



WPF

Урок №2

Работа с элементами управления

Содержание

1. Image.
2. Slider.
3. ComboBox.
4. ListBox.
5. Expander.
6. ProgressBar.
7. ScrollBar.
8. TextBlock.
9. RichTextBox.
10. TreeView.
11. ListView.
12. Примеры использования элементов управления.

1. Image

WPF предоставляет богатый набор базовых элементов управления. Для начала рассмотрим элемент управления Image. Он находится в наборе Common. Класс Image позволяет загружать и отображать изображения следующих типов: BMP, GIF, ICO, JPG, PNG, WDP и TIFF.

Анимация многокадровых изображений элементом управления Image не поддерживается. При отображении многокадрового изображения отображается только первый кадр.



До тех пор пока содержимое изображения не загружено, `ActualWidth` и `ActualHeight` элемента управления будут считаться как ноль, потому что содержимое изображений используется для определения окончательного размера и расположения элемента управления. Для элемента управления фиксированного размера могут быть заданы свойства `Width` и/или `Height`. Однако для сохранения пропорций носителя задайте свойство `Width` или `Height`, но не оба одновременно. Если задано одно свойство, то пропорции изображения сохраняются. В противном случае изображение может оказаться растянутым или перекошенным. Для управления растяжением изображения используются свойства `Stretch` и `StretchDirection`.

Источник изображения задается свойством `Source`. Есть несколько способов загрузить картинку в элемент управления.

Способ 1. Добавьте файл изображения в проект. Откройте свойства элемента управления. Отредактируйте свойство **Source**. Впишите относительный путь файла внутри проекта. Также можно вписать абсолютный путь файла и не добавлять его в проект.

Способ 2. Откройте код XAML. Найдите нужный элемент `Image` и вставьте в него свойство `Source` и впишите путь к файлу изображения.

```
<!-- Вариант 1 -->
<Image Margin="49,34,29,59" Name="image1" Stretch="Fill"
ImageFailed="image1_ImageFailed" Source="file:///d:/t.jpg" />
<!-- Вариант 2 -->
<Image Margin="49,34,29,59" Name="image2" Stretch="Fill"
ImageFailed="image1_ImageFailed" Source="d.jpg" />
<!-- Вариант 3 -->
<Image Name="image3" Width="200">
    <Image.Source>
        <BitmapImage DecodePixelWidth="200"
        UriSource="file:///d:/t.jpg" />
    </Image.Source>
</Image>
```



Способ 3. Также можно изменить загруженное изображение в коде. Для этого нужно создать объект `BitmapImage`, загрузить в него файл и присвоить этот объект свойству `Source`.

```
image3.Width = 20;

BitmapImage myBitmapImage = new BitmapImage();
myBitmapImage.BeginInit();
myBitmapImage.UriSource = new Uri(@"file:///d:/t.jpg");
myBitmapImage.DecodePixelWidth = 20;
myBitmapImage.EndInit();

image3.Source = myBitmapImage;
```

Рассмотрим основные члены класса `Image`.

Методы

Метод	Описание
<code>PointFromScreen</code>	Преобразовывает объект <code>Point</code> в экранных координатах в объект <code>Point</code> , представляющий текущую систему координат <code>Visual</code> . (Унаследовано от <code>Visual</code> .)
<code>PointToScreen</code>	Преобразовывает объект <code>Point</code> , представляющий текущую систему координат объекта <code>Visual</code> , в объект <code>Point</code> в экранных координатах. (Унаследовано от <code>Visual</code> .)
<code>BeginInit</code>	Запускает процесс инициализации данного элемента. (Унаследовано от <code>FrameworkElement</code> .)
<code>EndInit</code>	Указывает на завершение процесса инициализации элемента. (Унаследовано от <code>FrameworkElement</code> .)

Свойства

Свойство	Описание
<code>ActualHeight</code>	Возвращает визуализированную высоту данного элемента. Это свойство зависимостей. (Унаследовано от <code>FrameworkElement</code> .)
<code>ActualWidth</code>	Возвращает визуализированную ширину данного элемента. Это свойство зависимостей. (Унаследовано от <code>FrameworkElement</code> .)
<code>BaseUri</code>	Получает или задает базу URI (uniform resource identifier – универсальный код ресурса) для <code>Image</code>
<code>Height</code>	Возвращает или задает предполагаемую высоту элемента. Это свойство зависимостей. (Унаследовано от



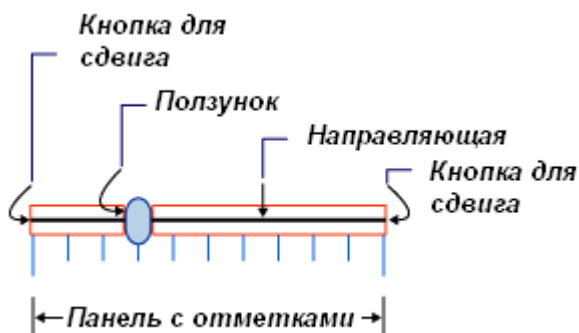
Свойство	Описание
	FrameworkElement.)
Source	Получает или задает ImageSource для изображения. Это свойство зависимости.
Stretch	Получает или задает значение, описывающее, как Image должен быть растянут, чтобы заполнить конечный прямоугольник. Это свойство зависимости.
StretchDirection	Получает или задает значение, показывающее, как масштабировано изображение. Это свойство зависимости.
Width	Возвращает или задает ширину элемента. Это свойство зависимостей. (Унаследовано от FrameworkElement.)

События

Событие	Описание
ImageFailed	Происходит в случае сбоя в изображении.

2. Slider

Элемент управления Slider позволяет пользователям выбрать значение из диапазона значений. Следующая иллюстрация представляет пример элемента управления Slider.



Можно настроить элемент управления Slider, настроив его свойства. Следующий список описывает некоторые атрибуты Slider, которые можно настроить:

Orientation – Ориентация Slider, может быть горизонтальной (Horizontal) или вертикальной (Vertical).

Delay – Возвращает или задает время ожидания в миллисекундах для кнопок повтора в нажатом состоянии перед выполнением команды



по перемещению ползунка, такой как команда DecreaseLarge. Это свойство зависимостей.

Interval – Возвращает или задает время в миллисекундах между командами повышения или понижения, когда пользователь нажимает RepeatButton Slider. Это свойство зависимостей.

Maximum – Возвращает или задает максимальное возможное значение Value элемента диапазона. Это свойство зависимостей.

Minimum – Возвращает или задает минимально возможное значение Value элемента диапазона. Это свойство зависимостей.

Value – Возвращает или задает текущую величину элемента управления диапазоном. Это свойство зависимостей.

Ползунок можно создать и изменить его свойства с помощью средств визуального проектирования или используя XAML.

```
<Slider Height="21" Name="slider1" VerticalAlignment="Bottom"
Margin="0,0,25,0" />

<Slider Name="slider2" Orientation="Vertical"
HorizontalAlignment="Right" Width="23" Margin="0,0,0,25" />
```

Следующий пример показывает, как создать Slider с TickBar, который отображает метки деления. Свойства TickPlacement и TickFrequency определяют расположение меток делений и интервал между ними. При перемещении Thumb, всплывающая подсказка отображает значение Slider. Свойство AutoToolTipPlacement определяет позицию, в которой появляется подсказка. Передвижения Thumb (ползунка) соответствуют расположению меток делений, так как значение IsSnapToTickEnabled установлено в true.

```
<Slider Name="slider1" Width="100"

Orientation="Horizontal" HorizontalAlignment="Left"

IsSnapToTickEnabled="True" Minimum="1" Maximum="9"

TickPlacement="BottomRight" TickFrequency="3"
```



```
AutoToolTipPrecision="2" AutoToolTipPlacement="BottomRight"

IsDirectionReversed="False"

IsMoveToPointEnabled="False"/>
```

Рассмотрим основные члены класса Slider.

Свойства

Свойство	Описание
Delay	Возвращает или задает время ожидания в миллисекундах для RepeatButton в нажатом состоянии перед выполнением команды по перемещению Thumb, такой как команда DecreaseLarge. Это свойство зависимостей.
Interval	Возвращает или задает время в миллисекундах между командами повышения или понижения, когда пользователь нажимает RepeatButton Slider. Это свойство зависимостей.
IsDirectionReversed	Возвращает или задает направление увеличения значения. Это свойство зависимостей.
IsMoveToPointEnabled	Возвращает или задает значение, которое показывает, передвигается ли Thumb Slider немедленно в положение щелчка мыши, который сделан при наведении курсора на дорожку Slider. Это свойство зависимостей.
IsSelectionRangeEnabled	Возвращает или задает значение, которое показывает, отображает ли Slider диапазон выбора вдоль Slider. Это свойство зависимостей.
IsSnapToTickEnabled	Возвращает или задает значение, которое показывает, перемещает ли Slider автоматически Thumb к ближайшей метке-галочке. Это свойство зависимостей.
LargeChange	Возвращает или задает значение для добавления или вычитания из Value элемента управления RangeBase. Это свойство зависимостей. (Унаследовано от RangeBase.)
MaximizeValue	Возвращает команду, которая задает для свойства Value объекта Slider значение Maximum.
Maximum	Возвращает или задает максимальное возможное значение Value элемента диапазона. Это свойство зависимостей. (Унаследовано от RangeBase.)
Minimum	Возвращает или задает минимальное возможное значение Value элемента диапазона. Это свойство зависимостей. (Унаследовано от RangeBase.)
MinimizeValue	Получает команду, устанавливающую Slider Value на значение Minimum.
Orientation	Возвращает или задает направление Slider. Это свой-



Свойство	Описание
	ство зависимостей.
SelectionEnd	Возвращает или задает наибольшее значение указанного выбора для Slider. Это свойство зависимостей.
SelectionStart	Возвращает или задает наименьшее значение указанного выбора для Slider. Это свойство зависимостей.
SmallChange	Возвращает или задает Value для добавления или вычитания из Value элемента управления RangeBase. Это свойство зависимостей. (Унаследовано от RangeBase.)
TickFrequency	Возвращает или задает интервал между метками. Это свойство зависимостей.
TickPlacement	Возвращает или задает позицию меток относительно Track Slider. Это свойство зависимостей.
Ticks	Возвращает или задает позицию меток для отображения для Slider. Это свойство зависимостей.
Value	Возвращает или задает текущую величину элемента управления диапазона. Это свойство зависимостей. (Унаследовано от RangeBase.)

События

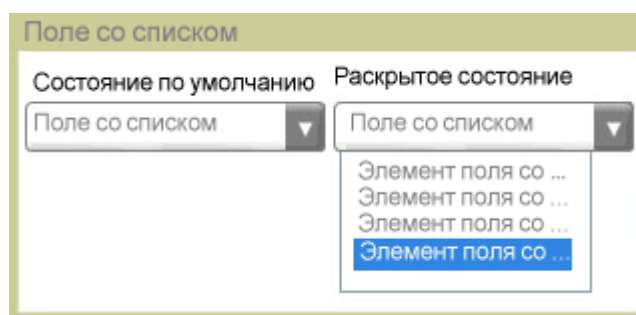
Событие	Описание
ValueChanged	Генерируется при изменении значения в диапазоне. (Унаследовано от RangeBase.)



3. ComboBox

ComboBox элемент управления, предназначенный для выбора значения, с раскрывающимся списком, который можно отображать и скрывать нажатием принадлежащей этому элементу управления кнопки со стрелкой. Элемент управления ComboBox позволяет пользователю выбирать элемент в раскрывающемся списке, а при желании также вводить новый текст в текстовое поле этого элемента управления.

На следующем рисунке показан ComboBox в разных состояниях.



Свойства `IsEditable` и `IsReadOnly` определяют поведение элемента управления ComboBox при выполнении пользователем одного из следующих действий:

- ввод строки для выбора элемента в ComboBox;
- ввод строки, не соответствующей ни одному из элементов ComboBox;
- выделение части строки в текстовом поле;
- копирование значения из текстового поля или вставка значения в текстовое поле.

Если `IsReadOnly` имеет значение `true` и `IsEditable` имеет значение `true` то:

- Выбор элемента в ComboBox путем ввода строки невозможен.
- Ввод строки, не соответствующей ни одному из элементов ComboBox, невозможен.
- Можно выделять часть строки в текстовом поле ComboBox.



- Можно скопировать строку из текстового поля элемента управления ComboBox, однако вставка строки в текстовое поле элемента управления ComboBox невозможна.

Если `IsReadOnly` имеет значение `true` и `IsEditable` имеет значение `false` то:

- Можно выбрать элемент в ComboBox путем ввода строки.
- Ввод строки, не соответствующей ни одному из элементов ComboBox, невозможен.
- Выделение части строки в текстовом поле ComboBox невозможно.
- Копирование или вставка строки в текстовом поле ComboBox невозможны.

Если `IsReadOnly` имеет значение `false` и `IsEditable` имеет значение `false` то:

- Можно выбрать элемент в ComboBox путем ввода строки.
- Ввод строки, не соответствующей ни одному из элементов ComboBox, невозможен.
- Выделение части строки в текстовом поле ComboBox невозможно.
- Копирование или вставка строки в текстовом поле ComboBox невозможны.

Если `IsReadOnly` имеет значение `true` и `IsEditable` имеет значение `true` то:

- Выбор элемента в ComboBox путем ввода строки невозможен.
- Ввод строки, не соответствующей ни одному из элементов ComboBox, невозможен.
- Можно выделять часть строки в текстовом поле ComboBox.
- Можно скопировать строку из текстового поля элемента управления ComboBox, однако вставка строки в текстовое поле элемента управления ComboBox невозможна.

