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**

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

*(наименование профессионального модуля)*

Выполнил:

обучающийся учебной группы № 420

А.В. Солуянов

*(И.О. Фамилия)*

**Москва**

**20­­­­22**

Специальность 09.02.03 «Программирование в компьютерных системах»

*(наименование специальности)*

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

*(наименование профессионального модуля)*

Федеральное государственное образовательное бюджетное

учреждение высшего образования

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

**(Финансовый университет)**

Колледж информатики и программирования

**ОТЧЕТ О ВЫПОЛНЕНИИ ЗАДАНИЯ № 7**

**Тема: “События C#”**

Студент: Солуянов А.В.

Группа: 3ПКС-420

Преподаватель: Сибирев И.В.

Дата:05.12.2022

Москва, 2022

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

**Пример работы проекта:**

Изображение выглядит как текст, электроника

Автоматически созданное описание

**Листинг кода:**

**C#:**

using System;

using System.Windows;

using System.Windows.Controls;

namespace WpfApp1

{

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

}

private void CE(object sender, RoutedEventArgs e)

{

TextBox.Clear();

}

private void BT1\_Click(object sender, RoutedEventArgs e)

{

TextBox.Text += 1;

}

private void BT2\_Click(object sender, RoutedEventArgs e)

{

TextBox.Text += 2;

}

private void BT3\_Click(object sender, RoutedEventArgs e)

{

TextBox.Text += 3;

}

private void BT4\_Click(object sender, RoutedEventArgs e)

{

TextBox.Text += 4;

}

private void BT5\_Click(object sender, RoutedEventArgs e)

{

TextBox.Text += 5;

}

private void BT6\_Click(object sender, RoutedEventArgs e)

{

TextBox.Text += 6;

}

private void BT7\_Click(object sender, RoutedEventArgs e)

{

TextBox.Text += 7;

}

private void BT8\_Click(object sender, RoutedEventArgs e)

{

TextBox.Text += 8;

}

private void BT9\_Click(object sender, RoutedEventArgs e)

{

TextBox.Text += 9;

}

private void BT0\_Click(object sender, RoutedEventArgs e)

{

TextBox.Text += 0;

}

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

{

TextBox.Text += "√";

}

private void divide(object sender, RoutedEventArgs e)

{

TextBox.Text += "/";

}

private void multiply(object sender, RoutedEventArgs e)

{

TextBox.Text += "\*";

}

private void substract(object sender, RoutedEventArgs e)

{

TextBox.Text += "-";

}

private void add(object sender, RoutedEventArgs e)

{

TextBox.Text += "+";

}

private void equal(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 point(object sender, RoutedEventArgs e)

{

TextBox.Text += ".";

}

private void clear\_one(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="Black" Height="544.51" Width="454.1">

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

<Grid.RowDefinitions>

<RowDefinition Height="420\*"/>

<RowDefinition Height="11\*"/>

</Grid.RowDefinitions>

<TextBox Name="TextBox" FontSize="65" FontFamily="OCR A Extended" Background="Yellow" HorizontalAlignment="Left" IsEnabled="False" Height="78" Margin="9,24,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="403" RenderTransformOrigin="0.496,0.433"/>

<Button Content="CE" Background="LightSeaGreen" HorizontalAlignment="Left" Margin="340,119,0,0" VerticalAlignment="Top" Width="72" Height="79" Click="CE" />

<Button Name ="BT1" Background="LightSeaGreen" Content="1" HorizontalAlignment="Left" Margin="9,304,0,0" VerticalAlignment="Top" Width="71" Height="80" Click="BT1\_Click"/>

<Button Name ="BT2" Background="LightSeaGreen" Content="2" HorizontalAlignment="Left" Margin="92,304,0,0" VerticalAlignment="Top" Width="71" Height="80" Click="BT2\_Click"/>

<Button Name ="BT3" Background="LightSeaGreen" Content="3" HorizontalAlignment="Left" Margin="175,304,0,0" VerticalAlignment="Top" Width="71" Height="80" Click="BT3\_Click"/>

<Button Name ="BT4" Background="LightSeaGreen" Content="4" HorizontalAlignment="Left" Margin="9,211,0,0" VerticalAlignment="Top" Width="71" Height="79" Click="BT4\_Click"/>

<Button Name ="BT5" Background="LightSeaGreen" Content="5" HorizontalAlignment="Left" Margin="92,211,0,0" VerticalAlignment="Top" Width="71" Height="79" Click="BT5\_Click"/>

<Button Name ="BT6" Background="LightSeaGreen" Content="6" HorizontalAlignment="Left" Margin="175,211,0,0" VerticalAlignment="Top" Width="71" Height="79" Click="BT6\_Click"/>

