Министерство образования Республики Беларусь

Учреждение образования

“Брестский государственный технический университет”

**Лабораторная работа №4**

**По дисциплине ТРиПС за 3 семестр**  
**Тема: «Привязка данных»**

**Выполнил:**

Студент группы ПО-6(1)  
 2-го курса

Мартынович Даниил

**Проверила:**

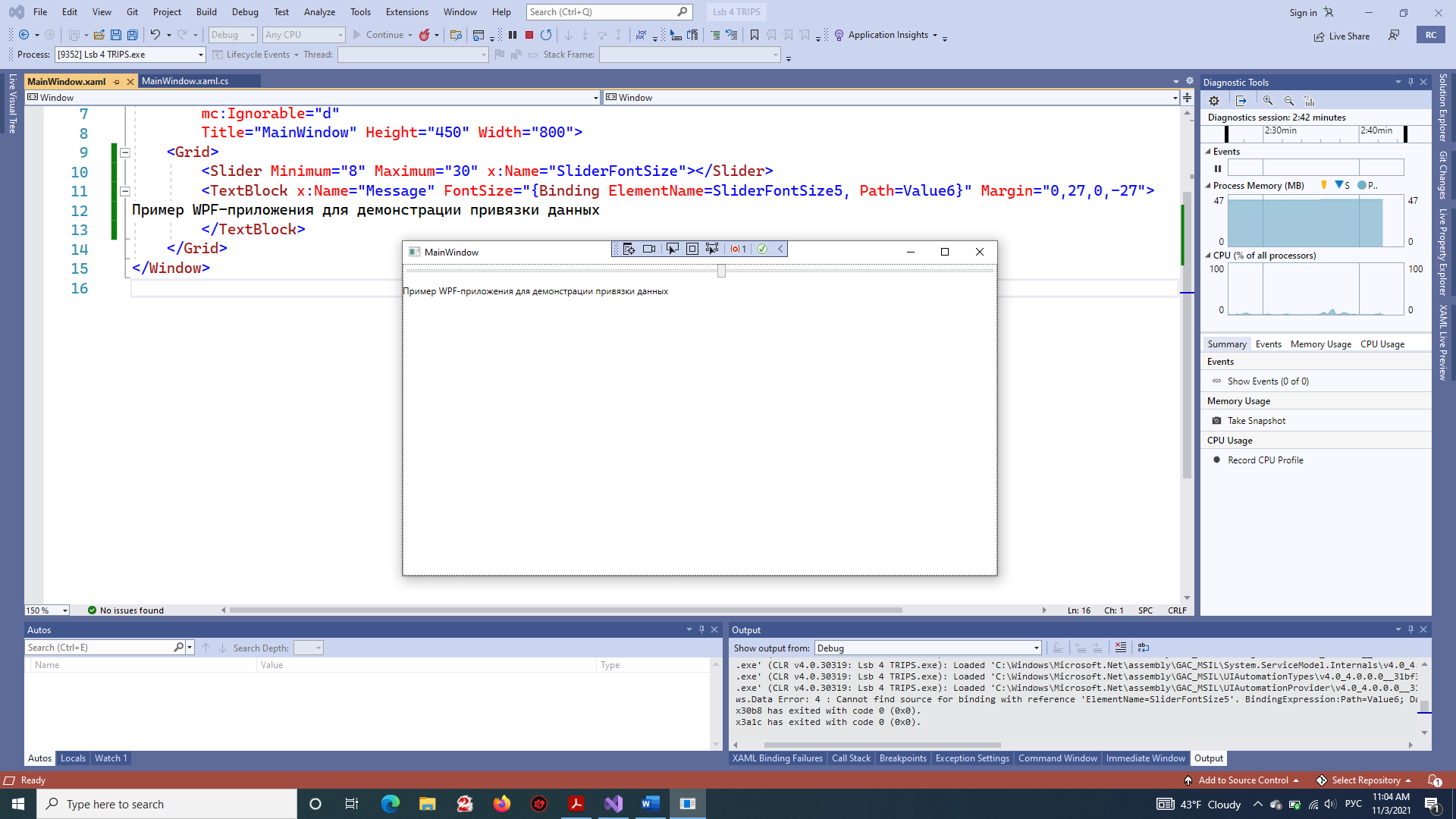
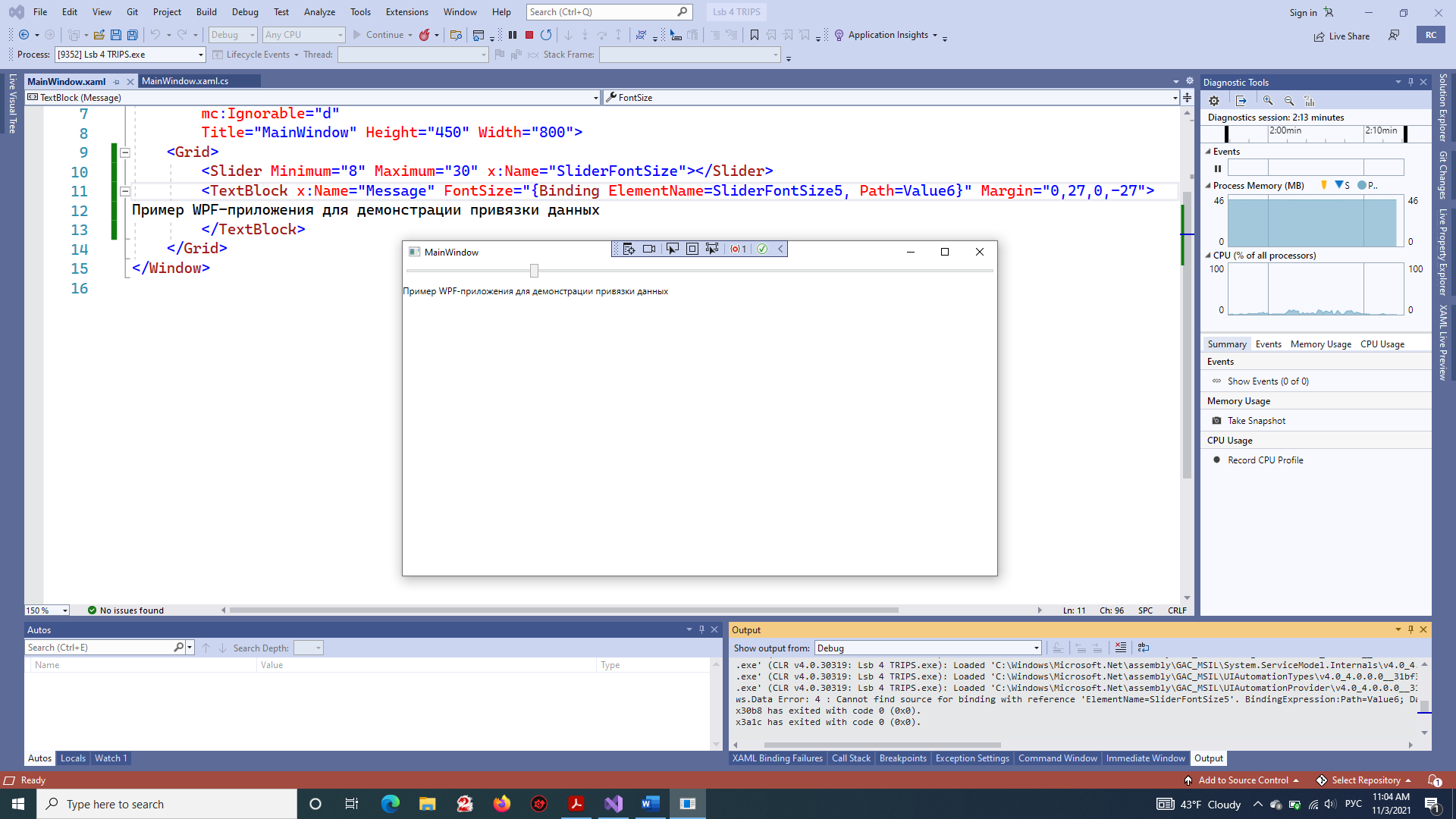
Войцехович О. Ю.

