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

Добагов Д.З.

*(И.О. Фамилия)*

Проверил:

руководитель практики от колледжа:

Сибирев И.В.

*(И.О. Фамилия)*

Альшакова Е.Л.

*(И.О. Фамилия)*

**Москва**

**20­­­­22**

**Перечень заданий/работ, выполненных в ходе учебной практики**

Специальность 09.02.03 «Программирование в компьютерных системах»

*(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 | Тема 1. Программирование разветвляющегося процесса | Разработать программу сравнения двух чисел соблюдая следующие этапы:  - создать алгоритм;  - написать программный код в соответствии с разработанными алгоритмами и проектом;  - выполнить отладку;  - разработать техническую документацию. |
| 2 | Тема 2. Программирование разветвляющихся алгоритмов с помощью команд передачи управления языка Assembler | Реализовать алгоритм ветвления на языке Assembler соблюдая следующие этапы:  - создать алгоритм;  - написать программный код в соответствии с разработанными алгоритмами и проектом;  - выполнить отладку;  - разработать техническую документацию. |
| 3 | Тема 3. Работа с флагами регистра флагов с использованием команд языка Assembler | Разработать программу с использованием регистра FLAGS соблюдая следующие этапы:  - создать алгоритм;  - написать программный код в соответствии с разработанными алгоритмами и проектом;  - выполнить отладку;  - разработать техническую документацию. |
| 4 | Тема 4. Программирование алгоритмов с использованием арифметических команд и флага переноса языка Assembler | Разработать программу решения квадратного уравнения соблюдая следующие этапы:  - создать алгоритм;  - написать программный код в соответствии с разработанными алгоритмами и проектом;  - выполнить отладку;  - разработать техническую документацию. |
| 5 | Тема 5. Выполнение операций сдвига и логических операций над числами с помощью команд языка Assembler | Разработать программу с логических преобразований чисел соблюдая следующие этапы:  - создать алгоритм;  - написать программный код в соответствии с разработанными алгоритмами и проектом;  - выполнить отладку;  - разработать техническую документацию. |
| 6 | Тема 6. Умножение знаковых чисел | Разработать программу вычисления произведений произвольных чисел соблюдая следующие этапы:  - создать алгоритм;  - написать программный код в соответствии с разработанными алгоритмами и проектом;  - выполнить отладку;  - разработать техническую документацию. |
| 7 | Тема 7. Работа с флагами CF, OF. Деление знаковых чисел | Разработать программу вычисления деления произвольных чисел соблюдая следующие этапы:  - создать алгоритм;  - написать программный код в соответствии с разработанными алгоритмами и проектом;  - выполнить отладку;  - разработать техническую документацию. |
| 8 | Тема 8. Циклы | Разработать программу вычисления значений функций разложением в ряд Тейлора соблюдая следующие этапы:  - создать алгоритм;  - написать программный код в соответствии с разработанными алгоритмами и проектом;  - выполнить отладку;  - разработать техническую документацию. |
| 9 | Тема 9. Работа с сегментами. Подпрограммы | Разработать программу с использованием подпрограмм соблюдая следующие этапы:  - создать алгоритм;  - написать программный код в соответствии с разработанными алгоритмами и проектом;  - выполнить отладку;  - разработать техническую документацию. |
| 10 | Тема 10. Обработка одномерных массивов с использованием команд языка Assembler | Разработать программу обработки одномерного массива соблюдая следующие этапы:  - создать алгоритм;  - написать программный код в соответствии с разработанными алгоритмами и проектом;  - выполнить отладку;  - разработать техническую документацию. |
| 11 | Тема 11. Работа со стеком | Разработать программу с использованием стека соблюдая следующие этапы:  - создать алгоритм;  - написать программный код в соответствии с разработанными алгоритмами и проектом;  - выполнить отладку;  - разработать техническую документацию. |
| 12 | Тема 12. Работа с файлами | Разработать программу работы с файлом соблюдая следующие этапы:  - создать алгоритм;  - написать программный код в соответствии с разработанными алгоритмами и проектом;  - выполнить отладку;  - разработать техническую документацию. |
| 13 | Тема 13. Ассемблерная вставка в С++ | Разработать программу на языке соблюдая следующие этапы:  - определить спецификации;  - создать алгоритм;  - написать программный код в соответствии с разработанными алгоритмами и проектом;  - выполнить отладку;  - разработать техническую документацию. |
| 14 | Тема 14. Разработка программы с использованием Windows Forms | Разработать программу с использованием Windows Forms соблюдая следующие этапы:  - определить спецификации;  - создать алгоритм;  - написать программный код в соответствии с разработанными алгоритмами и проектом;  - выполнить отладку;  - разработать техническую документацию. |
| 15 | Тема 15. Написание программы для произведения стереометрических расчетов | Разработать программу для произведения стереометрических расчетов соблюдая следующие этапы:  - определить спецификации;  - создать алгоритм;  - написать программный код в соответствии с разработанными алгоритмами и проектом;  - выполнить отладку;  - разработать техническую документацию. |
| 16 | Тема 16. Написание консольного приложения для вывода числового массива | Разработать консольное приложение для вывода числового массива соблюдая следующие этапы:  - определить спецификации;  - создать алгоритм;  - написать программный код в соответствии с разработанными алгоритмами и проектом;  - выполнить отладку;  - разработать техническую документацию. |