<Button Name ="BT7" Background="LightSeaGreen" Content="7" HorizontalAlignment="Left" Margin="9,119,0,0" VerticalAlignment="Top" Width="71" Height="79" Click="BT7\_Click"/>

<Button Name ="BT8" Background="LightSeaGreen" Content="8" HorizontalAlignment="Left" Margin="92,119,0,0" VerticalAlignment="Top" Width="71" Height="79" Click="BT8\_Click"/>

<Button Name ="BT9" Background="LightSeaGreen" Content="9" HorizontalAlignment="Left" Margin="175,119,0,0" VerticalAlignment="Top" Width="71" Height="79" Click="BT9\_Click"/>

<Button Name ="BT0" Background="LightSeaGreen" Content="0" HorizontalAlignment="Left" Margin="9,396,0,0" VerticalAlignment="Top" Width="154" Height="79" Click="BT0\_Click"/>

<Button Content="." Background="LightSeaGreen" HorizontalAlignment="Left" Margin="175,396,0,0" VerticalAlignment="Top" Width="71" Height="79" Click="point"/>

<Button Content="÷" Background="LightSeaGreen" HorizontalAlignment="Left" Margin="258,119,0,0" VerticalAlignment="Top" Width="70" Height="79" Click="divide"/>

<Button Content="x" Background="LightSeaGreen" HorizontalAlignment="Left" Margin="258,211,0,0" VerticalAlignment="Top" Width="70" Height="79" Click="multiply"/>

<Button Content="-" Background="LightSeaGreen" HorizontalAlignment="Left" Margin="258,304,0,0" VerticalAlignment="Top" Width="70" Height="80" Click="substract"/>

<Button Content="+" Background="LightSeaGreen" HorizontalAlignment="Left" Margin="258,396,0,0" VerticalAlignment="Top" Width="70" Height="79" Click="add"/>

<Button Content="=" Background="LightSeaGreen" HorizontalAlignment="Left" Margin="340,304,0,0" VerticalAlignment="Top" Width="72" Height="171" Click="equal"/>

<Button Content="С" Background="LightSeaGreen" HorizontalAlignment="Left" Margin="340,211,0,0" VerticalAlignment="Top" Width="72" Height="79" Click="clear\_one"/>

</Grid>

</Window>

**Вывод:** в ходе работы я научился использовать приём создания одного обработчика для нескольких событий, благодаря чему смог написать калькулятор через WPF.

Федеральное государственное образовательное бюджетное

учреждение высшего образования

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

**(Финансовый университет)**

Колледж информатики и программирования

**ОТЧЕТ О ВЫПОЛНЕНИИ ЗАДАНИЯ № 8**

**Тема: “Checkspelling”**

Студент: Солуянов А.В.

