



WPF

Microsoft MVP

Connor Park



Day 2

MVVM Pattern

Model-View- ViewModel(MVVM) Pattern overview

시작

- MVVM Pattern은 WPF (Windows Presentation Foundation) 및 Silverlight의 기능을 활용하여 사용자 인터페이스의 이벤트 중심 프로그래밍을 간소화하기 위해 Microsoft 아키텍처 Ken Cooper 및 Ted Peters가 개발

발표

- Microsoft의 WPF 및 Silverlight 아키텍트 중 한 명인 John Gossman은 2005년 자신의 블로그에서 MVVM을 발표

현재

- WPF, Xamarin.Forms, UWP 앱 개발에서 사용되며 웹에서도 비슷한 패턴인 MVC를 사용

핵심

- View와 ViewModel 사이에 Loosely-coupled 한 개발로 재사용성 증가
- 메모리의 효율적인 관리(Memory leak 방지)
- Unit Test 용이

MvvmLight, MvvmLightStd10(NuGet)

<http://www.mvmlight.net/>

대중적으로 사용되는 MVVM NuGet Package

Prism Framework를 사용하지 않는 경우 사용 추천

ViewModelLocator 사용

ViewModelBase

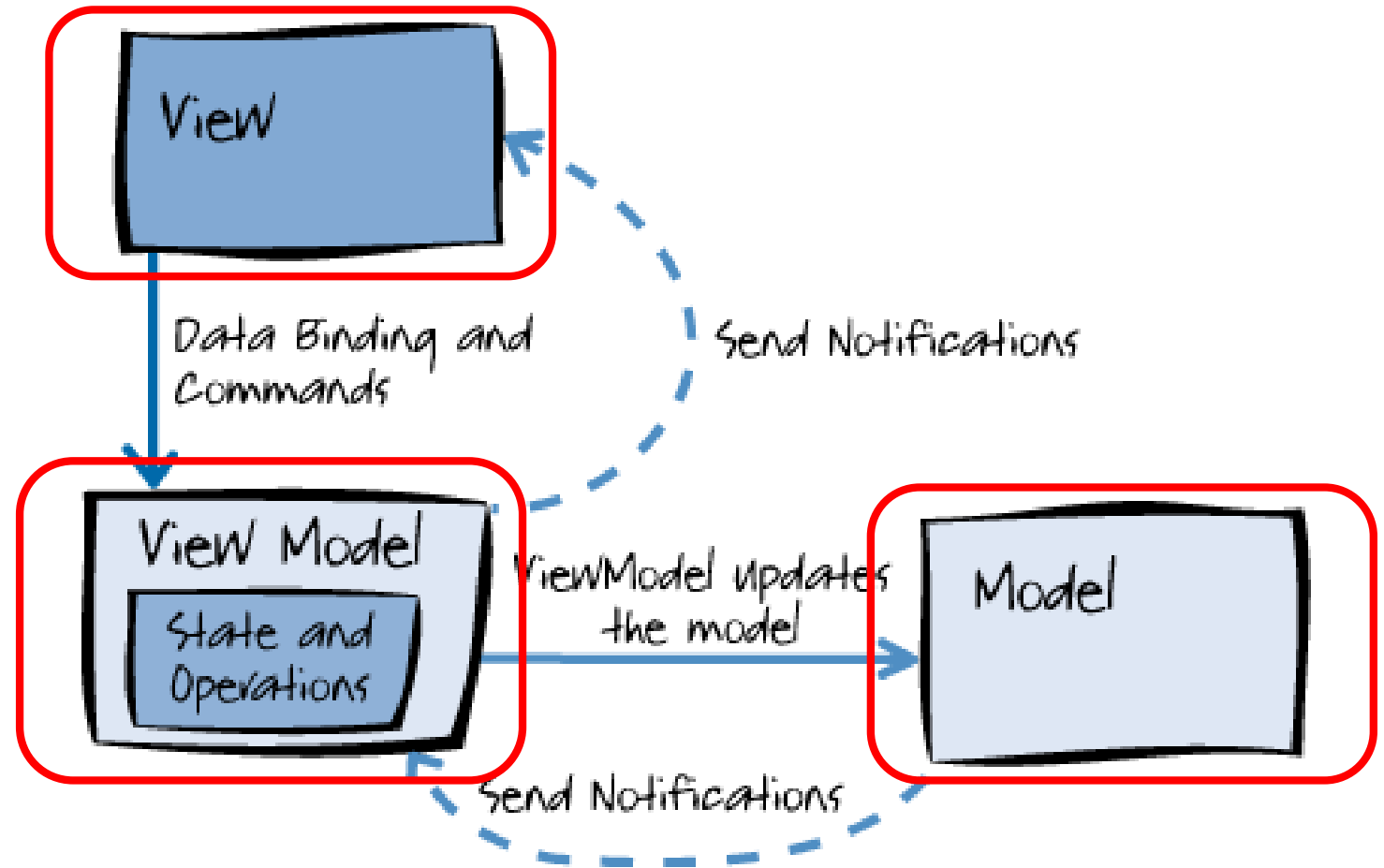
- MessengerInstance
- RaisePropertyChanged
- Set

Simpleloc 사용

- IoC : Inversion of Control, Dependency Inversion Principle, Dependency Injection

