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 № 4

по дисциплине

Разработка программных модулей

Тема: «Привязка данных»

Работу выполнил

Студент гр.4232

Салахов К.И

Принял

Преподаватель Лоповок Е.Е

Казань 2024

Цель работы

Научиться использовать привязку данных

Задания на лабораторную работу

Модифицируйте приложения, разработанные в предыдущей лабораторной работе: удалите как можно больше обработчиков событий и реализуйте ту же функциональность приложения с помощью привязки данных.

Результат выполнения работы

1. Графический редактор. Получилось реализовать без обработчиков событий всё, кроме увеличения ширины кисти.

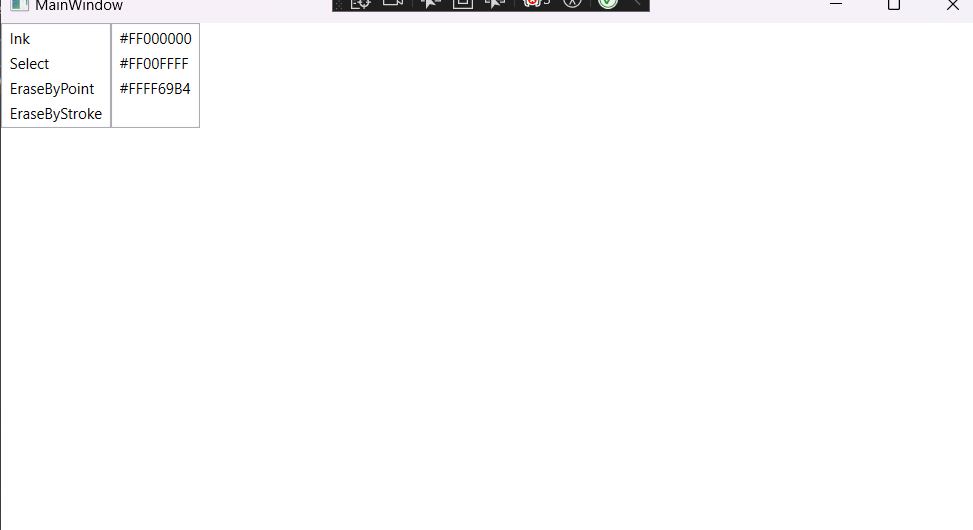


Рисунок 1 – Внешний вид графического редактора

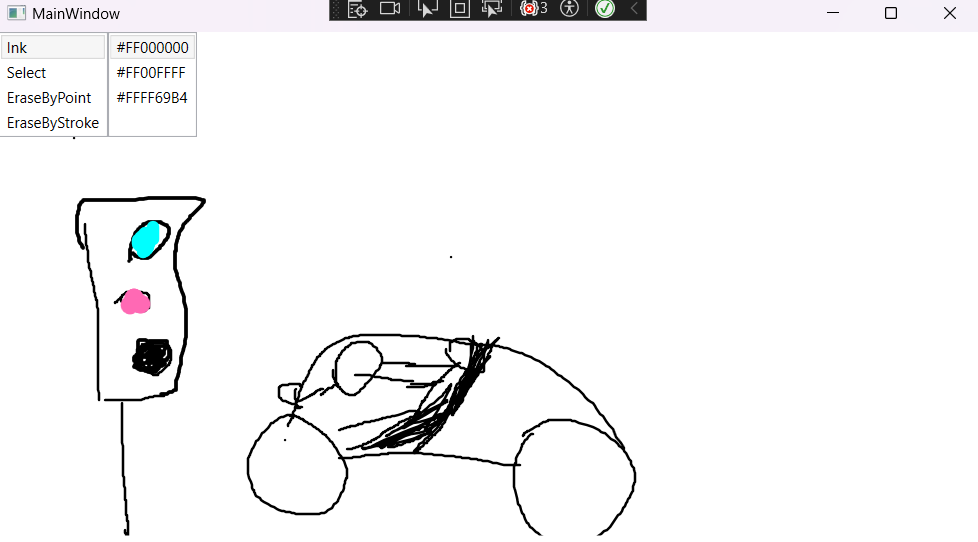


Рисунок 2 – Режим рисования с выбором разных цветов

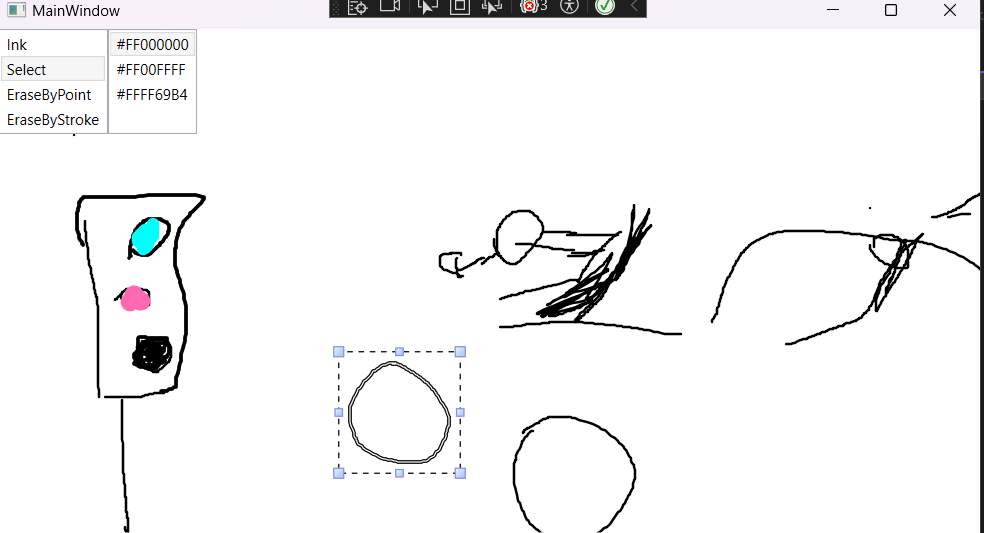


Рисунок 3 – Режим редактирования

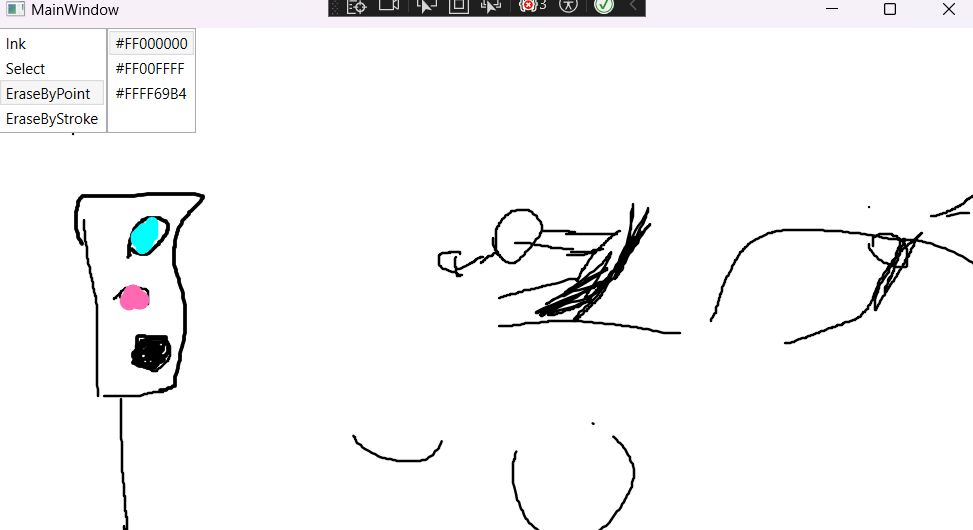


Рисунок 4 – Режим удаления по точкам

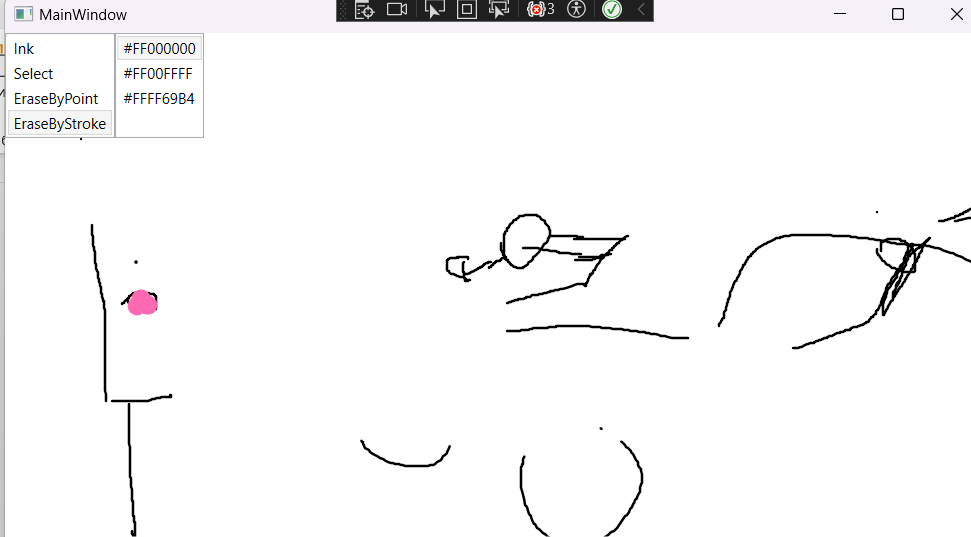


Рисунок 5 – Режим удаления фигуры

1. WPF-приложение с меню, панелью инструментов и строкой состояния. Получилось реализовать с помощью привязки. Обработчики событий использовались только для вывода сообщения через панель инструментов и закрытия окна.

