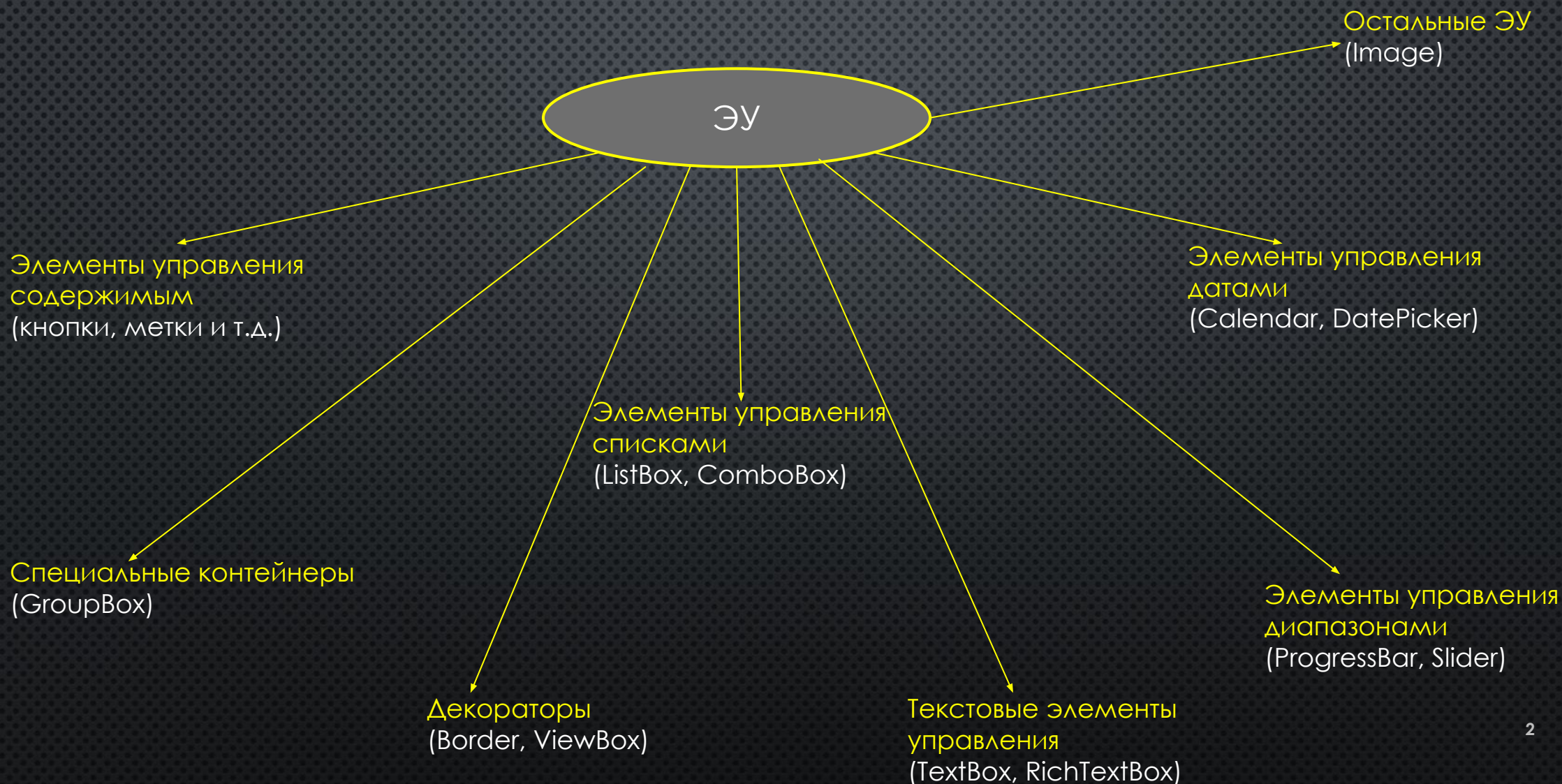


ЭЛЕМЕНТЫ УПРАВЛЕНИЯ

ГРУППЫ ЭЛЕМЕНТОВ



ОБЩИЕ СВОЙСТВА ЭУ

- **Name:** задать имя объекта для дальнейшего обращения к нему

```
<Button x:Name="button1" Width="60" Height="30" Content="Текст" Click="button1_Click" />
```

