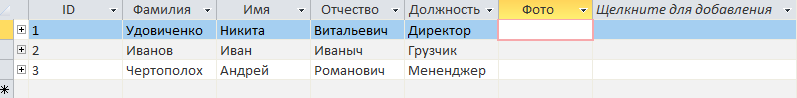


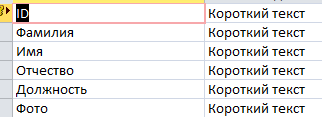


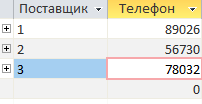


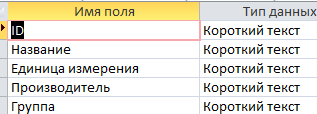




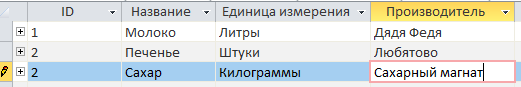












**Задание 1**

**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

{

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

}

private void CE(object sender, RoutedEventArgs e)

{

TextBox.Clear();

}

private void N1\_Click(object sender, RoutedEventArgs e)

{

TextBox.Text += 1;

}

private void N2\_Click(object sender, RoutedEventArgs e)

{

TextBox.Text += 2;

}

private void N3\_Click(object sender, RoutedEventArgs e)

{

TextBox.Text += 3;

}

private void N4\_Click(object sender, RoutedEventArgs e)

{

TextBox.Text += 4;

}

private void N5\_Click(object sender, RoutedEventArgs e)

{

TextBox.Text += 5;

}

private void N6\_Click(object sender, RoutedEventArgs e)

{

TextBox.Text += 6;

}

private void N7\_Click(object sender, RoutedEventArgs e)

{

TextBox.Text += 7;

}

private void N8\_Click(object sender, RoutedEventArgs e)

{

TextBox.Text += 8;

}

private void N9\_Click(object sender, RoutedEventArgs e)

{

TextBox.Text += 9;

}

private void N0\_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="Cornsilk" Height="360" Width="450">

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

<Grid.RowDefinitions>

<RowDefinition Height="420\*"/>

<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" HorizontalAlignment="Left" Margin="340,81,0,0" VerticalAlignment="Top" Width="72" Height="51" Click="CE" />

<Button Name ="N1" Content="1" HorizontalAlignment="Left" Margin="9,200,0,0" VerticalAlignment="Top" Width="71" Height="51" Click="N1\_Click"/>

<Button Name ="N2" Content="2" HorizontalAlignment="Left" Margin="92,200,0,0" VerticalAlignment="Top" Width="71" Height="51" Click="N2\_Click"/>

<Button Name ="N3" Content="3" HorizontalAlignment="Left" Margin="175,200,0,0" VerticalAlignment="Top" Width="71" Height="51" Click="N3\_Click"/>

<Button Name ="N4" Content="4" HorizontalAlignment="Left" Margin="9,140,0,0" VerticalAlignment="Top" Width="71" Height="51" Click="N4\_Click"/>

<Button Name ="N5" Content="5" HorizontalAlignment="Left" Margin="92,140,0,0" VerticalAlignment="Top" Width="71" Height="51" Click="N5\_Click"/>

<Button Name ="N6" Content="6" HorizontalAlignment="Left" Margin="175,140,0,0" VerticalAlignment="Top" Width="71" Height="51" Click="N6\_Click"/>

<Button Name ="N7" Content="7" HorizontalAlignment="Left" Margin="9,81,0,0" VerticalAlignment="Top" Width="71" Height="51" Click="N7\_Click"/>

<Button Name ="N8" Content="8" HorizontalAlignment="Left" Margin="92,81,0,0" VerticalAlignment="Top" Width="71" Height="51" Click="N8\_Click"/>

<Button Name ="N9" Content="9" HorizontalAlignment="Left" Margin="175,81,0,0" VerticalAlignment="Top" Width="71" Height="51" Click="N9\_Click"/>

<Button Name ="N0" Content="0" HorizontalAlignment="Left" Margin="9,259,0,0" VerticalAlignment="Top" Width="154" Height="51" Click="N0\_Click"/>

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

<Button Content="/" HorizontalAlignment="Left" Margin="258,200,0,0" VerticalAlignment="Top" Width="70" Height="51" Click="Button\_Click\_1"/>