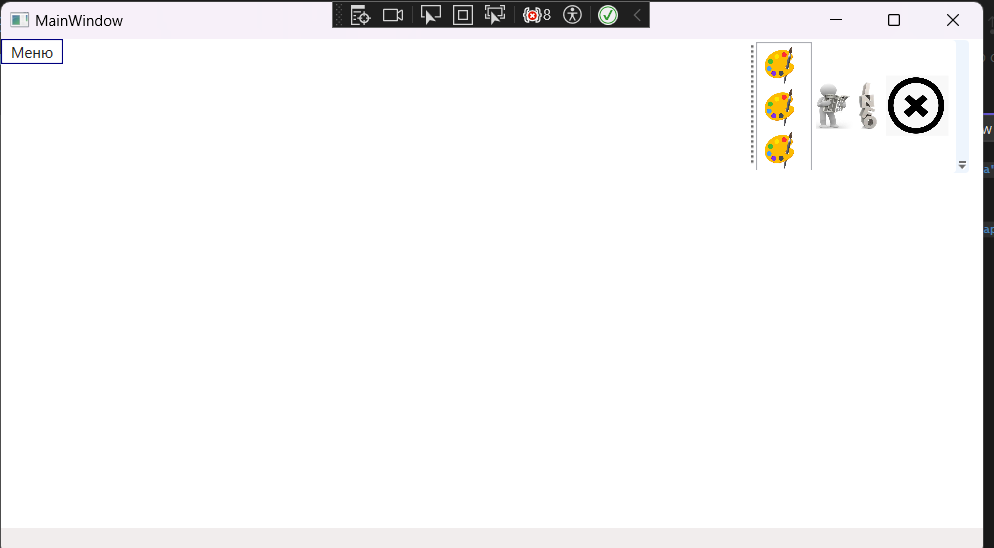


Рисунок 6 – Внешний вид приложения

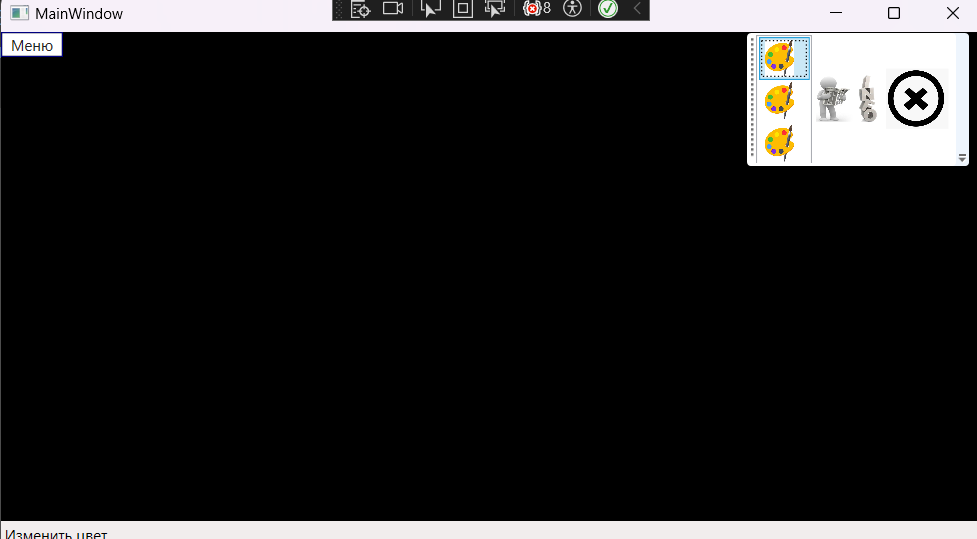


Рисунок 7 – Смена цвета через панель управления

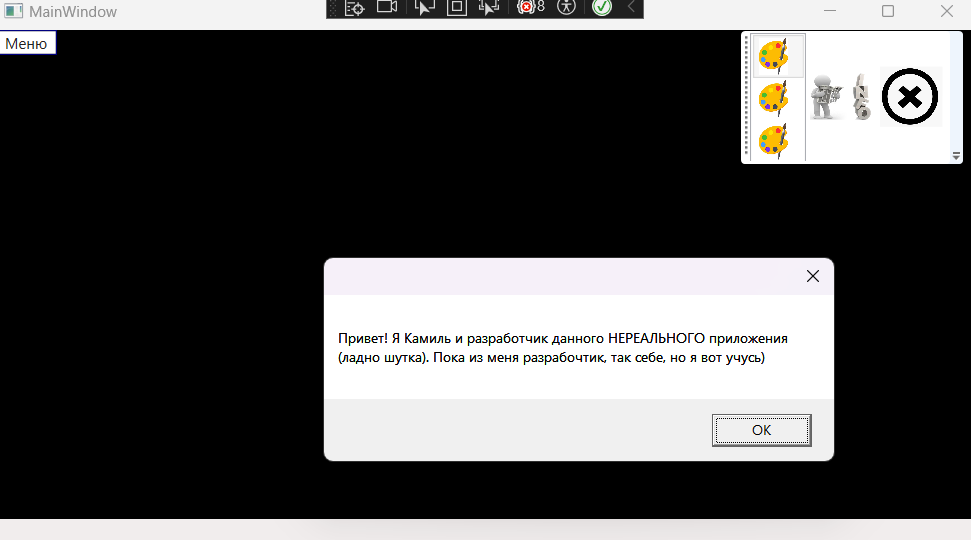


Рисунок 8 – Информация от разработчика через панель управления

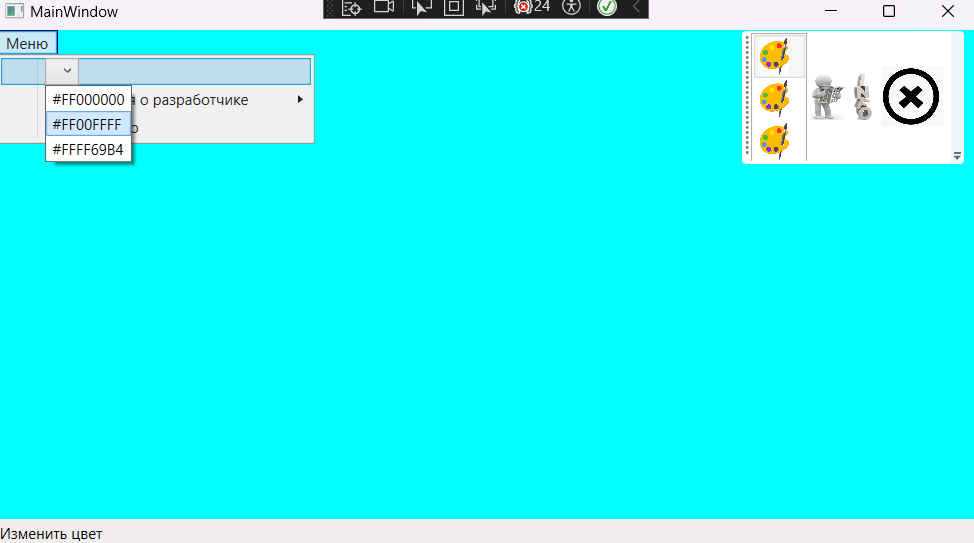


Рисунок 9 – Смена цвета через меню

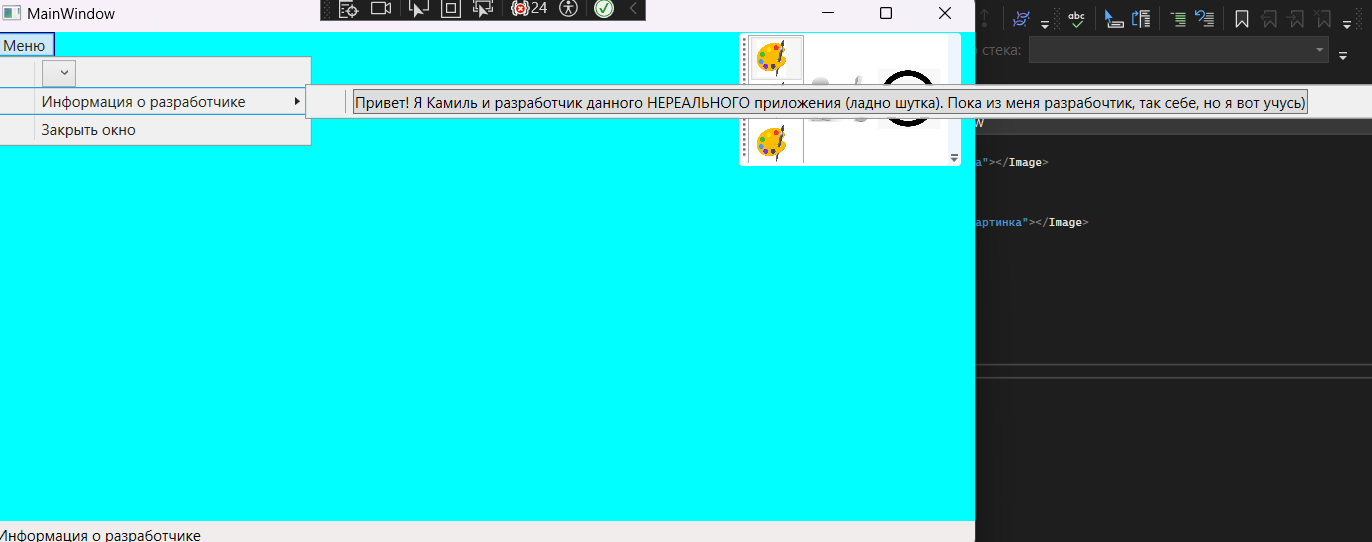


Рисунок 10 – Информация от разработчика через меню

Листинг

Графический редактор:

<Window x:Class="РПМ\_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:РПМ\_ЛР4"

mc:Ignorable="d"

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

<Window.Resources>

<x:Array x:Key="MyEditingModes" Type="{x:Type InkCanvasEditingMode}">

<x:Static Member="InkCanvasEditingMode.Ink"/>