Элемент управления ComboBox является элементом управления `ItemsControl`, поэтому он может отображать не только простые строки. Например, можно создать элемент управления ComboBox, содержащий



список изображений. При наличии в элементе управления ComboBox содержимого, отличного от строк, в элементе управления ComboBox может появиться бессмысленная строка, когда его раскрывающийся список скрыт. Для отображения строки в элементе управления ComboBox, содержащем нестроковые элементы, используйте вложенное свойство зависимостей `TextSearch...:Text` или `TextSearch...:TextPath`.

Рассмотрим основные члены класса `ComboBox`.

Методы

Метод	Описание
<code>PointFromScreen</code>	Преобразовывает объект <code>Point</code> в экранных координатах в объект <code>Point</code> , представляющий текущую систему координат <code>Visual</code> . (Унаследовано от <code>Visual</code> .)

Свойства

Свойство	Описание
<code>HasItems</code>	Получает значение, указывающее содержит ли <code>ItemsControl</code> элементы. Это свойство зависимостей. (Унаследовано от <code>ItemsControl</code> .)
<code>HandlesScrolling</code>	Возвращает значение, указывающее, поддерживает ли поле со списком прокрутку.
<code>IsDropDownOpen</code>	Возвращает или задает значение, указывающее, открыта ли в данный момент раскрывающаяся часть поля со списком. Это свойство зависимостей.
<code>IsEditable</code>	Возвращает или задает значение, разрешающее либо запрещающее редактировать текст в текстовом поле элемента управления <code>ComboBox</code> . Это свойство зависимостей.
<code>IsSelectionBoxHighlighted</code>	Возвращает значение, указывающее, выделен ли <code>SelectionBoxItem</code> .
<code>IsTextSearchEnabled</code>	Возвращает или задает значение, указывающее включен ли <code>TextSearch</code> в экземпляре <code>ItemsControl</code> . Это свойство зависимостей. (Унаследовано от <code>ItemsControl</code> .)
<code>ItemBindingGroup</code>	Возвращает или задает объект <code>BindingGroup</code> , который копируется в каждый элемент в объекте <code>ItemsControl</code> . (Унаследовано от <code>ItemsControl</code> .)
<code>ItemContainerStyle</code>	Возвращает или задает <code>Style</code> , применяемый к элементу контейнера, сгенерированному для каждого элемента. Это свойство зависимостей. (Унаследовано от <code>ItemsControl</code> .)
<code>ItemContainerStyleSelector</code>	Возвращает или задает пользовательскую логику выбо-



Свойство	Описание
	ра стиля для стиля, который может быть применен к каждому сгенерированному элементу контейнера. Это свойство зависимостей. (Унаследовано от ItemsControl.)
Items	Получает коллекцию, используемую для генерирования содержимого ItemsControl. (Унаследовано от ItemsControl.)
ItemsPanel	Возвращает или задает шаблон, определяющий панель, управляющую размещением элементов. Это свойство зависимостей. (Унаследовано от ItemsControl.)
ItemsSource	Возвращает или задает коллекцию, используемую для генерирования содержимого ItemsControl. Это свойство зависимостей. (Унаследовано от ItemsControl.)
ItemStringFormat	Возвращает или задает составную строку, которая определяет форматирование элементов в объектах ItemsControl, если они отображаются как строки. (Унаследовано от ItemsControl.)
ItemTemplate	Возвращает или задает DataTemplate, используемый для отображения каждого элемента. Это свойство зависимостей. (Унаследовано от ItemsControl.)
ItemTemplateSelector	Возвращает или задает пользовательскую логику для выбора шаблона, используемого для отображения каждого элемента. Это свойство зависимостей. (Унаследовано от ItemsControl.)
MaxDropDownHeight	Возвращает или задает максимальную высоту раскрываемой части поля со списком. Это свойство зависимостей.
SelectedIndex	Возвращает или задает индекс первого элемента в текущем выделении или минус единицу (-1), если выделение является пустым. Это свойство зависимостей. (Унаследовано от Selector.)
SelectedItem	Возвращает или задает первый элемент в текущем выделении или null, если выделение является пустым. Это свойство зависимостей. (Унаследовано от Selector.)
SelectedValue	Возвращает или задает значение SelectedItem, получаемое посредством SelectedValuePath. Это свойство зависимостей. (Унаследовано от Selector.)
SelectionBoxItem	Возвращает элемент, отображаемый в поле выбора. Это свойство зависимостей.
StaysOpenOnEdit	Возвращает или задает значение, указывающее, останется ли раскрываемый список элемента управления



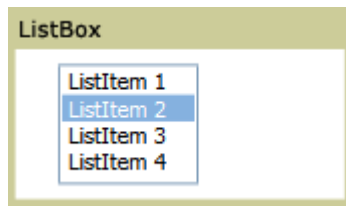
Свойство	Описание
	ComboBox открытым после того, как пользователь щелкнет вложенный элемент управления TextBox. Это свойство зависимостей.
Text	Возвращает или задает текст выделенного в данный момент элемента. Это свойство зависимостей.

События

Событие	Описание
SelectionChanged	Возникает при изменении текущего выделения в Selector. (Унаследовано от Selector.)

4. ListBox

Элемент управления ListBox представляет собой список выбираемых элементов. На следующем рисунке показан стандартный элемент управления ListBox.



ListBox является ItemsControl. Свойствами его содержимого являются Items и ItemsSource. Объект ListBox является элементом управления, содержащим коллекцию элементов. В списке ListBox видимы несколько элементов, в отличие от раскрывающегося списка ComboBox, в котором видим только один выбранный элемент, если только свойство IsDropDownOpen не установлено в true. Свойство SelectionMode определяет, можно ли выделять в списке ListBox несколько элементов одновременно.

Свойство SelectionMode определяет, сколько элементов можно выбрать одновременно. Можно установить это свойство в значения:

Single (Можно выбрать одновременно только один элемент.);

Multiple (Можно выбрать несколько элементов, не удерживая клавишу модификатора.);



Extended (Можно выбрать несколько последовательных элементов, удерживая клавишу SHIFT, или непоследовательных элементов, удерживая клавишу CTRL и щелкая элементы.).

Рассмотрим пример создания ListBox:

```
<ListBox Name="lb" Width="100" Height="55" SelectionChanged="PrintText"
SelectionMode="Single">
  <ListBoxItem>Item 1</ListBoxItem>
  <ListBoxItem>Item 2</ListBoxItem>
  <ListBoxItem>Item 3</ListBoxItem>
  <ListBoxItem>Item 4</ListBoxItem>
  <ListBoxItem>Item 5</ListBoxItem>
  <ListBoxItem>Item 6</ListBoxItem>
  <ListBoxItem>Item 7</ListBoxItem>
  <ListBoxItem>Item 8</ListBoxItem>
  <ListBoxItem>Item 9</ListBoxItem>
  <ListBoxItem>Item 10</ListBoxItem>
</ListBox>
```

В примере создается ListBox с набором элементов. Для него задаются конкретные размеры, одиночный режим выбора и подписывается обработчик PrintText на событие изменения выбранного элемента SelectionChanged. Обработчик может выглядеть примерно так

```
void PrintText(object sender, SelectionChangedEventArgs args)
{
    ListBoxItem lbi = ((sender as ListBox).SelectedItem as ListBoxItem);
    this.Text = "    You selected " + lbi.Content.ToString() + ".";
}
```

Тут содержание выбранного элемента выводится в заголовок окна. Рассмотрим основные члены класса ListBox.

Методы

Метод	Описание
PointFromScreen	Преобразовывает объект Point в экранных координатах в объект Point, представляющий текущую систему координат Visual. (Унаследовано от Visual.)
PointToScreen	Преобразовывает объект Point, представляющий текущую систему координат объекта Visual, в объект



Метод	Описание
	Point в экранных координатах. (Унаследовано от Visual.)
ScrollIntoView	Прокручивает объект, пока он не появится в представлении.
SelectAll	Выбирает все элементы в ListBox.
SetSelectedItems	Устанавливает коллекцию выбранных элементов списка.
ToString	Предоставляет строковое представление объекта ItemsControl. (Унаследовано от ItemsControl.)
UnselectAll	Отменяет весь выбор в ListBox.

Свойства

Свойство	Описание
AlternationCount	Возвращает или задает количество чередующихся контейнеров элементов в объекте ItemsControl, который позволяет чередующимся контейнерам иметь уникальный внешний вид. (Унаследовано от ItemsControl.)
GroupStyle	Получает коллекцию объектов GroupStyle, определяющих отображение каждого уровня групп. (Унаследовано от ItemsControl.)
GroupStyleSelector	Возвращает или задает метод, позволяющий предоставить пользовательскую логику выбора для GroupStyle для применения к каждой группе в коллекции. Это свойство зависимостей. (Унаследовано от ItemsControl.)
HandlesScrolling	Получает значение, указывающее, поддерживает ли список ListBox прокрутку. (Переопределяет Control...:HandlesScrolling.)
HasItems	Получает значение, указывающее содержит ли ItemsControl элементы. Это свойство зависимостей. (Унаследовано от ItemsControl.)
IsGrouping	Получает значение, указывающее использует ли данный элемент управления группирование. Это свойство зависимостей. (Унаследовано от ItemsControl.)
IsTextSearchEnabled	Возвращает или задает значение, указывающее включен ли TextSearch в экземпляре ItemsControl. Это свойство зависимостей. (Унаследовано от ItemsControl.)
ItemBindingGroup	Возвращает или задает объект BindingGroup, который копируется в каждый элемент в объекте ItemsControl. (Унаследовано от ItemsControl.)
Items	Получает коллекцию, используемую для генерирования содержимого ItemsControl. (Унаследовано от ItemsControl.)
ItemsSource	Возвращает или задает коллекцию, используемую для



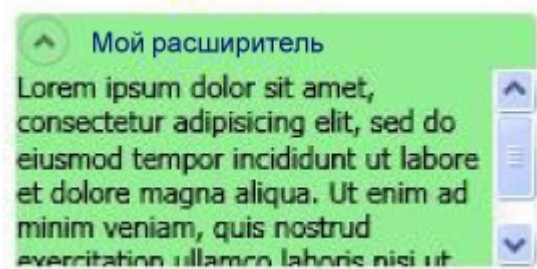
Свойство	Описание
	генерирования содержимого ItemsControl. Это свойство зависимостей. (Унаследовано от ItemsControl.)
SelectedIndex	Возвращает или задает индекс первого элемента в текущем выделении или минус единицу (-1), если выделение является пустым. Это свойство зависимостей. (Унаследовано от Selector.)
SelectedItem	Возвращает или задает первый элемент в текущем выделении или null, если выделение является пустым. Это свойство зависимостей. (Унаследовано от Selector.)
SelectedItems	Получает выбранные в данное время элементы. Это свойство зависимостей.
SelectedValue	Возвращает или задает значение SelectedItem, получаемое посредством SelectedValuePath. Это свойство зависимостей. (Унаследовано от Selector.)
SelectionMode	Возвращает или задает поведение выбора для списка ListBox. Это свойство зависимостей.

События

Событие	Описание
SelectionChanged	Возникает при изменении текущего выделения в Selector. (Унаследовано от Selector.)

5. Expander

Элемент управления Expander обеспечивает способ предоставления содержимого в разворачиваемой области, имеющей вид окна и включающей заголовок.



В следующем примере показано, как создать простой элемент управления Expander с текстовым блоком внутри.

```
<Expander Name="myExpander" Background="Tan"
          HorizontalAlignment="Left" Header="My Expander">
```



```
ExpandDirection="Down" IsExpanded="True" Width="100">
<TextBlock TextWrapping="Wrap">
    Lorem ipsum dolor sit amet, consectetur
    adipisicing elit, sed do eiusmod tempor incididunt ut
    labore et dolore magna aliqua
</TextBlock>
</Expander>
```

Свойства Content и Header элемента управления Expander могут также включать сложное содержимое, например объекты RadioButton и Image.

Можно задать содержимое области элемента управления Expander для развертывания в одном из четырех направлений (Down, Up, Left или Right), используя свойство ExpandDirection. Когда область содержимого свернута, отображается только Expander Header и его выключатель. Элемент управления Button, отображающий стрелку направления, используется в качестве выключателя для развертывания или свертывания области содержимого. При развертывании Expander пытается отобразить все содержимое в области, подобной окну.

Если элемент управления Expander находится внутри элемента управления макетом, наследуемого из Panel, например StackPanel, не указывайте свойство Height в элементе Expander, если для свойства ExpandDirection задано значение Down или Up. Так же не указывайте свойство Width в элементе Expander, если для свойства ExpandDirection задано значение Left или Right.

Если задается размерность элемента управления Expander в направлении отображения развернутого содержимого, то Expander принимает управление областью, которая используется содержимым, и отображает рамку вокруг нее. Рамка отображается даже тогда, когда содержимое свернуто. Чтобы установить размер развернутой области содержимого, задайте размерности для содержимого Expander или, если требуется возможность прокрутки, для элемента ScrollViewer, вмещающего содержимое.

Когда элемент управления Expander является последним элементом в DockPanel, Windows Presentation Foundation (WPF) автоматически уста-



навливает размерности равными оставшейся области Expander DockPanel. Чтобы избежать такой настройки по умолчанию, задайте для свойства LastChildFill в объекте DockPanel значение false или убедитесь, что Expander не является последним элементом в DockPanel.

Если содержимое слишком велико для размера области содержимого, то можно поместить содержимое Expander в элемент ScrollViewer, чтобы обеспечить прокрутку содержимого. Элемент управления Expander автоматически не предоставляет возможность прокрутки. На следующем рисунке показан элемент управления Expander, содержащий элемент управления ScrollViewer.

При помещении элемента управления Expander в элемент управления ScrollViewer задайте свойство размерности ScrollViewer, соответствующее направлению, в котором открывается содержимое Expander, по размеру области содержимого Expander. Например, если для свойства ExpandDirection в элементе управления Expander задается значение Down (область содержимого открывается вниз), то нужно задать для свойства Height в элементе управления ScrollViewer требуемую высоту для области содержимого. Если вместо этого задать размерность высоты самого содержимого, то ScrollViewer не распознает этот параметр и, таким образом, не предоставит прокручиваемое содержимое.