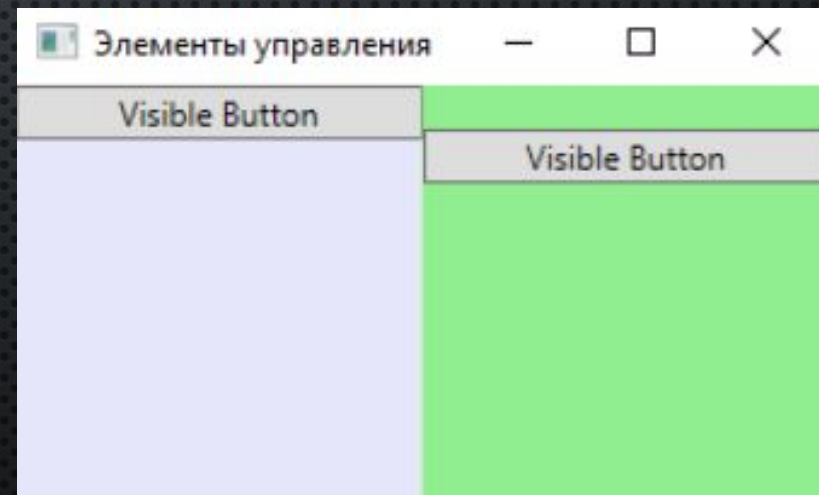
- **Visibility:** видимость объекта

Visible - элемент виден и участвует в компоновке.

Collapsed - элемент не виден и не участвует в компоновке.

Hidden - элемент не виден, но при этом участвует в компоновке

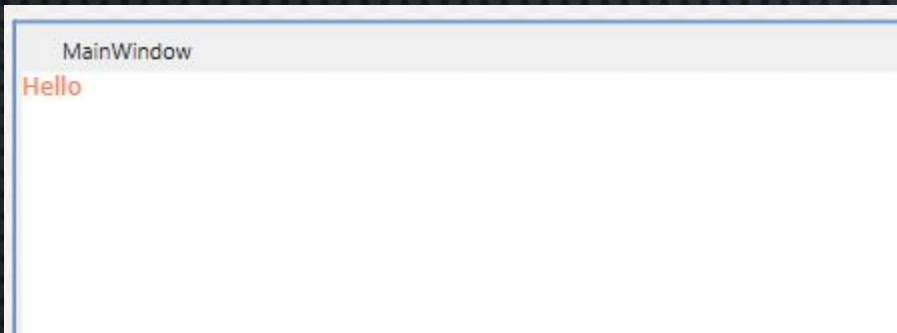
```
<Grid>
  <Grid.ColumnDefinitions>
    <ColumnDefinition Width="*" />
    <ColumnDefinition Width="*" />
  </Grid.ColumnDefinitions>
  <StackPanel Grid.Column="0" Background="Lavender">
    <Button Visibility="Collapsed" Content="Панель Collapsed" />
    <Button Height="20" Content="Visible Button" />
  </StackPanel>
  <StackPanel Grid.Column="1" Background="LightGreen">
    <Button Visibility="Hidden" Content="Панель Hidden" />
    <Button Height="20" Content="Visible Button" />
  </StackPanel>
</Grid>
```



ОБЩИЕ СВОЙСТВА ЭУ

- **FontFamily** - определяет семейство шрифта (например, **Arial, Verdana** и т.д.)
- **FontSize** - определяет высоту шрифта
- **FontStyle** - определяет наклон шрифта, принимает одно из трех значений - **Normal, Italic, Oblique**.
- **FontWeight** - определяет толщину шрифта и принимает ряд значений, как **Black, Bold** и др.

