**ЛАБОРАТОРНА РОБОТА № 1**

**Інтерфейс віконного додатку**

**Мета:** дослідження основних підходів та технологій реалізації інтерфейсу універсалього додатку Windows.

**Хід роботи**

**1. Завдання на лабораторну роботу**

Засобами інтегрованого середовища розробки Visual Studio побудувати

текстовий редактор. Потрібно реалізувати наступні функції:

1. Створення, відкриття та збереження файлів

2. Копіювання, переміщення тексту.

3. Виділення тексту та знищення тексту.

4. Відміну останньої дії.

5. Стрічку, меню та меню швидкого доступу для управління

функціональними компонентами додатку.

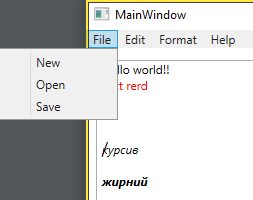
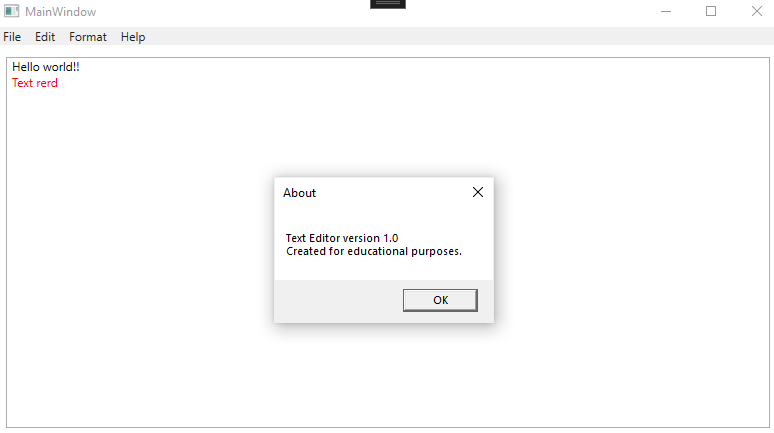
6. Персоналізувати параметри проекту

(назву додатку, оригінальну піктограму, вікно AboutBox тощо).

7. Виконати індивідуальне завдання за варіантом. (14.Колір тексту)

8. Додатково виконати завдання на самостійну роботу.( 3. Реалізувати кнопку відміни форматування для виділеного фрагменту тексту)

Репозиторій: https://github.com/ipz223-zdo/Human-machine-interface



**Висновок:** Я дослідив основні підходи та технології реалізації інтерфейсу універсалього додатку Windows.

Лістинг програми:

**MainWindow.xaml**

<Window x:Class="Task.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:Task"

mc:Ignorable="d"

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

<Grid>

<Menu VerticalAlignment="Top" IsMainMenu="True">

<MenuItem Header="File">

<MenuItem Header="New" Click="NewFile\_Click"/>

<MenuItem Header="Open" Click="OpenFile\_Click"/>

<MenuItem Header="Save" Click="SaveFile\_Click"/>

</MenuItem>

<MenuItem Header="Edit">

<MenuItem Header="Undo" Click="Undo\_Click"/>

<MenuItem Header="Copy" Click="Copy\_Click"/>

<MenuItem Header="Paste" Click="Paste\_Click"/>

<MenuItem Header="Cut" Click="Cut\_Click"/>

</MenuItem>

<MenuItem Header="Format">

<MenuItem Header="Text Color" Click="TextColor\_Click"/>

<MenuItem Header="Clear Formatting" Click="ClearFormatting\_Click"/>

</MenuItem>

<MenuItem Header="Help">

<MenuItem Header="About" Click="About\_Click"/>

</MenuItem>

</Menu>

<RichTextBox Name="textBox" Margin="10,30,10,10" />

</Grid>

</Window>

**MainWindow.xaml.cs**

namespace Task

{

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

}

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

{

textBox.Document.Blocks.Clear();

}

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

{

OpenFileDialog openFileDialog = new OpenFileDialog();

if (openFileDialog.ShowDialog() == true)

{

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

using (FileStream fileStream = new FileStream(openFileDialog.FileName, FileMode.Open))

{

range.Load(fileStream, DataFormats.Rtf);

}

}

}

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

{

SaveFileDialog saveFileDialog = new SaveFileDialog();

if (saveFileDialog.ShowDialog() == true)

{

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

using (FileStream fileStream = new FileStream(saveFileDialog.FileName, FileMode.Create))

{

range.Save(fileStream, DataFormats.Rtf);

}

}

}

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

{

textBox.Copy();

}

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

{

textBox.Paste();

}

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

{

textBox.Cut();

}

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

{

if (textBox.CanUndo)

textBox.Undo();

}

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

{

var selectedText = textBox.Selection;

if (!selectedText.IsEmpty)

{

selectedText.ApplyPropertyValue(TextElement.ForegroundProperty, Brushes.Red);

}

}

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

{

var selectedText = textBox.Selection;

if (!selectedText.IsEmpty)

{

selectedText.ApplyPropertyValue(TextElement.ForegroundProperty, Brushes.Black);

selectedText.ApplyPropertyValue(TextElement.FontWeightProperty, FontWeights.Normal);

}

}

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

{

MessageBox.Show("Text Editor version 1.0\nCreated for educational purposes.", "About");

}

}

}