В следующем примере показано, как создать элемент управления Expander программно, который имеет сложное содержимое и содержит элемент управления ScrollViewer.

```
void MakeExpander()
{
    StackPanel sp = new StackPanel();
    Grid.SetRow(sp, 0);
    Grid.SetColumn(sp, 1);
    sp.Background = Brushes.LightSalmon;

    TextBlock colTitle = new TextBlock();
    colTitle.Text = "EXPANDER CREATED FROM CODE";
    colTitle.HorizontalAlignment = HorizontalAlignment.Center;
    colTitle.Margin.Bottom.Equals(20);
}
```



```

sp.Children.Add(colTitle);

Expander exp = new Expander();

BulletDecorator bp = new BulletDecorator();
Image i = new Image();
BitmapImage bi= new BitmapImage();
bi.UriSource = new Uri(@"pack://application:,,/images/icon.jpg");
i.Source = bi;
i.Width = 10;
bp.Bullet = i;
TextBlock tb = new TextBlock();
tb.Text = "My Expander";
tb.Margin = new Thickness(20,0,0,0);
bp.Child = tb;
exp.Header = bp;

StackPanel spScroll = new StackPanel();
TextBlock tbc = new TextBlock();
tbc.Text =
    "Lorem ipsum dolor sit amet, consectetur adipisicing elit," +
    "sed do eiusmod tempor incididunt ut labore et dolore magna" +
    "aliqua. Ut enim ad minim veniam, quis nostrud exercitation" +
    "ullamco laboris nisi ut aliquip ex ea commodo consequat." +
    "Duis aute irure dolor in reprehenderit in voluptate velit" +
    "esse cillum dolore eu fugiat nulla pariatur. Excepteur sint"
+
    "occaecat cupidatat non proident, sunt in culpa qui officia" +
    "deserunt mollit anim id est laborum.";
tbc.TextWrapping = TextWrapping.Wrap;

spScroll.Children.Add(tbc);
ScrollViewer scr = new ScrollViewer();
scr.Content = spScroll;
scr.Height = 50;
exp.Content = scr;

exp.Width=200;
exp.HorizontalContentAlignment= HorizontalAlignment.Stretch;
sp.Children.Add(exp);
myGrid.Children.Add(sp);
}

```

Таким образом будет создан элемент который соответствует такому куду на XAML.

```

<Expander Width="200" HorizontalContentAlignment="Stretch">
    <Expander.Header>
        <BulletDecorator>
            <BulletDecorator.Bullet>
                <Image Width="10" Source="images\icon.jpg"/>
            </BulletDecorator.Bullet>
            <TextBlock Margin="20,0,0,0">My Expander</TextBlock>
        </BulletDecorator>
    </Expander.Header>
    <ScrollViewer Height="50">
        <StackPanel>
            <TextBlock>
                Lorem ipsum dolor sit amet, consectetur adipisicing elit,
                sed do eiusmod tempor incididunt ut labore et dolore magna
                aliqua. Ut enim ad minim veniam, quis nostrud exercitation
                ullamco laboris nisi ut aliquip ex ea commodo consequat.
                Duis aute irure dolor in reprehenderit in voluptate velit
                esse cillum dolore eu fugiat nulla pariatur. Excepteur sint
                occaecat cupidatat non proident, sunt in culpa qui officia
                deserunt mollit anim id est laborum.
            </TextBlock>
        </StackPanel>
    </ScrollViewer>
</Expander>

```



```

</Expander.Header>
<Expander.Content>
  <ScrollViewer Height="50">
    <TextBlock TextWrapping="Wrap">
      Lorem ipsum dolor sit amet, consectetur adipisicing elit,
      sed do eiusmod tempor incididunt ut labore et dolore magna
      aliqua. Ut enim ad minim veniam, quis nostrud exercitation
      ullamco laboris nisi ut aliquip ex ea commodo consequat.
      Duis aute irure dolor in reprehenderit in voluptate velit
      esse cillum dolore eu fugiat nulla pariatur. Excepteur sint
      occaecat cupidatat non proident, sunt in culpa qui officia
      deserunt mollit anim id est laborum.
    </TextBlock>
  </ScrollViewer>
</Expander.Content>
</Expander>

```

Можно выровнять содержимое, задавая свойства `HorizontalContentAlignment` и `VerticalContentAlignment` в элементе управления `Expander`. При задании этих свойств выравнивание применяется к заголовку и развернутому содержимому.

Модель содержимого: `Expander`, это `HeaderedContentControl`. Его свойством содержимого является `Content`, а его свойством заголовка является `Header`. Дополнительные сведения о модели содержимого для `Expander` см. в разделе Обзор модели содержимого элементов управления.

Рассмотрим основные члены класса `Expander`.

Свойства

Свойство	Описание
<code>ExpandDirection</code>	Получает или задает направление, в котором открывается окно содержимого <code>Expander</code> . Это свойство зависимости.
<code>IsExpanded</code>	Получает или задает, является ли видимым окно содержимого элемента <code>Expander</code> . Это свойство зависимости.

События

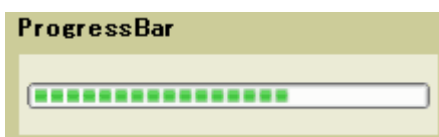
Событие	Описание
<code>Collapsed</code>	Происходит, когда закрывается окно содержимого элемента управления <code>Expander</code> и видимым является только <code>Header</code> .
<code>Expanded</code>	Событие происходит, когда открывается окно содержи-



Событие	Описание
	мого элемента управления Expander для отображения и его заголовка, и содержимого.

6. ProgressBar

ProgressBar отображает ход выполнения операции. Элемент управления ProgressBar состоит из окна, которое заполняется системным цветом выделения по мере выполнения операции.



Элемент управления ProgressBar состоит окна, которое по умолчанию заполняется слева направо по ходу выполнения операции. Элемент управления имеет диапазон и текущую позицию.

Объект ProgressBar переопределяет метаданные свойства Maximum и настраивает для него значение по умолчанию 100. Объект ProgressBar переопределяет метаданные свойства Focusable и настраивает для него значение по умолчанию false. Дополнительные сведения см. в разделе Общие сведения о свойствах зависимости.

В данном примере создается объект ProgressBar и используется анимация, чтобы показать ход выполнения операции.

```
<StatusBar Name="sbar"
    VerticalAlignment="Bottom" Background="Beige" >

    <StatusBarItem>
        <TextBlock>Downloading File</TextBlock>
    </StatusBarItem>
    <StatusBarItem>
        <ProgressBar Width="100" Height="20"
            Name="progressBar1">
            <ProgressBar.Triggers>
                <EventTrigger RoutedEvent="ProgressBar.Loaded">
                    <BeginStoryboard>
                        <Storyboard>
                            <DoubleAnimation
                                Storyboard.TargetName="progressBar1"
                                Storyboard.TargetProperty="Value"
                                From="0" To="100" Duration="0:0:5" />
                        </Storyboard>
                    </BeginStoryboard>
                </EventTrigger>
            </ProgressBar.Triggers>
```



```

        </ProgressBar>
    </StatusBarItem>
    <StatusBarItem>
        <Separator/>
    </StatusBarItem>
    <StatusBarItem>
        <TextBlock>Online</TextBlock>
    </StatusBarItem>
    <StatusBarItem HorizontalAlignment="Right">
        <Image Source="images\help.bmp" Width="16" Height="16"/>
    </StatusBarItem>
</StatusBar>

```

Тот же элемент созданный с помощью кода.

```

ProgressBar progbar = new ProgressBar();
progbar.IsIndeterminate = false;
progbar.Orientation = Orientation.Horizontal;
progbar.Width = 150;
progbar.Height = 15;
Duration duration = new Duration(TimeSpan.FromSeconds(10));
DoubleAnimation doubleanimation = new DoubleAnimation(100.0, duration);
progbar.BeginAnimation(ProgressBar.ValueProperty, doubleanimation);

```

Рассмотрим члены класса ProgressBar.

Свойства

Свойство	Описание
IsIndeterminate	Получает или задает значение, определяющее, показывает ли объект ProgressBar фактический или общий непрерывный ход выполнения. Это свойство зависимостей.
LargeChange	Возвращает или задает значение для добавления или вычитания из Value элемента управления RangeBase. Это свойство зависимостей. (Унаследовано от RangeBase.)
Maximum	Возвращает или задает максимальное возможное значение Value элемента диапазона. Это свойство зависимостей. (Унаследовано от RangeBase.)
Minimum	элемент диапазона возможный Minimum значения Value элемента диапазона. Это свойство зависимостей. (Унаследовано от RangeBase.)
Orientation	Получает или задает ориентацию объекта ProgressBar: горизонтальную или вертикальную. Это свойство зависимостей.
SmallChange	Возвращает или задает Value для добавления или вычитания из Value элемента управления RangeBase. Это свойство зависимостей. (Унаследовано от RangeBase.)
Value	Возвращает или задает текущую величину элемента



Свойство	Описание
	управления диапазона. Это свойство зависимостей. (Унаследовано от RangeBase.)

События

Событие	Описание
ValueChanged	Генерируется при изменении значения в диапазоне. (Унаследовано от RangeBase.)

7. ScrollBar

Элемент управления ScrollBar содержит элемент управления Track. Элемент управления Track состоит из элемента управления Thumb и двух элементов управления RepeatButton. Свойство Value элемента управления ScrollBar можно увеличивать и уменьшать нажатием RepeatButton или перемещением Thumb. Заданный по умолчанию диапазон значений свойства Value — от 0 до 1. Свойство Value представляет линейное расстояние элемента управления Thumb между конечными точками ScrollBar. Для изменения стандартного диапазона значений нужно задать свойства Minimum и Maximum. Свойство Orientation определяет, вертикально или горизонтально отображается ScrollBar, и для отображения элемента управления ScrollBar это свойство нужно обязательно задать.

Элемент управления Track в ScrollBar ориентирован так, чтобы его значение увеличивалось сверху вниз для вертикального ScrollBar или слева направо для горизонтального ScrollBar.

В следующем примере показано, как создать горизонтальный ScrollBar с диапазоном значений от 0 до 100.

```
<ScrollBar Orientation="Horizontal" Width="4in"
  Scroll="OnScroll" Minimum="1" Maximum="100" />
```

Рассмотрим члены класса ScrollBar.

Поля

Поле	Описание
DeferScrollTo+ HorizontalOffsetCommand	Команда, уведомляющая ScrollViewer о том, что пользователь перетаскивает Thumb горизонтальной полосы прокрутки ScrollBar до значения, заданного свойством Parameter.



Поле	Описание
DeferScrollTo+ VerticalOffsetCommand	Команда, уведомляющая ScrollView о том, что пользователь перетаскивает Thumb вертикальной полосы прокрутки ScrollBar до значения, заданного свойством Parameter.
LineDownCommand	Команда, прокручивающая ScrollBar на небольшое расстояние по вертикали с увеличением значения Track.
LineLeftCommand	Команда, прокручивающая ScrollBar на небольшое расстояние по горизонтали с уменьшением значения Track.
LineLeftCommand	Команда, прокручивающая ScrollBar на небольшое расстояние по горизонтали с уменьшением значения Track. LineRightCommand Команда, прокручивающая ScrollBar на небольшое расстояние по горизонтали с увеличением значения Track.
LineUpCommand	Команда, прокручивающая ScrollBar на небольшое расстояние по вертикали с уменьшением значения Track.
PageDownCommand	Команда, прокручивающая ScrollBar на большое расстояние по вертикали с увеличением значения Track.
PageLeftCommand	Команда, прокручивающая ScrollBar на большое расстояние по горизонтали с уменьшением значения Track.
PageRightCommand	Команда, прокручивающая ScrollBar на большое расстояние по горизонтали с увеличением значения Track.
PageUpCommand	Команда, прокручивающая ScrollBar на большое расстояние по вертикали с уменьшением значения Track.
ScrollHereCommand	Команда, прокручивающая ScrollBar до точки щелчка мыши, открывшего ContextMenu в ScrollBar.
ScrollToBottomCommand	Команда, прокручивающая ScrollBar до значения Maximum.
ScrollToEndCommand	Команда, прокручивающая содержимое до правого нижнего угла элемента управления ScrollView.
ScrollToHomeCommand	Команда, прокручивающая содержимое до левого верхнего угла элемента управления ScrollView.
ScrollToHorizontal+ OffsetCommand	Команда, прокручивающая горизонтальный ScrollBar в ScrollView до значения, заданного в Parameter.
ScrollToLeftEndCommand	Команда, прокручивающая ScrollBar до значения Minimum для горизонтального ScrollBar.
ScrollToRightEndCommand	Команда, прокручивающая ScrollBar до значения Maximum для горизонтального ScrollBar.
ScrollToTopCommand	Команда, прокручивающая ScrollBar до значения Maximum для вертикального ScrollBar.
ScrollToVertical+ OffsetCommand	Команда, прокручивающая вертикальный ScrollBar в ScrollView до значения, заданного в Parameter.



Свойства

Свойство	Описание
LargeChange	Возвращает или задает значение для добавления или вычитания из Value элемента управления RangeBase. Это свойство зависимостей. (Унаследовано от RangeBase.)
Maximum	Возвращает или задает максимальное возможное значение Value элемента диапазона. Это свойство зависимостей. (Унаследовано от RangeBase.)
Minimum	Возвращает или задает минимально возможное значение Value элемента диапазона. Это свойство зависимостей. (Унаследовано от RangeBase.)
Orientation	Возвращает или задает значение, указывающее, горизонтально или вертикально ориентирован элемент управления ScrollBar. Это свойство зависимостей.
SmallChange	Возвращает или задает Value для добавления или вычитания из Value элемента управления RangeBase. Это свойство зависимостей. (Унаследовано от RangeBase.)
Track	Возвращает или задает Track для элемента управления ScrollBar.
Value	Возвращает или задает текущую величину элемента управления диапазоном. Это свойство зависимостей. (Унаследовано от RangeBase.)

