Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №3

за 1 семестр

По дисциплине: «ТРИПС»

Тема: «Стили и триггеры»

Выполнил:

Студент 2 курса

Группы ПО-4(1)

Иваненко И. Л.

Проверил:

Войцехович О.Ю.

2020

Цель работы: “Ознакомится со стилями и триггерами WPF”

Задание 1: Модифицируйте WPF-приложение, разработанное ранее: используйте стили для однотипных элементов управления.

Код программы:

<Window x:Class="lab\_3\_\_task\_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:lab\_3\_\_task\_1"

mc:Ignorable="d"

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

<Window.Resources>

<Style x:Key="FrontButtons">

<Setter Property="Control.Width" Value="50px"/>

<Setter Property="Control.Height" Value="60px"/>

<Setter Property="Control.Margin" Value="15,0,0,0"/>

<Setter Property="Control.BorderThickness" Value="0"/>

</Style>

<Style x:Key="OptionsButtons">

<Setter Property="Control.Background">

<Setter.Value>

<LinearGradientBrush StartPoint="0,0" EndPoint="0,0.5">

<LinearGradientBrush.GradientStops>

<GradientStop Color="White" Offset="0"/>

<GradientStop Color="Gray" Offset="2"/>

</LinearGradientBrush.GradientStops>

</LinearGradientBrush>

</Setter.Value>

</Setter>

<Setter Property="Control.Height" Value="20px"/>

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

</Style>

</Window.Resources>

<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" Style="{StaticResource OptionsButtons}"/>

<Button Content="Сохранить" Style="{StaticResource OptionsButtons}"/>

<Button Content="Закрыть" Style="{StaticResource OptionsButtons}"/>

<Button Content="Обработать" Style="{StaticResource OptionsButtons}" />

<Button Content="О Программе" Style="{StaticResource OptionsButtons}" />

</DockPanel>

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

<TextBlock FontWeight="Bold" Margin="10, 0, 0, 0">1.txt</TextBlock>

<TextBlock Margin="10, 0, 0, 0">2.txt</TextBlock>

<TextBlock Margin="10, 0, 0, 0">3.txt</TextBlock>

</DockPanel>

</StackPanel>

</Grid>

</Grid>

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

<Button Content="A" Background="Yellow" Style="{StaticResource FrontButtons}"/>

<Button Content="B" Background="Red" Style="{StaticResource FrontButtons}"/>

<Button Content="C" Background="Green" Style="{StaticResource FrontButtons}"/>

<Button Content="D" Background="Blue" Style="{StaticResource FrontButtons}"/>

<Button Content="E" Background="Gray" Style="{StaticResource FrontButtons}"/>

<Button Content="F" Background="Moccasin" Style="{StaticResource FrontButtons}"/>

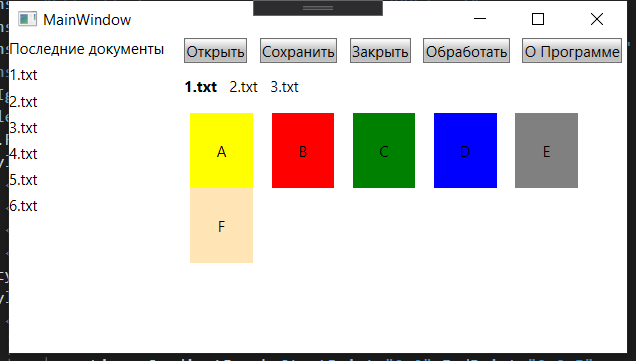
</WrapPanel>

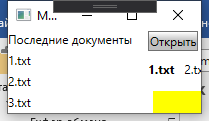
</Grid>

</Window>

Результат выполнения:







Задание 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 определиться с помощью стилей.