| 17 | Тема 17. Написание приложения для работы с символьным массивом | Разработать приложение для работы с символьным массивом соблюдая следующие этапы:  - определить спецификации;  - создать алгоритм;  - написать программный код в соответствии с разработанными алгоритмами и проектом;  - выполнить отладку;  - разработать техническую документацию. |
| 18 | Тема 18. Создание приложения с использованием компонента ListBox | Разработать приложение с использованием компонента ListBox соблюдая следующие этапы:  - определить спецификации;  - создать алгоритм;  - написать программный код в соответствии с разработанными алгоритмами и проектом;  - выполнить отладку;  - разработать техническую документацию. |
| 19 | Тема 19. Создание приложения с использованием компонента DataGridView и многострочным textBox | Разработать приложение с использованием компонента DataGridView и многострочным textBox соблюдая следующие этапы:  - определить спецификации;  - создать алгоритм;  - написать программный код в соответствии с разработанными алгоритмами и проектом;  - выполнить отладку;  - разработать техническую документацию. |
| 20 | Тема 20. Создание приложения с для произведения планиметрических расчетов с использованием классов | Разработать приложение с для произведения планиметрических расчетов с использованием классов соблюдая следующие этапы: - определить спецификации;  - создать алгоритм; - при необходимости провести проектирование; - написать программный код в соответствии с разработанными алгоритмами и проектом; - выполнить отладку; - разработать техническую документацию. |
| 21 | Тема 21. Рефакторинг и оптимизация ранее разработанного приложения для обработки числового массива с использованием классов | Произвести разработку приложения для обработки числового массива с использованием классов соблюдая следующие этапы: - определить спецификации;  - создать алгоритм; - при необходимости провести проектирование; - написать программный код в соответствии с разработанными алгоритмами и проектом; - выполнить отладку; - разработать техническую документацию. |
| 22 | Тема 22. Создание приложения с использованием свойства и методов класса String | Разработать приложение с использованием свойств и методов класса String соблюдая следующие этапы: - определить спецификации;  - создать алгоритм; - при необходимости провести проектирование; - написать программный код в соответствии с разработанными алгоритмами и проектом; - выполнить отладку; - разработать техническую документацию. |
| 23 | Тема 23. Создание приложения с использованием делегатов (часть 1) | Разработать приложение с использованием делегатов соблюдая следующие этапы: - определить спецификации;  - создать алгоритм. |
| 24 | Тема 24. Создание приложения с использованием делегатов (часть 2) | Разработать приложение с использованием делегатов соблюдая следующие этапы:  - при необходимости провести проектирование; - написать программный код в соответствии с разработанными алгоритмами и проектом; - выполнить отладку; - разработать техническую документацию. |
| 25 | Тема 25. Создание приложения с использованием событий | Разработать приложение с использованием событий соблюдая следующие этапы: - определить спецификации;  - создать алгоритм; - при необходимости провести проектирование; - написать программный код в соответствии с разработанными алгоритмами и проектом; - выполнить отладку; - разработать техническую документацию. |
| 26 | Тема 26. Создание приложения «Калькулятор» с использованием классов | Разработать приложение «Калькулятор» с использованием событий соблюдая следующие этапы: - определить спецификации;  - создать алгоритм; - при необходимости провести проектирование; - написать программный код в соответствии с разработанными алгоритмами и проектом; - выполнить отладку; - разработать техническую документацию. |
| 27 | Тема 27. Создание приложения для проверки орфографии | Разработать приложение для проверки орфографии соблюдая следующие этапы: - определить спецификации;  - создать алгоритм; - при необходимости провести проектирование; - написать программный код в соответствии с разработанными алгоритмами и проектом; - выполнить отладку; - разработать техническую документацию. |
| 28 | Тема 28. Создание приложения для организации взаимодействия с MS Word | Разработать приложение для организации взаимодействия с MS Word соблюдая следующие этапы: - определить спецификации;  - создать алгоритм; - при необходимости провести проектирование; - написать программный код в соответствии с разработанными алгоритмами и проектом; - выполнить отладку; - разработать техническую документацию. |
| 29 | Тема 29. Создание приложения с использованием функций MS Excel | Разработать приложение с использованием функций MS Excel соблюдая следующие этапы:  - определить спецификации;  - создать алгоритм; - при необходимости провести проектирование; - написать программный код в соответствии с разработанными алгоритмами и проектом; - выполнить отладку; - разработать техническую документацию. |
| 30 | Тема 30. Создание поточного приложения | Разработать поточное приложение соблюдая следующие этапы: - определить спецификации;  - создать алгоритм; - при необходимости провести проектирование; - написать программный код в соответствии с разработанными алгоритмами и проектом; - выполнить отладку; - разработать техническую документацию. |
| 31 | Тема 31. Защита отчета по практике | Защита сформированного отчета по практике. |

