Лаб 9 Чупеев А.Д. Варинат 15

using System;  
using System.Collections.Generic;  
using System.Diagnostics.SymbolStore;  
using System.Linq;  
using System.Text.RegularExpressions;  
using System.Windows;  
using System.Windows.Controls;  
using System.Windows.Documents;  
using System.Windows.Input;  
  
namespace Task\_1  
{  
 */// <summary>*  
 */// Interaction logic for MainWindow.xaml*  
 */// </summary>*  
public partial class MainWindow  
 {  
 public MainWindow()  
 {  
 InitializeComponent();  
 \_comboBox = ComboBoxTitles;  
 \_groupBox = GroupBoxFont;  
 }  
  
 private UIElement \_comboBox;  
 private UIElement \_groupBox;  
 private void ComboBoxTitles\_OnSelectionChanged(object sender, SelectionChangedEventArgs e)  
 {  
 var comboBox = (ComboBox)sender;  
 var selectedItem = (ComboBoxItem)comboBox.SelectedItem;  
 Application.Current.MainWindow.Title = selectedItem.Content.ToString();  
 }  
   
 private void ToggleButton\_OnChecked(object sender, RoutedEventArgs e)  
 {  
 var radioButton = (RadioButton)sender;  
 Application.Current.MainWindow.FontSize = Convert.ToInt32(radioButton.Content.ToString());  
 if (Menu != null) Menu.FontSize = Convert.ToInt32(radioButton.Content.ToString());  
 }  
  
 private void ButtonBase\_OnClick\_Font(object sender, RoutedEventArgs e)  
 {  
 if (Grid.Children.Contains(\_groupBox))  
 {  
 Grid.Children.Remove(\_groupBox);  
 }  
 else  
 {  
 Grid.Children.Add(\_groupBox);  
 }  
 }  
  
 private void ButtonBase\_OnClick\_Title(object sender, RoutedEventArgs e)  
 {  
 if (Grid.Children.Contains(\_comboBox))  
 {  
 Grid.Children.Remove(\_comboBox);  
 }  
 else  
 {  
 Grid.Children.Add(\_comboBox);  
 }  
 }  
 private void NumberValidationTextBoxC(object sender, TextCompositionEventArgs e)  
 {  
 var C = (TextBox)sender;  
 Regex regex = new Regex("[^0-9]+");  
 e.Handled = regex.IsMatch(e.Text);  
 if (e.Text == " ") e.Handled = true;  
 if (e.Text == "-" && C.Text == "") e.Handled = false;  
 if (e.Text == "." && C.Text.Count(x => x == '.') == 0) e.Handled = false;  
 }  
  
 private void TextBox\_OnKeyDown(object sender, KeyEventArgs e)  
 {  
 var C = (TextBox)sender;  
 if (e.Key == Key.**Enter** && C.Text != "")  
 {  
 var lbi = new ListBoxItem();  
 lbi.Content = C.Text;  
 ListBox1.Items.Add(lbi);  
 C.Text = "";  
 }  
 else if (e.Key == Key.**Enter**)  
 {  
 MessageBox.Show("Вы ввели пустую строку");  
 }  
 }  
   
 private void TextBox\_OnKeyDown\_ContextMenu(object sender, KeyEventArgs e)  
 {  
 var C = (TextBox)sender;  
 if (e.Key == Key.**Enter** && C.Text != "")  
 {  
 var lbi = new ListBoxItem();  
 lbi.Content = C.Text;  
 ListBox3.Items.Add(lbi);  
 C.Text = "";  
 }  
 else if (e.Key == Key.**Enter**)  
 {  
 MessageBox.Show("Вы ввели пустую строку");  
 }  
 }  
   
 private void MenuItem\_OnClick\_Delete(object sender, RoutedEventArgs e)  
 {  
 var lbis = new List<ListBoxItem>();  
 foreach (var listBox1SelectedItem in ListBox1.SelectedItems)  
 {  
 lbis.Add((ListBoxItem)listBox1SelectedItem);  
 }  
 lbis.ForEach(x => ListBox1.Items.Remove(x));  
 lbis.Clear();  
 foreach (var listBox1SelectedItem in ListBox2.SelectedItems)  
 {  
 lbis.Add((ListBoxItem)listBox1SelectedItem);  
 }  
 lbis.ForEach(x => ListBox2.Items.Remove(x));  
 lbis.Clear();  
 foreach (var listBox1SelectedItem in ListBox3.SelectedItems)  
 {  
 lbis.Add((ListBoxItem)listBox1SelectedItem);  
 }  
 lbis.ForEach(x => ListBox3.Items.Remove(x));  
 }  
  
