

Формально, к элементам управления относятся классы, унаследованные от класса [Control](#).

## Элементы управления содержимым

*Элементы управления содержимым* – это элементы, допускающие размещение единственного дочернего элемента, представляющего их содержимое. Элементы управления содержимым можно разбить на три подгруппы: кнопки, простые контейнеры, контейнеры с заголовком.

Все кнопки наследуются от абстрактного класса [ButtonBase](#). Этот класс содержит событие `Click`, булево свойство `IsPressed` и свойство `ClickMode` - оно определяет момент генерирования `Click`. Значениями `ClickMode` являются элементы одноимённого перечисления: `Release` (по умолчанию), `Press` и `Hover`.

`Button`, `RadioButton`, `CheckBox`.

Перейдем к разбору *простых контейнеров*, которые (согласно своему названию) предназначены для простого обрамления дочернего содержимого. Один из простых контейнеров - элемент [Label](#). (также `ToolTip`) Элемент управления [Frame](#) предназначен для изолирования своего содержимого от остальной части пользовательского интерфейса. Этот элемент напоминает аналог из HTML. Более того, [Frame](#) действительно может отображать HTML или XAML-страницу, если использовать его свойство `Source`:

```
<Frame Source="http://www.pinvoke.net" />
```

Последняя группа элементов управления содержимым – *контейнеры с заголовком*. Элемент управления [Expander](#) напоминает [GroupBox](#), но содержит в заголовке кнопку, которая позволяет спрятать и показать содержимое контейнера. Можно настроить направление «разворачивания» контейнера.

```
<Expander Header="Grammar">
    <StackPanel>
        <!-- содержимое StackPanel - три переключателя -->
    </StackPanel>
</Expander>
```

## Списковые элементы управления

*Списковые элементы управления* (кратко - *списки*) могут хранить и представлять коллекцию *элементов списка*. Для большинства списков доступна гибкая настройка внешнего вида, а также функции фильтрации, группировки и сортировки данных, однако в данном параграфе рассматриваются только простейшие аспекты работы со списками.

Списковые элементы управления можно разделить на три подгруппы:

1. Селекторы.
2. Меню.
3. Списки без категории.

*Селекторы* допускают индексацию и выбор элементов списка. К селекторам относятся элементы управления [ComboBox](#), [ListBox](#), [ListView](#), [DataGrid](#) и [TabControl](#).

Элемент управления [ListBox](#) отображает список и допускает множественный выбор элементов. Класс [ComboBox](#) позволяет выбрать один элемент из списка, отображая текущий выбор и раскрывая список элементов по требованию. Отдельный элемент [ComboBox](#) представлен объектом [ComboBoxItem](#) (унаследованным от [ListBoxItem](#)). Элемент управления [ListView](#) унаследован от [ListBox](#). Этот список позволяет настроить свой внешний вид при помощи свойства `View` с типом [ViewBase](#). Элемент управления [TabControl](#) - это простой набор страниц с закладками.

Перейдем к рассмотрению элементов управления из категории *меню* – [Menu](#) и [ContextMenu](#). Класс [Menu](#) может содержать коллекцию любых объектов, но ожидается, что будут использованы объекты [MenuItem](#) и [Separator](#). [Separator](#) – простой элемент, представляющий разделитель пунктов

меню. [MenuItem](#) — это контейнер с заголовком (наследник [HeaderedItemsControl](#)). В случае [MenuItem](#) заголовок определяет отображаемый пункт меню, а свойство содержимого `Items` может содержать элементы подменю. [MenuItem](#) поддерживает «горячие клавиши». Этот класс также содержит свойства `Icon` и `IsCheckable` и определяет события `Checked`, `Unchecked`, `SubmenuOpened`, `SubmenuClosed`, `Click`.

Элемент управления [ToolBar](#) представляет панель инструментов. Он просто группирует кнопки, разделители ([Separator](#)) и другие дочерние элементы на одной панели<sup>1</sup>. Хотя [ToolBar](#) можно разместить в произвольном месте, обычно один или несколько таких объектов размещают внутри элемента [ToolBarTray](#), что позволяет перетаскивать и переупорядочить панели инструментов. Элемент управления [StatusBar](#) группирует элементы подобно [ToolBar](#) или [Menu](#). Обычно элемент [StatusBar](#) размещается в нижней части окна.

### **Прочие элементы управления**

В WPF текстовыми элементами являются [TextBlock](#)<sup>2</sup>, [TextBox](#), [PasswordBox](#), [RichTextBox](#). Элемент [TextBlock](#) предназначен для отображения небольшой порции текста

Элемент [RichTextBox](#) — «продвинутая» версия [TextBox](#). Многие свойства у этих элементов общие, так как они унаследованы от одного базового класса [TextBoxBase](#). Содержимое [RichTextBox](#) сохраняется в свойстве `Document` типа [FlowDocument](#), который создан для поддержки XPS.

Элемент управления [PasswordBox](#) предназначен для ввода паролей. Можно сказать, что это упрощенная версия [TextBox](#) - не поддерживаются вырезание и копирование текста, не генерируются события `TextChanged` и `SelectionChanged`. В [PasswordBox](#) введенный пароль сохраняется в свойстве `Password`, а при изменении текста генерируется событие `PasswordChanged`. Символ, который отображается вместо букв пароля, настраивается при помощи свойства `PasswordChar`.

Элементы для представления диапазона [ProgressBar](#) и [Slider](#) хранят и отображают в некой форме числовое значение, попадающее в заданный диапазон. Оба элемента унаследованы от класса [RangeBase](#), имеющего свойства `Value`, `Minimum`, `Maximum` (все - типа `double`) и событие `ValueChanged`.

[ProgressBar](#) обычно используют для визуализации процесса выполнения длительной операции. Элементами для работы с датами являются [Calendar](#) и [DatePicker](#). Элемент [DatePicker](#) позволяет задать дату, набирая её с клавиатуры или применяя выпадающий элемент [Calendar](#).

---

<sup>1</sup> [ToolBar](#) переопределяет визуальные стили для кнопок и разделителей. Кнопка выглядит небольшой и плоской, а разделитель — как вертикальная черта.

<sup>2</sup> [TextBlock](#) — это не элемент управления, так как он не наследуется от [Control](#), а является прямым наследником [FrameworkElement](#).