События

Событие	Описание
ValueChanged	Генерируется при изменении значения в диапазоне. (Унаследовано от RangeBase.)

8. TextBlock

TextBlock поддерживает хостинг и отображение элементов потока содержимого Inline. Поддерживаемые элементы включают: AnchoredBlock, Bold, Hyperlink, InlineUIContainer, Italic, LineBreak, Run, Span и Underline. Для получения дополнительных сведений см. раздел Общие сведения о модели содержимого TextBlock.

TextBlock разработан упрощенным и специально приспособлен для объединения небольших частей потока содержимого в пользовательский интерфейс. TextBlock оптимизирован для однолинейного отображения и



обеспечивает хорошее отображение вплоть до нескольких очередей содержимого.

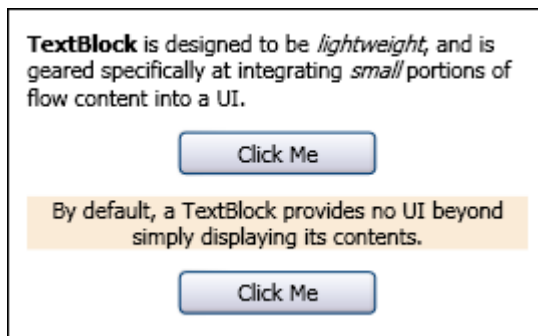
TextBlock не оптимизирован для сценариев, требующих отображения более нескольких очередей содержимого; для таких сценариев FlowDocument в связке с соответствующим элементом управления просмотром, является лучшим выбором, чем TextBlock с точки зрения производительности. После TextBlockFlowDocumentScrollView является следующим наиболее упрощенным элементом управления для отображения потока содержимого и просто обеспечивает область содержимого прокрутки минимальным Пользовательский интерфейс FlowDocumentPageViewer оптимизирован для режима просмотра "по одной странице" для потока содержимого. Наконец, FlowDocumentReader поддерживает богатейший набор функциональных возможностей для просмотра потока содержимого, но, соответственно, является более сложным.

Горизонтальное и вертикальное выравнивание текста в TextBlock выполняется с помощью свойств HorizontalContentAlignment и VerticalContentAlignment. Выравнивание TextBlock в разметке страницы выполнено с помощью свойств HorizontalAlignment и VerticalAlignment.

Рассмотрим пример использования TextBlock.

```
<TextBlock Name="textBlock1" TextWrapping="Wrap">
  <Bold>TextBlock</Bold> is designed to be <Italic>lightweight</Italic>,
  and is geared specifically at integrating <Italic>small</Italic> portions
  of flow content into a UI.
</TextBlock>
<Button Width="100" Margin="10">Click Me</Button>
<TextBlock Name="textBlock2"
  TextWrapping="Wrap" Background="AntiqueWhite" TextAlignment="Center"
>
  By default, a TextBlock provides no UI beyond simply displaying its
  contents.
</TextBlock>
<Button Width="100" Margin="10">Click Me</Button>
```

Получим такой результат.



Далее показано как добиться, того же результата кодом.

```
TextBlock textBlock1 = new TextBlock();
TextBlock textBlock2 = new TextBlock();

textBlock1.TextWrapping = textBlock2.TextWrapping = TextWrapping.Wrap;
textBlock2.Background = Brushes.AntiqueWhite;
textBlock2.TextAlignment = TextAlignment.Center;

textBlock1.Inlines.Add(new Bold(new Run("TextBlock")));
textBlock1.Inlines.Add(new Run(" is designed to be "));
textBlock1.Inlines.Add(new Italic(new Run("lightweight")));
textBlock1.Inlines.Add(new Run(", and is geared specifically at integrating "));
textBlock1.Inlines.Add(new Italic(new Run("small")));
textBlock1.Inlines.Add(new Run(" portions of flow content into a UI."));

textBlock2.Text =
    "By default, a TextBlock provides no UI beyond simply displaying its " +
    "contents.";
```

Рассмотрим основные члены класса TextBlock.

Методы

Метод	Описание
GetBaselineOffset	Возвращает значение вложенного свойстваBaselineOffset для указанного объекта зависимости.
GetFontFamily	Возвращает значение вложенного свойстваFontFamilyProperty для указанного объекта зависимости.
GetFontSize	Возвращает значение вложенного свойстваFontSize для указанного объекта зависимости.
GetFontStretch	Возвращает значение вложенного свойстваFontStretch для указанного объекта зависимости.
GetFontStyle	Возвращает значение вложенного свойстваFontStyle для указанного объекта зависимости.
GetFontWeight	Возвращает значение вложенного свойстваFontWeight для указанного объекта зависимости.
GetForeground	Возвращает значение вложенного свойстваForeground



Метод	Описание
	для указанного объекта зависимости.
GetLineHeight	Возвращает значение вложенного свойстваLineHeight для указанного объекта зависимости.
GetLineStackingStrategy	Возвращает значение вложенного свойстваLineStackingStrategy для указанного объекта зависимости.
GetPositionFromPoint	Возвращает TextPointer на позицию, ближайшую к указанному Point.
GetTextAlignment	Возвращает значение вложенного свойстваTextAlignment для указанного объекта зависимости.
SetBaselineOffset	Задание значения вложенного свойстваBaselineOffset для указанного объекта зависимости.
SetFontFamily	Задание значения вложенного свойстваFontFamily для указанного объекта зависимости.
SetFontSize	Задание значения вложенного свойстваFontSize для указанного объекта зависимости.
SetFontStretch	Задание значения вложенного свойстваFontStretch для указанного объекта зависимости.
SetFontStyle	Задание значения вложенного свойства FontStyle для указанного объекта зависимости.
SetFontWeight	Задание значения вложенного свойстваFontWeight для указанного объекта зависимости.
SetForeground	Задание значения вложенного свойстваForeground для указанного объекта зависимости.
SetLineHeight	Задание значения вложенного свойстваLineHeight для указанного объекта зависимости.
SetLineStackingStrategy	Задание значения вложенного свойстваLineStackingStrategy для указанного объекта зависимости.
SetTextAlignment	Задание значения вложенного свойстваTextAlignment для указанного объекта зависимости.

Поля

Поле	Описание
BackgroundProperty	Идентифицирует свойство зависимостиBackground.
BaselineOffsetProperty	Идентифицирует свойство зависимостиBaselineOffset.
FontFamilyProperty	Идентифицирует свойство зависимостиFontFamily.
FontSizeProperty	Идентифицирует свойство зависимостиFontSize.
FontStretchProperty	Идентифицирует свойство зависимостиFontStretch.
FontStyleProperty	Идентифицирует свойство зависимостиFontStyle.
FontWeightProperty	Идентифицирует свойство зависимости. FontWeight.



Поле	Описание
ForegroundProperty	Идентифицирует свойство зависимостиForeground.
IsHyphenationEnabledProperty	Идентифицирует свойство зависимостиIsHyphenationEnabled.
LineHeightProperty	Идентифицирует свойство зависимостиLineHeight.
LineStackingStrategyProperty	Идентифицирует свойство зависимостиLineStackingStrategy.
PaddingProperty	Идентифицирует свойство зависимостиPadding.
TextAlignmentProperty	Идентифицирует свойство зависимостиTextAlignment.
TextDecorationsProperty	Идентифицирует свойство зависимостиTextDecorations.
TextEffectsProperty	Идентифицирует свойство зависимостиTextEffects.
TextProperty	Идентифицирует свойство зависимостиText.
TextTrimmingProperty	Идентифицирует свойство зависимостиTextTrimming.
TextWrappingProperty	Идентифицирует свойство зависимостиTextWrapping.

Свойства

Свойства	Описание
BaselineOffset	Получает или задает количества, на которое каждая строка текста смещена от базовой линии. Это свойство зависимости.
BreakAfter	Получение LineBreakCondition, указывающего на то, как содержимое должно разбиваться после текущего элемента.
BreakBefore	Получение LineBreakCondition, указывающего на то, как содержимое должно разбиваться перед текущим элементом.
ContentEnd	Получение TextPointer к концу содержимого в TextBlock.
ContentStart	Получение TextPointer к началу содержимого в TextBlock.
FontFamily	Получает или задает предпочтительного семейства шрифтов верхнего уровня для TextBlock. Это свойство зависимости.
FontSize	Получает или задает размера шрифта верхнего уровня для TextBlock. Это свойство зависимости. Это свойство зависимости.
FontStretch	Получает или задает характеристик растягивания шрифтов верхнего уровня для TextBlock. Это свойство зависимости.
FontStyle	Получает или задает начертания шрифтов верхнего уровня для TextBlock. Это свойство зависимости.
FontWeight	Получает или задает насыщенности шрифтов верхнего уровня для TextBlock. Это свойство зависимости.
Foreground	Получает или задает Brush, которая применяется к текстовому содержимому TextBlock. Это свойство зависимости.
HostedElementsCore	Получение перечислителя, который может использо-



Свойства	Описание
	ваться для выполнения итерации элементов, размещенных этим <code>TextBlock</code> .
<code>Inlines</code>	Получение <code>InlineCollection</code> , содержащего элементы верхнего уровня <code>Inline</code> , составляющие содержимое <code>TextBlock</code> .
<code>IsHyphenationEnabled</code>	Получает или задает значение, указывающее на то, включена ли автоматическая расстановка переносов. Это свойство зависимости.
<code>LineHeight</code>	Получает или задает высоты каждой строки содержимого. Это свойство зависимости.
<code>LineStackingStrategy</code>	Получает или задает механизм, с помощью которого для каждой строки текста в рамках <code>TextBlock</code> . Это свойство зависимости.
<code>LogicalChildren</code>	Получение перечислителя, который может выполнить итерации логических дочерних элементов <code>TextBlock</code> . (Переопределяет <code>FrameworkElement...::LogicalChildren</code> .)
<code>Padding</code>	Получает или задает значение, указывающее толщину внутренних полей между границами области содержимого и содержимым, отображаемым <code>TextBlock</code> . Это свойство зависимости.
<code>Text</code>	Получает или задает текстового содержимого <code>TextBlock</code> . Это свойство зависимости.
<code>TextAlignment</code>	Получает или задает значения, указывающего на горизонтальное выравнивание содержимого текста. Это свойство зависимости.
<code>TextDecorations</code>	Получает или задает <code>TextDecorationCollection</code> , содержащего эффекты для применения к тексту <code>TextBlock</code> . Это свойство зависимости.
<code>TextEffects</code>	Получает или задает эффектов для применения к текстовому содержимому в данном элементе. Это свойство зависимости.
<code>TextTrimming</code>	Получает или задает поведения усечения для применения, когда область содержимого переполнена. Это свойство зависимости.
<code>TextWrapping</code>	Получает или задает способа обтекания текста <code>TextBlock</code> . Это свойство зависимости.
<code>Typography</code>	Получение действующего в настоящее время варианта оформления для содержимого данного элемента.



9. RichTextBox

Представляет элемент управления с расширенным редактированием для работы с объектами FlowDocument. FlowDocument — единственный поддерживаемый дочерний элемент объекта RichTextBox.

Для RichTextBox предусмотрена встроенная обработка событий MouseUp и MouseDown с восходящей маршрутизацией. Поэтому пользовательские обработчики событий, отслеживающие события MouseUp или MouseDown для объекта RichTextBox, не вызываются никогда. Если необходимо реагировать на такие события, отслеживайте события PreviewMouseUp и PreviewMouseDown с нисходящей маршрутизацией или зарегистрируйте обработчики с аргументом HandledEventsToo (последний вариант может быть реализован только в программном коде). Не помечайте событие как обработанное, если не собираетесь специально отключать собственную обработку этих событий в RichTextBox, и имейте в виду, что это существенно повлияет на Пользовательский интерфейс элемента управления. Хотя RichTextBox поддерживает копирование и вставку изображений, некоторые сценарии в настоящее время не поддерживаются и могут выполняться неправильно:

- Если источник изображения — графический метафайл, то вставка не производится;
- Если источник изображения указан относительным путем или ссылкой, то вставляется искаженное или пустое изображение;
- Если ссылка на источник изображения не оканчивается ожидаемым обозначением формата изображения (.PNG, .JPG, .GIF), то вставляется искаженное или пустое изображение;
- Если источник изображения скопирован из поля RTF неправильного формата, то вставляется ссылка на источник изображения, а не само изображение.



Вставка содержимого HTML в элемент RichTextBox может привести к непредсказуемым результатам, поскольку RichTextBox использует формат RTF, а не HTML.

Текст внутри элемента RichTextBox всегда переносится по строкам. Если текст переносить не следует, установите для свойства PageWidth объекта FlowDocument значение, превышающее ширину элемента RichTextBox. Тем не менее после достижения ширины страницы текст все же перейдет на следующую строку.

Горизонтальное и вертикальное выравнивание содержимого в элементе RichTextBox производится с помощью свойств HorizontalContentAlignment и VerticalContentAlignment. Выравнивание RichTextBox относительно разметки страницы производится с использованием свойств HorizontalAlignment и VerticalAlignment.

Полосы прокрутки в RichTextBox по умолчанию не отображаются. Чтобы сделать полосу прокрутки видимой, присвойте свойству VerticalScrollBarVisibility значение Visible или Auto.

Чтобы определить, когда изменяется текст в элементе TextBox или RichTextBox, обычно используется событие TextChanged, а не KeyDown, как можно было бы предположить. Пример см. в разделе Практическое руководство. Определение изменения текста в TextBox.

RichTextBox поддерживает широкий спектр команд клавиатуры.