<Button Content="\*" HorizontalAlignment="Left" Margin="258,259,0,0" VerticalAlignment="Top" Width="70" Height="51" Click="Button\_Click\_2"/>

<Button Content="-" Margin="0,0,102,0" VerticalAlignment="Center" Height="51" Click="Button\_Click\_3" HorizontalAlignment="Right" Width="70"/>

<Button Content="+" HorizontalAlignment="Left" Margin="258,81,0,0" VerticalAlignment="Top" Width="70" Height="51" Click="Button\_Click\_4"/>

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

<Button Content="С" HorizontalAlignment="Left" Margin="340,140,0,0" VerticalAlignment="Top" Width="72" Height="51" Click="Button\_Click\_6"/>

</Grid>

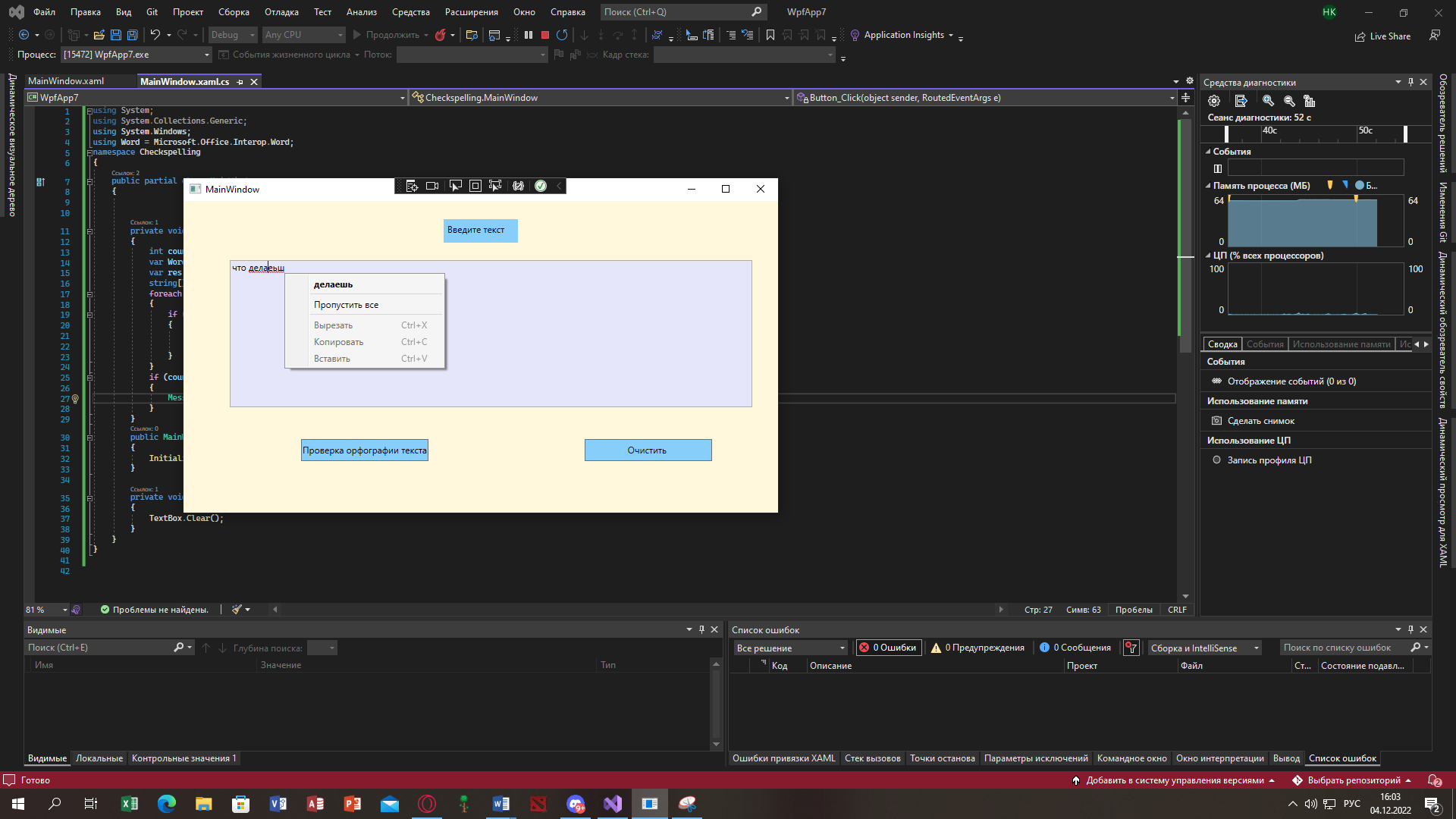
</Window>

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

**Задание 2**

**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**



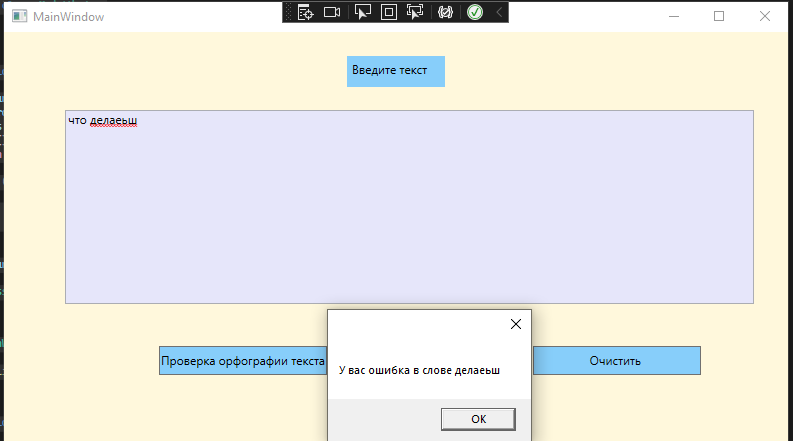


Рисунок 1. Вычисление ветвящийся функции

**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("У вас ошибка в слове " + s);

}

}

if (count == 0)

{

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

}

}

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="Cornsilk">