 private void MenuItem\_OnClick\_C(object sender, RoutedEventArgs e)  
 {  
 foreach (var listBox1Item in ListBox1.Items)  
 {  
 var lbi = (ListBoxItem)listBox1Item;  
 var num = Convert.ToDouble(lbi.Content,new System.Globalization.CultureInfo("en-US"));  
 if (Math.Truncate(num) % 2 == 0 && num > 0)  
 {  
 var lbiForLb2 = new ListBoxItem();  
 lbiForLb2.Content = lbi.Content;  
 ListBox2.Items.Add(lbiForLb2);  
 }  
 }  
 }  
  
 private void MenuItem\_OnClick\_E(object sender, RoutedEventArgs e)  
 {  
 foreach (var listBox1Item in ListBox1.Items)  
 {  
 var lbi = (ListBoxItem)listBox1Item;  
 var num = Convert.ToDouble(lbi.Content,new System.Globalization.CultureInfo("en-US"));  
 if (Math.Truncate(num) == 0 && num > 0)  
 {  
 var lbiForLb2 = new ListBoxItem();  
 lbiForLb2.Content = lbi.Content;  
 ListBox2.Items.Add(lbiForLb2);  
 }  
 }  
 }  
  
 private void MenuItem\_OnClick\_A(object sender, RoutedEventArgs e)  
 {  
 var k = 0;  
 foreach (var listBox3Item in ListBox3.Items)  
 {  
 var lbi = (ListBoxItem)listBox3Item;  
 if (!Regex.IsMatch(lbi.Content.ToString(),@"\d"))  
 {  
 k++;  
 }  
 }  
 MessageBox.Show($"Колл-во строк без чисел: {k}");  
 }  
 private void MenuItem\_OnClick\_D(object sender, RoutedEventArgs e)  
 {  
 var ans = "";  
 var nl = Environment.NewLine;  
 var i = 0;  
 foreach (var listBox3Item in ListBox3.Items)  
 {  
 var lbi = (ListBoxItem)listBox3Item;  
 if (i % 2 == 0)  
 {  
 ans += Regex.Replace(lbi.Content.ToString(), @" ", @"") + nl;  
 }  
 else  
 {  
 ans += Regex.Replace(lbi.Content.ToString(), @" ", @" ") + nl;  
 }  
 i++;  
 }  
 MessageBox.Show(ans);  
 }  
 private void MainWindow\_OnMouseLeftButtonDown(object sender, MouseButtonEventArgs e)  
 {  
 Application.Current.MainWindow.FontSize = 12;  
 Menu.FontSize = 12;  
 Application.Current.MainWindow.Title = "Изначальное название";  
 RadioButtonDefaultFont.IsChecked = true;  
 CheckBox\_comboBox.IsChecked = false;  
 CheckBox\_groupBox.IsChecked = false;  
 if (!Grid.Children.Contains(\_groupBox))  
 {  
 Grid.Children.Add(\_groupBox);  
 }  
   
 if (!Grid.Children.Contains(\_comboBox))  
 {  
 Grid.Children.Add(\_comboBox);  
 }  
 if (e.ClickCount >= 2) this.Close();  
 }  
 private void Window\_Close(object sender, ExecutedRoutedEventArgs e)  
 {  
 this.Close();  
 }  
 private void Window\_Start(object sender, ExecutedRoutedEventArgs e)  
 {  
 Application.Current.MainWindow.FontSize = 12;  
 Menu.FontSize = 12;  
 Application.Current.MainWindow.Title = "Изначальное название";  
 RadioButtonDefaultFont.IsChecked = true;  
 CheckBox\_comboBox.IsChecked = false;  
 CheckBox\_groupBox.IsChecked = false;  
 if (!Grid.Children.Contains(\_groupBox))  
 {  
 Grid.Children.Add(\_groupBox);  
 }  
   
 if (!Grid.Children.Contains(\_comboBox))  
 {  
 Grid.Children.Add(\_comboBox);  
 }  
 }  
  
