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

Гостищев Ф.Е.

*(подпись) (И.О. Фамилия)*

Проверил:

Руководитель практики от колледжа:

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

*(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 |

**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',\

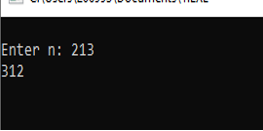
scanf,'scanf',\

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',\

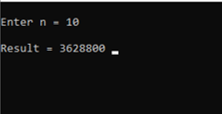
scanf,'scanf',\

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',\

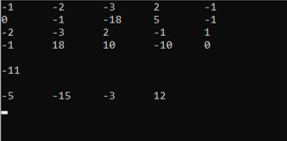
scanf,'scanf',\

exit, 'exit',\

setlocal, 'setlocal',\

system, 'system'

end data



**Практическая работа №5**

**Задание:**

Количество чисел, равных C1

**Решение:**

format PE console

include 'win32a.inc'

start:

macro prnt op1, op2 { ;макрос вывода, где op1 - массив, op2 - количество элементов

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,16 ;вывод массива

lea ebx, [m] ;массив

mov ecx,16 ;количество элементов в регистр ecx для цикла

mov eax,[ebx+4] ;запоминаем первый элемент для сравнения с остальными

proc function ;процедура для вычисления количества элементов

cycl:

cmp eax,[ebx] ; вычитаем первый элемент (eax) из текущего элемента массива

jz met ;если равно нулю, переход на метку met

jmp ex ;переход на следующий элемен массива

met:

inc [count] ;увеличиваем счётчик

ex:

add ebx,4

loop cycl

cinvoke printf,text1,[count] ;вывод

invoke sleep, 9000

endp

call function ;вызов процедуры

cinvoke system,p

invoke exit,0

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

fpp db '%d ', 0

spp db '%d',0

p db 'pause',0

m dd 6,1,10,-3,-5,1,-2,1,4,6,3,6,8,9,11,1

count dd 0

data import

library msvcrt,'MSVCRT.DLL',\

kernel32,'KERNEL32.DLL'

import kernel32,\

sleep,'Sleep'

import msvcrt,\

printf,'printf',\

scanf,'scanf',\

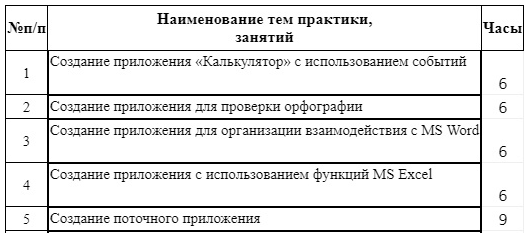
exit,'exit',\

setlocale,'setlocale',\

system,'system'

end data

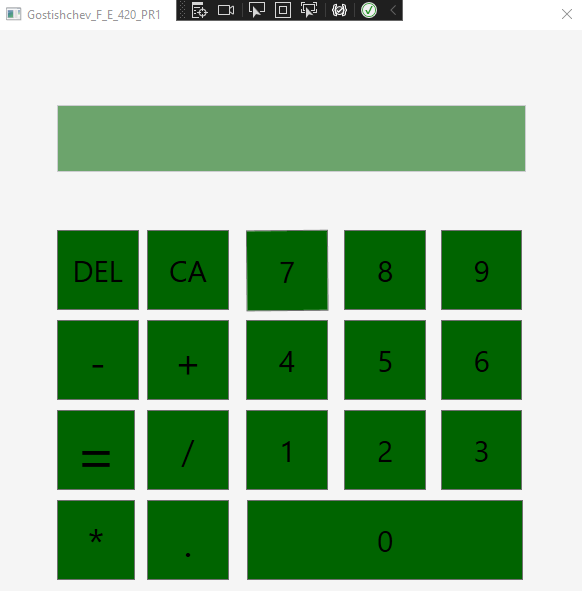


****

**Практическая работа №1 «Создание приложения «Калькулятор» с использованием событий»**

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

1. На рисунке снизу представлено окно WPF c дизайном самого калькулятора.