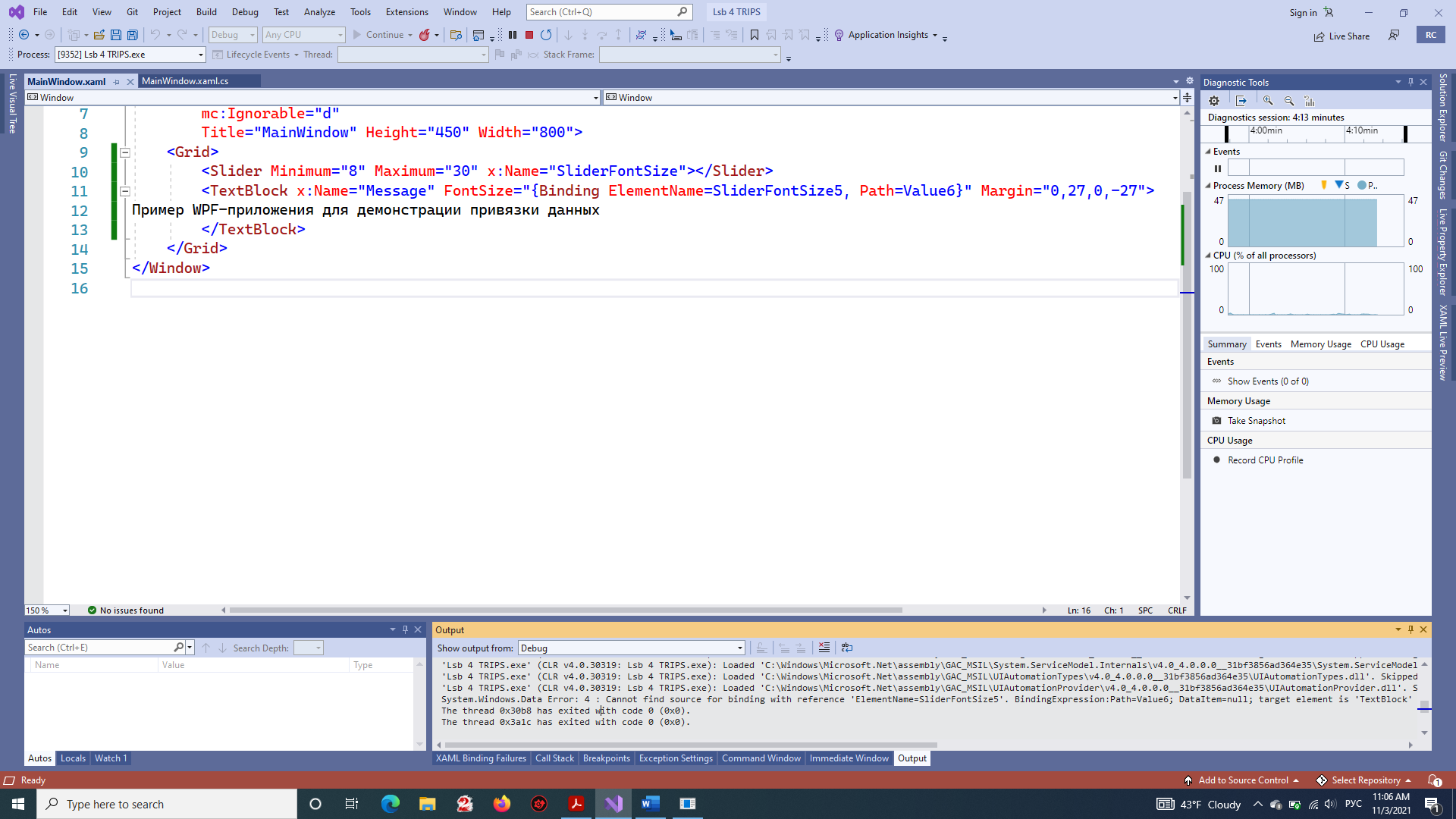
Брест 2021

Ход работы

**Задание 1**

|  |
| --- |
| Проверьте реакцию среды разработки на неверные значения параметров ElementName и Path. Проанализируйте сообщения, которые выводятся в окне вывода (Вид → Вывод) при построении и при запуске приложения. |





Текст перестал зависеть от слайдера. При изменении слайдер, текст не изменяет размер. Окно вывода показывает, что невозможно извлечь данные из свойства исходного объекта, для привязки их в графической системе (Cannot find source for binding with reference…).

**Задание 2**

|  |
| --- |
| Запустите приложение со следующим XAML-кодом:  <TextBox x:Name="t1" />  <TextBox x:Name="t2" Text="{Binding ElementName=t1, Path=Text}" />  <Slider x:Name="slider1" />  <Slider x:Name="slider2" Value="{Binding ElementName=slider1, Path=Value}" />  Определите различие в поведении полей t1 и t2 и модифицируйте код, чтобы устранить это различие. |

Изображение выглядит как стол

Автоматически созданное описание

Слайдеры ведут себя одинаково, то есть TwoWay - целевое свойство обновляется при изменении исходного свойства, исходное свойство обновляется при изменении целевого свойства. Основное свойство различие текст: текст во второй строке отображается только когда мы нажимаем на первую строку.