Код программы:

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

mc:Ignorable="d"

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

<Window.Resources>

<Style x:Key="Gradient">

<Setter Property="Control.Background">

<Setter.Value>

<LinearGradientBrush StartPoint="0,0" EndPoint="0,0.5">

<LinearGradientBrush.GradientStops>

<GradientStop Color="Blue" Offset="0"/>

<GradientStop Color="Red" Offset="2"/>

</LinearGradientBrush.GradientStops>

</LinearGradientBrush>

</Setter.Value>

</Setter>

</Style>

<Style x:Key="AllTextBoxes">

<Setter Property="Control.Opacity" Value="0.5"/>

</Style>

<Style x:Key="BigTextBox" BasedOn="{StaticResource AllTextBoxes}">

<Setter Property="Control.Height" Value="85px"/>

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

</Style>

<Style x:Key="SmallTextBox" BasedOn="{StaticResource AllTextBoxes}">

<Setter Property="Control.Height" Value="25px"/>

<Setter Property="Control.Margin" Value="0,40,0,0"/>

<Setter Property="Control.FontSize" Value="12"/>

</Style>

</Window.Resources>

<Grid>

<Grid>

<Grid.RowDefinitions>

<RowDefinition/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition/>

<ColumnDefinition/>

</Grid.ColumnDefinitions>

<StackPanel Grid.Row="0" Grid.Column="0" Style="{StaticResource Gradient}">

<TextBox Style="{StaticResource SmallTextBox}" MouseEnter="TextBox\_MouseEnter" MouseLeave="TextBox\_MouseLeave"></TextBox>

<TextBox Style="{StaticResource SmallTextBox}" MouseEnter="TextBox\_MouseEnter" MouseLeave="TextBox\_MouseLeave"></TextBox>

<TextBox Style="{StaticResource SmallTextBox}" MouseEnter="TextBox\_MouseEnter" MouseLeave="TextBox\_MouseLeave"></TextBox>

<TextBox Style="{StaticResource SmallTextBox}" MouseEnter="TextBox\_MouseEnter" MouseLeave="TextBox\_MouseLeave"></TextBox>

</StackPanel>

<StackPanel Grid.Row="0" Grid.Column="1" Style="{StaticResource Gradient}">

<TextBox Style="{StaticResource SmallTextBox}" MouseEnter="TextBox\_MouseEnter" MouseLeave="TextBox\_MouseLeave"></TextBox>

<TextBox Style="{StaticResource SmallTextBox}" MouseEnter="TextBox\_MouseEnter" MouseLeave="TextBox\_MouseLeave"></TextBox>

<TextBox Style="{StaticResource SmallTextBox}" MouseEnter="TextBox\_MouseEnter" MouseLeave="TextBox\_MouseLeave"></TextBox>

<TextBox Style="{StaticResource SmallTextBox}" MouseEnter="TextBox\_MouseEnter" MouseLeave="TextBox\_MouseLeave"></TextBox>

</StackPanel>

</Grid>

</Grid>

</Window>

Результат выполнения:





Задание 3

Рассмотрите случай, когда для одного и того же элемента управления срабатывают сразу несколько триггеров, устанавливающих для одного и того же свойства различные значения, и определите правило, по которому определяется приоритет применения элементов Setter этих триггеров.

Код программы:

<Window x:Class="lab\_3\_task\_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:lab\_3\_task\_3"

mc:Ignorable="d"

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

<Window.Resources>

<Style TargetType="Button">

<Style.Triggers>

<DataTrigger Binding="{Binding ElementName=slider1, Path=Value}" Value="0">

<Setter Property="IsEnabled" Value="True"></Setter>

</DataTrigger>

<DataTrigger Binding="{Binding ElementName=slider2, Path=Value}" Value="0">

<Setter Property="IsEnabled" Value="False"></Setter>

</DataTrigger>

</Style.Triggers>

</Style>

</Window.Resources>

<StackPanel Orientation="Horizontal" VerticalAlignment="Top">

<Button>Открыть</Button>

<Button>Обработать</Button>

<Slider Height="23" x:Name="slider1" Width="100" />

<Slider Height="23" x:Name="slider2" Width="100" />

</StackPanel>

</Window>

Результат выполнения:







В приоритете будет тот триггер, который был прописан позже.

Например, если поменять местами 2 триггера, то получим следующее:

Код программы:

<Window x:Class="lab\_3\_task\_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:lab\_3\_task\_3"

mc:Ignorable="d"

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

<Window.Resources>

<Style TargetType="Button">

<Style.Triggers>

<DataTrigger Binding="{Binding ElementName=slider2, Path=Value}" Value="0">

<Setter Property="IsEnabled" Value="False"></Setter>

</DataTrigger>

<DataTrigger Binding="{Binding ElementName=slider1, Path=Value}" Value="0">

<Setter Property="IsEnabled" Value="True"></Setter>

</DataTrigger>

</Style.Triggers>

</Style>

</Window.Resources>

<StackPanel Orientation="Horizontal" VerticalAlignment="Top">

<Button>Открыть</Button>

<Button>Обработать</Button>

<Slider Height="23" x:Name="slider1" Width="100" />

<Slider Height="23" x:Name="slider2" Width="100" />

</StackPanel>

</Window>

Результат выполнения:







Задание 4

Разработайте WPF-приложение с двумя многострочными текстовыми полями, кнопками «Открыть», «Очистить», «Закрыть» и выпадающим списком для задания внешнего вида текстовых полей. Задайте для текстовых полей одинаковый градиентный фон. Кнопка «Закрыть» должна быть доступна только в том случае, если в обоих текстовых полях нет текста. Задайте для кнопок различный внешний вид при наведении курсора и при нажатии на них. Внешний вид текстовых полей (тип шрифта, размер шрифта,

цвет шрифта) должен меняться в зависимости от значения, выбранного в выпадающем списке.

Код программы:

MainWindow.xaml:

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

mc:Ignorable="d"

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

<Window.Resources>

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

<Setter Property="Margin" Value="5,0,5,0"/>

<Setter Property="Background" Value="#282829"/>

<Setter Property="BorderThickness" Value="0"/>

<Setter Property="Foreground" Value="DimGray"/>

<Style.Triggers>

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

<Trigger.Setters>

<Setter Property="Foreground" Value="Red" />

</Trigger.Setters>

</Trigger>

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

<Setter Property="Foreground" Value="AliceBlue"/>

</Trigger>

</Style.Triggers>

</Style>

<Style TargetType="Button" x:Key="CloseButton" BasedOn="{StaticResource Buttons}">

<Style.Triggers>

<MultiDataTrigger>

<MultiDataTrigger.Conditions>

<Condition Binding="{Binding ElementName=MainWindowTextBox1, Path=Text}" Value=""/>

<Condition Binding="{Binding ElementName=MainWindowTextBox2, Path=Text}" Value=""/>

</MultiDataTrigger.Conditions>

<MultiDataTrigger.Setters>

<Setter Property="IsEnabled" Value="True"/>

</MultiDataTrigger.Setters>

</MultiDataTrigger>

</Style.Triggers>

<Setter Property="IsEnabled" Value="False"/>

</Style>

<Style TargetType="ComboBox">

<Setter Property="Margin" Value="15,0,0,0"/>

<Setter Property="Width" Value="100px"/>

<Setter Property="Foreground" Value="DimGray"/>

</Style>

<Style TargetType="TextBox" x:Key="MainWindowTextBox">

<Setter Property="Control.Background">

<Setter.Value>

<LinearGradientBrush StartPoint="0,0" EndPoint="0,0.5">

<LinearGradientBrush.GradientStops>

<GradientStop Color="#282829" Offset="2"/>

<GradientStop Color="LightGray" Offset="1"/>

</LinearGradientBrush.GradientStops>

</LinearGradientBrush>

</Setter.Value>

</Setter>

<Style.Triggers>

<DataTrigger Binding="{Binding ElementName=FontStyles, Path=SelectedIndex}" Value="0">

<DataTrigger.Setters>

<Setter Property="Foreground" Value="DimGray"/>

<Setter Property="FontSize" Value="12"/>

</DataTrigger.Setters>

</DataTrigger>

<DataTrigger Binding="{Binding ElementName=FontStyles, Path=SelectedIndex}" Value="1">

<DataTrigger.Setters>

<Setter Property="Foreground" Value="DarkOrange"/>

<Setter Property="FontSize" Value="12"/>

</DataTrigger.Setters>

</DataTrigger>

<DataTrigger Binding="{Binding ElementName=FontStyles, Path=SelectedIndex}" Value="2">

<DataTrigger.Setters>

<Setter Property="Foreground" Value="LightBlue"/>

<Setter Property="FontSize" Value="12"/>

</DataTrigger.Setters>

</DataTrigger>

<DataTrigger Binding="{Binding ElementName=FontStyles, Path=SelectedIndex}" Value="3">

<DataTrigger.Setters>

<Setter Property="Foreground" Value="Indigo"/>

<Setter Property="FontSize" Value="12"/>

</DataTrigger.Setters>

</DataTrigger>

</Style.Triggers>

</Style>

<Style TargetType="TextBox" x:Key="FontSizeTextBox">

<Setter Property="Width" Value="100px"/>

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

<Setter Property="Background" Value="#282829"/>

<Setter Property="Foreground" Value="DimGray"/>

<Setter Property="BorderBrush" Value="DimGray"/>

</Style>

</Window.Resources>

<Grid Background="#282829">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="0.15\*"/>