1. Код программы:

using System;

using System.Collections.Generic;

using System.ComponentModel;

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 CalculatorF\_WPF

{

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

}

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

{

if (screen.Text.Length == 0)

return;

screen.Text += "0";

}

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

{

screen.Text += "1";

}

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

{

screen.Text += "2";

}

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

{

screen.Text += "3";

}

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

{

screen.Text += "4";

}

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

{

screen.Text += "5";

}

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

{

screen.Text += "6";

}

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

{

screen.Text += "7";

}

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

{

screen.Text += "8";

}

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

{

screen.Text += "9";

}

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

{

if (screen.Text.Length != 0 && !new char[5]{'+','-','\*','/','.'}.Contains(screen.Text[screen.Text.Length-1]))

screen.Text += "+";

}

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

{

if (screen.Text.Length > 0 && new char[5] { '+', '-', '\*', '/', '.' }.Contains(screen.Text[screen.Text.Length - 1]))

{

return;

}

screen.Text += "-";

}

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

{

if (screen.Text.Length != 0 && !new char[5] { '+', '-', '\*', '/', '.' }.Contains(screen.Text[screen.Text.Length - 1]))

screen.Text += "\*";

}

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

{

if (screen.Text.Length != 0 && !new char[5] { '+', '-', '\*', '/', '.' }.Contains(screen.Text[screen.Text.Length - 1]))

screen.Text += "/";

}

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

{

if (screen.Text.Length == 0 || !new char[5] { '+', '-', '\*', '/', '.' }.Contains(screen.Text[screen.Text.Length - 1]))

if (screen.Text.Count(x => x == '.') < screen.Text.Count(x => x == '+' || x == '-' || x == '\*' || x == '/' ) + 1)

screen.Text += ".";

}

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

{

if (screen.Text.Trim().Length == 0)

{

return;

}

try

{

screen.Text = Eval(screen.Text).ToString().Replace(',', '.');

}

catch(OverflowException)

{

MessageBox.Show("Overflow", "Error", MessageBoxButton.OK, MessageBoxImage.Error);

}

catch(DivideByZeroException)

{

MessageBox.Show("Division by zero", "Error", MessageBoxButton.OK, MessageBoxImage.Error);

}

catch(Exception Ex)

{

MessageBox.Show("Error in the expression");

}

}

static Double Eval(String expression)

{

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

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

}

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

{

screen.Text = "";

}

protected override void OnClosing(CancelEventArgs e)

{

e.Cancel = MessageBox.Show("Do you really want to get out?", "Confirm the action", MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.No;

}

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

{

if(screen.Text.Length > 0)

screen.Text = screen.Text.Substring(0, screen.Text.Length - 1);

}

}

}

1. XAML

<Window x:Class="CalculatorF\_WPF.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:Calculator\_WPF"

mc:Ignorable="d"

Title="Gostishchev\_F\_E\_420\_PR1" Height="600" Width="600" WindowStartupLocation="CenterScreen" ResizeMode="NoResize" Background="WhiteSmoke">

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="50\*"/>

<ColumnDefinition Width="63\*"/>

<ColumnDefinition Width="67\*"/>

<ColumnDefinition Width="420\*"/>

</Grid.ColumnDefinitions>

<Button Content="7" Margin="73,200,0,0" FontSize="30" Click="Button\_Click\_7" Background="DarkGreen" Cursor="Hand" RenderTransformOrigin="0.5,0.5" Grid.Column="3" Height="81" VerticalAlignment="Top" HorizontalAlignment="Left" Width="82">

<Button.RenderTransform>

<TransformGroup>

<ScaleTransform/>

<SkewTransform AngleX="0.022"/>

<RotateTransform Angle="-0.406"/>

<TranslateTransform/>

</TransformGroup>

</Button.RenderTransform>

</Button>

<Button Content="0" Margin="74,470,0,0" FontSize="30" Click="Button\_Click\_0" Background="DarkGreen" Cursor="Hand" Grid.Column="3" Height="80" VerticalAlignment="Top" HorizontalAlignment="Left" Width="276"/>

<Button Content="1" HorizontalAlignment="Left" Margin="73,380,0,0" VerticalAlignment="Top" Width="82" Height="80" FontSize="30" Click="Button\_Click\_1" Background="DarkGreen" Cursor="Hand" Grid.Column="3"/>

<Button Content="2" HorizontalAlignment="Left" Margin="171,380,0,0" VerticalAlignment="Top" Width="82" Height="80" FontSize="30" Click="Button\_Click\_2" Grid.Column="3" Background="DarkGreen" Cursor="Hand"/>

<Button Content="3" HorizontalAlignment="Left" Margin="268,380,0,0" VerticalAlignment="Top" Width="81" Height="80" FontSize="30" Click="Button\_Click\_3" Grid.Column="3" Background="DarkGreen" Cursor="Hand"/>

<Button Content="4" HorizontalAlignment="Left" Margin="73,290,0,0" VerticalAlignment="Top" Width="82" Height="80" FontSize="30" Click="Button\_Click\_4" Background="DarkGreen" Cursor="Hand" Grid.Column="3"/>