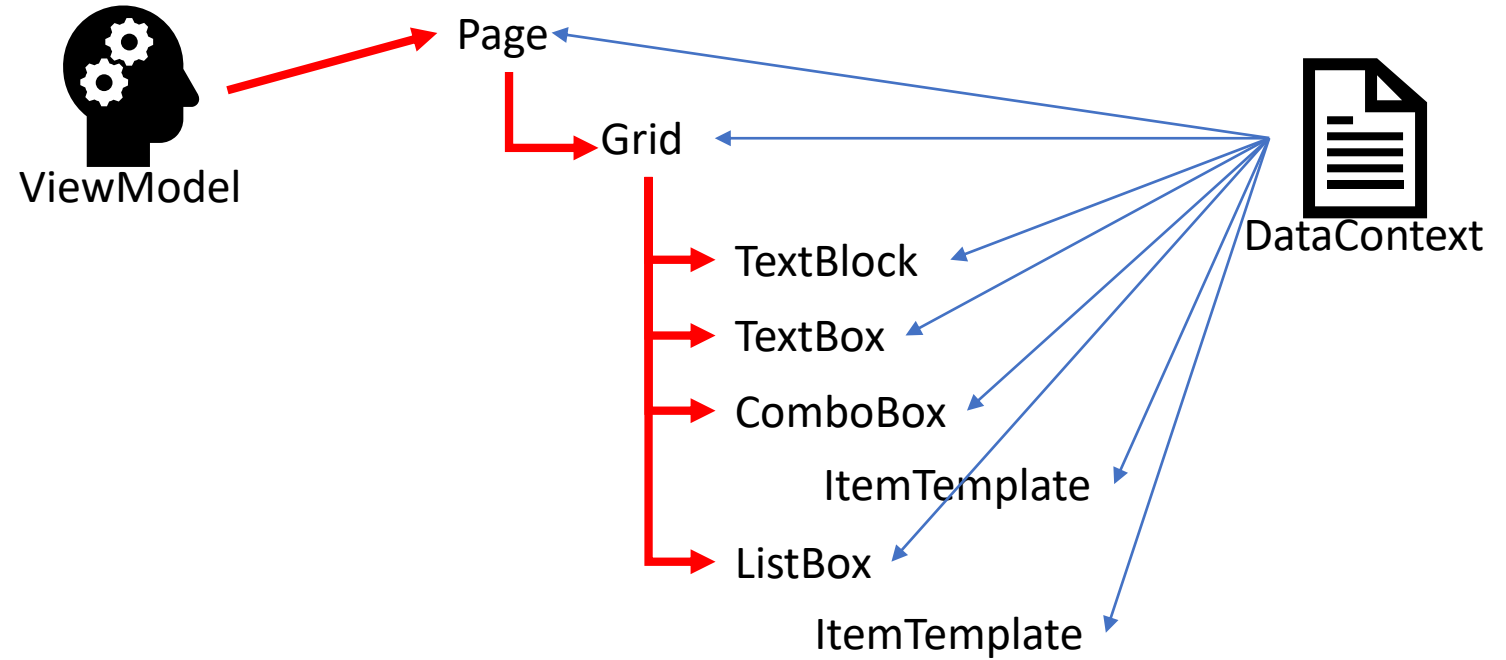
MVVM Pattern 구성

- Model : Person.cs
- View :
PersonDetailView.xaml
- ViewModel :
PersonDetailViewModel.cs
- 모델, 뷰와 뷰모델을
분리해서 개발
- 뷰모델은 뷰에서 하려고
하는 동작을 정의



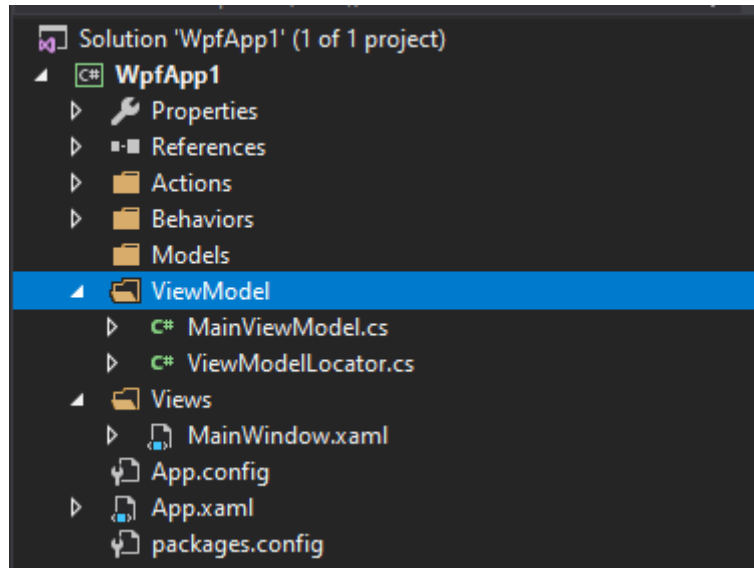
DataContext Property

- 대부분의 컨트롤들은 DataContext라는 프로퍼티를 가지고 있음
- ViewModel을 입력해서 사용
- VisualTreeHelper 상위 객체에서 하위객체로 상속



