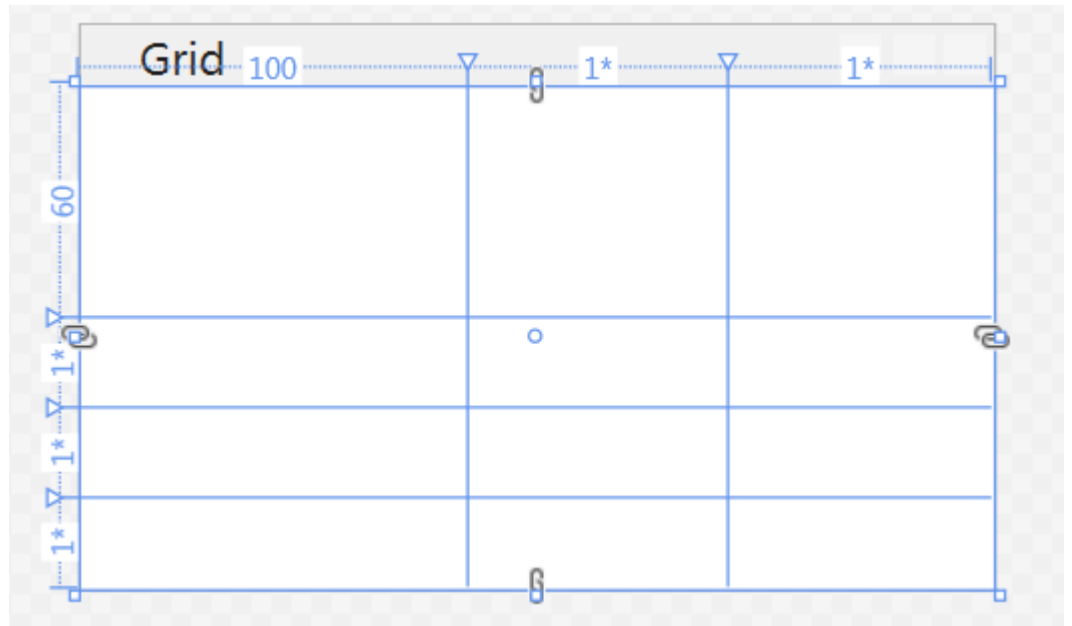


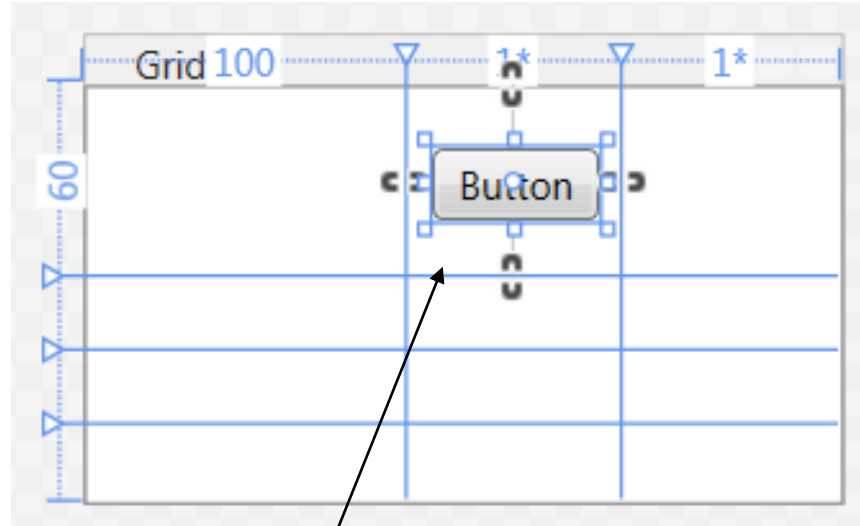
Windows Presentation Foundation

Grid

```
<Window x:Class="WpfApp3.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:WpfApp3"
  mc:Ignorable="d"
  Title="Grid" Height="145" Width="235">
  <Grid>
    <Grid.ColumnDefinitions>
      <ColumnDefinition Width="100"/>
      <ColumnDefinition />
      <ColumnDefinition />
    </Grid.ColumnDefinitions>
    <Grid.RowDefinitions>
      <RowDefinition Height="60"/>
      <RowDefinition />
      <RowDefinition />
      <RowDefinition />
    </Grid.RowDefinitions>
  </Grid>
</Window>
```