<RowDefinition/>

<RowDefinition/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition/>

</Grid.ColumnDefinitions>

<DockPanel Grid.Row="0" Grid.Column="0" VerticalAlignment="Top" Margin="3">

<Button Style="{StaticResource Buttons}">Открыть</Button>

<Button Style="{StaticResource Buttons}" Click="Clear\_Button\_Click">Очистить</Button>

<ComboBox x:Name="FontStyles" SelectedIndex="0">

<ComboBoxItem>Обычный</ComboBoxItem>

<ComboBoxItem>Лава</ComboBoxItem>

<ComboBoxItem>Вода</ComboBoxItem>

<ComboBoxItem>Космос</ComboBoxItem>

</ComboBox>

<TextBox x:Name="FontSize1" Style="{StaticResource FontSizeTextBox}" Text="{Binding ElementName=MainWindowTextBox1, Path=FontSize}"/>

<TextBox x:Name="FontSize2" Style="{StaticResource FontSizeTextBox}" Text="{Binding ElementName=MainWindowTextBox2, Path=FontSize}"/>

<Button Style="{StaticResource CloseButton}" HorizontalAlignment="Right" Click="Close\_Button\_Click">Закрыть</Button>

</DockPanel>

<TextBox x:Name="MainWindowTextBox1" Grid.Row="1" Grid.Column="0" Style="{StaticResource MainWindowTextBox}" FontSize="{Binding ElementName=FontSize1, Path=Text}"/>

<TextBox x:Name="MainWindowTextBox2" Grid.Row="2" Grid.Column="0" Style="{StaticResource MainWindowTextBox}" FontSize="{Binding ElementName=FontSize2, Path=Text}"/>

</Grid>

</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 TextEditor

{

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

}

private void Clear\_Button\_Click(object sender, RoutedEventArgs e)

{

MainWindowTextBox1.Clear();

MainWindowTextBox2.Clear();

}

private void Close\_Button\_Click(object sender, RoutedEventArgs e)

{

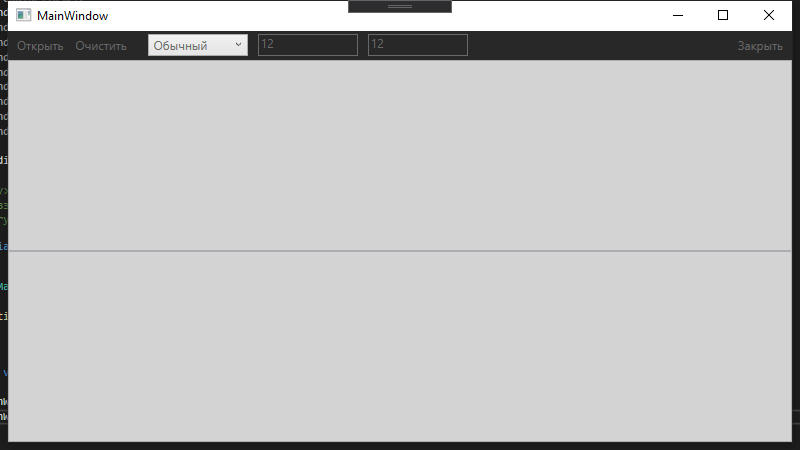
Close();