View and ViewModel connections

- 기본적인 방법
- ViewModelLocator를 이용하는 방법
- DesignTime 이용 방법



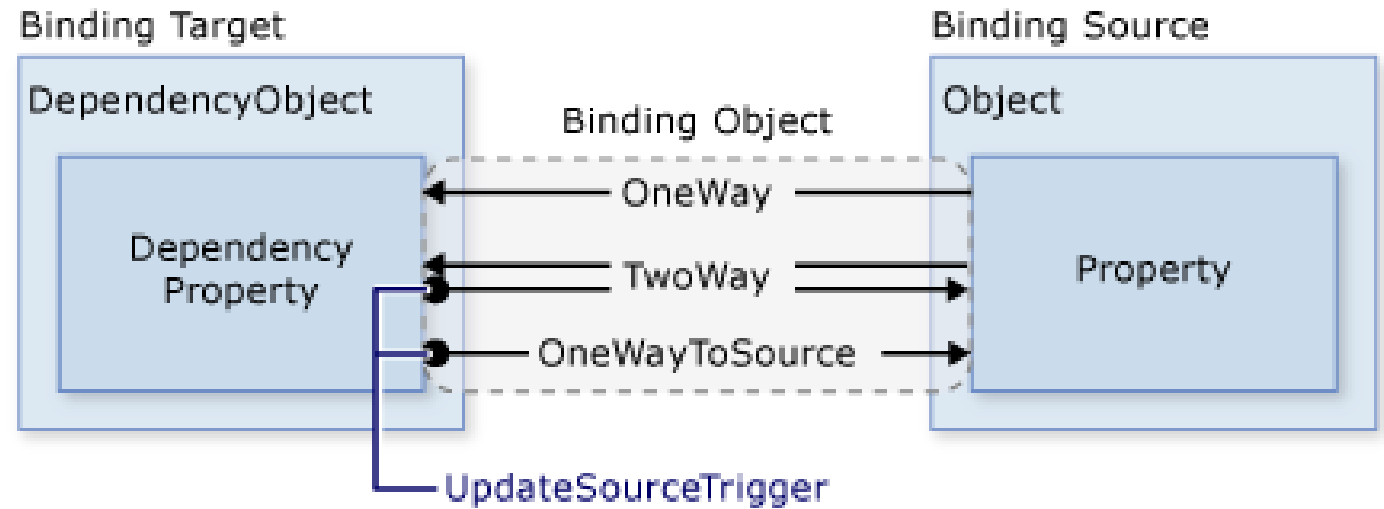
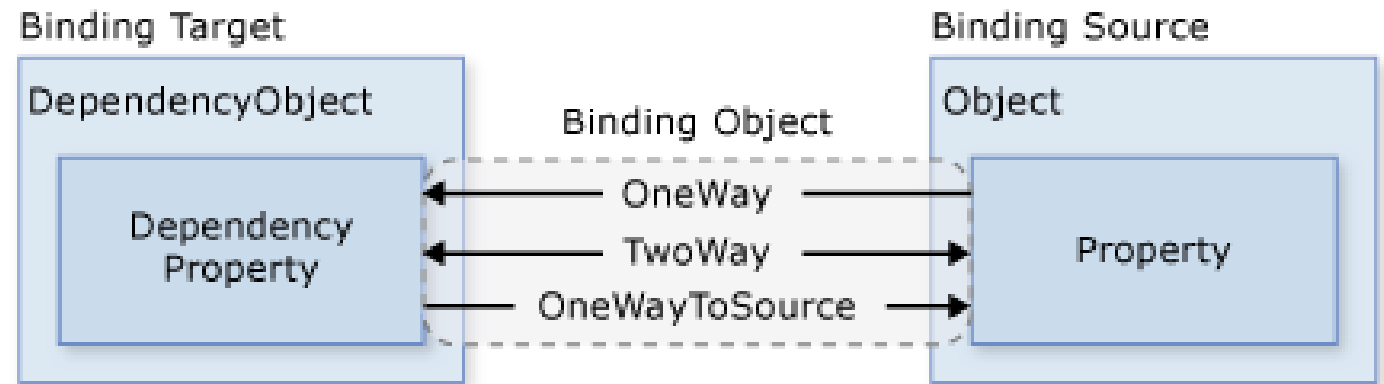
```
<d:DataContext>  
  <viewModel:MainViewModel/>  
</d:DataContext>
```

```
<Window.DataContext>  
  <Binding Mode="OneWay" Path="Main"  
Source="{StaticResource Locator}"/>  
</Window.DataContext>
```

```
if (IsInDesignMode)  
{  
  // Code runs in Blend --> create design time data.  
}  
else  
{  
  // Code runs "for real"  
  Init();  
}
```


Binding

- View
 - OneTime
 - OneWay
 - TwoWay
 - OneWayToSource
- ViewModel
 - INotifyPropertyChanged
 - MvvmLight, Prism BindableBase class