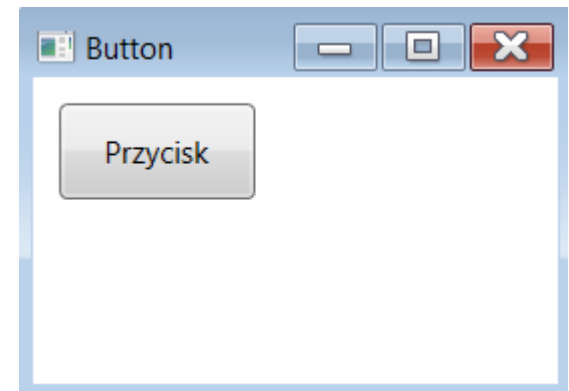
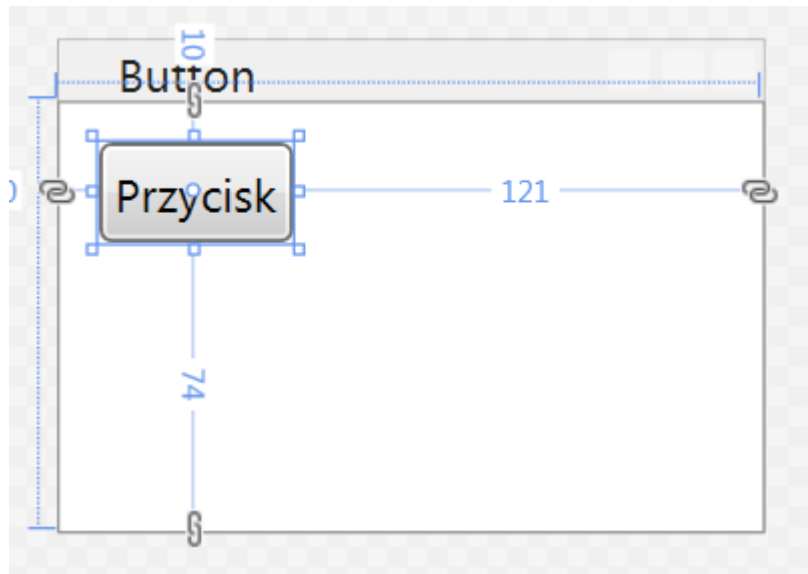


Grid



```
<Window x:Class="WpfApp3.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:WpfApp3"
  mc:Ignorable="d"
  Title="Grid" Height="145" Width="235">
  <Grid>
    <Grid.ColumnDefinitions>
      <ColumnDefinition Width="100"/>
      <ColumnDefinition />
      <ColumnDefinition />
    </Grid.ColumnDefinitions>
    <Grid.RowDefinitions>
      <RowDefinition Height="60"/>
      <RowDefinition />
      <RowDefinition />
      <RowDefinition />
    </Grid.RowDefinitions>
    <Button Content="Button" Grid.Row="0" Grid.Column="1" HorizontalAlignment="Center" VerticalAlignment="Center" Height="23" Width="52"/>
  </Grid>
</Window>
```

Button

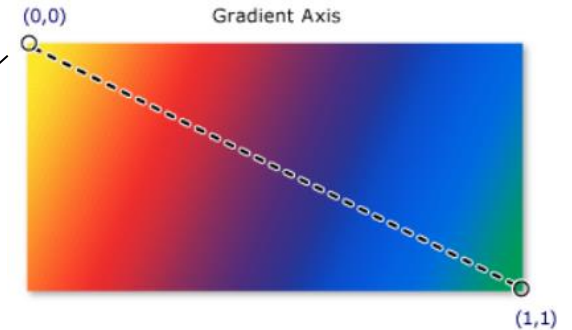


↓

```
<Grid>  
  <Button Content="Przycisk" Margin="10,10,121,74"/>  
</Grid>
```