<Grid>

<Label Content="Введите текст" HorizontalAlignment="Center" Margin="0,24,0,0" Background="LightSkyBlue" VerticalAlignment="Top" Height="31" Width="98"/>

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

<Button Content="Проверка орфографии текста" HorizontalAlignment="Left" Background="LightSkyBlue" Margin="155,314,0,0" VerticalAlignment="Top" Height="29" Click="Button\_Click" />

<Button Content="Очистить " HorizontalAlignment="Left" Margin="529,314,0,0" Background="LightSkyBlue" VerticalAlignment="Top" Height="29" Width="168" Click="Button\_Click\_1"/>

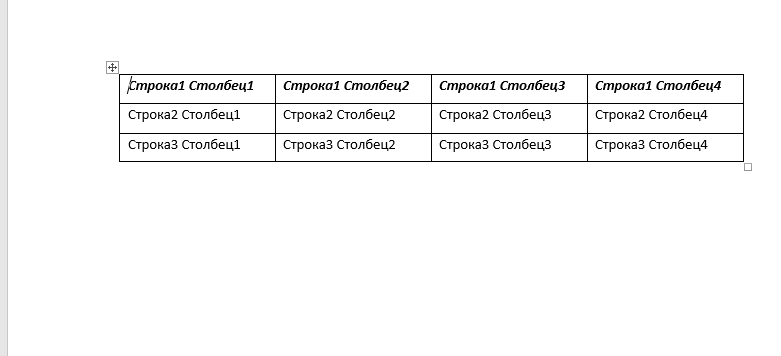
</Grid>

</Window>

Вывод: я разработал программу, для проверки орфографии.

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

**Ниже на рисунке представлен фрагмент работающего проекта**



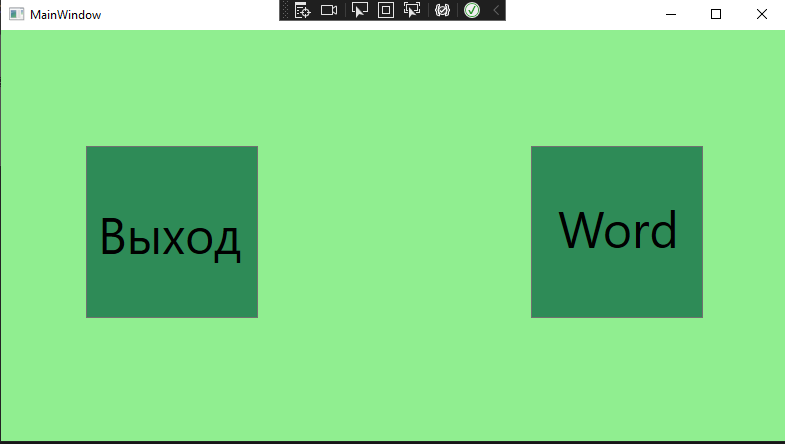


Рисунок 1. Вычисление ветвящийся функции

**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

{

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

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" Background="LightGreen" Height="450" Width="800">