- Binding.xaml

OneWay

OneWay Binding

TwoWay

TwoWay Binding

TwoWay Binding Result

OneWayToSource

□ OneWayToSource

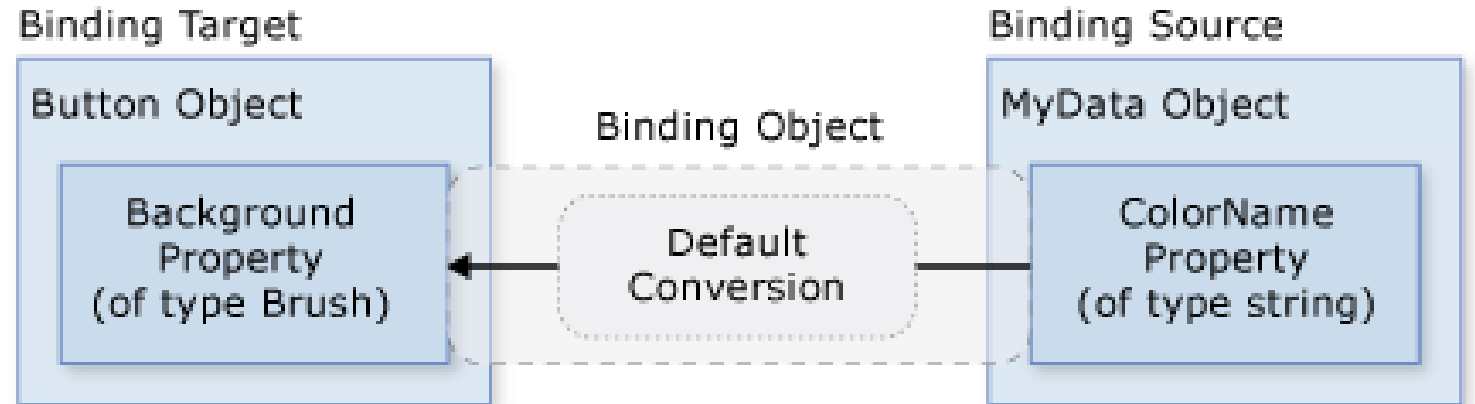
OneWayToSource Result

Code Binding

OneWay Binding with Code

IValueConverter

- ViewModel의 데이터를 View에 바인딩을 할 때 사용
- 원본 데이터 값을 변경시키지 않고, 보이는 형태를 변경



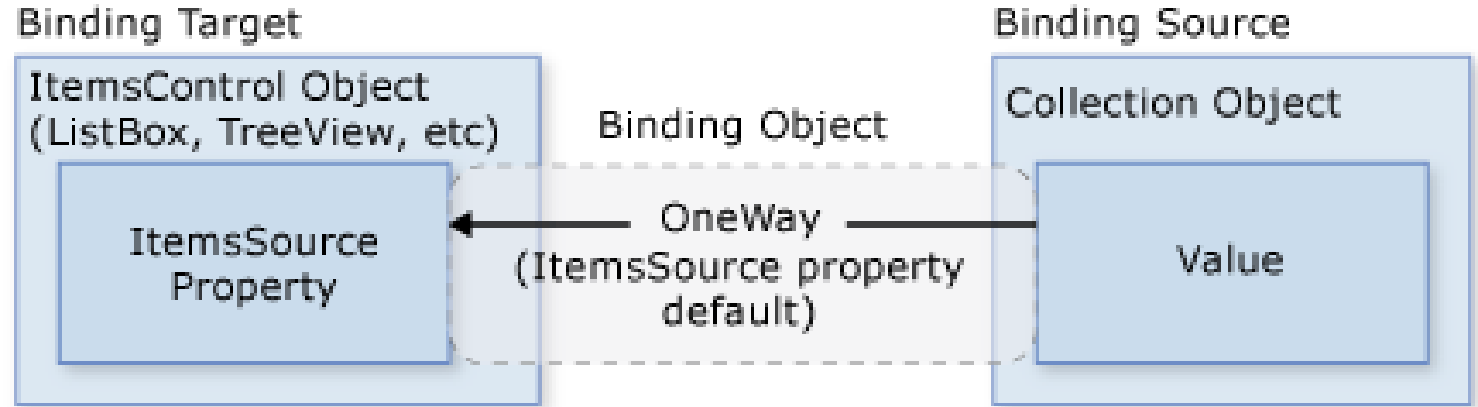
Grid

- Id 01
- Name 홍길동
- Age 1,111
- Address 서울
- Married 기혼

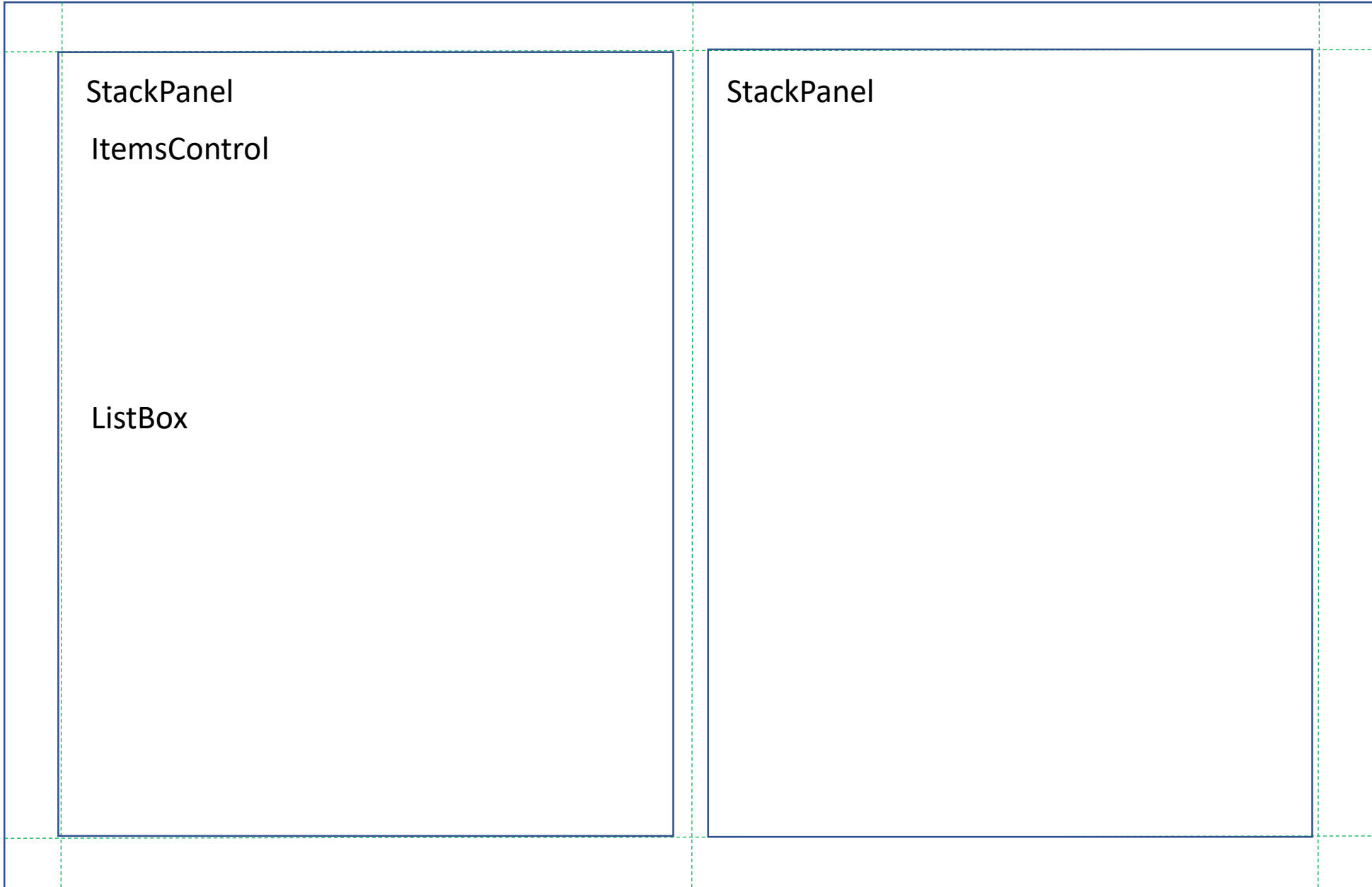
- ValueConverter.xam
|
- Grid를 추가, Row 5개, Col 2개 나눔
- TextBlock를 배치하고
- CurrentPerson 프로퍼티의 값을 Binding
- Age는 FormatString을 이용
- HasMarried 프로퍼티는 Converter를 만들고, True -> 기혼, False -> 미혼으로 표시

Binding to Collections

- 컬렉션 형태의 데이터 바인딩은 ItemsSource 프로퍼티를 이용합니다.
- 컬렉션 데이터의 변경 사항은 INotifyCollectionChanged 이벤트를 이용합니다.
- CollectionViewSource



Source collection type	Collection view type	Notes
IEnumerable	An internal type based on CollectionView	Cannot group items.
IList	ListCollectionView	Fastest.
IBindingList	BindingListCollectionView	