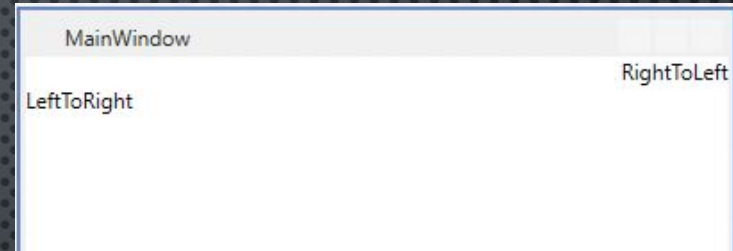
```
<Grid>  
  <TextBlock Text="Hello" FontFamily="Verdana" FontSize="13" Foreground="■"Coral"/>  
</Grid>
```



ОБЩИЕ СВОЙСТВА ЭУ

- **FlowDirection**: направление текста (слева-направо, справа-налево)

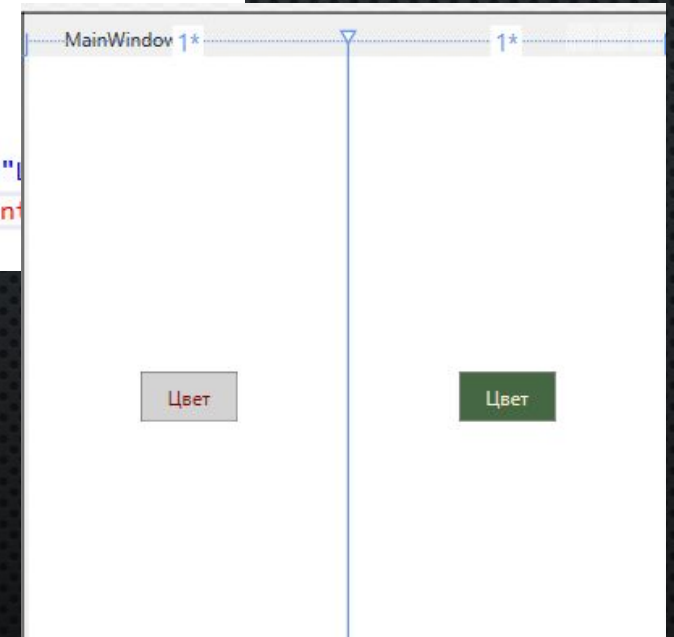
```
<Grid>
  <StackPanel>
    <TextBlock FlowDirection="RightToLeft">RightToLeft</TextBlock>
    <TextBlock FlowDirection="LeftToRight">LeftToRight</TextBlock>
  </StackPanel>
</Grid>
```



- **Background** и **Foreground**: цвет фона и текста элемента управления.

```
<Grid>
  <Grid.ColumnDefinitions>
    <ColumnDefinition></ColumnDefinition>
    <ColumnDefinition></ColumnDefinition>
  </Grid.ColumnDefinitions>
  <Button Grid.Column="0" Width="60" Height="30" Background="LightGray" Foreground="DarkRed" Content="Цвет"></Button>
  <Button Grid.Column="1" Width="60" Height="30" Background="#456743" Foreground="AntiqueWhite" Content="Цвет"></Button>
</Grid>
```

```
button1.Background = new SolidColorBrush(Colors.Red);
button1.Foreground = new SolidColorBrush(Color.FromRgb(0,255, 0));
```



BUTTON

- **IsDefault** и **IsCancel**: нажатие на клавиши Enter и Esc

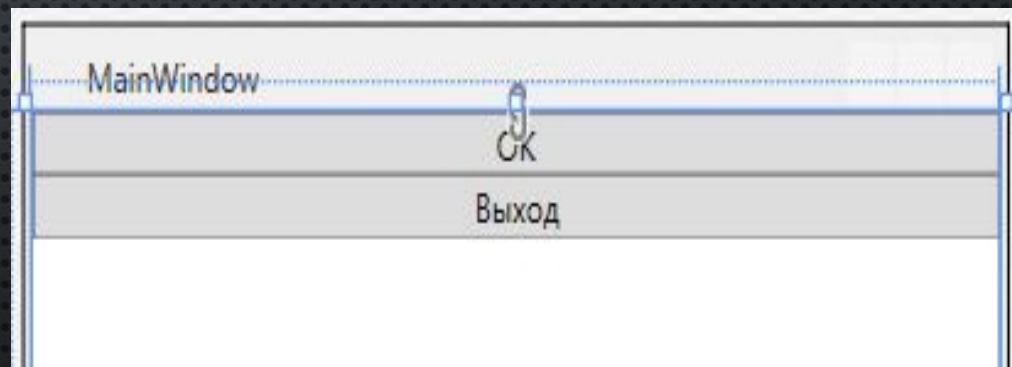
```
<StackPanel>
  <Button x:Name="acceptButton" Content="OK" IsDefault="True" Click="acceptButton_Click" />
  <Button x:Name="escButton" Content="Выход" IsCancel="True" Click="escButton_Click" />
</StackPanel>
```

```
using System.Windows;

namespace ControlsApp
{
    public partial class MainWindow : Window
    {
        public MainWindow()
        {
            InitializeComponent();
        }

        private void acceptButton_Click(object sender, RoutedEventArgs e)
        {
            MessageBox.Show("Действие выполнено");
        }

        private void escButton_Click(object sender, RoutedEventArgs e)
        {
            this.Close(); // закрытие окна
        }
    }
}
```