Команда изменения	Знак клавиши по умолчанию
AlignCenter	Ctrl+E
AlignJustify	Ctrl+J
AlignLeft	Ctrl+L
AlignRight	Ctrl+R
Backspace	Backspace
CorrectSpellingError	знак клавиши по умолчанию отсутствует
DecreaseFontSize	Ctrl+OemOpenBrackets
DecreaseIndentation	Ctrl+Shift+T
Delete	Delete
DeleteNextWord	Ctrl+Delete
DeletePreviousWord	Ctrl+Backspace
EnterLineBreak	Shift+Enter
EnterParagraphBreak	Enter



Команда изменения	Знак клавиши по умолчанию
IgnoreSpellingError	знак клавиши по умолчанию отсутствует
IncreaseFontSize	Ctrl+OemCloseBrackets
IncreaseIndentation	Ctrl+T
MoveDownByLine	Down
MoveDownByPage	PageDown
MoveDownByParagraph	Ctrl+Down
MoveLeftByCharacter	Left
MoveLeftByWord	Ctrl+Left
MoveRightByCharacter	Right
MoveRightByWord	Ctrl+Right
MoveToDocumentEnd	Ctrl+End
MoveToDocumentStart	Ctrl+Home
MoveToLineEnd	End
MoveToLineStart	Home
MoveUpByLine	Up
MoveUpByPage	PageUp
MoveUpByParagraph	Ctrl+Up
SelectDownByLine	Shift+Down
SelectDownByPage	Shift+PageDown
SelectDownByParagraph	Ctrl+Shift+Down
SelectLeftByCharacter	Shift+Left
SelectLeftByWord	Ctrl+Shift+Left
SelectRightByCharacter	Shift+Right
SelectRightByWord	Ctrl+Shift+Right
SelectToDocumentEnd	Ctrl+Shift+End
SelectToDocumentStart	Ctrl+Shift+Home
SelectToLineEnd	Shift+End
SelectToLineStart	Shift+Home
SelectUpByLine	Shift+Up
SelectUpByPage	Shift+PageUp
SelectUpByParagraph	Ctrl+Shift+Up
TabBackward	Shift+Tab
TabForward	Tab
ToggleBold	Ctrl+B
ToggleBullets	Ctrl+Shift+L
ToggleInsert	Insert
ToggleItalic	Ctrl+I
ToggleNumbering	Ctrl+Shift+N
ToggleSubscript	Ctrl+OemPlus
ToggleSuperscript	Ctrl+Shift+OemPlus
ToggleUnderline	Ctrl+U

Команды перемещения курсора и команды расширения выбора имеют один набор знаков клавиш по умолчанию, причем основное различие состоит в добавлении клавиши Shift, чтобы отличать команды вы-



бора от команд перемещения курсора. Например, команда `MoveLeftByCharacter` имеет знак клавиши по умолчанию `Left`, а соответствующая команда `SelectLeftByCharacter` имеет знак клавиши по умолчанию `Shift+Left`.

В общем, определения команд, предоставляемые классом `EditingCommands`, не используют параметры команды (параметр `parameter` ожидается в методе `Execute`).

Приведенный ниже пример отражает вызов команды изменения по объекту, который поддерживает данную команду.

В данном примере `RichTextBox` выступает как цель команды. Обратите внимание, что `RichTextBox` реализует интерфейс `IInputElement` (унаследованный от `FrameworkElement`), и что он включает исходную поддержку для большинства команд изменения.

Первым аргументом для метода `Execute` является параметр команды. Большинство команд изменения игнорируют параметр команды; в общем, данный параметр должен быть `null` (ничего не возвращает ссылка `null` для команд изменения).

Второй аргумент задает объект, на который будет направлена команда. Данный объект должен реализовать интерфейс `IInputElement` и включать обработчика для заданной команды. Обычно команда игнорируется при вызове объекта, который не обрабатывает данную команду.

```
RichTextBox rTB = new RichTextBox();  
EditingCommands.ToggleInsert.Execute(null, rTB);
```

В следующем примере объявляется объект `RichTextBox`, содержащий простой объект `FlowDocument`.

```
<RichTextBox Name="richTB">  
  <FlowDocument>  
    <Paragraph>  
      <Run>Paragraph 1</Run>  
    </Paragraph>  
    <Paragraph>  
      <Run>Paragraph 2</Run>  
    </Paragraph>  
  </FlowDocument>  
</RichTextBox>
```



```
<Paragraph>
  <Run>Paragraph 3</Run>
</Paragraph>
</FlowDocument>
</RichTextBox>
```

И тот же пример с помощью кода.

```
using System;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Media;
using System.Windows.Documents;
namespace SDKSample
{
    public partial class RichTextBoxExample : Page
    {
        public RichTextBoxExample()
        {
            StackPanel myStackPanel = new StackPanel();

            FlowDocument myFlowDoc = new FlowDocument();

            myFlowDoc.Blocks.Add(new Paragraph(new Run("Paragraph 1")));
            myFlowDoc.Blocks.Add(new Paragraph(new Run("Paragraph 2")));
            myFlowDoc.Blocks.Add(new Paragraph(new Run("Paragraph 3")));
            RichTextBox myRichTextBox = new RichTextBox();

            myRichTextBox.Document = myFlowDoc;

            myStackPanel.Children.Add(myRichTextBox);
            this.Content = myStackPanel;
        }
    }
}
```

Следующий код реализует метод, который принимает RichTextBox в качестве аргумента и возвращает строку, представляющую текстовое содержимое RichTextBox.

Метод создает новый TextRange из содержимого RichTextBox, с помощью ContentStart и ContentEnd для указания диапазона извлекаемого содержимого. Каждое из свойств ContentStart и ContentEnd возвращает TextPointer и доступно в основном FlowDocument, представляющем содержимое RichTextBox. TextRange предоставляет свойство Text, которое возвращает часть обычного текста TextRange в виде строки.



```
string StringFromRichTextBox(RichTextBox rtb)
{
    TextRange textRange = new TextRange(
        rtb.Document.ContentStart,
        rtb.Document.ContentEnd
    );

    return textRange.Text;
}
```

В следующем примере демонстрируется размещение пользовательского контекстного меню для объекта RichTextBox.

При реализации пользовательского контекстного меню для объекта RichTextBox обработка местоположения контекстного меню реализуется разработчиком. По умолчанию пользовательское контекстное меню открывается в центре объекта RichTextBox.

Чтобы переопределить поведение расположения по умолчанию, добавьте прослушиватель для события ContextMenuOpening. В следующем примере описывается программный способ переопределения.

```
richTextBox.ContextMenuOpening += new
ContextMenuEventHandler(richTextBox_ContextMenuOpening);
```

В следующем примере демонстрируется реализация соответствующего прослушивателя события ContextMenuOpening.

```
void richTextBox_ContextMenuOpening(object sender, ContextMenuEventArgs e)
{
    RichTextBox rtb = sender as RichTextBox;
    if (rtb == null) return;

    ContextMenu contextMenu = rtb.ContextMenu;
    contextMenu.PlacementTarget = rtb;

    contextMenu.Placement = PlacementMode.RelativePoint;

    TextPointer position = rtb.Selection.End;

    if (position == null) return;

    Rect positionRect = position.GetCharacterRect(LogicalDirection.Forward);
    contextMenu.HorizontalOffset = positionRect.X;
    contextMenu.VerticalOffset = positionRect.Y;
}
```



```
contextMenu.IsOpen = true;  
e.Handled = true;  
}
```

В этом примере показано использование стилей Windows Presentation Foundation (WPF) для замены стандартного узла содержимого для элемента управления RichTextBox.

Узел содержимого — это элемент, отображающий содержимое RichTextBox. Стандартный шаблон элемента управления RichTextBox задает элемент ScrollViewer в качестве узла содержимого.

В случаях, когда возможности прокрутки, предоставляемые ScrollViewer, являются нежелательными или ненужными, облегченный элемент AdornerDecorator может быть определен в качестве узла содержимого элемента управления RichTextBox. ScrollViewer и AdornerDecorator являются поддерживаемыми элементами узла содержимого.

Шаблон ControlTemplate элемента RichTextBox должен содержать только один элемент, обозначенный как элемент узла содержимого. Для обозначения элемента в качестве узла содержимого, назначьте ему специальное имя PART_ContentHost. Элемент узла содержимого должен быть либо ScrollViewer, либо AdornerDecorator. Элемент узла содержимого не может содержать ни одного дочернего элемента.

В следующем примере кода Язык XAML (Extensible Application Markup Language) определяется стиль, который переопределяет стандартный шаблон элемента управления для RichTextBox. Этот стиль обеспечивает совместимость с элементами, полученными из класса TextBoxBase. В этом примере элемент AdornerDecorator указывается в качестве узла содержимого.

```
<Window.Resources>  
  <Style x:Key="TextBoxNoScrollViewer" TargetType="{x:Type TextBoxBase}">  
    <Setter Property="Template">  
      <Setter.Value>  
        <ControlTemplate TargetType="{x:Type TextBoxBase}">  
          <Border
```



```

        CornerRadius="2"
        Background="{TemplateBinding Background}"
        BorderThickness="{TemplateBinding BorderThickness}"
        BorderBrush="{TemplateBinding BorderBrush}"
    >
        <AdornerDecorator
            x:Name="PART_ContentHost"
            Focusable="False"
        />
    </Border>
</ControlTemplate>
</Setter.Value>
</Setter>
</Style>
</Window.Resources>

```

В следующем примере кода XAML определяется элемент управления RichTextBox, который использует ранее объявленный стиль с помощью атрибута Style вместе со статической ресурсной ссылкой на атрибут Атрибут x:Key стиля.

```

<RichTextBox
    Grid.Column="0"

    VerticalScrollBarVisibility="Auto"
    HorizontalScrollBarVisibility="Auto"

    Style="{StaticResource TextBoxNoScrollViewer}"
>
    <FlowDocument>
        <Paragraph>
            RichTextBox styled not to use a ScrollViewer as the content host.
        </Paragraph>
    </FlowDocument>
</RichTextBox>

```

Рассмотрим основные члены класса RichTextBlock.

Методы

Метод	Описание
AppendText	Добавляет строку к содержимому текстового элемента управления. (Унаследовано от TextBoxBase.)
BeginChange	Начинает блок изменений. (Унаследовано от TextBoxBase.)
Copy	Копирует текущее выделение текста в элементе управления, поддерживающем редактирование текста, в Clipboard. (Унаследовано от TextBoxBase.)
Cut	Удаляет текущее выделение текста из элемента управления, поддерживающего редактирование текста, и ко-



Метод	Описание
	пишет его в Clipboard. (Унаследовано от TextBoxBase.)
DeclareChangeBlock	Создает блок изменений. (Унаследовано от TextBoxBase.)
EndChange	Завершает блок изменений. (Унаследовано от TextBoxBase.)
GetNextSpellingErrorPosition	Возвращает объект TextPointer, указывающий на следующую орфографическую ошибку в содержимом элемента RichTextBox.
GetPositionFromPoint	Возвращает объект TextPointer, указывающий на точку вставки, ближайшую к указанному положению TextPointer.
GetSpellingError	Возвращает объект SpellingError, связанный с орфографической ошибкой (если она существует) в указанной позиции содержимого элемента RichTextBox.
GetSpellingErrorRange	Возвращает объект TextRange, включающий слово с ошибкой (если оно существует) в указанной позиции содержимого элемента RichTextBox.
LineDown	Прокручивает содержимое элемента управления вниз на одну строку. (Унаследовано от TextBoxBase.)
LineLeft	Прокручивает содержимое элемента управления влево на одну строку. (Унаследовано от TextBoxBase.)
LineRight	Прокручивает содержимое элемента управления вправо на одну строку. (Унаследовано от TextBoxBase.)
LineUp	Прокручивает содержимое элемента управления вверх на одну строку. (Унаследовано от TextBoxBase.)
LockCurrentUndoUnit	Блокирует последний блок отмены в стеке отмены приложения. Это предотвращает слияние заблокированного блока с блоками отмены, добавленными позднее. (Унаследовано от TextBoxBase.)
PageDown	Прокручивает содержимое элемента управления вниз на одну страницу. (Унаследовано от TextBoxBase.)
PageLeft	Прокручивает содержимое элемента управления влево на одну страницу. (Унаследовано от TextBoxBase.)
PageRight	Прокручивает содержимое элемента управления вправо на одну страницу. (Унаследовано от TextBoxBase.)
PageUp	Прокручивает содержимое элемента управления вверх на одну страницу. (Унаследовано от TextBoxBase.)
Paste	Вставляет текущее содержимое буфера обмена на место текущего выделения в элементе управления, поддерживающем редактирование текста. (Унаследовано от TextBoxBase.)
Redo	Отменяет последнюю команду отмены. Иными словами, повторяет последний блок отмены в стеке отмены.



Метод	Описание
	(Унаследовано от TextBoxBase.)
ScrollToEnd	Прокручивает представление элемента управления, поддерживающего редактирование текста, в конец содержимого. (Унаследовано от TextBoxBase.)
ScrollToHome	Прокручивает представление элемента управления, поддерживающего редактирование текста, в начало окна просмотра. (Унаследовано от TextBoxBase.)
ScrollToHorizontalOffset	Прокручивает содержимое элемента управления, поддерживающего редактирование текста, на заданное смещение по горизонтали. (Унаследовано от TextBoxBase.)
ScrollToVerticalOffset	Прокручивает содержимое элемента управления, поддерживающего редактирование текста, на заданное смещение по вертикали. (Унаследовано от TextBoxBase.)
SelectAll	Выделяет все содержимое текстового элемента управления, поддерживающего редактирование текста. (Унаследовано от TextBoxBase.)
ShouldSerializeDocument	Возвращает значение, показывающее, должно ли быть сериализовано действительное значение свойства Document в процессе сериализации объекта RichTextBox.
Undo	Отменяет последнюю команду отмены. Иными словами, отменяет последний блок отмены в стеке отмены. (Унаследовано от TextBoxBase.)