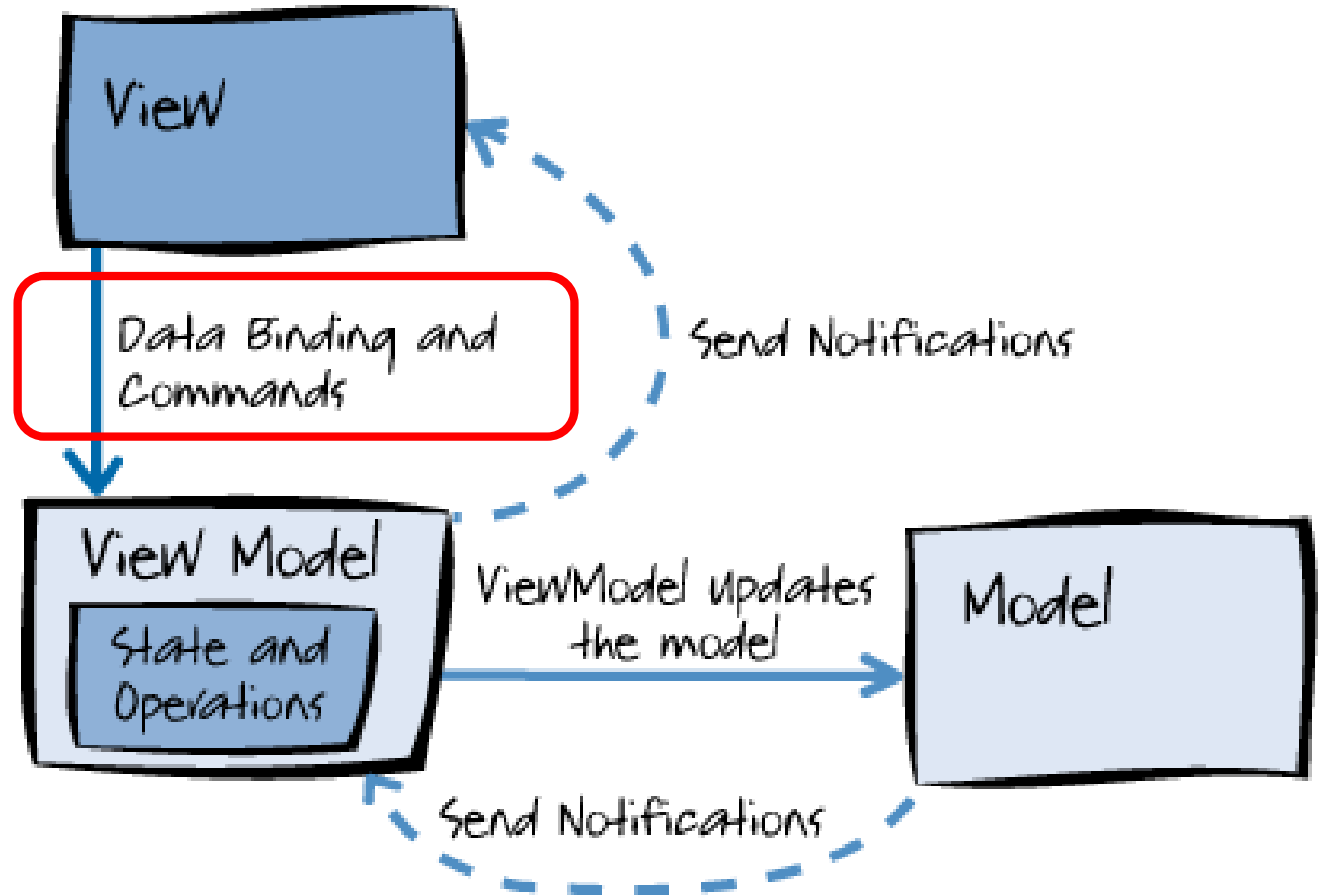


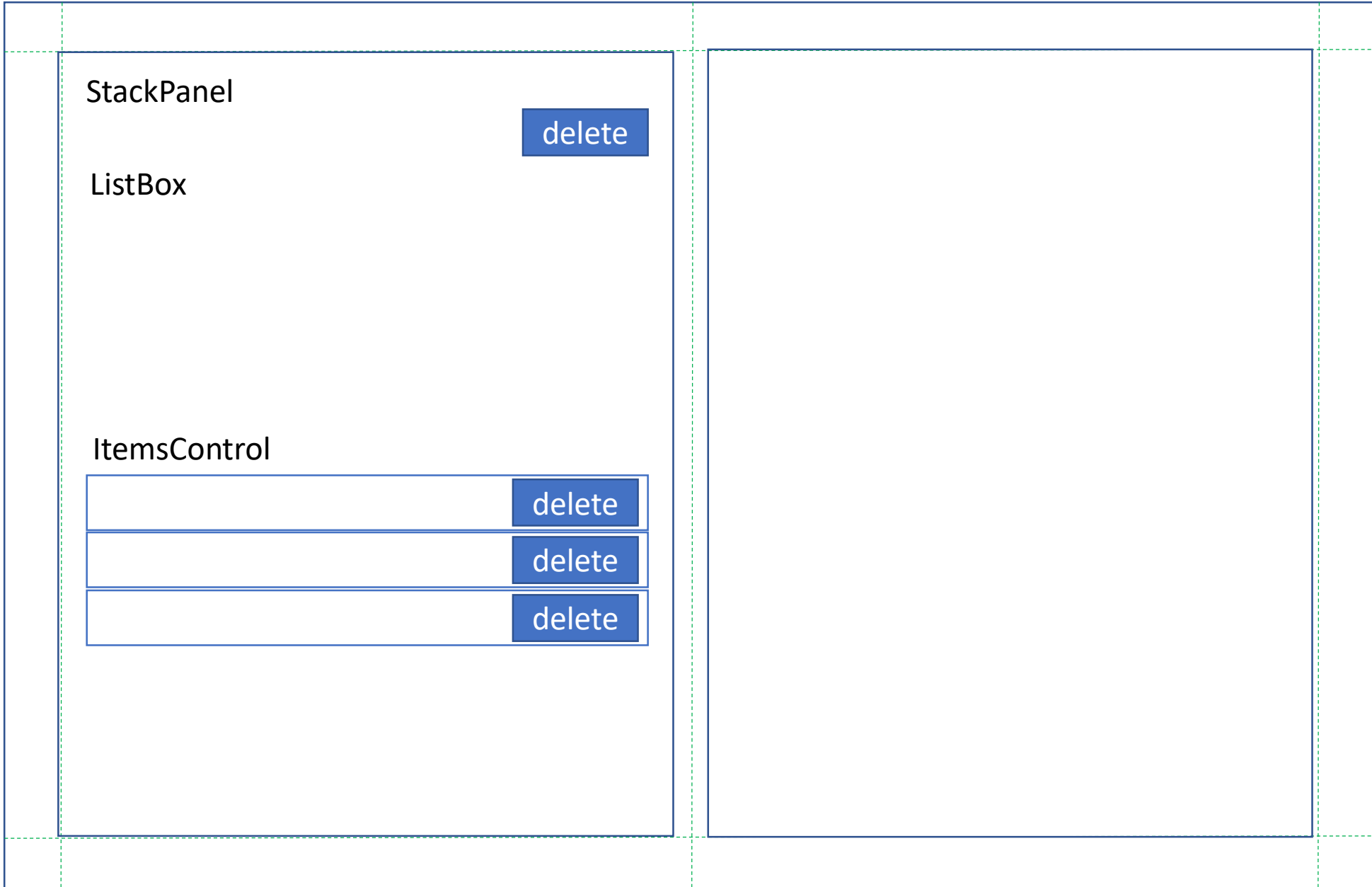
- BindingCollection.xml
- People List를 ItemsControl을 이용해서 출력
- People List를 ListBox를 이용해서 출력