Код после модификации:

<TextBox x:Name="t1" Text="{Binding ElementName=t2, Path=Text}" Margin="10,10,10,345"/>

<TextBox x:Name="t2" Margin="10,111,10,237" Text="{Binding ElementName=t1, Path=Text}" />

<Slider x:Name="slider1" Value="{Binding ElementName=slider2, Path=Value}" />

<Slider x:Name="slider2" Value="{Binding ElementName=slider1, Path=Value}" />

Изображение выглядит как текст, снимок экрана, компьютер

Автоматически созданное описание

|  |
| --- |
| **Задание 3**  Дополните пример №2 текстовым полем ввода TextBox, в котором пользователь может ввести размер шрифта, и задайте выражения привязки таким образом, чтобы значение ползунка, текст текстового поля и размер шрифта текстового блока соответствовали друг другу. |

Изображение выглядит как стол

Автоматически созданное описание

<Window x:Class="Lsb\_4\_TRIPS.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:Lsb\_4\_TRIPS"

mc:Ignorable="d"

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

<Grid>

<TextBox x:Name="t1" Text="{Binding ElementName=t2, Path=Text}" FontSize="{Binding ElementName=Size, Path=Text}" Margin="10,10,10,345"/>

<TextBox x:Name="t2" Text="{Binding ElementName=t1, Path=Text}" FontSize="{Binding ElementName=Size, Path=Text}" Margin="10,111,10,237"/>

<Slider x:Name="slider1" Maximum="100" Value="{Binding ElementName=Size, Path=Text}" Margin="0,248,0,-248" />

<Slider x:Name="slider2" Maximum="100" Value="{Binding ElementName=Size, Path=Text}" Margin="0,303,0,-303" />

<TextBox x:Name="Size" HorizontalAlignment="Left" Margin="183,378,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="49"/>

<Label Content="Размер шрифта" HorizontalAlignment="Left" Margin="70,374,0,0" VerticalAlignment="Top" Width="96"/>

</Grid>

</Window>

**Задание 4**

|  |
| --- |
| Модифицируйте приложения, разработанные в предыдущей лабораторной работе: удалите как можно больше обработчиков событий и реализуйте ту же функциональность приложения с помощью привязки данных.  Подсказки:  Свойство EditingMode (тип данных InkCanvasEditingMode) элемента управления InkCanvas нельзя напрямую связать с текстовым свойством выпадающего списка ComboBox или списка ListBox, т.к. в этом случае будет несовпадение типов. Для привязки данных необходимо, чтобы тип элементов списка совпадал с типом свойства EditingMode. Для этой цели необходимо добавить в ресурсы окна приложения (элемент Windows.Resources) массив (элемент x:Array) элементов типа InkCanvasEditingMode (артибут x:Type), данному ресурсу необходимо задать ключ (атрибут x:Key), который необходимо указать в свойстве ItemSource списка ListBox или выпадающего списка ComboBox. В этом случае можно будет осуществить привязку данных между свойством EditingMode и выделенным элементом списка:  <Window x:Class="Lab\_3\_zad\_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:Lab\_3\_zad\_2"  mc:Ignorable="d"  Title="Графический редактор" 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="Red" Height="5" Width="5"/>  <DrawingAttributes Color="Green" Height="5" Width="5"/>  <DrawingAttributes Color="Blue" Height="5" Width="5"/>  </x:Array>  </Window.Resources>  <Grid Background="WhiteSmoke">  <ComboBox Name="ChangeColor" ItemsSource="{StaticResource MyDrawingAttributes}" SelectedIndex="0" HorizontalAlignment="Left" Margin="36,30,0,0" VerticalAlignment="Top" Width="120">  <ComboBox.ItemTemplate>  <DataTemplate>  <TextBlock Text="{Binding Path=Color}"></TextBlock>  </DataTemplate>  </ComboBox.ItemTemplate>  </ComboBox >  <ComboBox x:Name="EditMode" SelectedIndex="0" Margin="244,27,0,0" ItemsSource="{StaticResource MyEditingModes}" Height="28" VerticalAlignment="Top" HorizontalAlignment="Left" Width="105"/>  <InkCanvas x:Name="Ink" Margin="0,84,0.2,0" DefaultDrawingAttributes="{Binding ElementName= ChangeColor, Path=SelectedValue}"  EditingMode="{Binding ElementName=EditMode, Path=SelectedValue}" />  <Slider x:Name="Slider" Height="26" Width="100" Minimum="5" Maximum="20" Value="0" Margin="435,28,0,0" ValueChanged="Slider\_ValueChanged" VerticalAlignment="Top" HorizontalAlignment="Left" />  <Label Content="Цвет" Margin="74,0,0,0" Height="30" VerticalAlignment="Top" HorizontalAlignment="Left" Width="44"/>  <Label Content="Режимы" HorizontalAlignment="Left" Margin="244,-4,0,0" VerticalAlignment="Top"/>  <Label Content="Толщина" Margin="435,-3,0,0" Height="26" VerticalAlignment="Top" HorizontalAlignment="Left" Width="66"/>  </Grid>  </Window> |

using System;