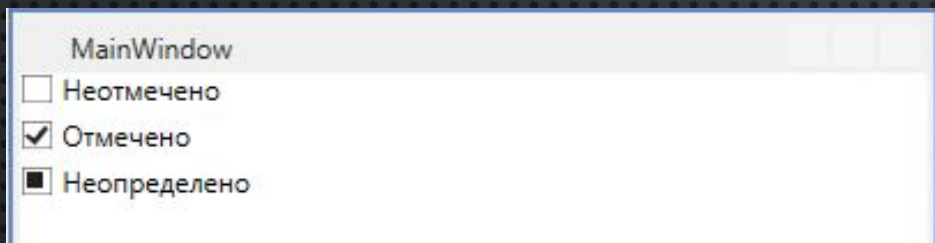


CHECKBOX

- **IsChecked**: состояние флажка (отмечен (true), не отмечен (false), неопределенно (x:Null))

```
<StackPanel x:Name="stackPanel">
    <CheckBox x:Name="checkBox1" IsThreeState="True" IsChecked="False" Height="20" Content="Неотмечено" />
    <CheckBox x:Name="checkBox2" IsThreeState="True" IsChecked="True" Height="20" Content="Отмечено" />
    <CheckBox x:Name="checkBox3" IsThreeState="True" IsChecked="{x:Null}" Height="20" Content="Неопределено"/>
</StackPanel>
```

- **IsThreeState**: свойство указывает, что флажок может находиться в трех состояниях



События **Checked**, **Unchecked** и **Intermediate**.


```
<StackPanel x:Name="stackPanel">
    <CheckBox x:Name="checkBox" IsChecked="False" Height="20" Content="Флажок" IsThreeState="True" Unchecked="checkBox_Unchecked"
Indeterminate="checkBox_Indeterminate" Checked="checkBox_Checked" />
```

```
private void checkBox_Unchecked(object sender, RoutedEventArgs e)
{
    MessageBox.Show(checkBox.Content.ToString() + " не отмечен");
}
```


RADIOBUTTON

- **IsChecked**: состояние флажка (отмечен (true), не отмечен(false))
- **GroupName**: объединение элементов по группам

```
<StackPanel x:Name="stackPanel">
  <RadioButton GroupName="Languages" Content="C#" IsChecked="True" />
  <RadioButton GroupName="Languages" Content="VB.NET" />
  <RadioButton GroupName="Languages" Content="C++" />
  <RadioButton GroupName="Technologies" Content="WPF" IsChecked="True" />
  <RadioButton GroupName="Technologies" Content="WinForms" />
  <RadioButton GroupName="Technologies" Content="ASP.NET" />
</StackPanel>
```



```
-----MainWindow-----
 C#
 VB.NET
 C++
 WPF
 WinForms
 ASP.NET
```