<Button Content="5" Margin="171,290,0,0" VerticalAlignment="Top" Height="80" FontSize="30" Click="Button\_Click\_5" Grid.Column="3" Background="DarkGreen" Cursor="Hand" HorizontalAlignment="Left" Width="82"/>

<Button Content="6" HorizontalAlignment="Left" Margin="268,290,0,0" VerticalAlignment="Top" Width="81" Height="80" FontSize="30" Click="Button\_Click\_6" Grid.Column="3" Background="DarkGreen" Cursor="Hand"/>

<Button Content="8" HorizontalAlignment="Left" Margin="171,200,0,0" VerticalAlignment="Top" Width="82" Height="80" FontSize="30" Click="Button\_Click\_8" Grid.Column="3" Background="DarkGreen" Cursor="Hand"/>

<Button Content="9" Margin="268,200,0,0" VerticalAlignment="Top" Height="80" FontSize="30" Click="Button\_Click\_9" Grid.Column="3" Background="DarkGreen" Cursor="Hand" HorizontalAlignment="Left" Width="81"/>

<TextBox Name="screen" TextAlignment="Center" IsEnabled="False" HorizontalAlignment="Left" Height="67" Margin="10,75,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="469" FontSize="40" Grid.ColumnSpan="3" Background="DarkGreen" Grid.Column="1"/>

<Button HorizontalContentAlignment="Center" Content="+" HorizontalAlignment="Left" Margin="39,290,0,0" VerticalAlignment="Top" Width="82" Height="80" FontSize="40" Click="Button\_Click\_Plus" Grid.Column="2" Background="DarkGreen" Cursor="Hand" Grid.ColumnSpan="2"/>

<Button HorizontalContentAlignment="Center" Content="-" HorizontalAlignment="Left" Margin="10,290,0,0" VerticalAlignment="Top" Width="82" Height="80" FontSize="40" Click="Button\_Click\_Minus" Background="DarkGreen" Cursor="Hand" Grid.ColumnSpan="2" Grid.Column="1"/>

<Button HorizontalContentAlignment="Center" Content="\*" Margin="10,470,38,0" VerticalAlignment="Top" Height="80" FontSize="40" Click="Button\_Click\_Mul" Background="DarkGreen" Cursor="Hand" Grid.ColumnSpan="2" Grid.Column="1"/>

<Button HorizontalContentAlignment="Center" Content="/" HorizontalAlignment="Left" Margin="39,380,0,0" VerticalAlignment="Top" Width="82" Height="80" FontSize="32" Click="Button\_Click\_Div" Background="DarkGreen" Cursor="Hand" Grid.ColumnSpan="2" Grid.Column="2"/>

<Button HorizontalContentAlignment="Center" Content="." HorizontalAlignment="Left" Margin="39,470,0,0" VerticalAlignment="Top" Width="82" Height="80" FontSize="40" Click="Button\_Click\_Float" Grid.Column="2" Background="DarkGreen" Cursor="Hand" Grid.ColumnSpan="2"/>

<Button Content="=" Margin="10,380,38,0" VerticalAlignment="Top" Height="80" FontSize="60" Click="Button\_Click\_Result" Background="DarkGreen" Cursor="Hand" Grid.ColumnSpan="2" Grid.Column="1"/>

<Button HorizontalContentAlignment="Center" Content="CA" HorizontalAlignment="Left" Margin="39,200,0,0" VerticalAlignment="Top" Width="82" Height="80" FontSize="30" Click="Button\_CA\_Click" Background="DarkGreen" Cursor="Hand" Grid.ColumnSpan="2" Grid.Column="2"/>

<Button HorizontalContentAlignment="Center" Content="DEL" HorizontalAlignment="Left" Margin="10,200,0,0" VerticalAlignment="Top" Width="82" Height="80" FontSize="30" Click="Button\_DEL\_Click" Background="DarkGreen" Cursor="Hand" Grid.ColumnSpan="2" Grid.Column="1"/>

</Grid>

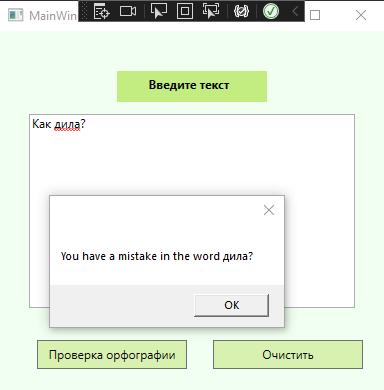
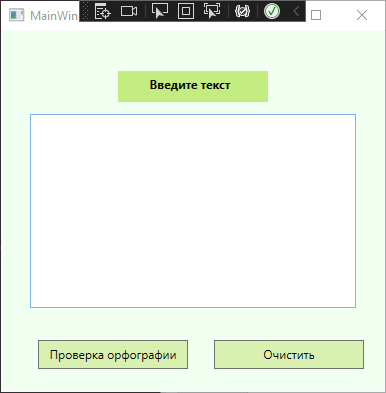
</Window>