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;

namespace Lab\_3\_zad\_2

{

/// <summary>

/// Interaction logic for MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

}

private void Slider\_ValueChanged(object sender, RoutedPropertyChangedEventArgs<double> e)

{

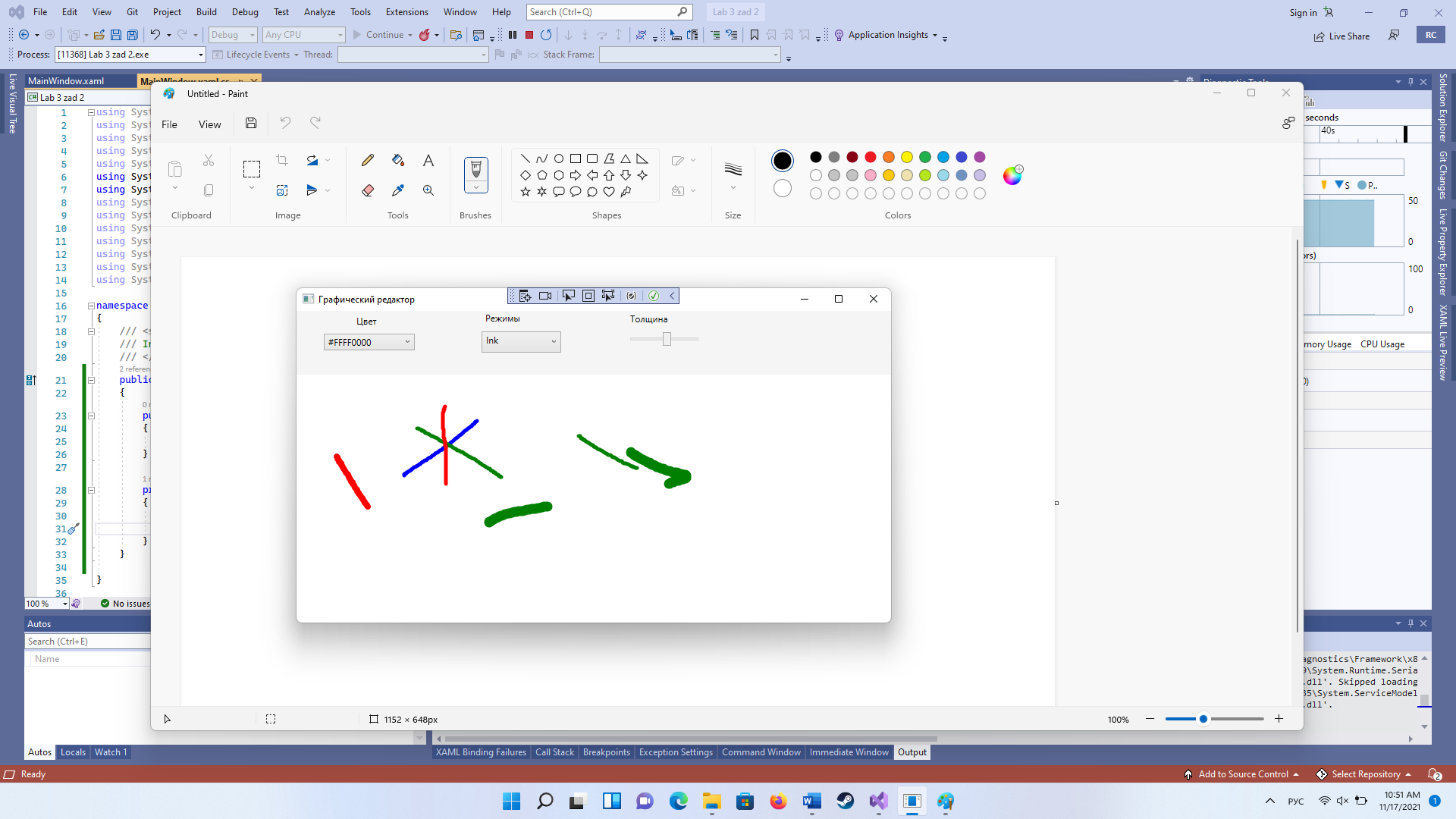
Ink.DefaultDrawingAttributes.Width = Slider.Value;

Ink.DefaultDrawingAttributes.Height = Slider.Value;

}

}

}



Вывод: научился использовать привязку данных.