Программное добавление RadioButton

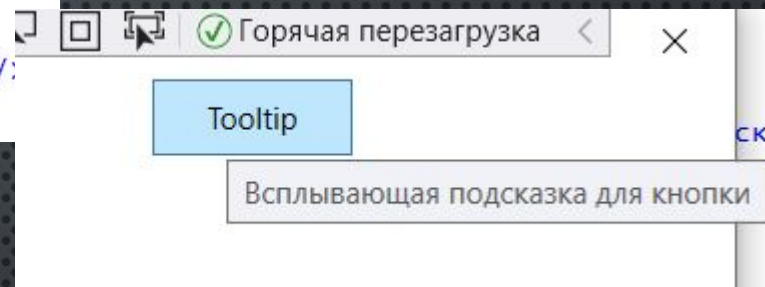
```
public MainWindow()
{
    InitializeComponent();

    RadioButton rb = new RadioButton { IsChecked = true, GroupName = "Languages", Content = "JavaScript" };
    stackPanel.Children.Add(rb);
}
```


ВСПЛЫВАЮЩИЕ ПОДСКАЗКИ TOOLTIP

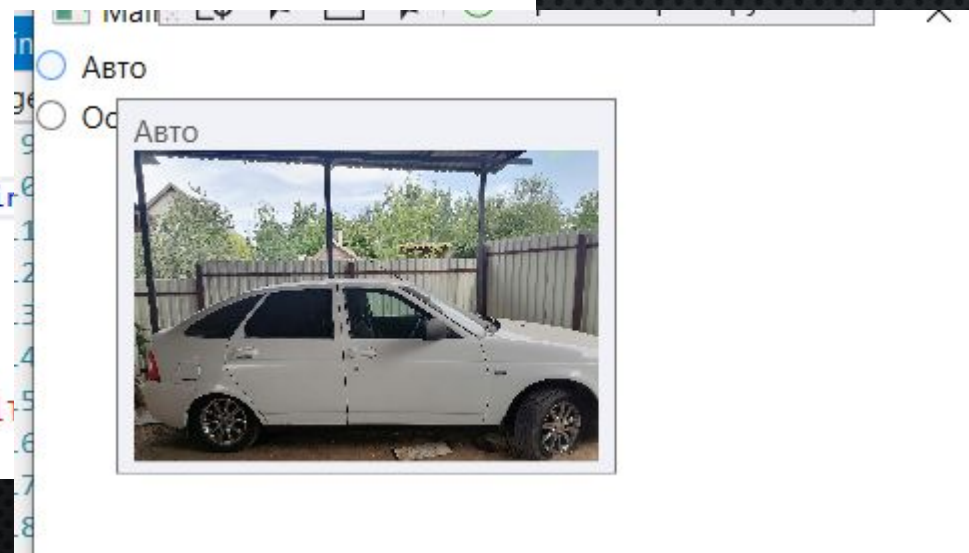
Элемент **ToolTip** представляет всплывающую подсказку при наведении на какой-нибудь элемент.

```
<StackPanel x:Name="stackPanel">  
    <Button Content="Tooltip" ToolTip="Всплывающая подсказка для кнопки" Height="30" Width="80" />  
</StackPanel>
```



СЛОЖНАЯ ПОДСКАЗКА

```
<StackPanel>  
    <RadioButton GroupName="Photos" Content="Авто" Height="20">  
        <RadioButton.ToolTip>  
            <ToolTip Width="200" Height="150">  
                <StackPanel>  
                    <TextBlock Text="Авто" />  
                    <Image Source="C:\Users\zairo\source\repos\Mycop\Mycop\bir" />  
                </StackPanel>  
            </ToolTip>  
        </RadioButton.ToolTip>  
    </RadioButton>  
    <RadioButton GroupName="Photos" Content="Остальные фото" Height="20" ToolTip="..." />  
</StackPanel>
```



ВСПЛЫВАЮЩИЕ ПОДСКАЗКИ TOOLTIP

Программное создание подсказки при наведении на кнопку:

```
<Button x:Name="button1" Content="Hello" />
```

```
public MainWindow()  
{  
    InitializeComponent();  
    Tooltip tooltip = new Tooltip();  
    StackPanel tooltipPanel = new StackPanel();  
    tooltipPanel.Children.Add(new TextBlock { Text = "Заголовок", FontSize = 16 });  
    tooltipPanel.Children.Add(new TextBlock { Text = "Текст" });  
    tooltip.Content = tooltipPanel;  
    button1.Tooltip = tooltip;  
}
```