***Вывод***: Я разработал калькулятор на C# c помощью одного обработчика для нескольких событий.

**Практическая работа №2 «Создание приложения для проверки орфографии»**

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

1. Фрагменты окна WPF Checkspelling



1. Код программы

using System;

using System.Collections.Generic;

using System.Windows;

using System.Windows.Controls;

using Word = Microsoft.Office.Interop.Word;

namespace CheckSpelling

{

public partial class MainWindow

{

private void Button\_Click\_Check(object sender, RoutedEventArgs A)

{

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("You have a mistake in the word " + S);

}

}

if (count == 0)

{

MessageBox.Show("Your text is completely correct");

}

}

public MainWindow()

{

InitializeComponent();

}

private void Button\_Click\_Cleaner(object sender, RoutedEventArgs A)

{

TextBox.Clear();

}

}

}

1. 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="400" Width="400" Background="Honeydew">

<Grid>

<Label x:Name="Label" Content=" Введите текст" HorizontalAlignment="Center" Margin="0,40,0,0" Background="#FFC3ED80" VerticalAlignment="Top" Height="31" Width="150" FontWeight="Bold" ScrollViewer.CanContentScroll="True"/>

<TextBox SpellCheck.IsEnabled="True" Name ="TextBox" HorizontalAlignment="Center" Margin="0,83,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="326" Height="194"/>

<Button Content="Проверка орфографии" HorizontalAlignment="Left" Background="#FFD8F1B1" Margin="37,309,0,0" VerticalAlignment="Top" Height="29" Click="Button\_Click\_Check" Width="150" />

<Button Content="Очистить" HorizontalAlignment="Left" Margin="213,309,0,0" Background="#FFD8F1B1" VerticalAlignment="Top" Height="29" Width="150" Click="Button\_Click\_Cleaner"/>

