Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №2

за 1 семестр

По дисциплине: «ТРИПС»

Тема: «Диспетчеры компановки. Привязка данных»

Выполнил:

Студент 2 курса

Группы ПО-4(1)

Иваненко И. Л.

Проверил:

Войцехович О.Ю.

2020

Лабораторная работа №2

Диспетчеры компановки. Привязка данных

Цель работы: изучить диспетчеры компановки и привязку данных в графической системе WPF.

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.

Код программы:

<Window x:Class="lab2.\_1.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:lab2.\_1"

mc:Ignorable="d"

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

<Grid>

<Slider Minimum="8" Maximum="30" x:Name="SliderFontSize"></Slider>

<TextBlock Margin="0, 30, 0, 0" x:Name="Message" FontSize="{Binding ElementName=SliontSize, Path=feis}">

Ppppppppp

</TextBlock>

</Grid>

</Window>

Неверный Path:  
System.Windows.Data Error: 40 : BindingExpression path error: 'feis' property not found on 'object' ''Slider' (Name='SliderFontSize')'. BindingExpression:Path=feis; DataItem='Slider' (Name='SliderFontSize'); target element is 'TextBlock' (Name='Message'); target property is 'FontSize' (type 'Double')

Неверный ElementName:

System.Windows.Data Error: 4 : Cannot find source for binding with reference 'ElementName=SliontSize'. BindingExpression:Path=feis; DataItem=null; target element is 'TextBlock' (Name='Message'); target property is 'FontSize' (type 'Double')

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.

Различие:

Если изменить текст в поле t1, то этот же текст сразу отобразится в поле t2, если же изменить текст в поле t2, то этот текст отобразится в поле t1 только после нажатия на это поле.

Решение:

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

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

<Slider x:Name="slider1"/>

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

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.

Код программы:

<Window x:Class="lab2.\_3.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:lab2.\_3"

mc:Ignorable="d"

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

<Grid>

<Slider Minimum="8" Maximum="30" x:Name="SliderFontSize" Value="{Binding ElementName=tbox, Path=Text}" Margin="0,0,-0.4,0"></Slider>

<TextBlock x:Name="Message" FontSize="{Binding ElementName=SliderFontSize, Path=Value}" Margin="0,40,-0.4,0">

Dksaodfjeofjewiof

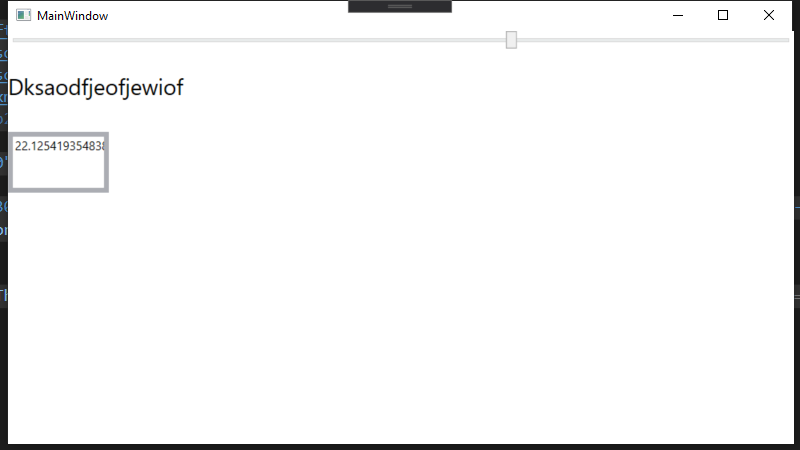
</TextBlock>

<TextBox x:Name="tbox" BorderThickness="5" Margin="0,101,684.6,251" Text="{Binding ElementName=SliderFontSize, Path=Value}"/>

</Grid>

</Window>

Результат выполнения:



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.

Удалил обработчики событий выбора цвета и выбора метода редактирования:

MainWindow.xaml:

<Window x:Class="GraphicalDesigner\_v2.\_0.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:GraphicalDesigner\_v2.\_0"

mc:Ignorable="d"

Title="Graphical Designer v2.0" Height="450" Width="800">

<Window.Resources>

<x:Array x:Key="EditingModes" 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="DrawingAttributes\_arr" Type="{x:Type DrawingAttributes}">

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

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

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

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

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

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

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

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

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

</x:Array>

</Window.Resources>

<Grid>

<InkCanvas Name ="Ink" EditingMode="{Binding ElementName=EditingModelsList, Path=SelectedValue}" DefaultDrawingAttributes="{Binding ElementName=ColorsList, Path=SelectedValue}"/>

<Grid Name="TopPanel" VerticalAlignment="Top" HorizontalAlignment="Left" Width="800" Height="80" Background="Gainsboro">