**СОДЕРЖАНИЕ**

**ПР26. Создание приложения «Калькулятор» с использованием классов.**

**ПР27. Создание приложения для проверки орфографии.**

**ПР28. Создание приложения для организации взаимодействия с MS Word.**

**ПР29. Создание приложения с использованием функций MS Excel.**

**ПР30. Создание поточного приложения.**

Федеральное государственное образовательное бюджетное

учреждение высшего образования

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

**(Финансовый университет)**

Колледж информатики и программирования

**ОТЧЕТ О ВЫПОЛНЕНИИ ЗАДАНИЯ № 26**

**Тема: “События C#”**

Студент: Добагов Д.З.

Группа: 3ПКС-420

Преподаватель: Сибирев И.В.

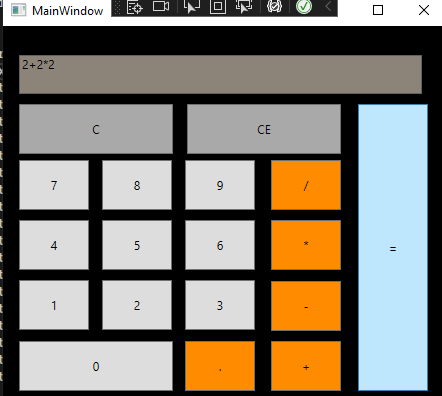
Дата:29.11.2022

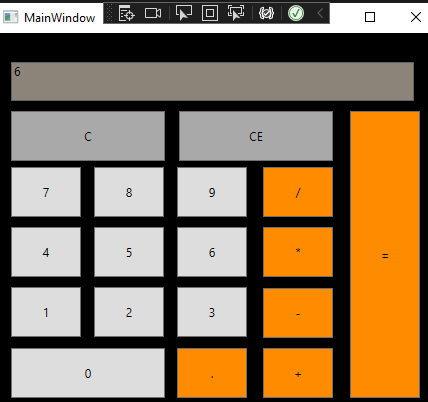
Москва, 2022

**Цель Работы:** разработать калькулятор на языке C#.

**Решение:**

Пример вывода консоли:





Код:

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="Black" Height="420" Width="460">

<Grid Margin="10,20,10,-6">

<Grid.RowDefinitions>

<RowDefinition Height="420\*"/>

<RowDefinition Height="11\*"/>

</Grid.RowDefinitions>

<TextBox Name="TextBox" Background="AntiqueWhite" HorizontalAlignment="Left" IsEnabled="False" Height="39" Margin="06,9,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="403" RenderTransformOrigin="0.496,0.433"/>

<Button Content="C" Background="DarkGray" HorizontalAlignment="Left" Margin="6,58,0,0" VerticalAlignment="Top" Width="154" Height="50" Click="CE" />

<Button Name ="N1" Content="1" HorizontalAlignment="Left" Margin="6,234,0,0" VerticalAlignment="Top" Width="70" Height="50" Click="N1\_Click"/>