</Grid>

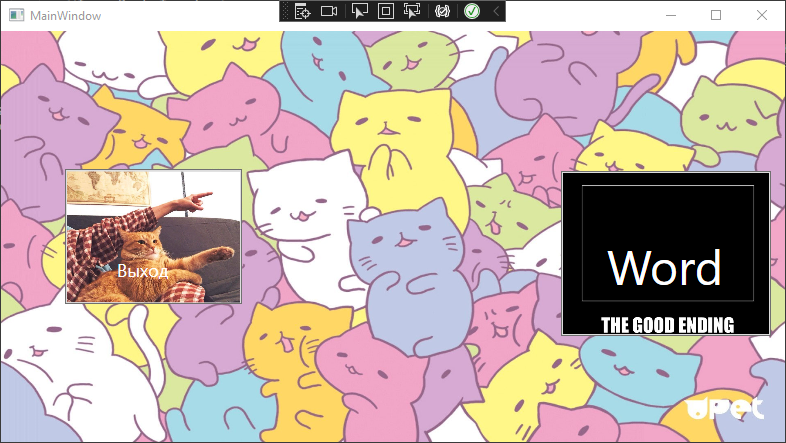
</Window>

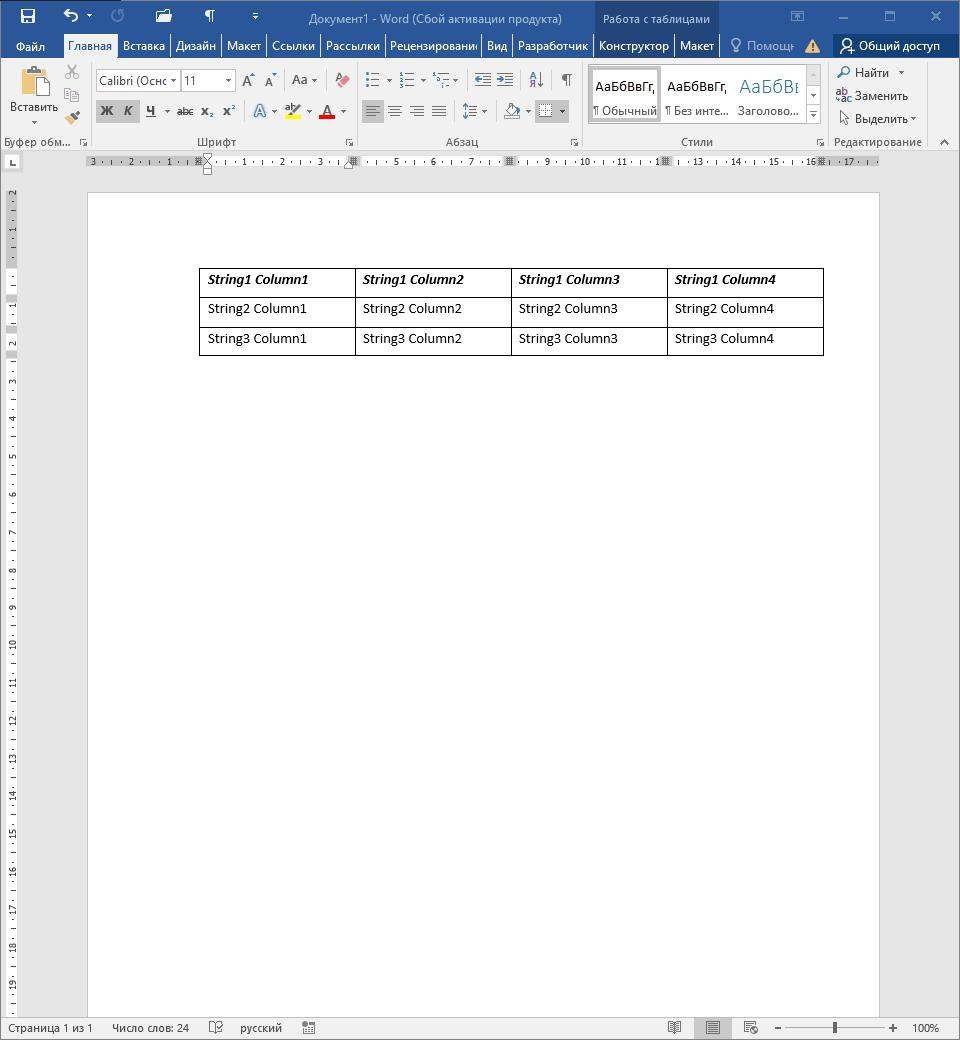
Вывод: я разработал программу, для проверки орфографии.

**Практическая №3 «Создание приложения для организации взаимодействия с MS Word»**

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

1. Фрагменты приложения WPF





1. 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\_spreadsheet

{

/// <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 = "String" + i + " Column" + 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();

}

}

}

1. XAML

<Window x:Class="Word\_spreadsheet.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\_spreadsheet"

mc:Ignorable="d"

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

<Grid>

<Image Source="/background-chat-vk-25.jpg" Stretch="Fill" RenderTransformOrigin="0.505,0.535"/>

<Button HorizontalAlignment="Left" Margin="560,140,0,0" VerticalAlignment="Top" RenderTransformOrigin="0.609,0.413" Height="165" Width="210" Click="Vord">

<Image Source="/Good\_ending.png" Height="171" Width="211"></Image>

</Button>

<Button HorizontalAlignment="Left" Margin="64,0,0,0" VerticalAlignment="Center" Height="135" Width="177" Click ="Exit">

<Image Source="C:\Users\lordf\OneDrive\Изображения\DIckpics\Kotik.png" RenderTransformOrigin="0.5,0.5" Height="131" Width="220">

<Image.RenderTransform>

<TransformGroup>

<ScaleTransform/>

<SkewTransform AngleY="0.182"/>

<RotateTransform/>

<TranslateTransform Y="0.397"/>

</TransformGroup>

</Image.RenderTransform>

</Image>

</Button>

<Viewbox HorizontalAlignment="Left" Margin="550,130,0,0" VerticalAlignment="Top" Height="184" Width="188" Stretch="Fill"/>

<Label Content="Выход" HorizontalAlignment="Left" Margin="111,222,0,0" VerticalAlignment="Top" Height="63" Width="130" Foreground="White" FontSize="18"/>

<Label Content="Word" FontSize ="48" Foreground="White" Background="{x:Null}" Margin="601,197,50,149"/>

</Grid>

</Window>

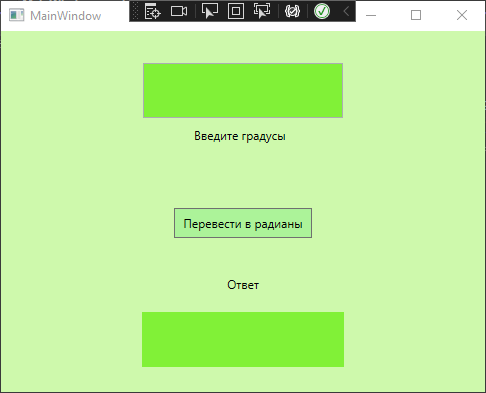
Вывод: я разработал программу для создания таблицы в ворде.