<TextBlock Text="Choose color:" FontFamily="Arial" FontSize="18" Margin="20, 10, 0, 0"/>

<ComboBox FontFamily="Arial" FontSize="16" Name="ColorsList" Width="200" Height="30" HorizontalAlignment="Left" Margin="20,15,0,0" SelectedIndex="0" ItemsSource="{StaticResource DrawingAttributes\_arr}">

<ComboBox.ItemTemplate>

<DataTemplate>

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

</DataTemplate>

</ComboBox.ItemTemplate>

</ComboBox>

<TextBlock Text="Choose mode:" FontFamily="Arial" FontSize="18" Margin="280,10,0,0"/>

<ComboBox FontFamily="Arial" FontSize="16" HorizontalAlignment="Center" Background="Gainsboro" Margin="-40, 15, 0, 0" Name="EditingModelsList" Width="200" Height="30" SelectedIndex="0" ItemsSource="{StaticResource EditingModes}"/>

<TextBlock Text="Choose width:" FontFamily="Arial" FontSize="18" Margin="500,10,0,0"/>

<Slider Name="ChooseWidth" Maximum="100" Value="1" TickPlacement="Both" TickFrequency="10" Margin="400,35,0,0" Width="200" Foreground="Black" ValueChanged="BrushWidthChange"/>

</Grid>

<Button Name="CloseWindow" Width="40" Height="40" BorderThickness="0" VerticalAlignment="Top" HorizontalAlignment="Right" Background="Gainsboro" Click="CloseWindow\_Click">

<Image Source="https://emojio.ru/images/apple-b/274c.png"></Image>

</Button>

</Grid>

</Window>

MainWindow.xaml.cs:

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 GraphicalDesigner\_v2.\_0

{

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

}

//choosing brush width

private void BrushWidthChange(object sender, RoutedPropertyChangedEventArgs<double> e)

{

Ink.DefaultDrawingAttributes.Width = ChooseWidth.Value;

Ink.DefaultDrawingAttributes.Height = ChooseWidth.Value;

}

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

{

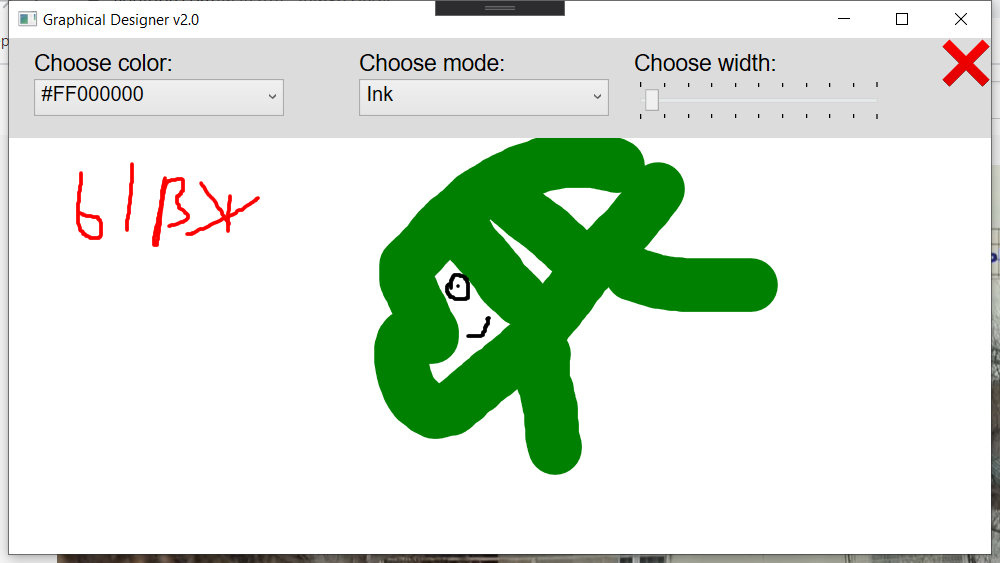
Close();

}

}

}

Результат выполнения:



Задание по диспетчерам компановки:  
Код программы:

<Window x:Class="imdone.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:imdone"

mc:Ignorable="d"

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

<Grid>

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="Auto"/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="130px"/>

<ColumnDefinition Width="Auto"/>

</Grid.ColumnDefinitions>

<DockPanel Grid.Row="0" Grid.Column="0" HorizontalAlignment="Left" LastChildFill="False">