<Button Name ="N2" Content="2" HorizontalAlignment="Left" Margin="89,234,0,0" VerticalAlignment="Top" Width="70" Height="50" Click="N2\_Click"/>

<Button Name ="N3" Content="3" HorizontalAlignment="Left" Margin="172,234,0,0" VerticalAlignment="Top" Width="70" Height="50" Click="N3\_Click"/>

<Button Name ="N4" Content="4" HorizontalAlignment="Left" Margin="6,174,0,0" VerticalAlignment="Top" Width="70" Height="50" Click="N4\_Click"/>

<Button Name ="N5" Content="5" HorizontalAlignment="Left" Margin="89,174,0,0" VerticalAlignment="Top" Width="70" Height="50" Click="N5\_Click"/>

<Button Name ="N6" Content="6" HorizontalAlignment="Left" Margin="172,174,0,0" VerticalAlignment="Top" Width="70" Height="50" Click="N6\_Click"/>

<Button Name ="N7" Content="7" HorizontalAlignment="Left" Margin="6,114,0,0" VerticalAlignment="Top" Width="70" Height="50" Click="N7\_Click"/>

<Button Name ="N8" Content="8" HorizontalAlignment="Left" Margin="89,114,0,0" VerticalAlignment="Top" Width="70" Height="50" Click="N8\_Click"/>

<Button Name ="N9" Content="9" HorizontalAlignment="Left" Margin="172,114,0,0" VerticalAlignment="Top" Width="70" Height="50" Click="N9\_Click"/>

<Button Name ="N0" Content="0" HorizontalAlignment="Left" Margin="6,295,0,0" VerticalAlignment="Top" Width="154" Height="50" Click="N0\_Click"/>

<Button Content="." Background="DarkOrange" HorizontalAlignment="Left" Margin="172,295,0,0" VerticalAlignment="Top" Width="70" Height="50" Click="Button\_Click\_5"/>

<Button Content="/" Background="DarkOrange" HorizontalAlignment="Left" Margin="258,114,0,0" VerticalAlignment="Top" Width="70" Height="50" Click="Button\_Click\_1"/>

<Button Content="\*" Background="DarkOrange" HorizontalAlignment="Left" Margin="258,174,0,0" VerticalAlignment="Top" Width="70" Height="50" Click="Button\_Click\_2"/>

<Button Content="-" Background="DarkOrange" HorizontalAlignment="Left" Margin="258,235,0,0" VerticalAlignment="Top" Width="70" Height="50" Click="Button\_Click\_3"/>

<Button Content="+" Background="DarkOrange" HorizontalAlignment="Left" Margin="258,295,0,0" VerticalAlignment="Top" Width="70" Height="50" Click="Button\_Click\_4"/>

<Button Content="=" Background="DarkOrange" HorizontalAlignment="Left" Margin="345,58,0,0" VerticalAlignment="Top" Width="70" Height="287" Click="ravno"/>

<Button Content="CE" Background="DarkGray" HorizontalAlignment="Left" Margin="174,58,0,0" VerticalAlignment="Top" Width="154" Height="50" Click="Button\_Click\_6"/>

</Grid>

</Window>

**Вывод:** Я разработал калькулятор на языке С#.

Федеральное государственное образовательное бюджетное

учреждение высшего образования

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

**(Финансовый университет)**

Колледж информатики и программирования

**ОТЧЕТ О ВЫПОЛНЕНИИ ЗАДАНИЯ № 27**

**Тема: “Checkspelling”**

Студент: Добагов Д.З.

Группа: 3ПКС-420

Преподаватель: Сибирев И.В.

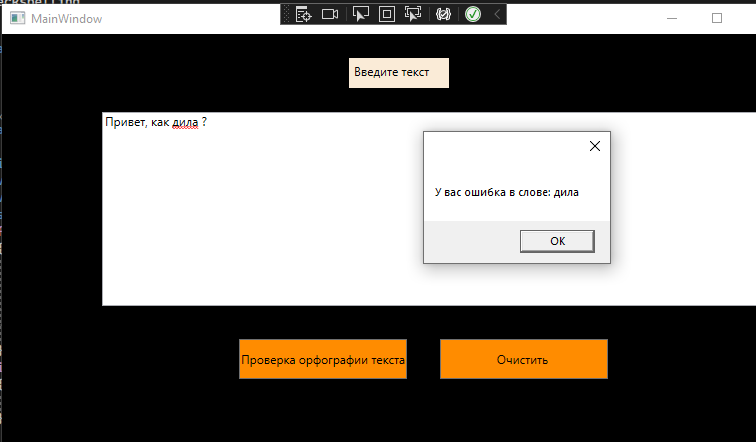
Дата:30.11.2022

Москва, 2022

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

**Решение:**

Пример вывода консоли:



Код:

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(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="Black">

<Grid>

<Label Content="Введите текст" HorizontalAlignment="Center" Margin="010,24,0,0" Background="AntiqueWhite" VerticalAlignment="Top" Height="30" Width="100"/>

<TextBox SpellCheck.IsEnabled="True" Name ="TextBox" HorizontalAlignment="Left" Margin="100,78,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="670" Height="194"/>

<Button Content="Проверка орфографии текста" HorizontalAlignment="Left" Background="DarkOrange" Margin="237,305,0,0" VerticalAlignment="Top" Height="40" Click="Button\_Click" />

<Button Content="Очистить " HorizontalAlignment="Left" Margin="438,305,0,0" Background="DarkOrange" VerticalAlignment="Top" Height="40" Width="168" Click="Button\_Click\_1"/>

</Grid>

</Window>

**Вывод:** Я разработал программу на языке С#, которая выполняет проверку орфографии в тексте.

Федеральное государственное образовательное бюджетное

учреждение высшего образования

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

**(Финансовый университет)**

Колледж информатики и программирования

**ОТЧЕТ О ВЫПОЛНЕНИИ ЗАДАНИЯ № 28**

**Тема: “Создание таблицы в Word”**

Студент: Добагов Д.З.

Группа: 3ПКС-420

Преподаватель: Сибирев И.В.

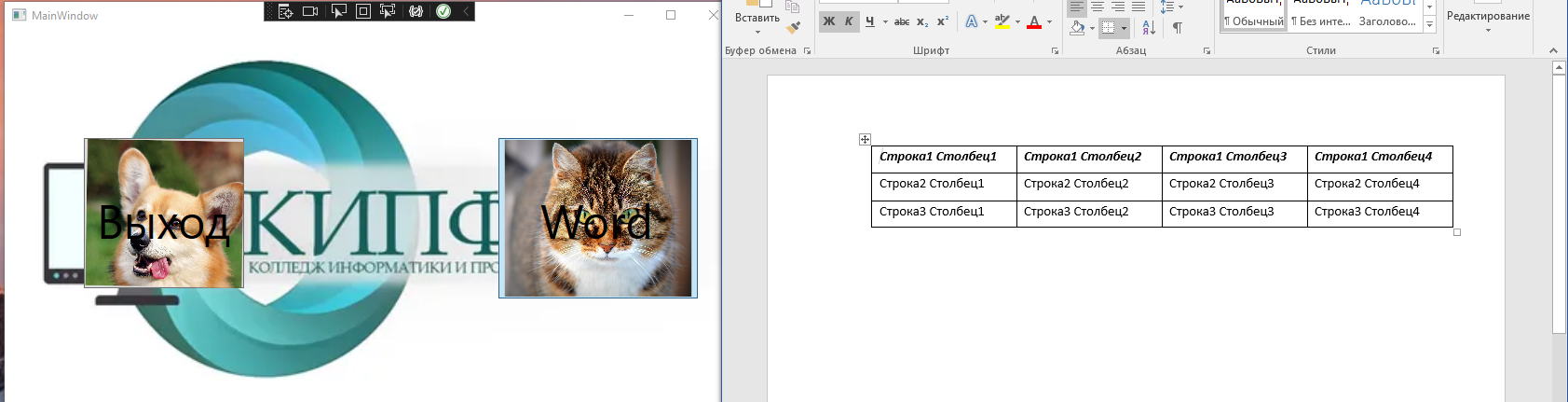
Дата:2.12.2022

Москва, 2022

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

**Решение:**

Пример вывода консоли:



Код:

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

<Grid>

<Image Stretch="Fill" RenderTransformOrigin="0.505,0.535" Source="/i.webp.png"/>

<Button HorizontalAlignment="Left" Margin="530,116,0,0" VerticalAlignment="Top" RenderTransformOrigin="0.609,0.413" Height="172" Width="214" Click="Vord">

<Image Source="/402712.jpg"/>

</Button>

<Button HorizontalAlignment="Left" Margin="85,116,0,0" VerticalAlignment="Top" Height="161" Width="172" Click ="Exit">