Свойства

Свойство	Описание
AcceptsReturn	Возвращает или задает значение, указывающее, как элемент управления, поддерживающий редактирование текста, отвечает на нажатие пользователем клавиши ВВОД. (Унаследовано от TextBoxBase.)
AcceptsTab	Возвращает или задает значение, указывающее, как элемент управления, поддерживающий редактирование текста, отвечает на нажатие пользователем клавиши TAB. Это свойство зависимостей. (Унаследовано от TextBoxBase.)
AutoWordSelection	Возвращает или задает значение, указывающее, производится ли автоматическое выделение всего слова, когда пользователь выделяет часть его с помощью мыши. Это свойство зависимостей. (Унаследовано от TextBoxBase.)
CanRedo	Возвращает значение, указывающее, может ли быть вы-



Свойство	Описание
	полнен повтор для последней операции отмены. (Унаследовано от TextBoxBase.)
CanUndo	Возвращает значение, указывающее, может ли быть выполнена отмена последнего действия. (Унаследовано от TextBoxBase.)
CaretPosition	Получает или задает положение точки ввода.
Document	Получает или задает объект FlowDocument, представляющий содержимое элемента RichTextBox.
ExtentHeight	Возвращает размер видимой области содержимого по вертикали. (Унаследовано от TextBoxBase.)
ExtentWidth	Возвращает размер видимой области содержимого по горизонтали. (Унаследовано от TextBoxBase.)
FontFamily	Возвращает или задает семейство шрифтов элемента управления. Это свойство зависимостей. (Унаследовано от Control.)
FontSize	Возвращает или задает размер шрифта. Это свойство зависимостей. (Унаследовано от Control.)
FontStretch	Возвращает или задает степень сжатия или расширения шрифта на экране. Это свойство зависимостей. (Унаследовано от Control.)
FontStyle	Возвращает или задает стиль шрифта. Это свойство зависимостей. (Унаследовано от Control.)
FontWeight	Возвращает или задает плотность или толщину указанного шрифта. Это свойство зависимостей. (Унаследовано от Control.)
HorizontalOffset	Возвращает или задает текущую позицию прокрутки по горизонтали. (Унаследовано от TextBoxBase.)
HorizontalOffset	Возвращает или задает текущую позицию прокрутки по горизонтали. (Унаследовано от TextBoxBase.)
IsReadOnly	Возвращает или задает значение, указывающее, является ли элемент управления, поддерживающий редактирование текста, доступным работающему с ним пользователю только для чтения. Это свойство зависимостей. (Унаследовано от TextBoxBase.)
IsUndoEnabled	Возвращает или задает значение, указывающее, поддерживает ли элемент управления, используемый для редактирования текста, операцию отмены. Это свойство зависимостей. (Унаследовано от TextBoxBase.)
Selection	Получает объект TextSelection, содержащий текущий выделенный фрагмент элемента RichTextBox.
SpellCheck	Возвращает объект SpellCheck, предоставляющий доступ к ошибкам правописания в текстовом содержимом TextBoxBase или RichTextBox. (Унаследовано от TextBoxBase.)



Свойство	Описание
UndoLimit	Возвращает или задает количество действий, хранящихся в очереди отмены. (Унаследовано от TextBoxBase.)
VerticalOffset	Возвращает или задает позицию прокрутки по вертикали. (Унаследовано от TextBoxBase.)
VerticalScrollBarVisibility	Возвращает или задает значение, указывающее, отображается ли вертикальная полоса прокрутки. Это свойство зависимостей. (Унаследовано от TextBoxBase.)
ViewportHeight	Возвращает размер прокручиваемой области содержимого по вертикали. (Унаследовано от TextBoxBase.)
ViewportWidth	Возвращает размер прокручиваемой области содержимого по горизонтали. (Унаследовано от TextBoxBase.)

События

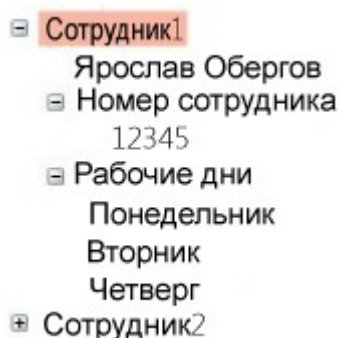
Событие	Описание
SelectionChanged	Генерируется при изменении текстового выделения. (Унаследовано от TextBoxBase.)
TextChanged	Генерируется при изменении содержимого текстового элемента. (Унаследовано от TextBoxBase.)



10. TreeView.

TreeView представляет элемент управления, отображающий иерархические данные в структуре дерева с элементами, которые могут разворачиваться и сворачиваться.

Объект TreeView относится к типу ItemsControl. Свойствами содержимого являются Items и ItemsSource.



Элементы управления, которые наследуют от объекта ItemsControl, содержат коллекцию объектов. Примером объекта ItemsControl является объект ListBox. Для заполнения объекта ItemsControl можно использовать свойство ItemsSource или Items.

Свойство ItemsSource объекта ItemsControl позволяет использовать любой тип, реализующий интерфейс IEnumerable как содержимое объекта ItemsControl. Свойство ItemsSource обычно используется для отображения коллекции данных или привязки объекта ItemsControl к объекту коллекции.

Если не требуется использовать объект, который реализует интерфейс IEnumerable для заполнения объекта ItemsControl, можно добавить элементы с помощью свойства Items. Элементы в объекте ItemsControl могут иметь типы, отличные друг от друга. Например, объект ListBox может содержать один элемент, который является строкой, и другой элемент, являющийся объектом Image.

Элемент управления TreeView — это объект ItemsControl, который выполняет вложение элементов с помощью элементов управления TreeViewItem. В следующем примере создается объект TreeView.



```
<TreeView Name="myTreeViewEvent" >
  <TreeViewItem Header="Employee1" IsSelected="True">
    <TreeViewItem Header="Jesper Aaberg"/>
    <TreeViewItem Header="Employee Number">
      <TreeViewItem Header="12345"/>
    </TreeViewItem>
    <TreeViewItem Header="Work Days">
      <TreeViewItem Header="Monday"/>
      <TreeViewItem Header="Tuesday"/>
      <TreeViewItem Header="Thursday"/>
    </TreeViewItem>
  </TreeViewItem>
  <TreeViewItem Header="Employee2">
    <TreeViewItem Header="Dominik Paiha"/>
    <TreeViewItem Header="Employee Number">
      <TreeViewItem Header="98765"/>
    </TreeViewItem>
    <TreeViewItem Header="Work Days">
      <TreeViewItem Header="Tuesday"/>
      <TreeViewItem Header="Wednesday"/>
      <TreeViewItem Header="Friday"/>
    </TreeViewItem>
  </TreeViewItem>
</TreeView>
```

Элемент управления `TreeView` содержит иерархию элементов управления `TreeViewItem`. Элемент управления `TreeViewItem` — это объект `HeaderedItemsControl`, который имеет свойство `Header` и содержит коллекцию `Items`.

При определении `TreeView` с помощью Язык XAML (Extensible Application Markup Language) можно явно определить содержимое `Header` элемента управления `TreeViewItem` и элементы, составляющие его коллекцию. На предыдущем рисунке показан этот метод.

Можно также указать `ItemsSource` в качестве источника данных и затем указать свойства `HeaderTemplate` и `ItemTemplate` для определения содержимого `TreeViewItem`.

Для определения макета элемента управления `TreeViewItem` можно также использовать объекты `HierarchicalDataTemplate`. Дополнительные сведения и пример см. в разделе Инструкция по Использованию `SelectedValue`, `SelectedValuePath` и `SelectedItem`.



Если элемент не является элементом управления `TreeViewItem`, он автоматически включается в элемент управления `TreeViewItem` при отображении элемента управления `TreeView`.

Если пользователь разворачивает узел `TreeViewItem`, для свойства `IsExpanded` устанавливается значение `true`. Можно также развернуть или свернуть `TreeViewItem` без участия пользователя, установив для свойства `IsExpanded` значение `true` (развернуть) или `false` (свернуть). При изменении этого свойства создается событие `Expanded` или `Collapsed`.

При вызове метода `BringIntoView` в элементе управления `TreeViewItem`, `TreeViewItem` и его родительский элемент управления `TreeViewItem` разворачиваются. Если элемент управления `TreeViewItem` скрыт или частично скрыт, элемент управления `TreeView` прокручивается, чтобы сделать его видимым.

Когда пользователь щелкает элемент управления `TreeViewItem`, чтобы его выделить, создается событие `Selected`, и для свойства `IsSelected` элемента управления устанавливается значение `true`. Элемент `TreeViewItem` также становится свойством `SelectedItem` элемента управления `TreeView`. И наоборот, при изменении выбора элемента управления `TreeViewItem` создается событие `Unselected` и для свойства `IsSelected` устанавливается значение `false`.

Свойство `SelectedItem` элемента управления `TreeView` доступно только для чтения; поэтому его нельзя установить явно. Свойство `SelectedItem` устанавливается, если пользователь щелкает элемент управления `TreeViewItem` или свойству `IsSelected` элемента управления `TreeViewItem` присваивается значение `true`.

Для установки значения свойству `SelectedItem` используются свойства `SelectedValue` и `SelectedValuePath`. Дополнительные сведения см. в разделе Инструкция по Использованию `SelectedValue`, `SelectedValuePath` и `SelectedItem`.

Можно зарегистрировать обработчик событий для события `SelectedItemChanged`, чтобы определить, когда изменяется выбранный



объект `TreeViewItem`. Параметр `RoutedPropertyChangedEventArgs<(Of <(T>)>)`, который предоставляется обработчику событий, указывает значение `OldValue`, которое является предыдущим выбором, и значение `NewValue`, представляющее текущий выбор. Любое из двух значений может быть `null`, если предыдущий или текущий выбор не выполнен приложением или пользователем.

По умолчанию стиль элемента управления `TreeView` помещает его в объект `StackPanel`, содержащий элемент управления `ScrollViewer`. При установке свойств `Width` и `Height` для элемента управления `TreeView` эти значения используются для изменения размера объекта `StackPanel`, отображающего элемент управления `TreeView`. Если содержимое для отображения больше, чем область отображения, автоматически отображается элемент управления `ScrollViewer`, чтобы пользователь мог прокручивать содержимое `TreeView`.

Для настройки внешнего вида элемента управления `TreeViewItem` установите для свойства `Style` пользовательский класс `Style`.

В следующем примере показано, как установить значения свойств `Foreground` и `FontSize` для элемента управления `TreeViewItem` с помощью свойства `Style`.

```
<Style TargetType="{x:Type TreeViewItem}">
  <Setter Property="Foreground" Value="Blue"/>
  <Setter Property="FontSize" Value="12"/>
</Style>
```

В содержимое `Header` элемента управления `TreeViewItem` можно включить несколько объектов. Чтобы включить несколько объектов в содержимое `Header`, разместите объекты внутри элемента управления макета, такого как `Panel` или `StackPanel`.

В следующем примере показано определение свойства `Header` для элемента управления `TreeViewItem` в качестве объектов `CheckBox` и `TextBlock`, которые включены в элемент управления `DockPanel`.



```
<TreeViewItem>
  <TreeViewItem.Header>
    <DockPanel>
      <CheckBox/>
      <TextBlock>
        TreeViewItem Text
      </TextBlock>
    </DockPanel>
  </TreeViewItem.Header>
</TreeViewItem>
```

Если элемент управления TreeView содержит множество элементов, для их загрузки может потребоваться значительное время, что может привести к задержке в работе пользовательского интерфейса. Время загрузки можно сократить, присвоив вложенному свойству VirtualizingStackPanel...:IsVirtualizing значение true. Пользовательский интерфейс также может медленно реагировать на прокрутку пользователем элемента управления TreeView с помощью колесика мыши или путем перетаскивания ползунка полосы прокрутки. Производительность элемента управления TreeView при прокрутке можно повысить, присвоив вложенному свойству VirtualizingStackPanelVirtualizationMode()()() значение Recycling.

В приведенном ниже примере создается элемент управления TreeView, для оптимизации производительности свойству VirtualizingStackPanel...:IsVirtualizing которого присваивается значение true, а свойству VirtualizingStackPanelVirtualizationMode()()() — значение Recycling.

```
<StackPanel>
  <StackPanel.Resources>
    <src:TreeViewData x:Key="dataItems"/>

    <HierarchicalDataTemplate DataType="{x:Type src:ItemsForTreeView}"
                             ItemsSource="{Binding
Path=SecondLevelItems}">

      <TextBlock Text="{Binding Path=TopLevelName}"/>

      <HierarchicalDataTemplate.ItemTemplate>
        <DataTemplate>
          <TextBlock Text="{Binding}"/>
        </DataTemplate>
      </HierarchicalDataTemplate.ItemTemplate>
    </HierarchicalDataTemplate>
  </StackPanel.Resources>
</StackPanel>
```



```

</HierarchicalDataTemplate.ItemTemplate>

<HierarchicalDataTemplate.ItemContainerStyle>
  <Style TargetType="TreeViewItem">
    <Setter Property="Foreground" Value="Navy"/>
  </Style>
</HierarchicalDataTemplate.ItemContainerStyle>
</HierarchicalDataTemplate>
</StackPanel.Resources>

<TreeView Height="200" ItemsSource="{Binding Source={StaticResource
dataItems}}">
  VirtualizingStackPanel.IsVirtualizing="True"
  VirtualizingStackPanel.VirtualizationMode="Recycling">
  <TreeView.ItemContainerStyle>

    <Style TargetType="TreeViewItem">
      <Setter Property="IsExpanded" Value="True"/>
      <Setter Property="Foreground" Value="Green"/>
    </Style>
  </TreeView.ItemContainerStyle>
</TreeView>
</StackPanel>

```

В следующем примере показаны данные, использовавшиеся в предыдущем примере.

```

public class TreeViewData : ObservableCollection<ItemsForTreeView>
{
    public TreeViewData()
    {
        for (int i = 0; i < 100; ++i)
        {
            ItemsForTreeView item = new ItemsForTreeView();
            item.TopLevelName = "item " + i.ToString();
            Add(item);
        }
    }
}

public class ItemsForTreeView
{
    public string TopLevelName { get; set; }
    private ObservableCollection<string> level2Items;

    public ObservableCollection<string> SecondLevelItems
    {
        get
        {
            if (level2Items == null)
            {
                level2Items = new ObservableCollection<string>();
            }
            return level2Items;
        }
    }
}

```



```

    }
}

public ItemsForTreeView()
{
    for (int i = 0; i < 10; ++i)
    {
        SecondLevelItems.Add("Second Level " + i.ToString());
    }
}
}

```

Рассмотрим основные члены класса `TreeView`.