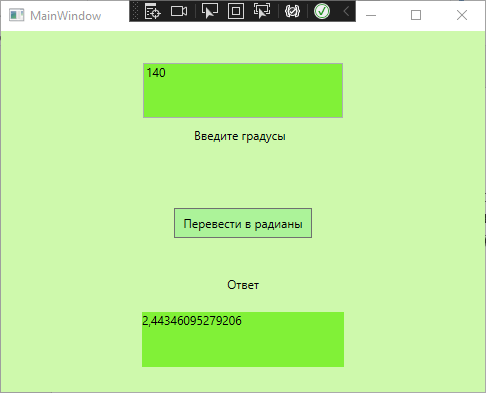
**Практическая работа №4 «Создание приложения с использованием функции MS Excel»**

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

Excel не используя функцию нахождения числа Pi.

1. Фрагменты окна WPF





1. 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 Using\_Excel\_functions

{

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

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.Radians(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.Radians(Convert.ToDouble(TextBox.Text));

TxtBLM.Text = Convert.ToString(rad);

XL.Quit();

}

}

}

1. XAML

<Window x:Class="Using\_Excel\_functions.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:Using\_Excel\_functions"

mc:Ignorable="d"

Title="MainWindow" Height="400" Width="500" Background="LightSkyBlue">

<Grid Background="#FFCEF9AC">

<Grid.RowDefinitions>

<RowDefinition Height="131\*"/>

<RowDefinition Height="86\*"/>

</Grid.RowDefinitions>

<TextBox Name="TextBox" Background="#FF81F137" HorizontalAlignment="Center" Margin="0,32,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="200" Height="55"/>

<Label Content="Введите градусы&#xD;&#xA;" HorizontalAlignment="Center" VerticalAlignment="Center" Height="35" Width="108"/>

<Button Content="Перевести в радианы" HorizontalAlignment="Center" Margin="0,177,0,0" VerticalAlignment="Top" Click="Button\_Click" Height="30" Width="138" Background="#FFACF399"/>

<TextBlock Name="TxtBLM" Background="#FF81F137" TextWrapping="Wrap" VerticalAlignment="Top" Height="55" Width="202" Grid.Row="1" Margin="0,63,0,0" HorizontalAlignment="Center"/>

<Label Content="Ответ" HorizontalAlignment="Center" Margin="0,22,0,0" VerticalAlignment="Top" Grid.Row="1"/>

</Grid>

</Window>

Вывод: я создал собственный проект с использованием функций MS

Excel.