<TextBlock DockPanel.Dock="Top">Последние документы</TextBlock>

<TextBlock DockPanel.Dock="Top" Margin="0,5,0,0">1.txt</TextBlock>

<TextBlock DockPanel.Dock="Top" Margin="0,5,0,0">2.txt</TextBlock>

<TextBlock DockPanel.Dock="Top" Margin="0,5,0,0">3.txt</TextBlock>

<TextBlock DockPanel.Dock="Top" Margin="0,5,0,0">4.txt</TextBlock>

<TextBlock DockPanel.Dock="Top" Margin="0,5,0,0">5.txt</TextBlock>

<TextBlock DockPanel.Dock="Top" Margin="0,5,0,0">6.txt</TextBlock>

</DockPanel>

<Grid Grid.Row="0" Grid.Column="1" VerticalAlignment="Top" Height="120px">

<StackPanel Orientation="Vertical">

<DockPanel LastChildFill="False">

<Button Content="Открыть" Height="20px"/>

<Button Content="Сохранить" Margin="10, 0, 0, 0" Height="20px"/>

<Button Content="Закрыть" Margin="10, 0, 0, 0" Height="20px"/>

<Button Content="Обработать" Margin="10, 0, 0, 0" Height="20px"/>

<Button Content="О Программе" Margin="10, 0, 0, 0" Height="20px"/>

</DockPanel>

<DockPanel LastChildFill="False" Margin="0,10,0,0">

<TextBlock FontWeight="Bold">1.txt</TextBlock>

<TextBlock>2.txt</TextBlock>

<TextBlock>3.txt</TextBlock>

</DockPanel>

</StackPanel>

</Grid>

</Grid>

<WrapPanel Margin="130px,60px,0,0">

<Button Content="A" Width="50px" Height="60px" Margin="10,0,0,0" Background="Red"/>

<Button Content="B" Width="50px" Height="60px" Margin="10,0,0,0" Background="Green"/>

<Button Content="C" Width="50px" Height="60px" Margin="10,0,0,0" Background="Blue"/>

<Button Content="D" Width="50px" Height="60px" Margin="10,0,0,0" Background="Yellow"/>

<Button Content="E" Width="50px" Height="60px" Margin="10,0,0,0" Background="Gray"/>

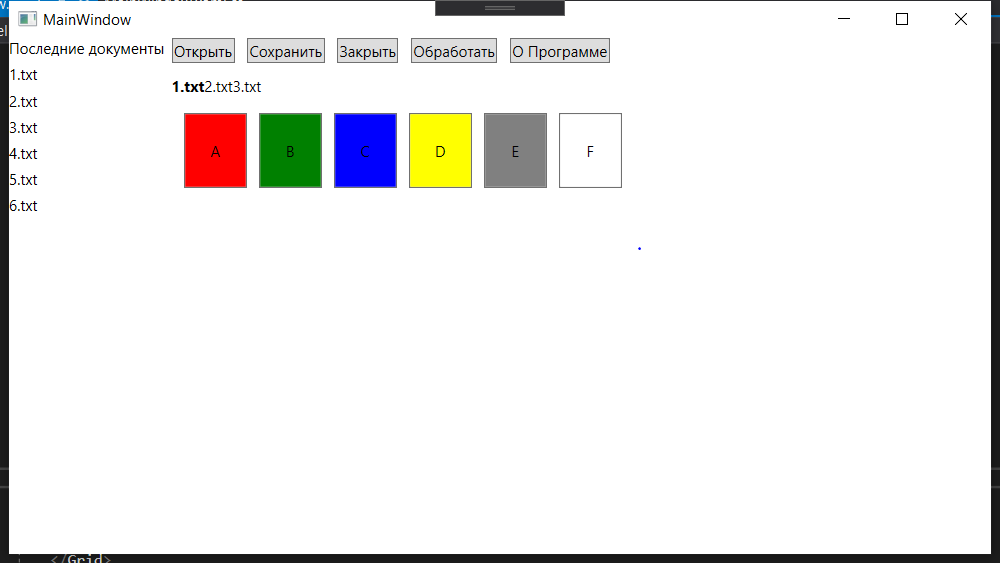
<Button Content="F" Width="50px" Height="60px" Margin="10,0,0,0" Background="White"/>

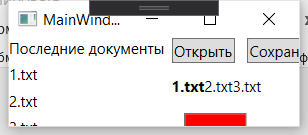
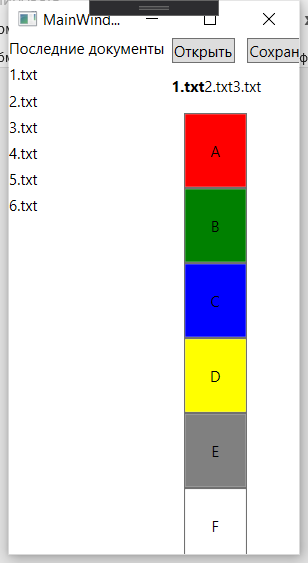
</WrapPanel>

</Grid>

</Window>

Результат выполнения:





Вывод: В ходе выполнения данной работы я ознакомился с привязкой данных и диспетчерами компоновки в WPF.