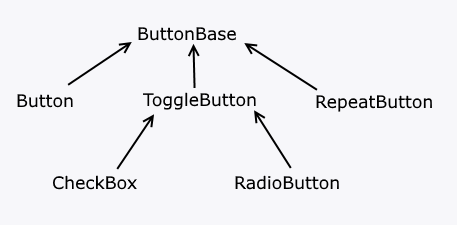
## Кнопки

 WPF кнопки представлены целым рядом классов, которые наследуются от базового класса ButtonBase:



### Button

Элемент Button представляет обычную кнопку:

|  |  |
| --- | --- |
| 1 | <Button x:Name="button1" Width="60" Height="30" Background="LightGray" /> |

От класса ButtonBase кнопка наследует ряд событий, например, Click, которые позволяют обрабатывать пользовательский ввод.

Чтобы связать кнопку с обработчиком события нажатия, нам надо определить в самой кнопке атрибут Click. А значением этого атрибута будет название обработчика в коде C#. А затем в самом коде C# определить этот обработчик.

Например, код xaml:

|  |  |
| --- | --- |
| 1 | <Button x:Name="button1" Width="60" Height="30" Content="Нажать" Click="Button\_Click" /> |

И обработчик в коде C#:

|  |  |
| --- | --- |
| 1  2  3  4 | private void Button\_Click(object sender, RoutedEventArgs e)  {      MessageBox.Show("Кнопка нажата");  } |

Либо можно не задавать обработчик через атрибут, а стандартным образом для C# прописать в коде: button1.Click+=Button\_Click;

К унаследованным свойствам кнопка имеет такие свойства как **IsDefault** и **IsCancel**, которые принимают значения true и false.

Если свойство IsDefault установлено в true, то при нажатии клавиши Enter будет вызываться обработчик нажатия этой кнопки.

Аналогично если свойство IsCancel будет установлено в true, то при нажатии на клавишу Esc будет вызываться обработчик нажатия этой кнопки.

Например, определим код xaml:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13 | <Window x:Class="ControlsApp.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:ControlsApp"          mc:Ignorable="d"          Title="Элементы управления" Height="250" Width="300">      <StackPanel>          <Button x:Name="acceptButton" Content="ОК" IsDefault="True" Click="acceptButton\_Click" />          <Button x:Name="escButton" Content="Выход" IsCancel="True" Click="escButton\_Click" />      </StackPanel>  </Window> |

А в коде MainWindow.xaml.cs определим следующий код C#:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22 | 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          }      }  } |

Теперь при нажатии на клавишу Enter будет отображаться сообщение, а при нажатии на Esc будет происходить выход из приложения и закрытие окна.

### RepeatButton

Отличительная особенность элемента RepeatButton - непрерывная генерация события Click, пока нажата кнопка. Интервал генерации события корректируется свойствами **Delay** и **Interval**.

Сам по себе элемент RepeatButton редко используется, однако он может служить основой для создания ползунка в элементах ScrollBar и ScrollViewer, в которых нажатие на ползунок инициирует постоянную прокрутку.

### ToggleButton

Представляет элементарный переключатель. Может находиться в трех состояниях - true, false и "нулевом" (неотмеченном) состоянии, а его значение представляет значение типа bool? в языке C#. Состояние можно установить или получить с помощью свойства **IsChecked**. Также добавляет три события - **Checked** (переход в отмеченное состояние), **Unchecked** (снятие отметки) и **Intermediate** (если значение равно null). Чтобы отрабатывать все три события, надо установить свойство **IsThreeState="True"**

ToggleButton, как правило, сам по себе тоже редко используется, однако при этом он служит основой для создания других более функциональных элементов, таких как checkbox и radiobutton.