 private void UIElement\_OnMouseMove(object sender, MouseEventArgs e)  
 {  
   
 StatusBar.Items.Clear();  
 if (sender.Equals(ComboBoxTitles))  
 {  
 StatusBar.Items.Add("ComboBox");  
 }  
 if (sender.Equals(GroupBoxFont))  
 {  
 StatusBar.Items.Add("GroupBox");  
 }  
  
 if (sender.Equals(StackPanelCheck))  
 {  
 StatusBar.Items.Add("StackPanel");  
 }  
  
 if (sender.Equals(Menu))  
 {  
 StatusBar.Items.Add("Menu");  
 }  
 StatusBar.Visibility = Visibility.**Visible**;  
 }  
  
 private void UIElement\_OnMouseLeave(object sender, MouseEventArgs e)  
 {  
 StatusBar.Visibility = Visibility.**Hidden**;  
 }  
 }  
}

<Window x:Class="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:Task\_1"  
 mc:Ignorable="d"  
 Background="Aqua"  
 FontSize="12"  
 MouseLeftButtonDown="MainWindow\_OnMouseLeftButtonDown"  
 Title="Изначальный название" Height="500" Width="550">  
 <Window.Resources>  
 <ResourceDictionary>  
 <RoutedUICommand x:Key="Close1"/>  
 <RoutedUICommand x:Key="Start1"/>  
 </ResourceDictionary>  
 </Window.Resources>  
 <Window.InputBindings>  
 <KeyBinding Gesture="ALT+X" Command="{StaticResource Close1}" />  
 <KeyBinding Gesture="ALT+I" Command="{StaticResource Start1}" />  
 </Window.InputBindings>  
 <Window.CommandBindings>  
 <CommandBinding Command="{StaticResource Close1}" Executed="Window\_Close" />  
 <CommandBinding Command="{StaticResource Start1}" Executed="Window\_Start" />  
 </Window.CommandBindings>  
 <Grid x:Name="Grid">  
 <ComboBox Width="200"   
 Height="30"   
 VerticalAlignment="**Top**"  
 FontSize="22"  
 Margin="10 30"  
 x:Name="ComboBoxTitles"  
 MouseMove="UIElement\_OnMouseMove"  
 MouseLeave="UIElement\_OnMouseLeave"  
 SelectionChanged="ComboBoxTitles\_OnSelectionChanged"  
 HorizontalAlignment="**Left**">  
 <ComboBoxItem Content="Название 1"></ComboBoxItem>  
 <ComboBoxItem Content="Название 2"></ComboBoxItem>  
 <ComboBoxItem Content="Название 3"></ComboBoxItem>  
 <ComboBoxItem Content="Название 4"></ComboBoxItem>  
 <ComboBoxItem Content="Название 5"></ComboBoxItem>  
 <ComboBoxItem Content="Название 6"></ComboBoxItem>  
 <ComboBoxItem Content="Название 7"></ComboBoxItem>  
 <ComboBoxItem Content="Название 8"></ComboBoxItem>  
 </ComboBox>  
 <GroupBox Header="Размер шрифта"   
 Padding="5"  
 x:Name="GroupBoxFont"  
 HorizontalAlignment="**Left**"  
 VerticalAlignment="**Top**"  
 Margin="10 60"  
 MouseMove="UIElement\_OnMouseMove"  
 MouseLeave="UIElement\_OnMouseLeave"  
 Width="200" Height="220">  
 <StackPanel>  
 <RadioButton  
 Margin="3"  
 Content="12"  
 IsChecked="True"  
 x:Name="RadioButtonDefaultFont"  
 Checked="ToggleButton\_OnChecked"></RadioButton>  
 <RadioButton Margin="3"   
 Checked="ToggleButton\_OnChecked"   
 Content="10"></RadioButton>  
 <RadioButton Margin="3"   
 Checked="ToggleButton\_OnChecked"  
 Content="8"></RadioButton>  
 <RadioButton Margin="3"  
 Checked="ToggleButton\_OnChecked"  
 Content="6"></RadioButton>  
 <RadioButton Margin="3"  
 Checked="ToggleButton\_OnChecked"  
 Content="4"></RadioButton>  
 <RadioButton Margin="3"  
 Checked="ToggleButton\_OnChecked"  