Группа: 3ПКС-420

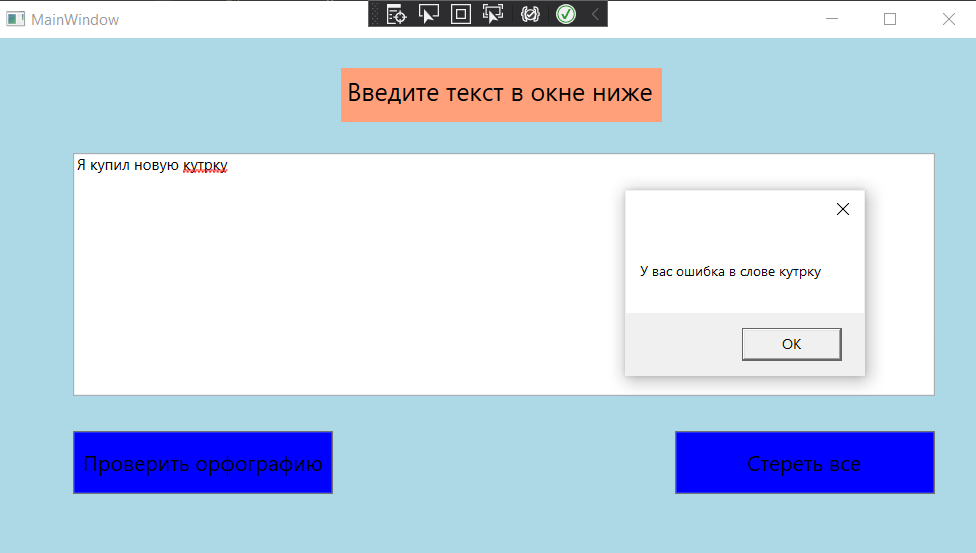
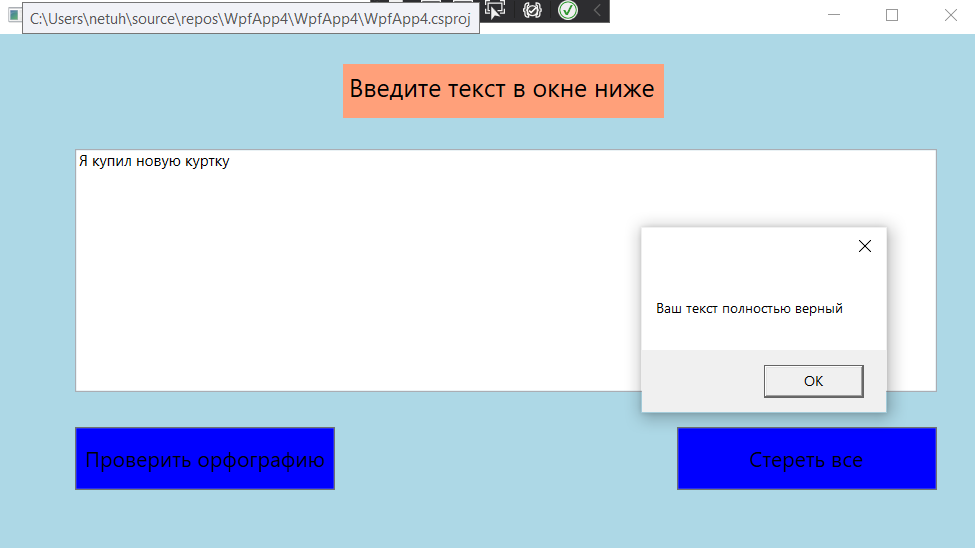
Преподаватель: Сибирев И.В.

Дата:05.12.2022

Москва, 2022

**Цель работы**: Разработать программу, которая предлагает пользователю ввести какие-либо слова, предложения в текстовое поле и после нажатия соответствующей кнопки проверить орфографию введенного текста. Для непосредственной проверки орфографии воспользуемся функцией **Checkspelling** объектной библиотеки MS Word.

**Пример работы проекта:**

**** ****

**Листинг кода**:

**C#:**

using System.Windows;

using Word = Microsoft.Office.Interop.Word;

namespace Checkspelling

{

public partial class MainWindow

{

private void Check(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 Clear(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="LightBlue">

<Grid>

<Label Content="Введите текст в окне ниже" FontSize="20" HorizontalAlignment="Left" Margin="275,24,0,0" Background="LightSalmon" VerticalAlignment="Top" Height="43" Width="257"/>

<TextBox SpellCheck.IsEnabled="True" Name ="TextBox" HorizontalAlignment="Left" Margin="61,92,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="689" Height="194"/>

<Button Content="Проверить орфографию" FontSize="17" HorizontalAlignment="Left" Background="Blue" Margin="61,314,0,0" VerticalAlignment="Top" Height="51" Click="Check" Width="208" />

<Button Content="Стереть все" FontSize="17" HorizontalAlignment="Left" Margin="542,314,0,0" Background="blue" VerticalAlignment="Top" Height="51" Width="208" Click="Clear"/>

</Grid>

</Window>

**Вывод**: в ходе работы я смог разработать программу, которая предлагает пользователю ввести какие-либо слова, предложения в текстовое поле и после нажатия соответствующей кнопки проверить орфографию введенного

Федеральное государственное образовательное бюджетное

учреждение высшего образования

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

**(Финансовый университет)**

Колледж информатики и программирования

**ОТЧЕТ О ВЫПОЛНЕНИИ ЗАДАНИЯ № 9**

**Тема: “Создание таблицы в Word”**

Студент: Солуянов А.В.

Группа: 3ПКС-420

Преподаватель: Сибирев И.В.

Дата:05.12.2022

Москва, 2022

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

**Пример работы проекта:**

Изображение выглядит как текст, кот, внутренний, млекопитающее

Автоматически созданное описаниеИзображение выглядит как стол

Автоматически созданное описание

**Листинг кода:**

**C#:**

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 = "A" + i + " a" + 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\netuh\Downloads\cat.jpg" Stretch="Fill" RenderTransformOrigin="0.505,0.535"/>

<Button HorizontalAlignment="Left" Margin="530,165,0,0" VerticalAlignment="Top" RenderTransformOrigin="0.609,0.413" Height="86" Width="214" Click="Vord">

<Image Source="C:\Users\netuh\Downloads\wrd.jpg" Height="86" Width="200"></Image>