<Image Height="233" Width="430" RenderTransformOrigin="0.5,0.5" Source="/porodistye-sobaki-krasivye-kartinki-19.jpg">

<Image.RenderTransform>

<TransformGroup>

<ScaleTransform/>

<SkewTransform AngleY="0.182"/>

<RotateTransform Angle="0.927"/>

<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>

**Вывод:** Я разработал программу, которая при нажатии на кнопку создаёт таблицу в MS Word.

Федеральное государственное образовательное бюджетное

учреждение высшего образования

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

**(Финансовый университет)**

Колледж информатики и программирования

**ОТЧЕТ О ВЫПОЛНЕНИИ ЗАДАНИЯ № 29**

**Тема: “Использование функций MS Excel”**

Студент: Добагов Д.З.

Группа: 3ПКС-420

Преподаватель: Сибирев И.В.

Дата:04.12.2022

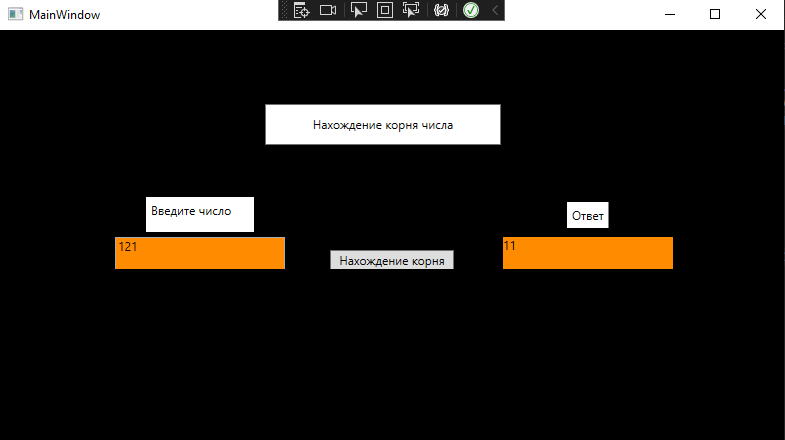
Москва, 2022

**Цель Работы:** создайте собственные проекты с использованием функций MS

Excel не используя функцию нахождения числа Pi.

**Решение:**

Пример вывода консоли:



Код:

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

TxtBlck.Text = Convert.ToString(rad);

XL.Quit();

}

}

}

Xaml:

<Window x:Class="ExcelForm.MainWindow"

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

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

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

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

xmlns:local="clr-namespace:ExcelForm"

mc:Ignorable="d"

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

<Grid>

<Grid.RowDefinitions>

<RowDefinition/>

<RowDefinition Height="172.016"/>

</Grid.RowDefinitions>

<TextBox Name="TextBox" Background="DarkOrange" HorizontalAlignment="Left"

Margin="115,207,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="170"

Height="55"/>

<Button Content="Нахождение корня числа" Background="White" HorizontalAlignment="Left"

Margin="265,74,0,0" VerticalAlignment="Top" Height="41" Width="236"/>

<Label Content="Введите число&#xD;&#xA;" Background="White" HorizontalAlignment="Left"

Margin="146,167,0,0" VerticalAlignment="Top" Height="35" Width="108"/>

<Button Content="Нахождение корня" HorizontalAlignment="Center"

Margin="0,220,0,0" VerticalAlignment="Top" Click="Button\_Click" Height="20"

Width="124"/>

<TextBlock Name="TxtBlck" Background="DarkOrange" HorizontalAlignment="Left"

Margin="503,207,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Height="55"

Width="170"/>

<Label Content="Ответ" Background="White" HorizontalAlignment="Left" Margin="567,172,0,0"

VerticalAlignment="Top"/>

</Grid>

</Window>

**Вывод:** Я разработал программу на языке С#, которая с использованием функций MS Excel находит корень числа.

Федеральное государственное образовательное бюджетное

учреждение высшего образования

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

**(Финансовый университет)**

Колледж информатики и программирования

**ОТЧЕТ О ВЫПОЛНЕНИИ ЗАДАНИЯ № 30**

**Тема: “Текстовый редактор”**

Студент: Добагов Д.З.