<x:Static Member="InkCanvasEditingMode.Select"/>

<x:Static Member="InkCanvasEditingMode.EraseByPoint"/>

<x:Static Member="InkCanvasEditingMode.EraseByStroke"/>

</x:Array>

<x:Array x:Key="MyDrawingAttributes" Type="{x:Type DrawingAttributes}">

<DrawingAttributes Color="Black" Width="3" Height="3"/>

<DrawingAttributes Color="Aqua" Width="10" Height="10"/>

<DrawingAttributes Color="HotPink" Width="15" Height="15"/>

</x:Array>

</Window.Resources>

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="Auto"/>

<RowDefinition Height="\*"/>

</Grid.RowDefinitions>

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

<ListBox x:Name="lbEditingModes" ItemsSource="{StaticResource MyEditingModes}" />

<ListBox x:Name="lbColors" ItemsSource="{StaticResource MyDrawingAttributes}">

<ListBox.ItemTemplate>

<DataTemplate>

<TextBlock Text="{Binding Path=Color}"></TextBlock>

</DataTemplate>

</ListBox.ItemTemplate>

</ListBox>

</StackPanel>

<InkCanvas EditingMode="{Binding SelectedValue, ElementName=lbEditingModes}" Margin="0,0,0,10" Grid.Row="1" DefaultDrawingAttributes="{Binding ElementName=lbColors, Path=SelectedValue}" />

</Grid>

</Window>

WPF-приложение с меню, панелью инструментов и строкой состояния:

Xaml:

<Window x:Class="РПМ\_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:РПМ\_ЛР4"

mc:Ignorable="d"

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

<Window.Resources>

<Style TargetType="TextBlock">

<Style.Triggers>

<DataTrigger Binding="{Binding ElementName=Меню, Path=IsMouseOver}" Value="True">