</Button>

<Button HorizontalAlignment="Left" Margin="85,165,0,0" VerticalAlignment="Top" Height="86" Width="172" Click ="Exit">

<Image Source="C:\Users\netuh\Downloads\exit1.jpg" Height="87" Width="176" RenderTransformOrigin="0.5,0.5">

<Image.RenderTransform>

<TransformGroup>

<ScaleTransform/>

<SkewTransform AngleY="0.182"/>

<RotateTransform/>

<TranslateTransform Y="0.397"/>

</TransformGroup>

</Image.RenderTransform>

</Image>

</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="94,165,0,0" FontSize ="50" VerticalAlignment="Top" Height="76" Width="154"/>

</Grid>

</Window>

Федеральное государственное образовательное бюджетное

учреждение высшего образования

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

**(Финансовый университет)**

Колледж информатики и программирования

**ОТЧЕТ О ВЫПОЛНЕНИИ ЗАДАНИЯ № 10**

**Тема: “Использование функций Excel”**

Студент: Солуянов А.В.

Группа: 3ПКС-420

Преподаватель: Сибирев И.В.

Дата:05.12.2022

Москва, 2022

**Цель работы:** В программе буквально в трех строчках программного кода демонстрируем обращение к одной простой функции MS Excel, а именно получению значения числа **Pi** = 3,14.

Число **Pi** представлено в классе **Math** языка C#. Но цель данной про­граммы - показать легкость доступа к функциям MS Excel. Если у нас получится обращение к этой элементарной функции, то мы открываем для воз­можности управления большой библиотекой объектов MS Excel.

**Пример работы проекта:**

Изображение выглядит как текст

Автоматически созданное описание

**Листинг проекта:**

**C#:**

using System;

using System.Windows;

namespace ExcelForm

{

public partial class MainWindow

{

public MainWindow()

{

InitializeComponent();

}

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

{

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

var a = XL.WorksheetFunction.Acos(Convert.ToDouble(TextBox.Text));

TxtBlck.Text = Convert.ToString(a);

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="LightGray">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="131\*"/>

<RowDefinition Height="86\*"/>

</Grid.RowDefinitions>

<TextBox Name="TextBox" Background="LightGoldenrodYellow" HorizontalAlignment="Left" Margin="115,207,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="170" Height="55"/>

<Label Content="Вычисление арккосинуса числа через функцию excel&#xD;&#xA;" FontSize="20" HorizontalAlignment="Left" Margin="146,87,0,0" VerticalAlignment="Top" Height="45" Width="505"/>

<Label Content="Введите нужное число&#xD;&#xA;(-1&lt;=x&lt;=1)" FontSize="15" HorizontalAlignment="Left" Margin="115,151,0,0" VerticalAlignment="Top" Height="51" Width="168"/>

<Button Content="Вычислить арккосинус" FontSize="15" Background="LightBlue" HorizontalAlignment="Left" Margin="314,207,0,0" VerticalAlignment="Top" Click="Button\_Click" Height="37" Width="172"/>

<TextBlock Name="TxtBlck" Background="LightGoldenrodYellow" HorizontalAlignment="Left" Margin="503,207,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Height="55" Width="170"/>

<Label Content="Ответ" FontSize="15" HorizontalAlignment="Left" Margin="569,161,0,0" VerticalAlignment="Top" Width="49"/>

</Grid>

</Window>

**Вывод:** в ходе работы я научился использовать функции Excel при написании проекта WPF.

Федеральное государственное образовательное бюджетное

учреждение высшего образования

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

**(Финансовый университет)**

Колледж информатики и программирования

**ОТЧЕТ О ВЫПОЛНЕНИИ ЗАДАНИЯ № 11**

**Тема: “Текстовый редактор”**

Студент: Солуянов А.В.

Группа: 3ПКС-420

Преподаватель: Сибирев И.В.