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

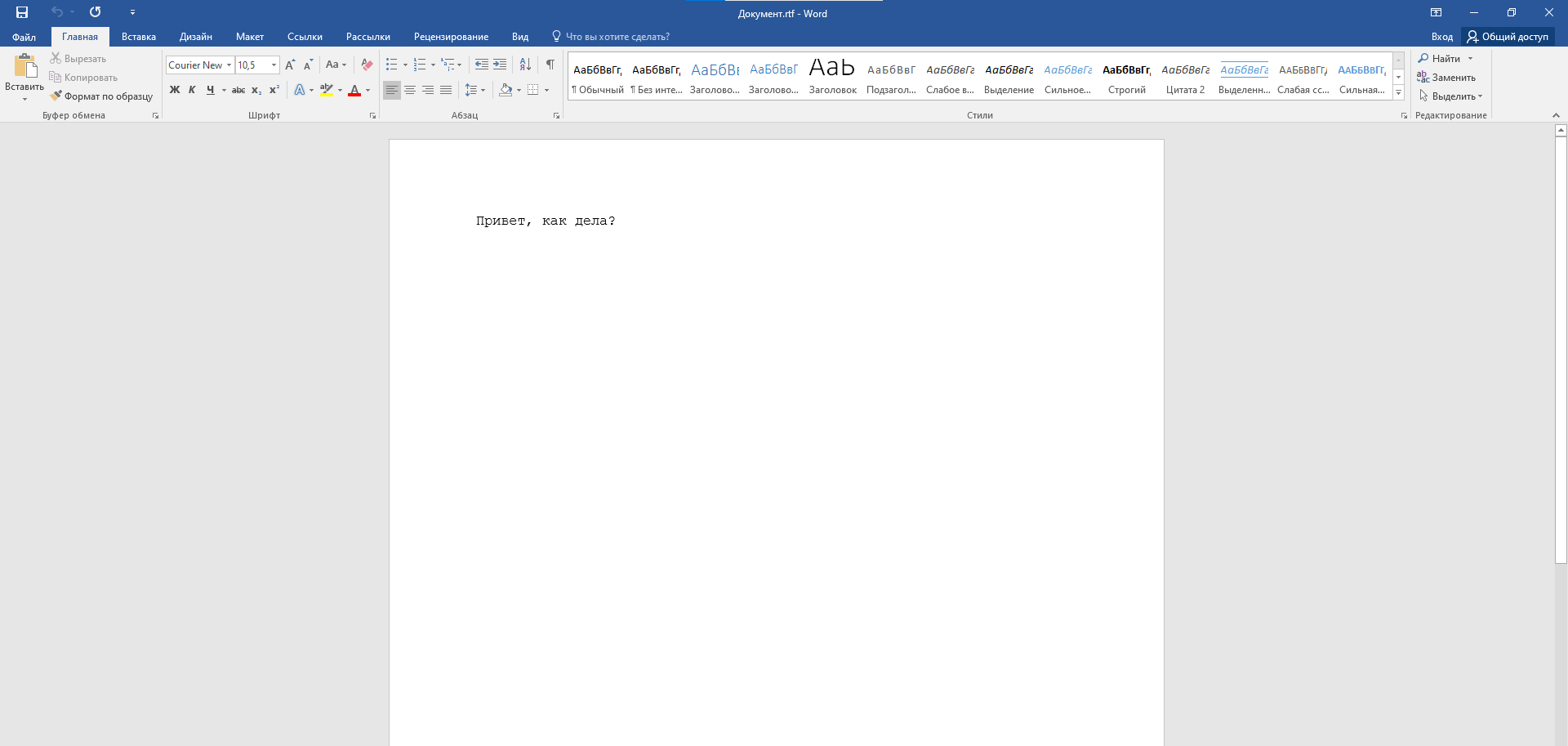
Сохранение информации в бинарный файл

Печать текстового документа.

**Решение:**

Пример вывода консоли:





Код:

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";

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";

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";

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";

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!");

}

}

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

{

rtxtbx.Document.Blocks.Clear();

odlg.DefaultExt = ".rtf";

odlg.Filter = "Текстовый документ";

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="Black">

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

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

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

<MenuItem Header="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="DarkOrange" VerticalAlignment="Top" Width="692" Height="249" Margin="0,89,0,0">

<FlowDocument>

<Paragraph>

<Run/>

</Paragraph>

</FlowDocument>

</RichTextBox>

</Grid>

</Window>

**Вывод:** Я написал программу на языке C#, которая реализует

работу с файлами в различных кодировках.