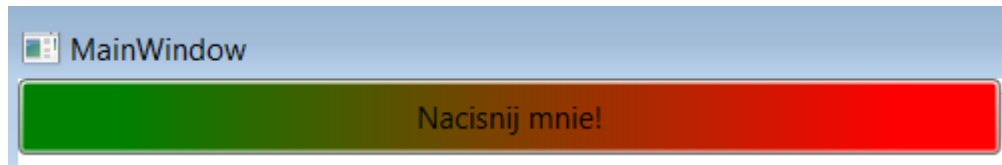
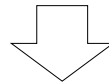
↗

Button



A diagonal linear gradient

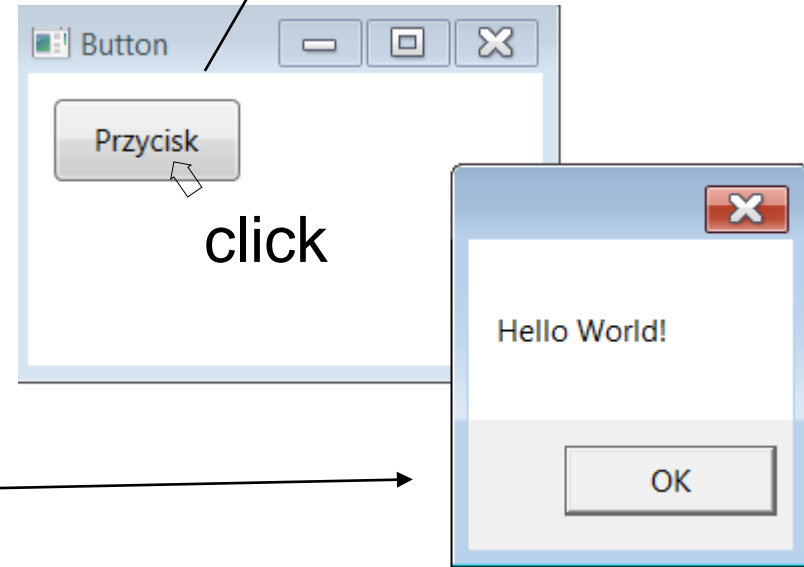
```
<Button>  
  <Button.Background>  
    <LinearGradientBrush StartPoint="0,0.4"  
      EndPoint="1,0.4">  
      <GradientStop Color="Red" Offset="0.9" />  
      <GradientStop Color="Green" Offset="0.1" />  
    </LinearGradientBrush>  
  </Button.Background>  
  Nacisnij mnie!  
</Button>
```



Button

```
<Grid>  
  <Button Content="Przycisk" Margin="10,10,121,74" Click="Button_Click"/>  
</Grid>
```

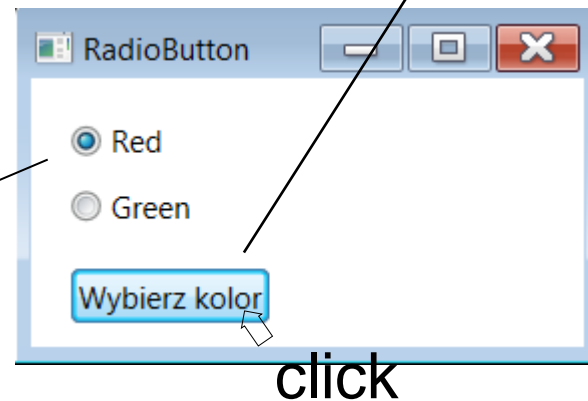
```
namespace WpfApp3  
{  
  /// <summary>  
  /// Interaction logic for MainWindow.xaml  
  /// </summary>  
  public partial class MainWindow : Window  
  {  
    public MainWindow()  
    {  
      InitializeComponent();  
    }  
  
    private void Button_Click(object sender, RoutedEventArgs e)  
    {  
      MessageBox.Show("Hello World!");  
    }  
  }  
}
```