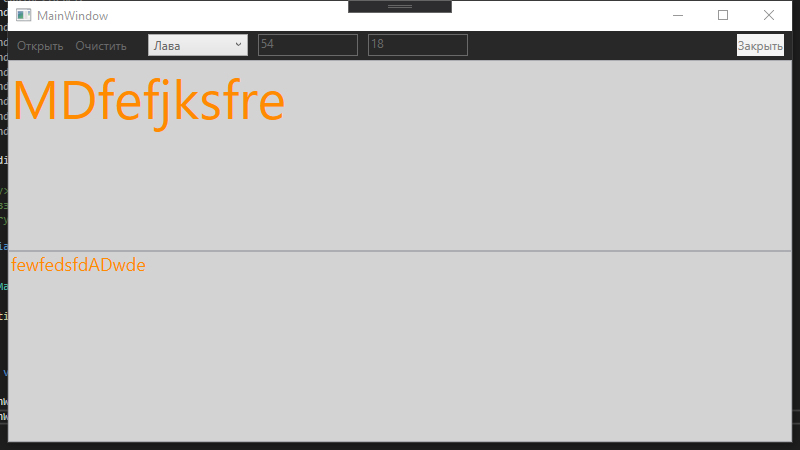
}

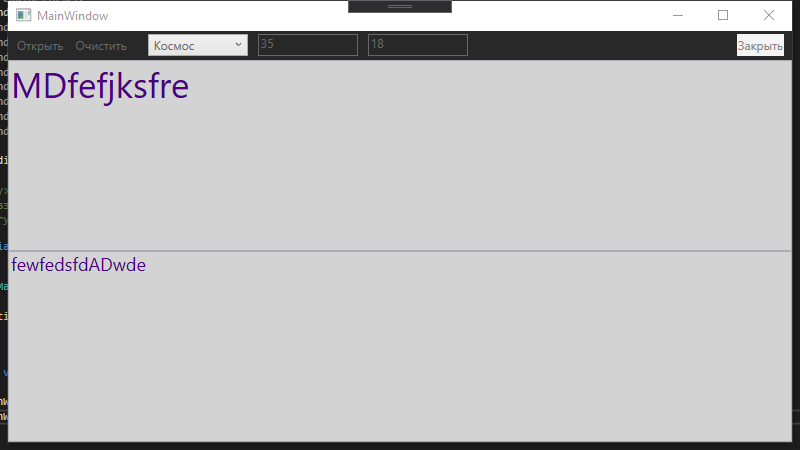
}

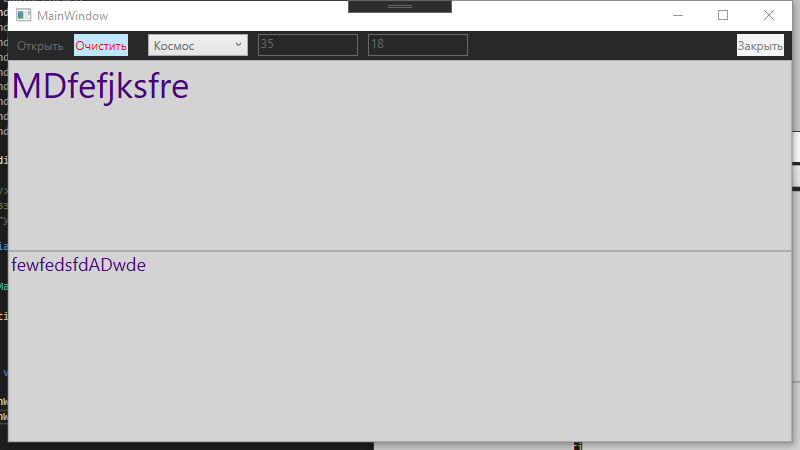
}

Результат выполнения:

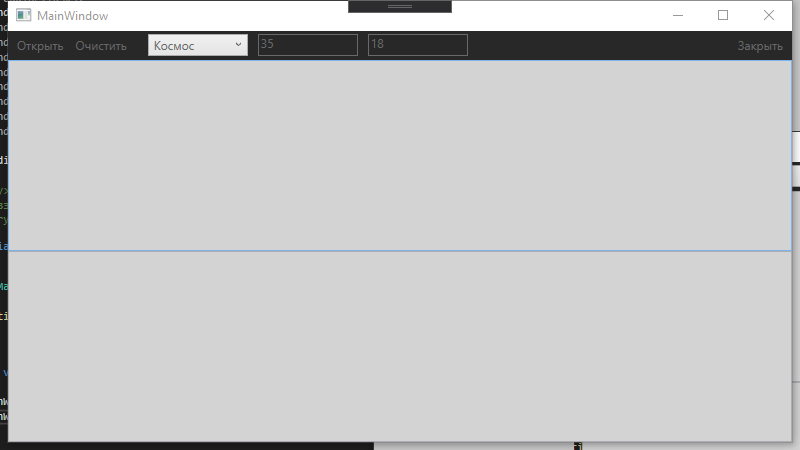








Очистим оба окна – кнопка “закрыть доступна”



Вывод: В ходе выполнения данной лабораторной работы я изучил триггеры и стили в графической системе WPF.