<Setter Property="Text" Value="Меню" />

</DataTrigger>

<DataTrigger Binding="{Binding ElementName=menu\_color, Path=IsMouseOver}" Value="True">

<Setter Property="Text" Value="Изменить цвет" />

</DataTrigger>

<DataTrigger Binding="{Binding ElementName=Color\_TolBar, Path=IsMouseOver}" Value="True">

<Setter Property="Text" Value="Изменить цвет" />

</DataTrigger>

<DataTrigger Binding="{Binding ElementName=инфа, Path=IsMouseOver}" Value="True">

<Setter Property="Text" Value="Информация о разработчике" />

</DataTrigger>

<DataTrigger Binding="{Binding ElementName=закрыть, Path=IsMouseOver}" Value="True">

<Setter Property="Text" Value="Закрыть окно" />

</DataTrigger>

<DataTrigger Binding="{Binding ElementName=закрыть\_картинка, Path=IsMouseOver}" Value="True">

<Setter Property="Text" Value="Закрыть окно" />

</DataTrigger>

<DataTrigger Binding="{Binding ElementName=закрыть\_картинка, Path=IsMouseOver}" Value="True">

<Setter Property="Text" Value="Закрыть окно" />

</DataTrigger>

<DataTrigger Binding="{Binding ElementName=инфа\_картинка, Path=IsMouseOver}" Value="True">

<Setter Property="Text" Value="Информация о разработчике" />

</DataTrigger>

</Style.Triggers>

</Style>

<x:Array x:Key="MyBackgroundBrushes" Type="{x:Type Brush}">

<SolidColorBrush Color="Black"/>

<SolidColorBrush Color="Aqua"/>

<SolidColorBrush Color="HotPink"/>

</x:Array>

<x:Array x:Key="информация" Type="{x:Type Button}">

<Button>Привет! Я Камиль и разработчик данного НЕРЕАЛЬНОГО приложения (ладно шутка). Пока из меня разрабочтик, так себе, но я вот учусь)</Button>

</x:Array>

</Window.Resources>

<Grid>

<WrapPanel x:Name="wparpanel" Background="{Binding ElementName=Color\_TolBar, Path=SelectedValue }">

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

<MenuItem Header="\_Меню" x:Name="Меню">

<ComboBox SelectedValue="{Binding ElementName=wparpanel,Path=Background, Mode=OneWayToSource}" x:Name="menu\_color" ItemsSource="{StaticResource MyBackgroundBrushes}"/>

<MenuItem Header="\_Информация о разработчике" ItemsSource="{StaticResource информация}" x:Name="инфа"></MenuItem>

<MenuItem Header="\_Закрыть окно" Click="Button\_Click\_1" x:Name="закрыть"/>

</MenuItem>

</Menu>

</WrapPanel>

<ToolBar Margin="594,0,10,305" Background="White">

<ListBox x:Name="Color\_TolBar" ItemsSource="{StaticResource MyBackgroundBrushes}" Height="109" Width="45" SelectedValue="{Binding ElementName=wparpanel,Path=Background, Mode=OneWayToSource}">

<ListBox.ItemTemplate>

<DataTemplate>

<Image Source="C:\Users\Камиль\OneDrive\Изображения\Снимки экрана\Снимок экрана 2024-03-29 105311.png" Height="30"></Image>

</DataTemplate>

</ListBox.ItemTemplate>

</ListBox>

<Button Click="Button\_Click">

<Image Source="C:\Users\Камиль\OneDrive\Изображения\Снимки экрана\инфа.png" Height="50" Width="50" x:Name="инфа\_картинка"></Image>

</Button>

<Button Click="Button\_Click\_1">

<Image Source="C:\Users\Камиль\OneDrive\Изображения\Снимки экрана\закрыть.png" Height="50" Width="50" x:Name="закрыть\_картинка"></Image>

</Button>

</ToolBar>

<StatusBar VerticalAlignment="Top" Height="43" Margin="0,391,0,0">

<TextBlock x:Name="input\_name\_el" Width="500" Margin="0 0 0 20" Height="40"/>

</StatusBar>

</Grid>

</Window>

Xaml.cs:

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;

namespace РПМ\_ЛР4

{

/// <summary>

/// Interaction logic for MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

}

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

{

Close();

}

}

}