<Grid>

<Button HorizontalAlignment="Left" Background="SeaGreen" Margin="530,116,0,0" VerticalAlignment="Top" RenderTransformOrigin="0.609,0.413" Height="172" Width="172" Click="Vord">

<Label Content="Word" HorizontalAlignment="Center" FontSize ="50" VerticalAlignment="Center" Height="86" Width="128"/>

</Button>

<Button HorizontalAlignment="Left" Background="SeaGreen" Margin="85,116,0,0" VerticalAlignment="Top" Height="172" Width="172" Click ="Exit">

</Button>

<Label Content="Выход" HorizontalAlignment="Left" Margin="92,165,0,0" FontSize ="50" VerticalAlignment="Top" Height="76" Width="154"/>

</Grid>

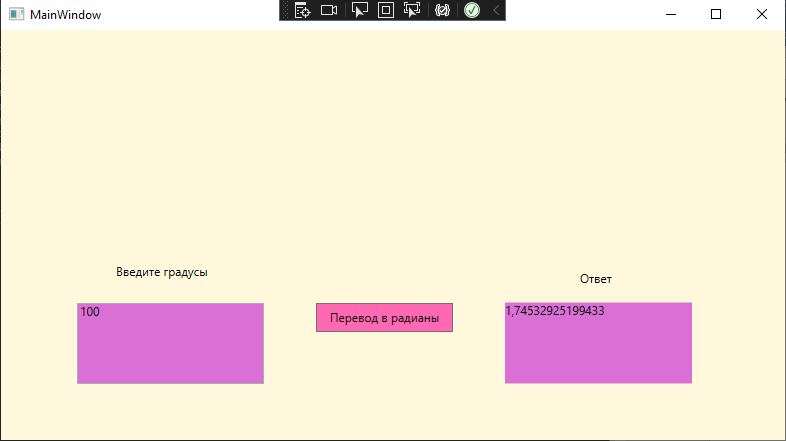
</Window>

Вывод: я разработал программу для создания таблицы в ворде

**Цель работы**: создать 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

{

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

}

private void ToRad(object sender, EventArgs e)

{

// Создание экземпляра класса Excel.Application:

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

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

XL.Quit();

}

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

{

// Создание экземпляра класса Excel.Application:

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

var rad = XL.WorksheetFunction.Radians(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="Cornsilk">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="131\*"/>

<RowDefinition Height="86\*"/>

</Grid.RowDefinitions>

<TextBox Name="TextBox" Background="Orchid" HorizontalAlignment="Left" Margin="76,25,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="187" Height="81" Grid.Row="1"/>

<Label Content="Введите градусы&#xD;&#xA;" HorizontalAlignment="Left" Margin="110,228,0,0" VerticalAlignment="Top" Height="52" Width="119" Grid.RowSpan="2"/>

<Button Content="Перевод в радианы" Background="HotPink" HorizontalAlignment="Left" Margin="315,25,0,0" VerticalAlignment="Top" Click="Button\_Click" Height="29" Width="137" Grid.Row="1"/>

<TextBlock Name="TxtBlck" Background="Orchid" HorizontalAlignment="Left" Margin="504,25,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Height="81" Width="187" Grid.Row="1"/>

<Label Content="Ответ" HorizontalAlignment="Left" Margin="574,235,0,0" VerticalAlignment="Top" Height="39" Width="46" Grid.RowSpan="2"/>

