ПРИЛОЖЕНИЕ 1. ШАБЛОН ОТЧЕТА

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Казанский национальный исследовательский технический университет

им. А.Н. Туполева – КАИ»

Институт компьютерных технологий и защиты информации

Отделение СПО ИКТЗИ (Колледж информационных технологий)

ЛАБОРАТОРНАЯ РАБОТА № 5

по дисциплине

Разработка программных модулей

Тема: «Использование стилей в WPF-приложениях»

Работу выполнил

Студент гр.4232

Салахов К.И

Принял

Преподаватель Лоповок Е.Е

Казань 2024

Цель

Научиться использовать стили в WPF-приложениях.

Задания на лабораторную работу

1. Модифицируйте WPF-приложение, разработанное в 3-ей лабораторной работе: используйте стили для однотипных элементов управления.
2. Разработайте приложение MultiEdit для одновременной работы с несколькими текстами. Окно должно быть разделено на две части с одинаковыми градиентами. В каждой части окна должно быть несколько многострочных текстовых полей: одно из них большого размера с крупным шрифтом, а остальные маленького размера с мелким шрифтом. То текстовое окно, в котором пользователь набирает текст, должно быть большим, остальные текстовые поля должны быть маленькими. Внешний вид однотипных элементов управления должен определяться с помощью стилей.

Результат выполнения работы

Модифицированное WPF-приложение (Рисунок 1–4). Функционал приложения остался неизменным, но для однотипных элементво были использованы стили.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 1 – Изменение цвета через меню

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 2 – Информация от разработчика из меню

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описание

Рисунок 3 – Изменениe цвета из ToolBar

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описание

Рисунок 4 – Информация от разработчика из ToolBar

Приложение MultiEdit (Рисунок 5-7).

Изображение выглядит как снимок экрана, текст, линия, Прямоугольник

Автоматически созданное описание

Рисунок 5 - Окно, разделённое на две части с одинаковыми градиентами и textbox

Изображение выглядит как текст, снимок экрана, линия, Шрифт

Автоматически созданное описание

Рисунок 6 – Текстовое окно, в котором пользователь набирает текст увеличивается

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описание

Рисунок 7 – Остальные текстовые поля остаются маленькими

Листинг

1 задние:

<Window x:Class="РПМ\_5\_ЛАБА.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:РПМ\_5\_ЛАБА"

mc:Ignorable="d"

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

<Window.Resources>

<Style TargetType="RadioButton">

<EventSetter Event="Click" Handler="color\_Click"></EventSetter>

</Style>

<Style TargetType="Image">

<Setter Property="Width" Value="70"></Setter>

<Setter Property="Height" Value="70"></Setter>

</Style>

<Style x:Key="Info" TargetType="Button">

<EventSetter Event="Click" Handler="MenuItem\_Click\_1"/>

</Style>

<Style TargetType="Button" x:Key="colorChanged">

<EventSetter Event="Click" Handler="Button\_Click"></EventSetter>

</Style>

<Style TargetType="Button" x:Key="close">

<EventSetter Event="Click" Handler="MenuItem\_Click\_2"></EventSetter>

</Style>

<Style TargetType="MenuItem" x:Key="MenuClose">

<EventSetter Event="Click" Handler="MenuItem\_Click\_2"></EventSetter>

</Style>

<Style TargetType="MenuItem" x:Key="MenuInfo">

<EventSetter Event="Click" Handler="MenuItem\_Click\_1"></EventSetter>

</Style>

<Style TargetType="Grid">

<EventSetter Event="MouseLeave" Handler="STROKA\_MouseLeave"/>

</Style>

</Window.Resources>

<Grid>

<WrapPanel>

<Menu Background="White" BorderBrush="Navy" BorderThickness="1" Grid.ColumnSpan="2">

<MenuItem Header="\_Меню" MouseMove="MenuItem\_MouseMove">

<ComboBox SelectedIndex="0">

<ComboBoxItem Content="Изменить цвет" MouseMove="Color\_changed\_MouseMove" />

<RadioButton Content="Чёрный" x:Name="color1" MouseMove="black\_MouseMove"></RadioButton>

<RadioButton Content="Бирюзовый" x:Name="color2" MouseMove="turquoise\_MouseMove" ></RadioButton>

<RadioButton Content="Розовый" x:Name="color3" MouseMove="Hot\_Pink\_MouseMove"></RadioButton>

</ComboBox>

<MenuItem Style="{StaticResource MenuInfo}" Header="\_Получить информацию о разработчике" MouseMove="Info\_MouseMove"/>

<MenuItem Style="{StaticResource MenuClose}" Header="\_Закрыть окно" MouseMove="Close\_MouseMove"/>

</MenuItem>

</Menu>

</WrapPanel>

<ToolBar Margin="348,10,10,357" Background="White">

<Button Style="{StaticResource colorChanged}" x:Name="biryzovie" MouseMove="turquoise\_MouseMove" >

<Image Source="C:\Users\Камиль\OneDrive\Изображения\Снимки экрана\бирюзовый.png"></Image>

</Button>

<Button Style="{StaticResource colorChanged}" x:Name="black" MouseMove="black\_MouseMove">