Дата:05.12.2022

Москва, 2022

**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

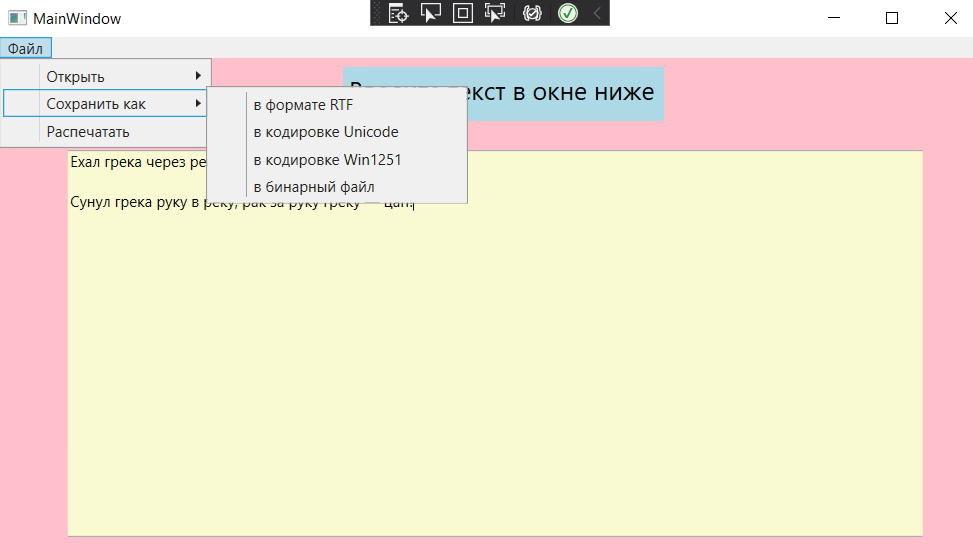
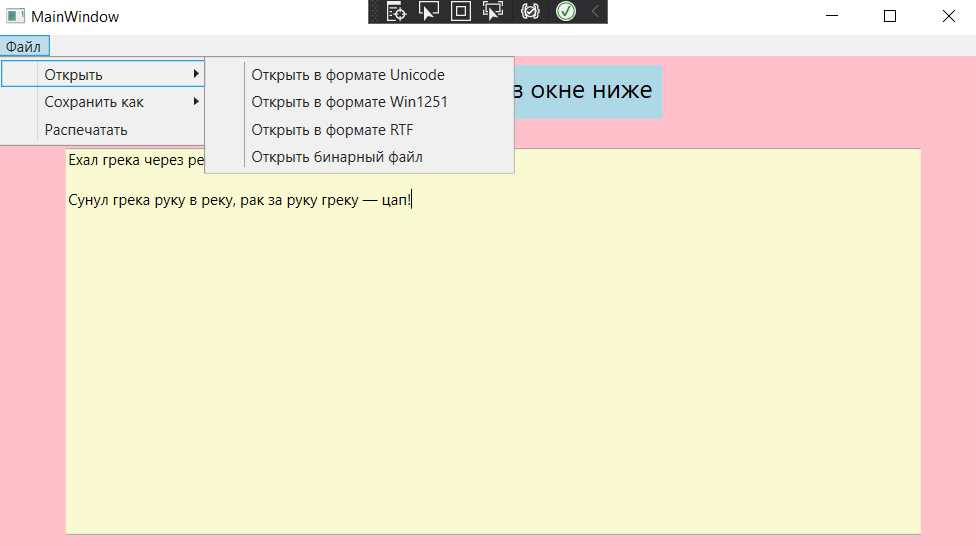
Сохранение информации в файл с кодировкай Win1251

Сохранение информации в файл с разрешением RTF

Сохранение информации в бинарный файл

Печать текстового документа.

**Пример работы проекта:**

****

**Листинг программы:**

**C#:**

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 = "Файл";

odlg.FileName = "Файл";

}

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("Документ был сохранен");

}

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("Документ был сохранен");

}

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("Документ был сохранен");

}

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("Документ был сохранён на рабочем столе");

}

}

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("Документ был 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("Документ был 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("Документ был 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("Документ был открыт");

}

private void PrintWindow(object sender, RoutedEventArgs e)

{

PrintDialog p = new PrintDialog();

if (p.ShowDialog() == true)

{

p.PrintVisual(grid1, "Производится 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="Pink">

<Grid Name="grid1">

<Grid.RowDefinitions>

<RowDefinition Height="177\*"/>

<RowDefinition Height="40\*"/>

</Grid.RowDefinitions>

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

<MenuItem Header="Файл">

<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>

<Label Content="Введите текст в окне ниже" FontSize="20" HorizontalAlignment="Left" Margin="275,24,0,0" Background="LightBlue" VerticalAlignment="Top" Height="43" Width="257"/>

<RichTextBox Name="rtxtbx" HorizontalAlignment="Center" Background="LightGoldenrodYellow" VerticalAlignment="Top" Width="692" Height="310" Margin="55,90,46.6,0" Grid.RowSpan="2">

<FlowDocument>

<Paragraph>

<Run/>

</Paragraph>

</FlowDocument>

</RichTextBox>

</Grid>

</Window>

**Вывод:** в ходе работы я смог написать программу на языке C#, которая реализует работу с файлами в различных кодировках.