Министерство образования и молодежной политики Свердловской области

ГАПОУ СО «Екатеринбургский колледж транспортного строительства»

Отчет по ТРПО. Лабораторная работа   
**Основные элементы управления WPF**

Выполнили: Максимов К. Э.

Группа: ПР-31

Преподаватель: Бушуева Е. В.

2024

Задание 1

<Window x:Class="Lab\_3\_WPF.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:Lab\_3\_WPF"

mc:Ignorable="d"

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

<DockPanel>

<Menu DockPanel.Dock="Top">

<MenuItem Header="Меню">

<MenuItem Header="Изменить цвет фона" Click="ChangeColor\_Click"/>

<MenuItem Header="Получить информацию о разработчике" Click="AboutDeveloper\_Click"/>

<MenuItem Header="Выход" Click="Exit\_Click"/>

</MenuItem>

</Menu>

<ToolBar DockPanel.Dock="Top">

<Button Content="Изменить цвет фона" Click="ChangeColor\_Click" MouseEnter="Button\_MouseEnter" MouseLeave="Button\_MouseLeave"/>

<Button Content="Получить информацию о разработчике" Click="AboutDeveloper\_Click" MouseEnter="Button\_MouseEnter" MouseLeave="Button\_MouseLeave"/>

<Button Content="Выход" Click="Exit\_Click" MouseEnter="Button\_MouseEnter" MouseLeave="Button\_MouseLeave"/>

</ToolBar>

<StatusBar DockPanel.Dock="Bottom">

<TextBlock x:Name="statusBarText"/>

</StatusBar>

<Grid x:Name="mainGrid"/>

</DockPanel>

</Window>

using System.Windows.Media;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Input;

namespace Lab\_3\_WPF

{

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

}

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

{

mainGrid.Background = new SolidColorBrush(Colors.LightBlue);

statusBarText.Text = "Цвет фона изменен";

}

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

{

MessageBox.Show("Разработчик: Кирилл");

statusBarText.Text = "Информация о разработчике";

}

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

{

this.Close();

}

private void Button\_MouseEnter(object sender, MouseEventArgs e)

{

Button button = sender as Button;

statusBarText.Text = $"Вы навели на кнопку: {button.Content}";

}

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

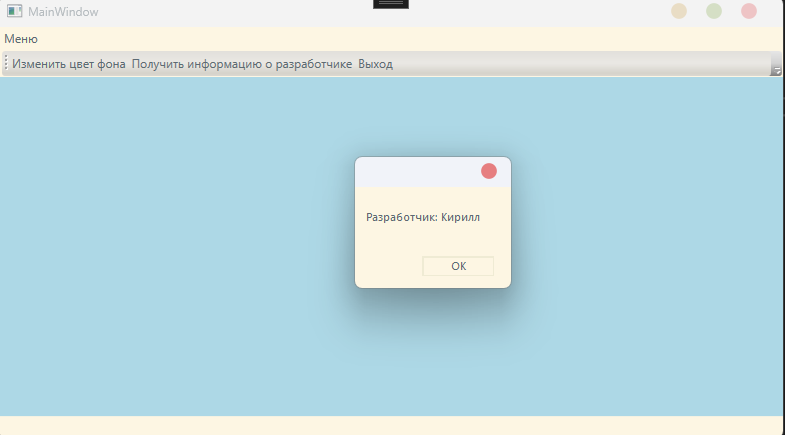
{

statusBarText.Text = "";

}

}

}

Задание 2

<Window x:Class="WpfApp\_2.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:WpfApp\_2"

mc:Ignorable="d"

Title="Графический редактор" Height="450" Width="800">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="Auto"/>

<RowDefinition Height="\*"/>

</Grid.RowDefinitions>

<StackPanel Orientation="Horizontal" Grid.Row="0">

<ComboBox Name="colorComboBox" Width="120" Height="20" Margin="10" MaxDropDownHeight="200"/>