RadioButton

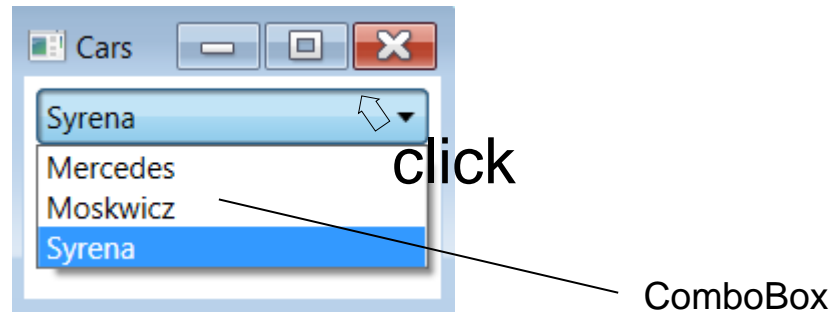
```
<Grid>  
  <RadioButton x:Name="btnRed" Content="Red" HorizontalAlignment="Left" Margin="16,17,0,0" VerticalAlignment="Top"/>  
  <RadioButton x:Name="btnGreen" Content="Green" HorizontalAlignment="Left" Margin="16,43,0,0" VerticalAlignment="Top"/>  
  <Button Content="Wybierz kolor" HorizontalAlignment="Left" Margin="16,76,0,0" VerticalAlignment="Top" Click="Button_Click"/>  
</Grid>
```

```
namespace WpfApp3  
{  
  /// <summary>  
  /// Interaction logic for MainWindow.xaml  
  /// </summary>  
  public partial class MainWindow : Window  
  {  
    public MainWindow()  
    {  
      InitializeComponent();  
    }  
  
    private void Button_Click(object sender, RoutedEventArgs e)  
    {  
      if (btnRed.IsChecked == true)  
      {  
        MessageBox.Show("Red");  
      }  
      else if (btnGreen.IsChecked == true)  
      {  
        MessageBox.Show("Green");  
      }  
    }  
  }  
}
```

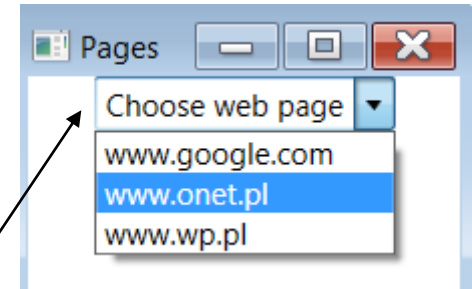


ComboBox

```
<Grid>  
  <StackPanel Margin="5">  
    <ComboBox>  
      <ComboBoxItem>Mercedes</ComboBoxItem>  
      <ComboBoxItem>Moskwicz</ComboBoxItem>  
      <ComboBoxItem IsSelected="True">Syrena</ComboBoxItem>  
    </ComboBox>  
  </StackPanel>  
</Grid>
```



ComboBox



```
<ComboBox x:Name = "cbWebPages" Width="120"
```

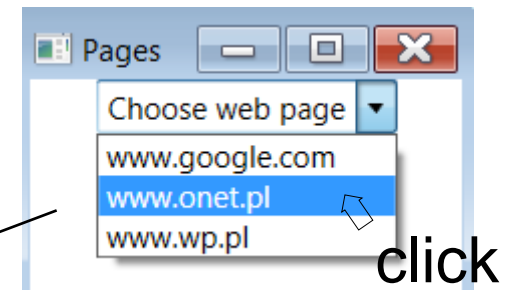
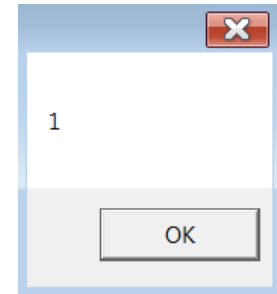
```
Focusable="False" IsEditable="True" IsReadOnly="True" Text = "Choose web page"
```

```
SelectionChanged="cbWebPages_SelectionChanged">
```

