WPF コントロールライブラリ

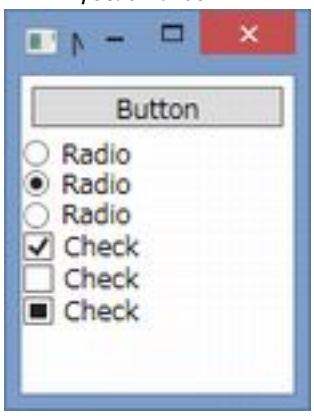
[WPF][Silverlight]

- WPF は UI パッケージに期待する標準コントロールの大部分を提供します (DataGrid など、少数の例外もあり)。
- コンテンツモデルとテンプレートがビジュアルの大幅なカスタマイズを可能とするため、コントロールの重要な部分は、コントロールが提供するもでると対話モデルということになる。

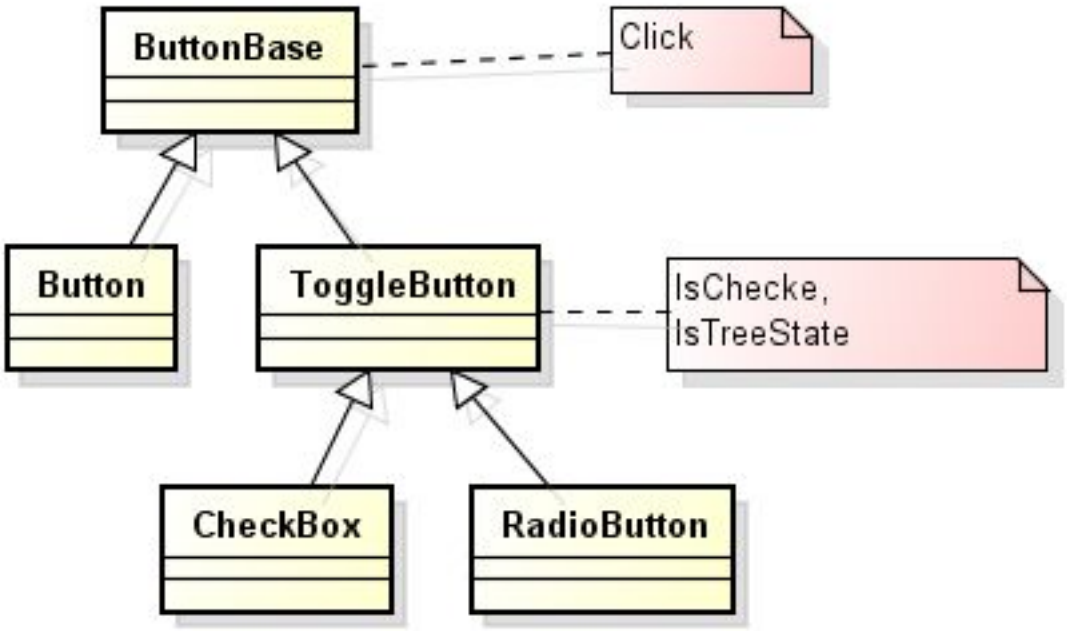
ボタン

- 基本的にボタンは、「クリック可能」なもの
- ボタンは、標準のボタンの規定のルックアンドフィール以外は、BottonBase のクリックイベント以外に重要な物は追加しません。
- CheckBox と RadioButton は、いずれも IsChecked(データモデル) プロパティと IsThreeState(対話モデル) プロパティをサポートするトグルボタンを表す。
- IsThreeState が True の場合、Checked、Unchecked、Indeterminate(IsThreeState が False の場合この状態にはならない) を切り替えることが出来る。

```
<StackPanel Name="MainPanel">
    <Button Margin="5" VerticalAlignment="Top">Button</Button>
    <RadioButton>Radio</RadioButton>
    <RadioButton IsChecked="True">Radio</RadioButton>
    <RadioButton>Radio</RadioButton>
    <CheckBox>Check</CheckBox>
    <CheckBox>Check</CheckBox>
    <CheckBox IsThreeState="True">Check</CheckBox>
</StackPanel>
```



クラス階層



リスト

・WPF はテンプレートのサポートが充実しているため、ListBox、ComboBox、DomainUpDown、ラジオボタンリストのような物までが、基本コントロールの上位に異なるテンプレートを適用するだけで実現できる。
・WPF には、ListBox、ComboBox、ListView、TreeView の 4 つの基本リストコントロールがある。

ソース

・すべてのリストコントロールには、2 つのソースのいずれかを使用して項目を代入できる。

■ Items プロパティ

・リスト内部のデータ項目リストにデータを追加

```
var list = new ListBox();
list.Items.Add("a");
list.Items.Add("b");
list.Items.Add("c");
```

■ ItemSource プロパティ

・リストが表示するデータ項目のコレクションをリストコントロールに提供

リストコントロールの外部にあるデータを維持できる

```
string[] items = { "1", "2", "3" };
var list = new ListBox();
list.ItemsSource = items;
```

ListBox と ComboBox

・オブジェクトモデルの観点からいうとこれらはほぼ同一
・ItemSource プロパティを使用することが推奨され、IEnumerable を実装する任意の型をソースとして使用できる。
・.NET3.0 からこれらのシナリオで使用するために特別に設計された ObservableCollection<T> という新しいコレクションが提供される。
・ObservableCollection<T> は、リストシナリオのデータソースとして変更追跡のための複数のインターフェースを実装している
・まったく新しいテンプレートを記述しなくてもコントロールの外観を調整出来るようになる一連のプロパティを提供。