 Content="2"></RadioButton>  
 </StackPanel>  
 </GroupBox>  
 <StackPanel x:Name="StackPanelCheck"  
 Width="280"  
 Height="100"  
 Margin="10 30"  
 MouseMove="UIElement\_OnMouseMove"  
 MouseLeave="UIElement\_OnMouseLeave"  
 HorizontalAlignment="**Right**"  
 VerticalAlignment="**Top**">  
 <CheckBox Height="20" x:Name="CheckBox\_comboBox" Click="ButtonBase\_OnClick\_Title" Content="Заблакировать ComboBox" />  
 <CheckBox Height="20" x:Name="CheckBox\_groupBox" Click="ButtonBase\_OnClick\_Font" Content="Заблакирвоать GroupBox" />  
 </StackPanel>  
 <Menu VerticalAlignment="**Top**"  
 x:Name="Menu"  
 FontSize="12"  
 MouseMove="UIElement\_OnMouseMove"  
 MouseLeave="UIElement\_OnMouseLeave"  
 Height="20">  
 <MenuItem Header="ListBox1">  
 <ListBox SelectionMode="**Multiple**"  
 x:Name="ListBox1"  
 Height="60"  
 Width="70">  
 <ListBoxItem Content="1.0"></ListBoxItem>  
 <ListBoxItem Content="2.0"></ListBoxItem>  
 <ListBox.ContextMenu>  
 <ContextMenu>  
 <MenuItem Header="Удалить" Click="MenuItem\_OnClick\_Delete"></MenuItem>  
 </ContextMenu>  
 </ListBox.ContextMenu>  
 </ListBox>  
 </MenuItem>  
 <MenuItem Header="ListBox2">  
 <ListBox x:Name="ListBox2"  
 SelectionMode="**Multiple**"  
 Width="70"  
 Height="60"  
 Padding="0">  
 <ListBoxItem> 3.0 </ListBoxItem>  
 <ListBoxItem> 4.0 </ListBoxItem>  
 <ListBox.ContextMenu>  
 <ContextMenu>  
 <MenuItem Header="Удалить" Click="MenuItem\_OnClick\_Delete"></MenuItem>  
 </ContextMenu>  
 </ListBox.ContextMenu>  
 </ListBox>  
 </MenuItem>  
 <MenuItem Header="TextBox">  
 <TextBox Background="White"  
 PreviewTextInput="NumberValidationTextBoxC"  
 KeyDown="TextBox\_OnKeyDown"  
 Width="70">  
 </TextBox>  
 </MenuItem>  
 <MenuItem Click="MenuItem\_OnClick\_C"   
 Header="C">  
 </MenuItem>  
 <MenuItem Click="MenuItem\_OnClick\_E"  
 Header="E">  
 </MenuItem>  
 <MenuItem Header="Удалить"  
 Click="MenuItem\_OnClick\_Delete">  
 </MenuItem>  
 <Menu.ContextMenu>  
 <ContextMenu >  
 <MenuItem StaysOpenOnClick="True" Header="ListBox3">  
 <ListBox SelectionMode="**Multiple**"  
 x:Name="ListBox3"  
 HorizontalAlignment="**Left**"  
 VerticalAlignment="**Top**"  
 Width="70">  
 <ListBoxItem Content="5.0"></ListBoxItem>  
 <ListBoxItem Content="6.0"></ListBoxItem>  
 <ListBox.ContextMenu>  
 <ContextMenu>  
 <MenuItem Header="Удалить" Click="MenuItem\_OnClick\_Delete"></MenuItem>  
 </ContextMenu>  
 </ListBox.ContextMenu>  
 </ListBox>  
 </MenuItem>  
 <MenuItem StaysOpenOnClick="True" Header="TextBox3">  
 <TextBox Background="White"  
 HorizontalAlignment="**Right**"  
 VerticalAlignment="**Top**"  
 KeyDown="TextBox\_OnKeyDown\_ContextMenu"  
 Width="70">  
 </TextBox>  
 </MenuItem>  
 <MenuItem Header="Удалить" Click="MenuItem\_OnClick\_Delete"></MenuItem>  
 <MenuItem Header="A" Click="MenuItem\_OnClick\_A"></MenuItem>  
 <MenuItem Header="D" Click="MenuItem\_OnClick\_D"></MenuItem>  
 </ContextMenu>  
 </Menu.ContextMenu>  
 </Menu>  
 <StatusBar x:Name="StatusBar"   
 VerticalAlignment="**Bottom**"  
 Height="20"  
 Visibility="**Hidden**">  
 Здесь будет подсказка...  
 </StatusBar>  
 </Grid>  
</Window>