Методы

Метод	Описание
<code>AddChild</code>	Добавляет заданный объект в качестве дочернего объекта <code>ItemsControl</code> . (Унаследовано от <code>ItemsControl</code> .)
<code>AddText</code>	Добавляет заданную текстовую строку к объекту <code>ItemsControl</code> . (Унаследовано от <code>ItemsControl</code> .)

Поля

Поле	Описание
<code>SelectedItemChangedEvent</code>	Определяет перенаправленное событие <code>SelectedItemChanged</code> .
<code>SelectedItemProperty</code>	Определяет свойство зависимостей <code>SelectedItem</code> .
<code>SelectedValuePathProperty</code>	Идентифицирует свойство зависимостей <code>SelectedValuePath</code> .
<code>SelectedValueProperty</code>	Определяет свойство зависимостей <code>SelectedValue</code> .

Свойства

Свойство	Описание
<code>AlternationCount</code>	Возвращает или задает количество чередующихся контейнеров элементов в объекте <code>ItemsControl</code> , который позволяет чередующимся контейнерам иметь уникальный внешний вид. (Унаследовано от <code>ItemsControl</code> .)
<code>HasItems</code>	Получает значение, указывающее содержит ли <code>ItemsControl</code> элементы. Это свойство зависимостей. (Унаследовано от <code>ItemsControl</code> .)
<code>IsTextSearchEnabled</code>	Возвращает или задает значение, указывающее включен ли <code>TextSearch</code> в экземпляре <code>ItemsControl</code> . Это свойство зависимостей. (Унаследовано от <code>ItemsControl</code> .)
<code>Items</code>	Получает коллекцию, используемую для генерирования содержимого <code>ItemsControl</code> . (Унаследовано от <code>ItemsControl</code> .)
<code>ItemsPanel</code>	Возвращает или задает шаблон, определяющий панель, управляющую размещением элементов. Это свойство за-



Свойство	Описание
	зависимостей. (Унаследовано от ItemsControl.)
ItemsSource	Возвращает или задает коллекцию, используемую для генерирования содержимого ItemsControl. Это свойство зависимостей. (Унаследовано от ItemsControl.)
LogicalChildren	Получает перечислитель для логических дочерних объектов объекта ItemsControl. (Унаследовано от ItemsControl.)
SelectedItem	Получает выбранный элемент в TreeView. Это свойство зависимостей.
SelectedValue	Получает значение свойства, указанное SelectedValuePath для SelectedItem. Это свойство зависимостей.
SelectedValuePath	Возвращает или задает путь, используемый для получения SelectedValue, принадлежащего SelectedItem, в TreeView. Это свойство зависимостей.

События

Событие	Описание
SelectedItemChanged	Происходит при изменении свойства SelectedItem.



11. ListView.

Элемент управления ListView предоставляет инфраструктуру для отображения набора элементов данных в различных макетах или представлениях. Например, у пользователя может возникнуть потребность в отображении элементов данных в таблице, а также в сортировке ее столбцов.

Элемент управления ListView является ItemsControl, который является производным от ListBox. Как правило, его элементы являются членами коллекции данных и представляются как объекты ListViewItem. ListViewItem является ContentControl и может содержать только один дочерний элемент. Однако дочерним элементом может быть любой визуальный элемент.

Чтобы задать режим просмотра содержимого элемента управления ListView, необходимо задать значение свойства View. Одним из режимов, предоставляемых Windows Presentation Foundation (WPF), является GridView, отображающий коллекцию элементов данных в таблице с настраиваемыми столбцами.

В следующем примере показано определение GridView для элемента управления ListView, в котором отображаются сведения о сотруднике.

```
<ListView ItemsSource="{Binding Source=
    {StaticResource EmployeeInfoDataSource}}">

    <ListView.View>

        <GridView AllowsColumnReorder="true"
            ColumnHeaderToolTip="Employee Information">

            <GridViewColumn DisplayMemberBinding=
                "{Binding Path=FirstName}"
                Header="First Name" Width="100"/>

            <GridViewColumn DisplayMemberBinding=
                "{Binding Path=LastName}"
                Width="100">
                <GridViewColumnHeader>Last Name
                <GridViewColumnHeader.ContextMenu>
                <ContextMenu
                    MenuItem.Click="LastNameCM_Click">
```



```

Name="LastNameCM">
    <MenuItem Header="Ascending" />
    <MenuItem Header="Descending" />
</ContextMenu>
</GridViewColumnHeader.ContextMenu>
</GridViewColumnHeader>
</GridViewColumn>

<GridViewColumn DisplayMemberBinding=
    "{Binding Path=EmployeeNumber}"
    Header="Employee No." Width="100"/>
</GridView>

</ListView.View>
</ListView>

```

На следующем рисунке показано, как будут отображаться данные для предыдущего примера.

First Name	Last Name	Employee No.
Jesper	Aaberg	12345
Dominik	Paiha	98765
Yale	Li	23875
Muru	Subramani	49392

Можно создать пользовательский режим представления путем определения класса, наследуемого от класса `ViewBase`. Класс `ViewBase` предоставляет инфраструктуру, которая необходима для создания пользовательского режима представления.

Элемент управления `ListView` содержит объекты `ListViewItem`, представляющие отображаемые элементы данных. Можно использовать следующие свойства для определения содержимого и стиля элементов данных:

- * Для элемента управления `ListView` используйте свойства `ItemTemplate`, `ItemTemplateSelector` и `ItemContainerStyle`.

- * Для элемента управления `ListViewItem` используйте свойства `ContentTemplate` и `ContentTemplateSelector`.

Чтобы избежать проблем выравнивания между ячейками в `GridView`, не используйте `ItemContainerStyle` для задания свойств или добавление содержимого, которое влияет на ширину элемента в `ListView`. Например, может возникнуть проблема выравнивания при задании свойства `Margin` в `ItemContainerStyle`. Чтобы задать свойства или определить



содержимое, которое влияет на ширину элементов в GridView, используйте свойства класса GridView и его связанных классов, таких как GridViewColumn.

Дополнительные сведения о том, как использовать GridView и его вспомогательные классы, см. в разделе Общие сведения о GridView.

Если определить ItemContainerStyle для элемента управления ListView и также определить ItemTemplate, необходимо включить ContentPresenter в стиль для правильной работы ItemTemplate.

Не используйте свойства HorizontalContentAlignment и VerticalContentAlignment для содержимого ListView, которое отображается с помощью GridView. Чтобы задать выравнивание содержимого в столбце GridView, определите свойство CellTemplate.

Два элемента управления ListView не могут совместно использовать один режим в одно и то же время. При попытке использования одного режима более, чем одним элементом управления ListView возникает исключение.

Чтобы указать режим представления, который может одновременно использоваться более, чем одним ListView, используйте шаблоны или стили.



12. Примеры использование элементов управления

Примеры программ, которые относятся, к этому уроку располагаются в директории Source.

Пример 1. Работа с Image. (Source\1-2-WPF_UI\WPF_UI)

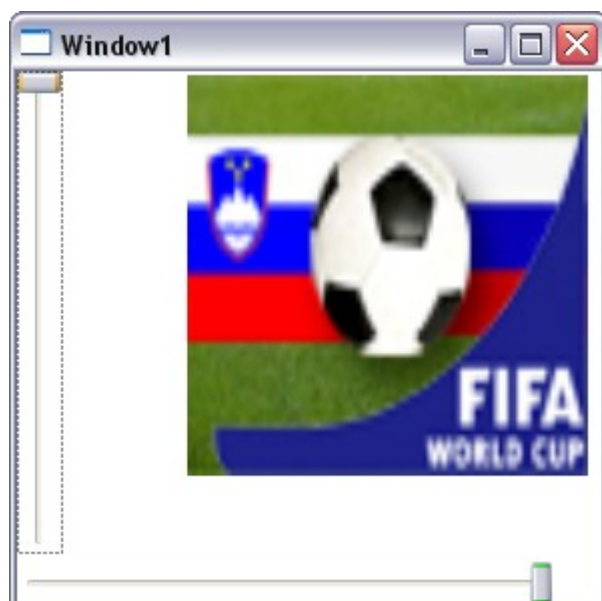
В этом примере показаны приемы работы с элементом Image.



В окне размещено несколько рисунков. Показано не сколько способов указания на файл рисунков. Также показан пример загрузки файла рисунка в коде (функция button1_Click).

Пример 2. Работа с Slider. (Source\1-2-WPF_UI\SliderEx)

Размеры рисунка меняются в зависимости от движения ползунков.

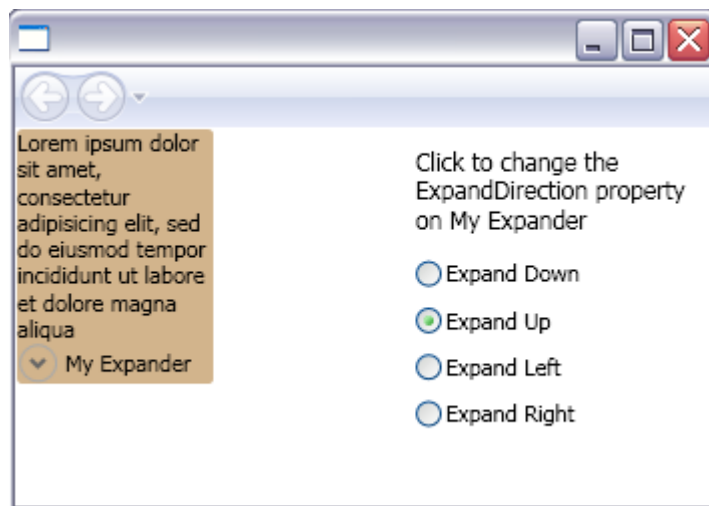




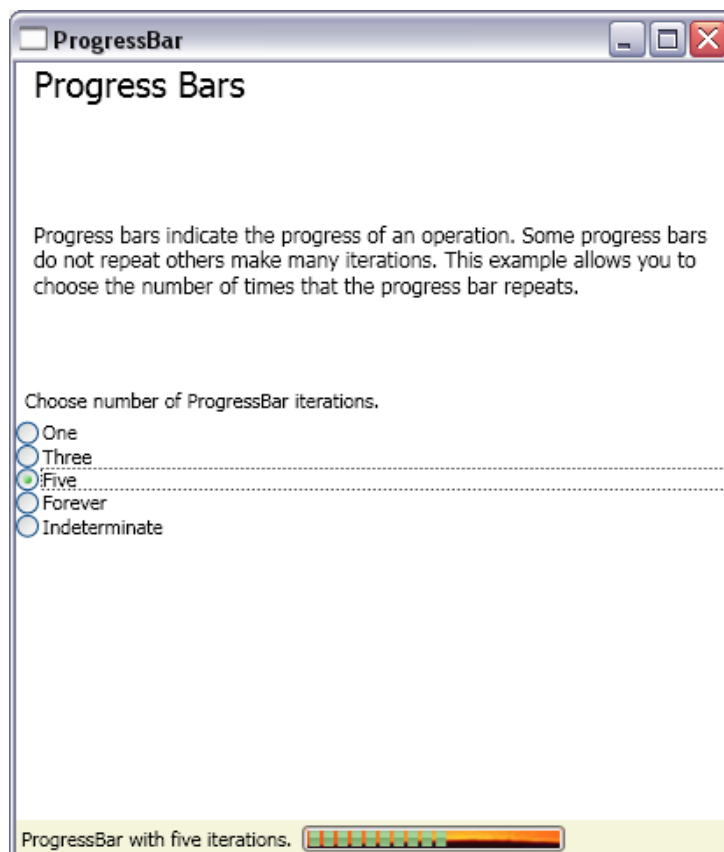
Обработчики движений меняют размеры картинки после изменения положения ползунков.

Пример 3. Работа с Expander. (D:\R\WPF-урок 2\Source\3-ExpanderExample)

В данном примере при выборе RadioButton меняются свойства компонента Expander.



Пример 4. Работа с ProgressBar. (Source\4-ProgressBar)



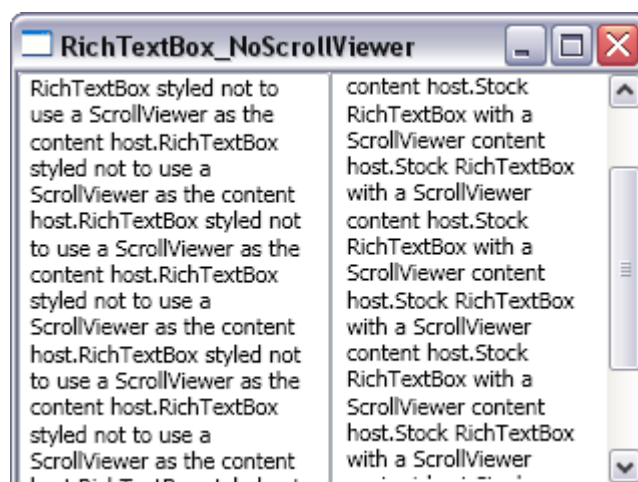


В данном примере при выборе `RadioButton` меняются свойства компонента `ProgressBar`. Показано несколько вариантов работы компонента, создание компонента в коде, работа с повторениями, загрузка картинок для оформления компонента и анимация.

