**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

****

**МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**Факультет информационных технологий**

**Кафедра Информатики и информационных технологий**

**направление подготовки 09.03.02 «Информационные системы и технологии»,**

**профиль «Цифровая трансформация»**

**ЛАБОРАТОРНАЯ РАБОТА №4**

**Дисциплина:** Технологии прикладного программирования

**Выполнил:** студент группы 231-337

Сильченко Александр Алексеевич

**Дата, подпись** 16/02/2024  ***\_\_\_\_\_\_\_\_\_\_\_***

(Дата) (Подпись)

**Проверила:** Полубояринова А.С.\_\_\_\_\_\_\_\_\_\_\_

**(Оценка)**

**Дата, подпись** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  ***\_\_\_\_\_\_\_\_\_\_\_***

(Дата) (Подпись)

**Замечания:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Москва**

**2024**

Текст задачи

Разработать программу "Заметки".

Для реализации данной программы необходимо:

1. Создать многострочное окно ввода используя многострочный TextBox (свойства TextWrapping="Wrap" AcceptsReturn="True");

2. При закрытие программы заметки должны автоматически сохранятся по указаному адресу (File.WriteAllText(@"Source", text1.Text), работа с File идет в пространстве имен System.IO;

3. Перед сохранением проводить проверку наличия пустых строк и их удаления;

4. При запуске программы в TextBox автоматически загружаются ранее сохраненные заметки (textBox.Text = File.ReadAllText(@"Source");)

5. Настроить изменение цвета панели (используя ComboBox со списком цветов);

6. Настроить изменение размера и семейства шрифта (используя ComboBox).

Код вёрстки окон и код программной логики

<Window x:Class="ЛР4.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:ЛР4"

mc:Ignorable="d"

Title="Заметки" Height="350" Width="500" Closing="Form1\_Closing" Initialized="Opening">

<Grid>

<StackPanel Orientation="Horizontal">

<ComboBox x:Name="Color" SelectionChanged="color\_settings" SelectedValuePath="color\_path" Text="Цвет"

Height="20" VerticalAlignment="Top" Width="100" IsEditable="True" IsReadOnly="True">

</ComboBox>

<ComboBox x:Name="Font" Height="20" VerticalAlignment="Top" Width="100" IsEditable="True" Text="Шрифт"

SelectionChanged="Font\_settings">

</ComboBox>

<ComboBox x:Name="Font\_Size" SelectionChanged="size\_settings" Height="20" IsEditable="True"

VerticalAlignment="Top" Width="100" Text="Размер">

</ComboBox>

</StackPanel>

<TextBox x:Name="Text" FontFamily="Times New Roman" TextWrapping="Wrap" AcceptsReturn="True" Grid.Row="0" Grid.Column="0" Text="" Margin="0,20,0,0"/>

</Grid>

</Window>

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 System.Xml.Linq;

using System.Data.SqlTypes;

namespace ЛР4

{

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

Color.ItemsSource = new Color\_Item[]

{

new Color\_Item {color = "Чёрный"},

new Color\_Item {color = "Синий"},

new Color\_Item {color = "Красный"},

new Color\_Item {color = "Желтый"},

new Color\_Item {color = "Зеленый"}

};

Font.ItemsSource = new Font\_Item[]

{

new Font\_Item {font = "Times New Roman"},

new Font\_Item {font = "Segoe UI"},

new Font\_Item {font = "Arial"},

new Font\_Item {font = "Bahnschrift SemiCondensed"},

new Font\_Item {font = "Bodoni MT Black"}

};

Font\_Size.ItemsSource = new Size\_Item[]

{

new Size\_Item {size = "8"},

new Size\_Item {size = "10"},

new Size\_Item {size = "12"},

new Size\_Item {size = "14"},

new Size\_Item {size = "16"},

new Size\_Item {size = "18"},

new Size\_Item {size = "20"},

new Size\_Item {size = "22"},

new Size\_Item {size = "24"},

new Size\_Item {size = "26"},

new Size\_Item {size = "28"},

new Size\_Item {size = "30"}

};

}

private void Form1\_Closing(object sender, System.ComponentModel.CancelEventArgs e)

{

string replace = Text.Text;

while (replace.Contains("\r\n\r\n")){

replace = replace.Replace("\r\n\r\n", "\r\n");

Text.Text = replace;

}

if (replace.StartsWith("\r\n") == true)

{

replace = replace.TrimStart('\r', '\n');

Text.Text = replace;

}

if (replace.EndsWith("\r\n") == true)

{

replace = replace.TrimEnd('\r', '\n');

Text.Text = replace;

}

File.WriteAllText(@"Source", Text.Text);

}

private void Opening(object sender, EventArgs e)

{

Text.Text = File.ReadAllText(@"Source");

}

private void color\_settings(object sender, SelectionChangedEventArgs e)

{

int color\_index = 0;

if(Color.SelectedItem is Color\_Item color\_value)

{

color\_index = Color.SelectedIndex;

}

switch (color\_index)

{

case 0:

Text.Foreground = Brushes.Black; break;

case 1:

Text.Foreground = Brushes.Blue; break;

case 2:

Text.Foreground = Brushes.Red; break;

case 3:

Text.Foreground = Brushes.Yellow; break;

case 4:

Text.Foreground = Brushes.Green; break;

}

}

private void Font\_settings(object sender, EventArgs e)

{

int font\_index = 0;

if (Font.SelectedItem is Font\_Item font\_value)

{

font\_index = Font.SelectedIndex;

}

switch (font\_index)

{

case 0:

Text.FontFamily = new FontFamily("Times New Roman"); break;

case 1:

Text.FontFamily = new FontFamily("Segoe UI"); break;

case 2:

Text.FontFamily = new FontFamily("Arial"); break;

case 3:

Text.FontFamily = new FontFamily("Bahnschrift SemiCondensed"); break;

case 4:

Text.FontFamily = new FontFamily("Bodoni MT Black"); break;

}

}

private void size\_settings(object sender, EventArgs e)

{

int size\_index = 0;

if (Font\_Size.SelectedItem is Size\_Item size\_value)

{

size\_index = Font\_Size.SelectedIndex;

}

switch (size\_index)

{

case 0:

Text.FontSize = 8; break;

case 1:

Text.FontSize = 10; break;

case 2:

Text.FontSize = 12; break;

case 3:

Text.FontSize = 14; break;

case 4:

Text.FontSize = 16; break;

case 5:

Text.FontSize = 18; break;

case 6:

Text.FontSize = 20; break;

case 7:

Text.FontSize = 22; break;

case 8:

Text.FontSize = 24; break;

case 9:

Text.FontSize = 26; break;

case 10:

Text.FontSize = 28; break;

case 11:

Text.FontSize = 30; break;

}

}

}

}

public class Color\_Item

{

public string color { get; set; } = "";

public override string ToString() => $"{color}";

}

public class Font\_Item

{

public string font { get; set; } = "";

public override string ToString() => $"{font}";

}

public class Size\_Item

{

public string size { get; set; } = "";

public override string ToString() => $"{size}";

}