- Live Visual Tree를 확인해서 복잡성을 확인

Command

- 사용자 인터렉션을 뷰모델에 전달
- View
 - Command property
 - CommandParameter property
 - EventTrigger
 - InvokeCommandAction
- ViewModel
 - ICommand Interface
 - DelegateCommand, RelayCommand





- Command.xaml
- People를 ListBox에 출력
- ListBox에서
아이템을 선택 후
delete 버튼을
클릭하면 선택된
아이템 삭제
- ItemsControl의
ItemTemplate에
delete 버튼 추가
- delete 버튼을
클릭하면 선택된
아이템 삭제

WPF deep-dive 1

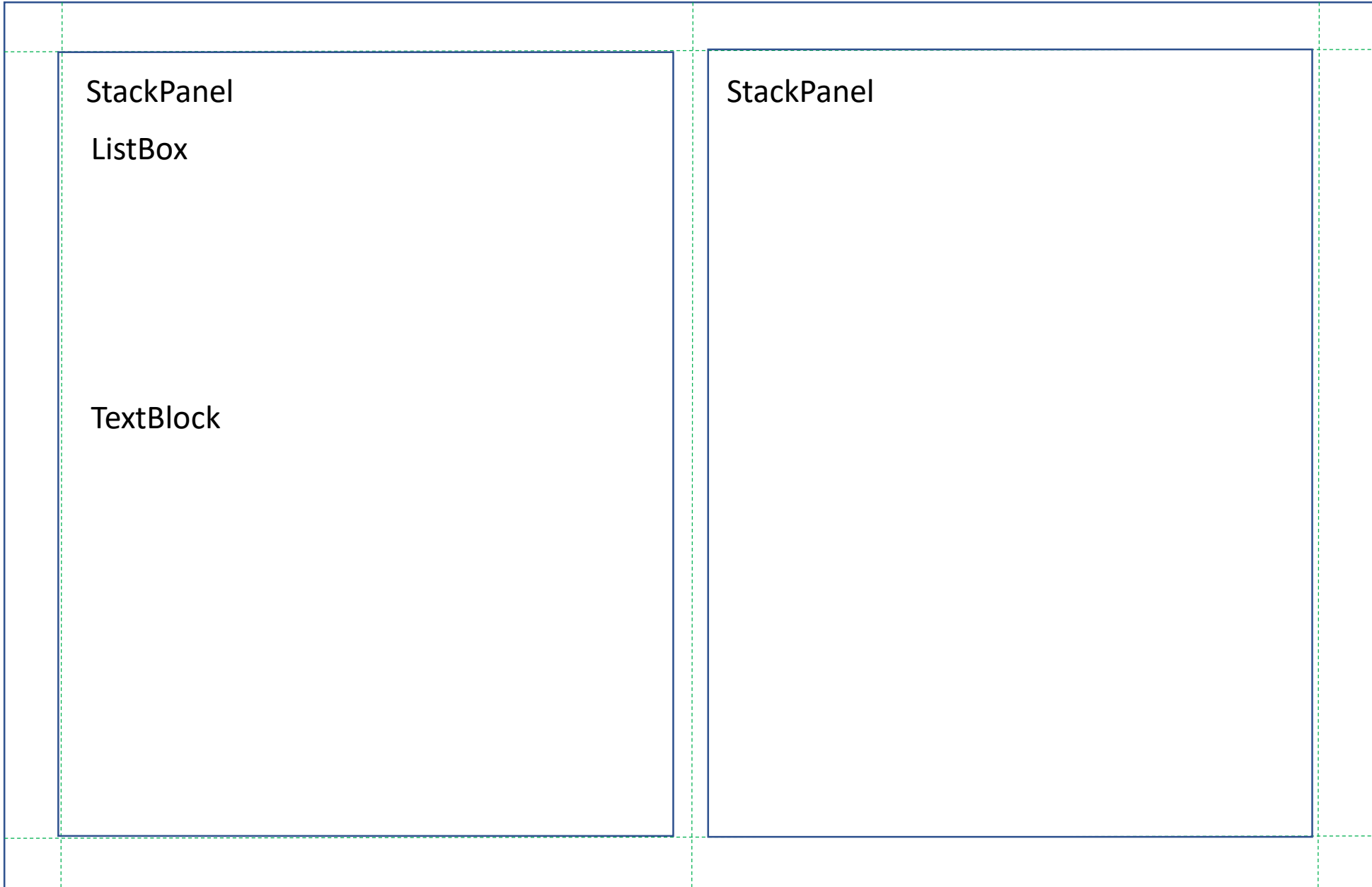
Item Selection

- ItemsSource(IEnumerable)
- SelectedItem(object)
- SelectedIndex(int)
- SelectedValue(object)
- SelectedValuePath(string)
- SelectionChanged(event)
- SelectionMode(enum)
- DisplayMemberPath(string)