<Image Source="C:\Users\Камиль\OneDrive\Изображения\Снимки экрана\Черный.png"></Image>

</Button>

<Button Style="{StaticResource colorChanged}" x:Name="pink" MouseMove="Hot\_Pink\_MouseMove" >

<Image Source="C:\Users\Камиль\OneDrive\Изображения\Снимки экрана\Хот пинк.png"></Image>

</Button>

<Separator/>

<Button Style="{StaticResource Info}" MouseMove="Info\_MouseMove">

<Image Source="C:\Users\Камиль\OneDrive\Изображения\Снимки экрана\инфа.png"></Image>

</Button>

<Separator/>

<Button Style="{StaticResource close}" MouseMove="Close\_MouseMove">

<Image Source="C:\Users\Камиль\OneDrive\Изображения\Снимки экрана\закрыть.png"></Image>

</Button>

</ToolBar>

<StatusBar DockPanel.Dock="Bottom" Margin="0,380,10,10">

<TextBlock x:Name="str\_left" Text="" />

<StatusBarItem HorizontalAlignment="Right" >

<TextBlock x:Name="stroka" Text=""/>

</StatusBarItem>

</StatusBar>

</Grid>

</Window>

using System.Text;

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 static System.Net.Mime.MediaTypeNames;

namespace РПМ\_5\_ЛАБА

{

/// <summary>

/// Interaction logic for MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

}

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

{

MessageBox.Show("Привет! Я Камиль и разработчик данного НЕРЕАЛЬНОГО приложения (ладно шутка). Пока из меня разрабочтик, так себе, но я вот учусь)");

}

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

{

Close();

}

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

{

if (color1.IsChecked == true)

{

this.Background = Brushes.Black;

}

else if (color2.IsChecked == true)

{

this.Background = Brushes.Aqua;

}

else if (color3.IsChecked == true)

{

this.Background = Brushes.HotPink;

}

}

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

{

Button button = (Button)sender;

if (button.Name == "biryzovie")

{

this.Background = Brushes.Aqua;

}

else if (button.Name == "black")

{

this.Background = Brushes.Black;

}

else if (button.Name == "pink")

{

this.Background = Brushes.HotPink;

}

}

private void STROKA\_MouseLeave(object sender, MouseEventArgs e)

{

this.stroka.Text = "";

this.str\_left.Text = "";

}

private void menu\_MouseMove(object sender, MouseEventArgs e)

{

this.stroka.Text = "Меню";

}

private void turquoise\_MouseMove(object sender, MouseEventArgs e)

{

this.stroka.Text = "Изменить цвет на бирюзовый";

}

private void Color\_changed\_MouseMove(object sender, MouseEventArgs e)

{

this.stroka.Text = "Изменить цвет";

}

private void MenuItem\_MouseMove(object sender, MouseEventArgs e)

{

this.str\_left.Text = "Меню";

}

private void black\_MouseMove(object sender, MouseEventArgs e)

{

this.stroka.Text = "Изменить цвет на чёрный";

}

private void Hot\_Pink\_MouseMove(object sender, MouseEventArgs e)

{

this.stroka.Text = "Изменить цвет на розовый";

}

private void Info\_MouseMove(object sender, MouseEventArgs e)

{

this.stroka.Text = "Получить информацию от разработчика";

}

private void Close\_MouseMove(object sender, MouseEventArgs e)

{

this.stroka.Text = "Закрыть окно";

}

}

}

2 задание:

<Window x:Class="РПМ\_5\_ЛАБА.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:РПМ\_5\_ЛАБА"

mc:Ignorable="d"

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

<Window.Resources>

<LinearGradientBrush x:Key="GradientBackground" StartPoint="0,0" EndPoint="1,1">

<GradientStop Color="LightBlue" Offset="0.0" />

<GradientStop Color="Blue" Offset="1.0" />

</LinearGradientBrush>

<Style TargetType="TextBox" >

<Style.Triggers>

<Trigger Property="IsFocused" Value="True">

<Trigger.Setters>

<Setter Property="FontSize" Value="30"></Setter>

<Setter Property="Height" Value="50"></Setter>

<Setter Property="Width" Value="400"></Setter>

<Setter Property="FontWeight" Value="Bold"></Setter>

</Trigger.Setters>

</Trigger>

</Style.Triggers>

<Setter Property="Width" Value="300"></Setter>

<Setter Property="Height" Value="30"></Setter>

<Setter Property="Margin" Value="0,0,0,10"></Setter>

</Style>

</Window.Resources>

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*" />

<ColumnDefinition Width="\*" />

</Grid.ColumnDefinitions>

<Border Grid.Column="0" Background="{StaticResource GradientBackground}">

<Grid VerticalAlignment="Center" HorizontalAlignment="Center">

<StackPanel>

<TextBox/>

<TextBox/>

<TextBox/>

</StackPanel>

</Grid>

</Border>

<Border Grid.Column="1" Background="{StaticResource GradientBackground}">

<Grid VerticalAlignment="Center" HorizontalAlignment="Center">

<StackPanel>

<TextBox/>

<TextBox />

<TextBox/>

</StackPanel>

</Grid>

</Border>

</Grid>

</Window>