ComboBox

```
namespace WpfApp
{
    /// <summary>
    /// Interaction logic for MainWindow.xaml
    /// </summary>
    public partial class MainWindow : Window
    {
        public MainWindow()
        {
            InitializeComponent();
        }

        private void cbWebPages_SelectionChanged(object sender, EventArgs e)
        {
            MessageBox.Show(cbWebPages.SelectedIndex.ToString());
        }
    }
}
```

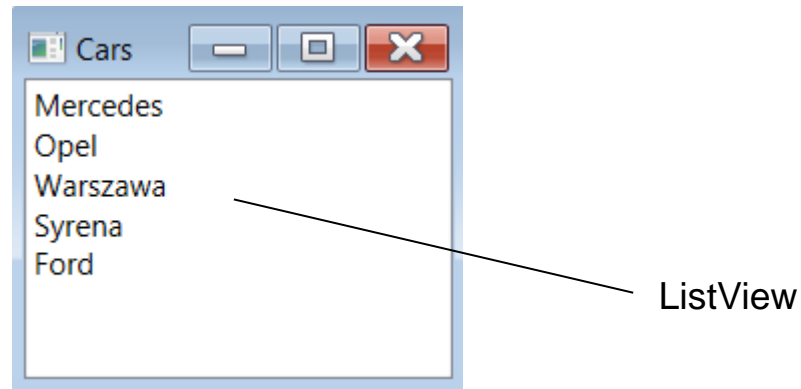


`SelectionChanged="cbWebPages_SelectionChanged">`

ComboBox

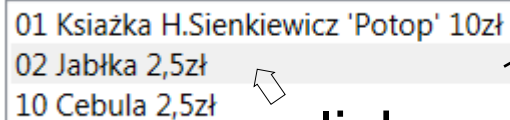
Listview

```
<Grid>  
  <ListView>  
    <ListViewItem Content="Mercedes"></ListViewItem>  
    <ListViewItem Content="Opel"></ListViewItem>  
    <ListViewItem Content="Warszawa"></ListViewItem>  
    <ListViewItem Content="Syrena"></ListViewItem>  
    <ListViewItem Content="Ford"></ListViewItem>  
  </ListView>  
</Grid>
```



ListView

```
<ListView x:Name="ItemsListView" Height="300" Width="400" PreviewMouseLeftButtonUp="listView_Click">
```

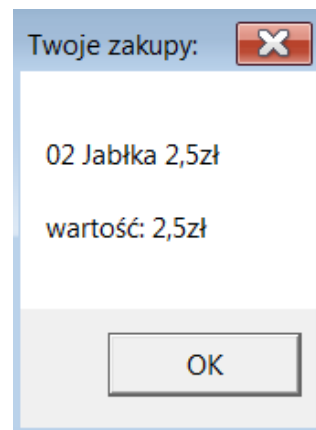


01 Książka H.Sienkiewicz 'Potop' 10zł
02 Jabłka 2,5zł
10 Cebula 2,5zł

click

ListView

```
// zdarzenie kliknięcia na wiersz ListView  
private void listView_Click(object sender, RoutedEventArgs e)  
{  
    // pobranie wiersza  
    var item = (sender as ListView).SelectedItem;  
    if (item != null)  
    {  
        // musimy znaleźć towar z listy towarów o danym Id  
        // (pobranym z klikniętego wiersza ListView)  
        // żeby dodać go do listy zakupów  
  
        // komunikat  
        MessageBox.Show(zakupioneTowary, "Twoje zakupy:");  
    }  
}
```



second Window

prawy myszy na nazwę projektu

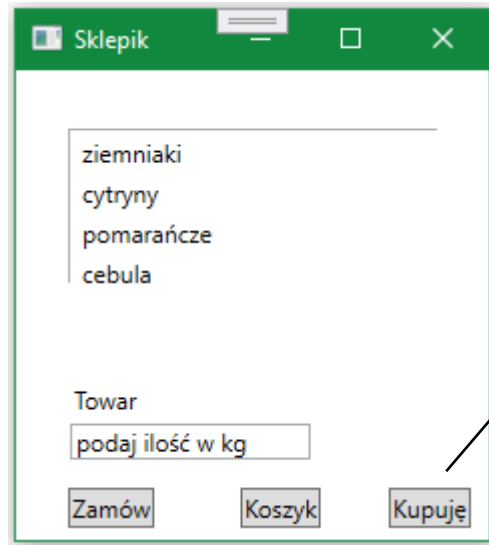
The image illustrates the steps to add a new window to a project in Visual Studio. It consists of three main parts:

- Top Right:** A screenshot of the Solution Explorer showing a project named "sklep1". A right-click context menu is open over the project name, with the "Dodaj" (Add) option highlighted. An arrow points from the text "prawy myszy na nazwę projektu" to this menu.
- Middle:** A screenshot of the "Dodaj nowy element" (Add New Item) dialog box. The "Okno (WPF)" (Window (WPF)) option is selected and circled in red. The "Typ: Elementy C#" (Type: C# Items) is also visible. An arrow points from the "Dodaj" option in the context menu to this dialog.
- Bottom Right:** A screenshot of the Solution Explorer showing the project structure. A new window named "Window1.xaml" has been added to the project, and it is circled in red. An arrow points from the "Dodaj" button in the dialog box to this new window.

Additional details from the screenshots include:

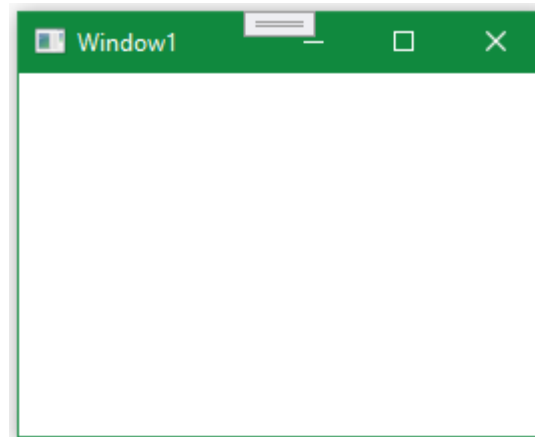
- The "Dodaj" context menu options: Kompiluj, Kompiluj ponownie, Wyczyść, Analiza i oczyszczanie kodu, Pakuj, Opublikuj..., Zawęż zakres do tego elementu, Nowy widok — Eksplorator rozwiązań, Edytuj plik projektu, Dodaj, Zarządzaj pakietami NuGet..., Zarządzaj kluczami tajnymi użytkownika, Usuń nieużywane odwołania..., Synchronizuj przestrzeń nazw, Ustaw jako projekt startowy, Debuguj, Wytnij (Ctrl+X), Usuń (Del), Zmień nazwę (F2), Zwolnij projekt, Załaduj zależności bezpośrednie, Załaduj całe drzewo zależności, Kopiuj pełną ścieżkę, Otwórz folder w Eksploratorze plików, Otwórz w terminalu, Zaprojektuj w programie Blend..., Właściwości (Alt+Enter).
- The "Dodaj nowy element" dialog categories: Zainstalowane, Sieć Web, Xamarin.Forms, Grafika, Online.
- The "Dodaj" button in the dialog box is also circled in red.

second Window



1 odwołanie

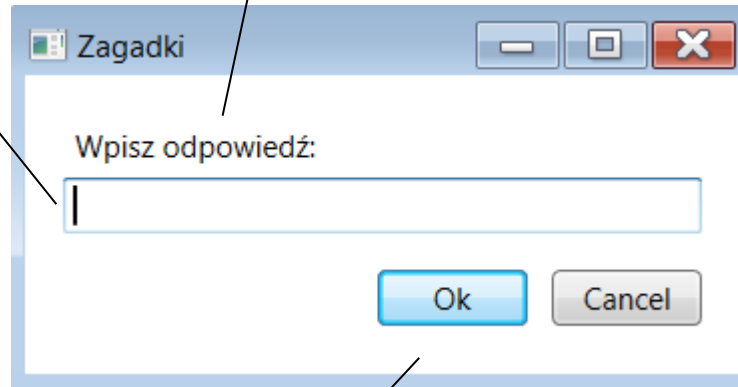
```
private void Button_Click(object sender, RoutedEventArgs e)
{
    Window1 win1 = new Window1();
    win1.Show();
}
```