**Практическая работа №5 «Создание поточного приложения»**

**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

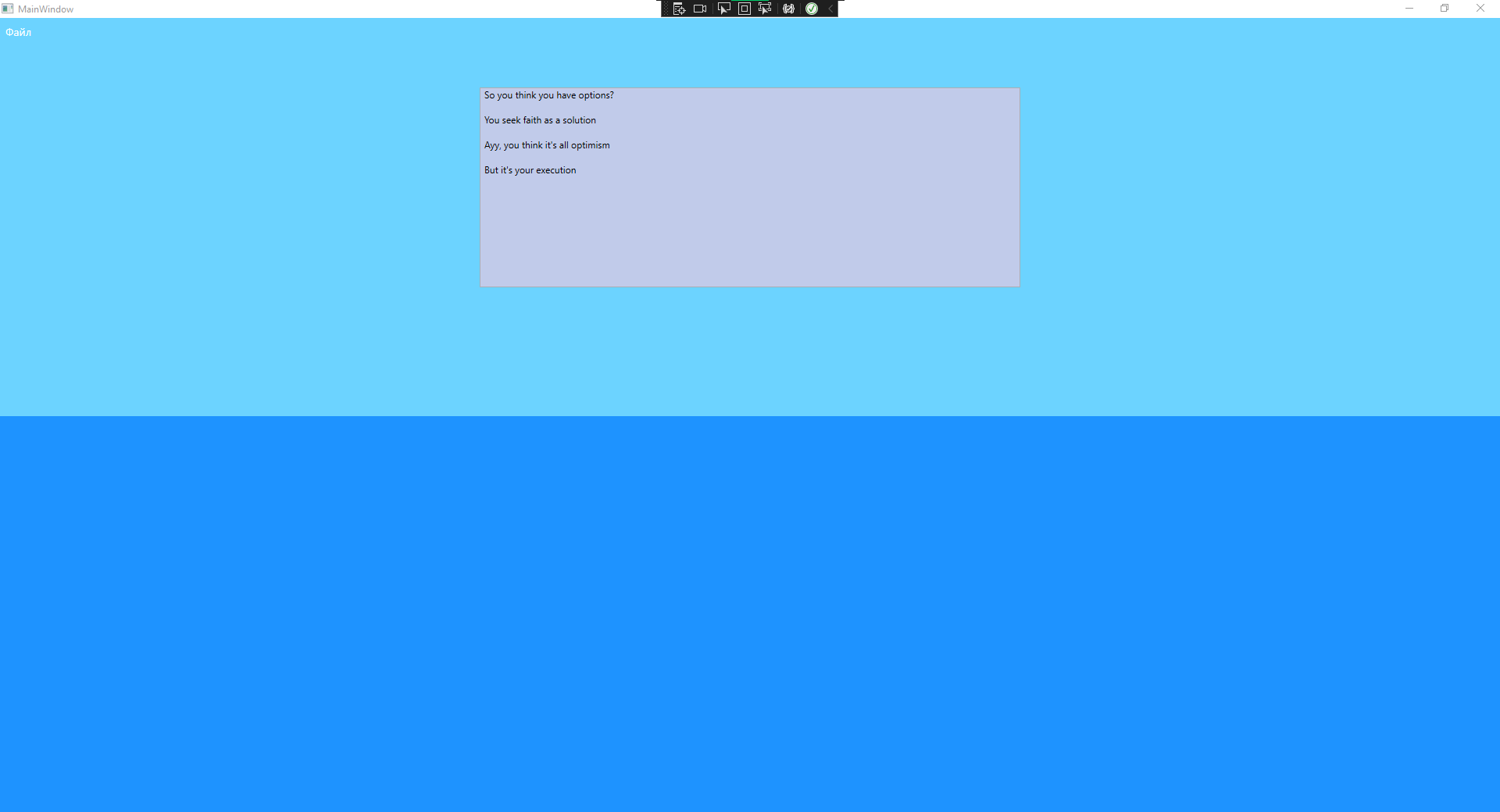
Сохранение информации в файл с кодировкай Win1251

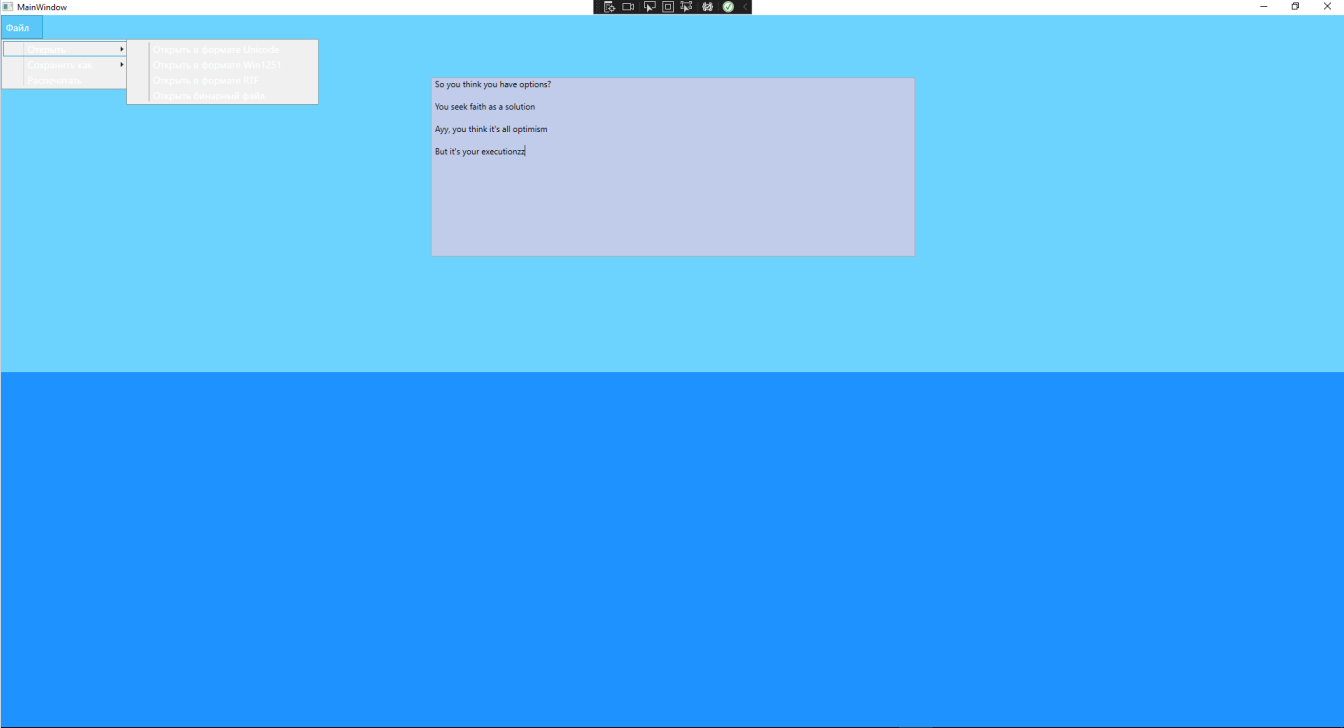
Сохранение информации в файл с разрешением RTF