</Grid>

</Window>

Вывод: я разработал программу для создания таблицы в ворде

**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

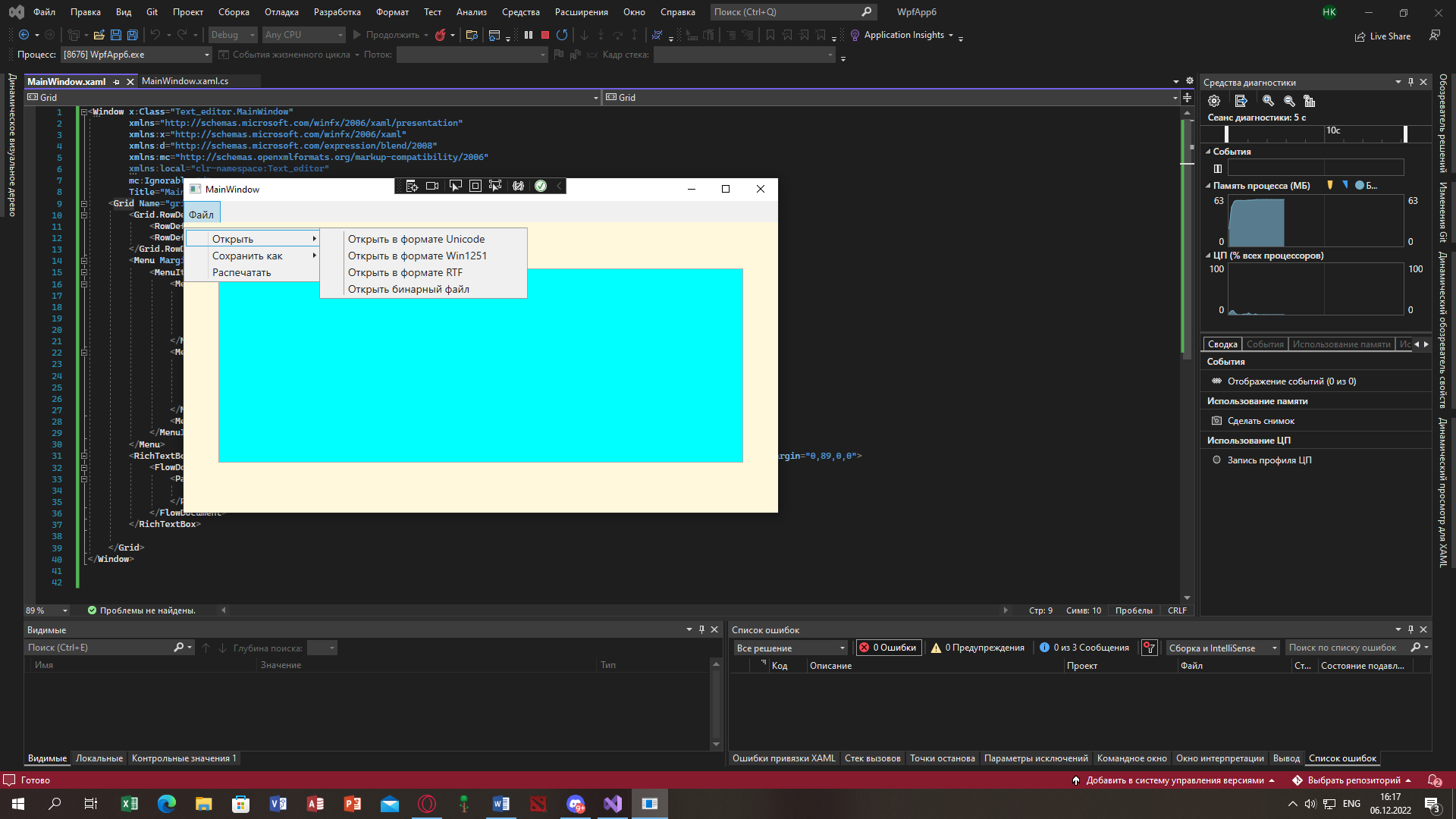
Сохранение информации в файл с кодировкай Win1251