InputBox

```
<TextBox Name="TextBoxOdpowiedz" Grid.Column="1" Grid.Row="1" MinWidth="250" />
```

```
<Label Grid.Column="1">Wpisz odpowiedź:</Label>
```



```
<Button IsDefault="True" Name="btnOk" Click="btnOk_Click" MinWidth="60" Margin="0,0,10,0">_Ok</Button>  
<Button IsCancel="True" MinWidth="60" Name="btnCancel" Click="btnCancel_Click">_Cancel</Button>
```

InputBox

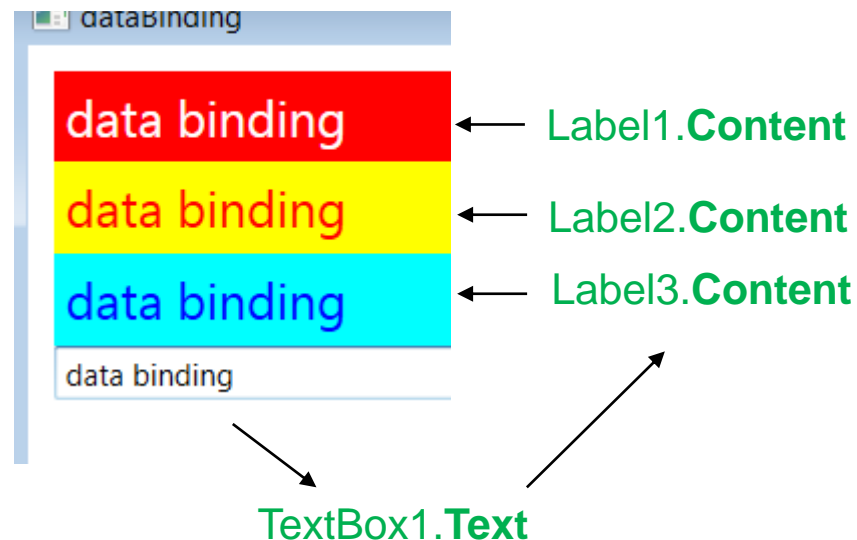
```
// obsługa przycisku OK
private void btnOk_Click(object sender, RoutedEventArgs e)
{
    // zapisujemy odpowiedź użytkownika w polu suma
    suma = TextBoxOdpowiedz.Text;
    // zamykamy okno
    this.Close();
}

// obsługa przycisku Cancel
private void btnCancel_Click(object sender, RoutedEventArgs e)
{
    // zamykamy okno
    this.Close();
}
```


Data binding (wiązanie danych)

```
<Grid>
  <!-- FocusManager - ustawienie focus na TextBox1 (kursor jest wewnątrz
       od momentu uruchomienia programu -->
  <StackPanel Margin="10" FocusManager.FocusedElement="{Binding ElementName=TextBox1}">
    <!-- bindujemy (wiążemy) atrybut Content Labela z atrybutem Text elementu TextBox1-->
    <Label x:Name="Label1" Content="{Binding Path=Text, ElementName=TextBox1}" />
    <Label x:Name="Label2" Content="{Binding Path=Text, ElementName=TextBox1}" />
    <Label x:Name="Label3" Content="{Binding Path=Text, ElementName=TextBox1}" />
    <TextBox x:Name="TextBox1" />
  </StackPanel>
</Grid>
```

Atrybut **Content** (każdego elementu Label) jest związany z atrybutem **Text** elementu TextBox1, czyli wszystko co wpiszemy w TextBox1 ukaże się w Labelach



Źródłem danych dla elementu Label jest element TextBox

Przydatne linki

<https://wpf-tutorial.com/> WPF tutorial (po polsku)

[How to put delay before doing an operation in WPF](#) wprowadzenie opóźnienia do programu