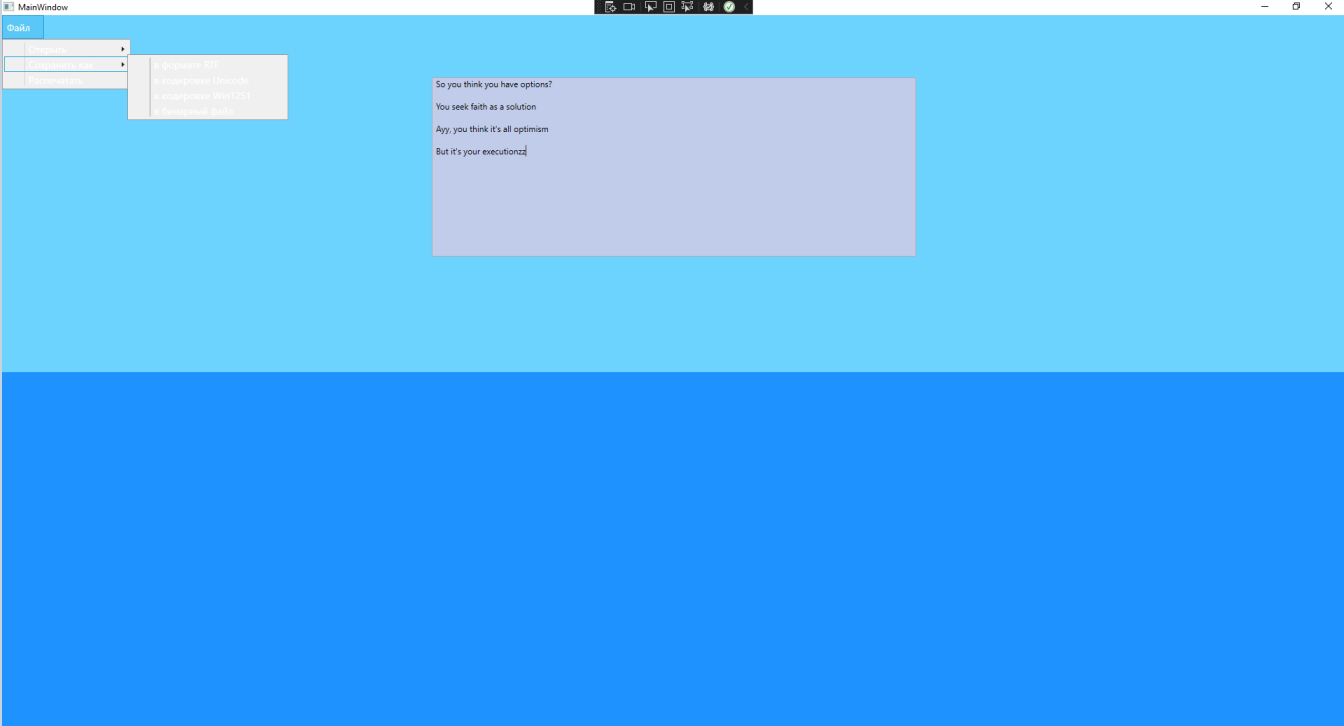
Сохранение информации в бинарный файл

Печать текстового документа.

1. Фрагменты проекта







1. 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");

}

}

}

}

1. 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="LightSkyBlue">

<Grid Name="grid1" Background="#FF1E93FF">

<Grid.RowDefinitions>

<RowDefinition Height="177\*"/>

<RowDefinition Height="40\*"/>

</Grid.RowDefinitions>

<Menu Margin="0,0,0,320" Background="#FF6CD3FF">

<MenuItem Header="Файл" Foreground="White" Height="34" Width="60" 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="#FFC1CBEA" VerticalAlignment="Top" Width="692" Height="256" Margin="0,89,0,0">

<FlowDocument>

<Paragraph>

<Run/>

</Paragraph>

</FlowDocument>

</RichTextBox>

</Grid>

</Window>

Вывод: я разработал примитивный текстовый редактор на c#.