Каждый `RadioButton` имеет свой обработчик, в котором выполняется примерно такой код.

```
//очистка контейнера
sbar.Items.Clear();
//создание надписи и добавление к контейнеру
Label lbl = new Label();
lbl.Background = new LinearGradientBrush(Colors.LightBlue,
Colors.SlateBlue, 90);
lbl.Content = "ProgressBar with one iteration.";
sbar.Items.Add(lbl);
//<Snippet1>
//создание индикатора
ProgressBar progbar = new ProgressBar();
progbar.IsIndeterminate = false;
progbar.Orientation = Orientation.Horizontal;
progbar.Width = 150;
progbar.Height = 15;
//создание анимации
Duration duration = new Duration(TimeSpan.FromSeconds(10));
DoubleAnimation doubleanimation = new DoubleAnimation(100.0,
duration);
progbar.BeginAnimation(ProgressBar.ValueProperty,
doubleanimation);
//</Snippet1>
//добавление к контейнеру
sbar.Items.Add(progbar);
```

Пример 5. RichTextBox без скрола. (Source\5-RichTextBox_NoScrollViewer)





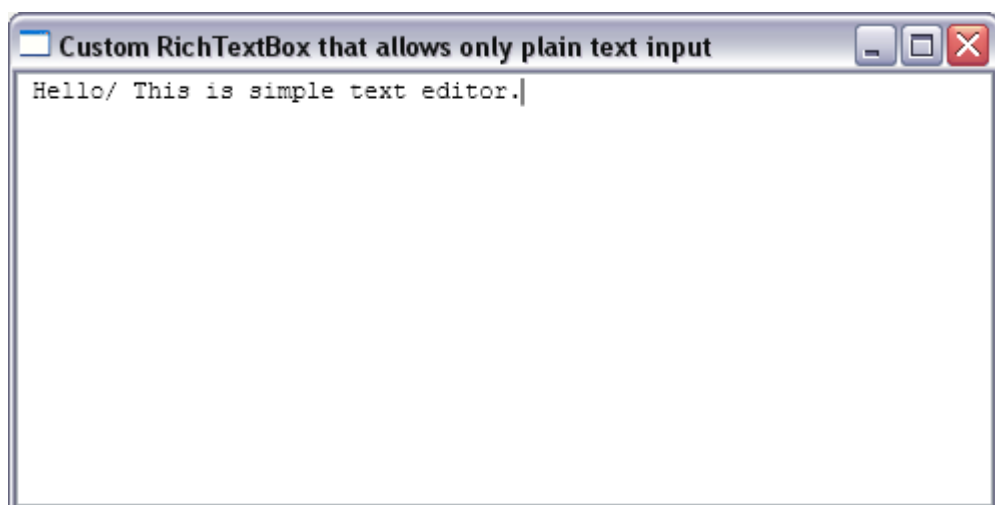
Показано создание RichTextBox с скроллингом и без на основе стиля.

Пример 6. RichTextBox с подсветкой гиперссылок. (Source\6-RichTextBox_HyperlinkAutoDetect)



В примере создан собственный компонент MyRichTextBox на основе стандартного RichTextBox. Он реагирует на нажатия клавиш, ищет слова и подсвечивает гиперссылку www.microsoft.com.

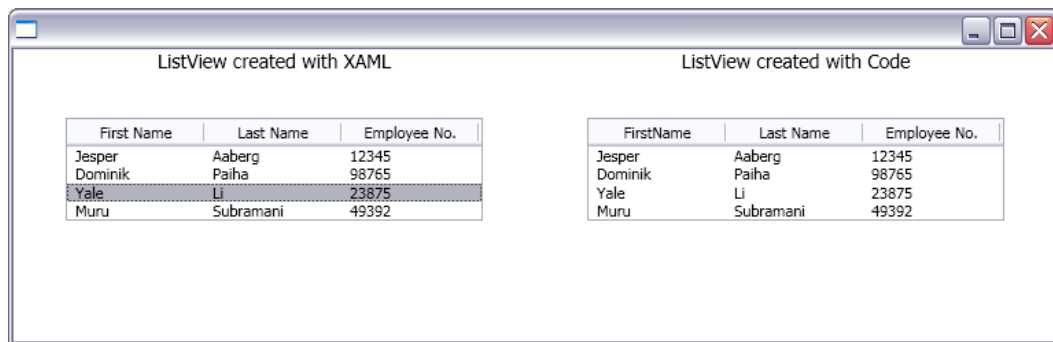
Пример 7. Простой редактор текста на основе RichTextBox (Source\7-CustomRichTextBox)



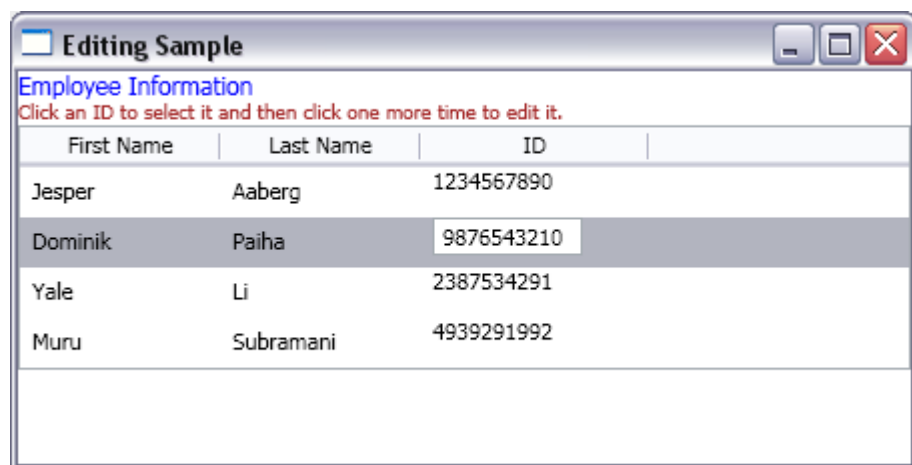
Разработан собственный элемент управления, который обрабатывает стандартные операции в контекстном меню.

Пример 8. Создание ListView в коде (Source\8-ListViewCode)

Показан пример создание компонента через XAML и с помощью кода.

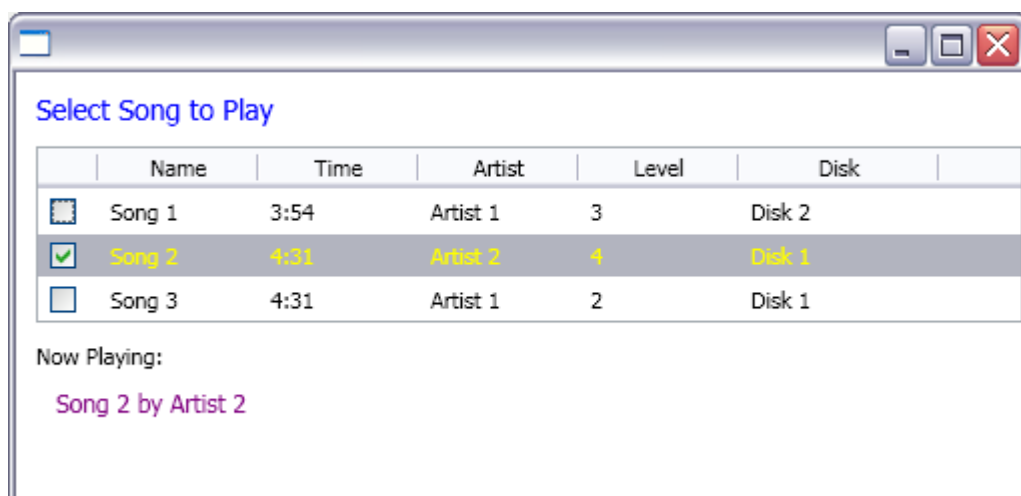


Пример 9. ListView с ячейками доступными для редактирования.
(Source\9-ListViewEditable)



Создан класс EditBox, который дает возможность редактировать содержимое ячеек ListView.

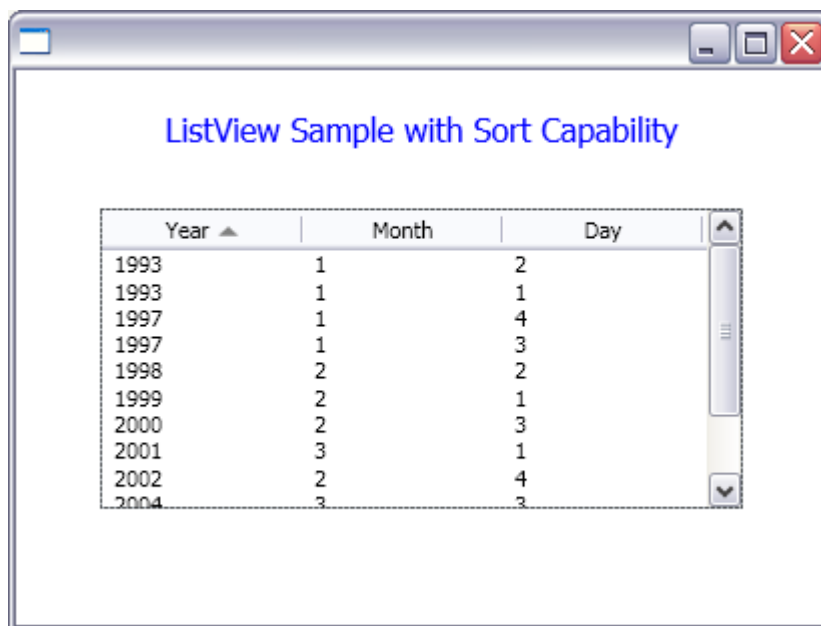
Пример 10. ListView с CheckBox. (Source\10-ListViewCheckBox)



В примере создается стиль для первой колонки с переключателями. На основе этого стиля создается таблица с колонкой с переключателями.

**Пример 11.** ListView с сортировкой. (\Source\11-ListViewSort)

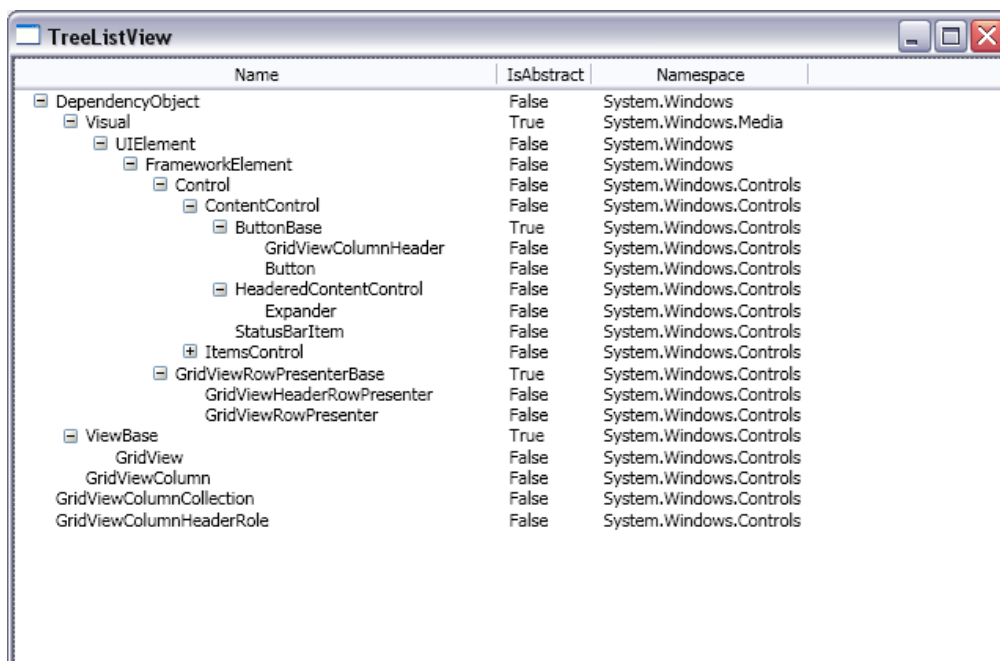
Нажатие на заголовки столбцов инициирует сортировку содержимого таблицы. Окно имеет обработчик клика по заголовку и функцию сортировки.

**Пример 12.** ListView с изменяемым видом. (Source\12-ListViewCustomView)



На основе класса `ViewBase` создано несколько видов для `ListView`. По нажатию на пункты контекстного меню меняется вид `ListView`.

Пример 13. Работа с `TreeView` (Source\13-TreeView)



Создается класс `TreeView` для отображения дерева в виде колонки элемента `GridView`.

Домашнее задание:

Галерея картинок. Написать программу для просмотра изображений. Пользователь выбирает папку, из которой нужно просмотреть картинки. Далее содержимое папки считывается (отобразить прогресс с помощью `ProgressBar`). Загружается первая картинка (размер регулируется `Slider`). Пользователь перелистывает картинки или может выбрать одну из них в `ListBox`. Информация о просмотре и файле картинки отображается в `Expander`.

Текстовый редактор. Напишите простой (а-ля Блокнот), но полнофункциональный текстовый редактор.

Календарь. Напишите приложение календарь с использованием `TreeView` и `ListView`. В `TreeView` есть 4 категории (Зима, Весна, Лето,



Осень) в каждой категории соответствующие месяцы. При выборе месяца ListView должен отобразить календарь на этот месяц.

Головоломка. Напишите игру Головоломка (как гаджет Головоломка в Windows 7, это пятнашки, у которых вместо цифр части картинки).