<Slider Name="sizeSlider" Width="120" Margin="10" Minimum="1" Maximum="10"/>

<RadioButton Name="drawRadioButton" Content="Рисование" IsChecked="True" Margin="10"/>

<RadioButton Name="editRadioButton" Content="Редактирование" Margin="10"/>

<RadioButton Name="deleteRadioButton" Content="Удаление" Margin="10"/>

</StackPanel>

<Canvas Name="myCanvas" Background="White" MouseDown="Canvas\_MouseDown" MouseMove="Canvas\_MouseMove" MouseUp="myCanvas\_MouseUp" Grid.Row="1"/>

</Grid>

</Window>

using System.Windows;

using System.Windows.Controls;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Shapes;

namespace WpfApp\_2

{

public partial class MainWindow : Window

{

private Shape \_currentShape;

private Point \_startPoint;

public MainWindow()

{

InitializeComponent();

foreach (var prop in typeof(Colors).GetProperties())

{

colorComboBox.Items.Add(prop.Name);

}

}

private void Canvas\_MouseDown(object sender, MouseButtonEventArgs e)

{

\_startPoint = e.GetPosition(myCanvas);

if (drawRadioButton.IsChecked == true)

{

if (colorComboBox.SelectedItem != null)

{

\_currentShape = new Ellipse

{

Stroke = (Brush)new BrushConverter().ConvertFromString(colorComboBox.SelectedItem.ToString()),

StrokeThickness = sizeSlider.Value

};

Canvas.SetLeft(\_currentShape, \_startPoint.X);

Canvas.SetTop(\_currentShape, \_startPoint.Y);

myCanvas.Children.Add(\_currentShape);

}

else

{

MessageBox.Show("Пожалуйста, выберите цвет перед рисованием.");

}

}

else if (editRadioButton.IsChecked == true || deleteRadioButton.IsChecked == true)

{

\_currentShape = null;

foreach (var child in myCanvas.Children)

{

if (child is Ellipse ellipse)

{

var mousePos = e.GetPosition(myCanvas);

var left = Canvas.GetLeft(ellipse);

var top = Canvas.GetTop(ellipse);

var right = left + ellipse.Width;

var bottom = top + ellipse.Height;

if (mousePos.X >= left && mousePos.X <= right && mousePos.Y >= top && mousePos.Y <= bottom)

{

\_currentShape = ellipse;

break;

}

}

}

if (\_currentShape != null)

{

if (editRadioButton.IsChecked == true)

{

}

else if (deleteRadioButton.IsChecked == true)

{

myCanvas.Children.Remove(\_currentShape);

}

}

}

}

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

{

if (e.LeftButton == MouseButtonState.Pressed && \_currentShape != null)

{

if (drawRadioButton.IsChecked == true)

{

var pos = e.GetPosition(myCanvas);

var width = pos.X - \_startPoint.X;

var height = pos.Y - \_startPoint.Y;

if (Keyboard.IsKeyDown(Key.LeftShift) || Keyboard.IsKeyDown(Key.LeftCtrl))

{

var size = Math.Max(width, height);

\_currentShape.Width = size < 0 ? 0 : size;

\_currentShape.Height = size < 0 ? 0 : size;

}

else

{

\_currentShape.Width = width < 0 ? 0 : width;

\_currentShape.Height = height < 0 ? 0 : height;

}

}

else if (editRadioButton.IsChecked == true)

{

var pos = e.GetPosition(myCanvas);

var offsetX = pos.X - \_startPoint.X;

var offsetY = pos.Y - \_startPoint.Y;

Canvas.SetLeft(\_currentShape, Canvas.GetLeft(\_currentShape) + offsetX);

Canvas.SetTop(\_currentShape, Canvas.GetTop(\_currentShape) + offsetY);

\_startPoint = pos;

}

}

}

private void myCanvas\_MouseUp(object sender, MouseButtonEventArgs e)

{

if (editRadioButton.IsChecked == true)

{

\_currentShape = null;

}

}

}

}