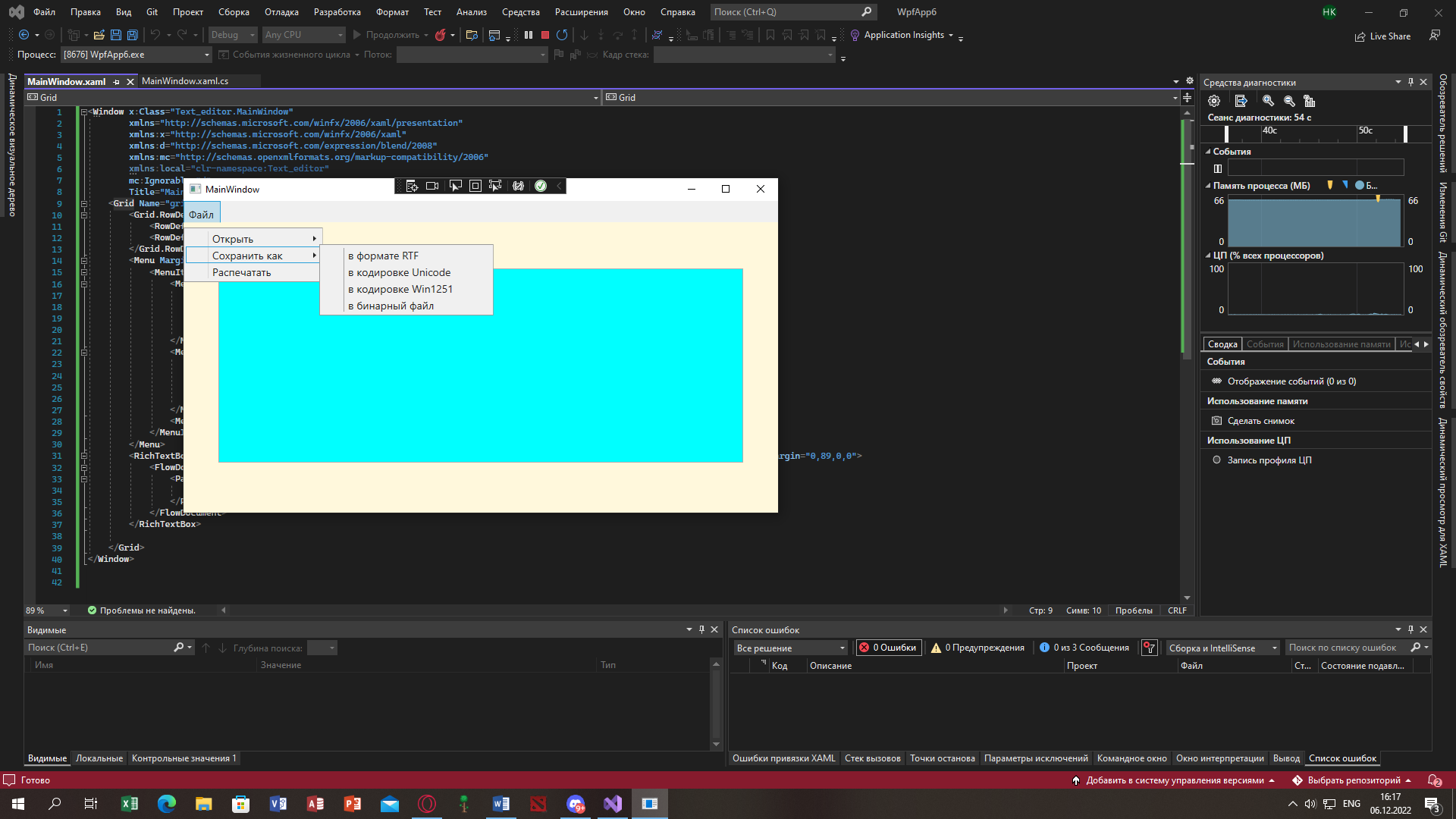
Сохранение информации в файл с разрешением RTF

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="450" Width="800" Background="Cornsilk">

<Grid Name="grid1" Margin="0,0,0,-16">

<Grid.RowDefinitions>

<RowDefinition Height="177\*"/>

<RowDefinition Height="40\*"/>

</Grid.RowDefinitions>

<Menu Margin="0,0,0,320">

<MenuItem Header="Файл" Background="Coral" Height="35" Width="49" FontSize="14">

<MenuItem Header="Открыть">

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

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

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

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

</MenuItem>

<MenuItem Header="Сохранить как">

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

<MenuItem Header="в кодировке Unicode" Click="Save\_as\_Unicode"></MenuItem>

<MenuItem Header="в кодировке Win1251" Click="Save\_as\_Win1251"></MenuItem>

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

</MenuItem>

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

</MenuItem>

</Menu>

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

<FlowDocument>

<Paragraph>

<Run/>

</Paragraph>

</FlowDocument>

</RichTextBox>

</Grid>

</Window>

Вывод: я разработал простенький текстовый редактор на c#.