Id	Name	Age	Address	HasMarried
1	지수	23	주소1	False
2	제니	23	주소2	True
3	로제	21	주소3	False
4	리사	22	주소4	True

```
<ListBox ItemsSource="{Binding People}"  
  SelectedItem="{Binding CurrentPerson, Mode=TwoWay}"  
  DisplayMemberPath="Name"/>
```

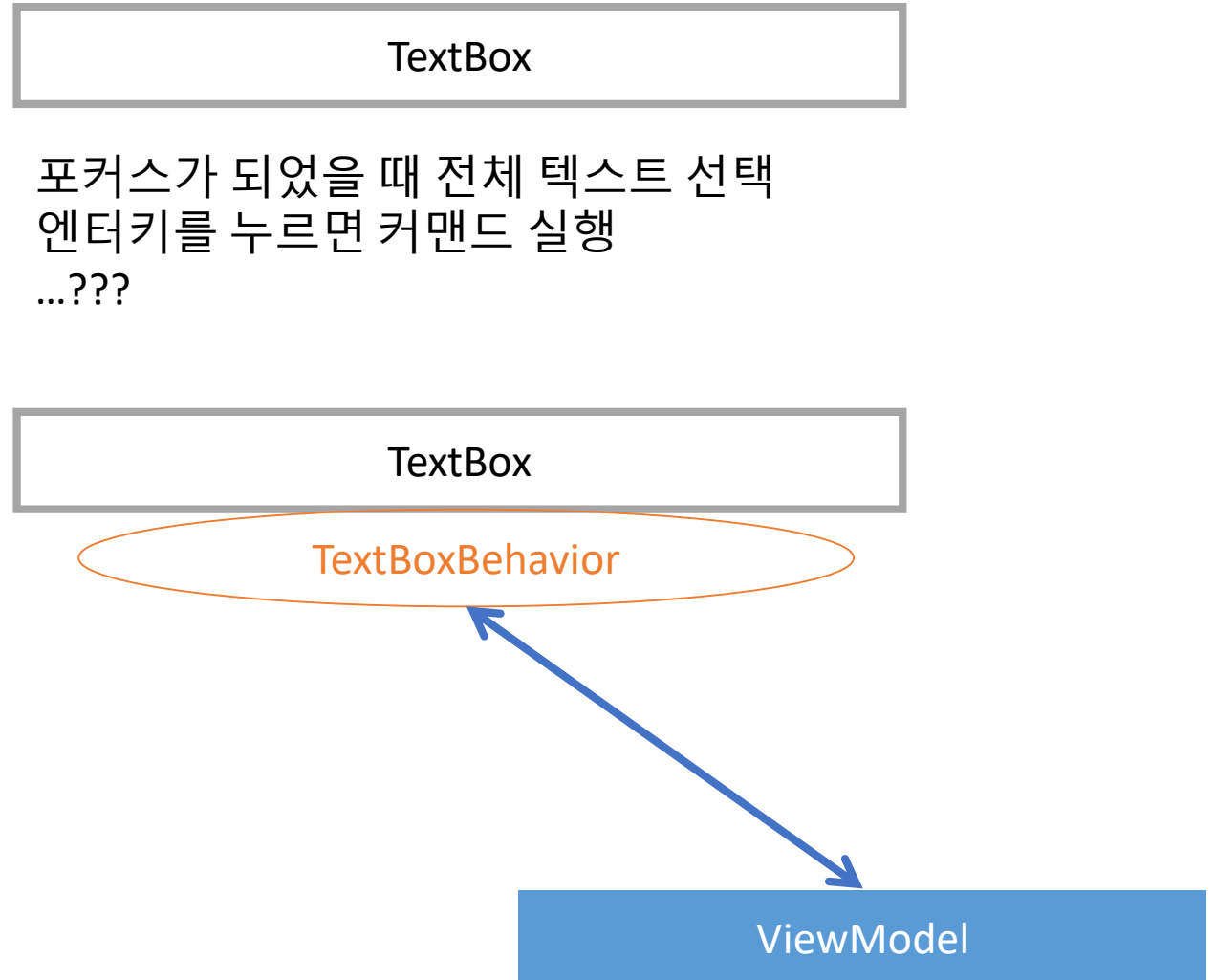
```
<ListBox ItemsSource="{Binding People}"  
  SelectedValue="{Binding CurrentPersonId, Mode=TwoWay}"  
  SelectedValuePath="Id"  
  DisplayMemberPath="Name"/>
```



- ItemSelection.xaml
- People 바인딩
- ListBox에 EventTrigger, InvokeCommandAction을 이용
- SelectionChanged 이벤트에 SelectionChangedCommand 연결
- ListBox에 선택된 아이템의 값을 TextBlock에 출력

Behavior

- 컨트롤을 뷰모델에서 쉽게 제어하기 위해 사용
- 바인딩만으로 해결하지 못하는 문제 해결
- 비즈니스 로직과 관련 없는 UI 조작
- OnAttached, OnDetaching
- AssociatedObject
- 재사용성 높임



Grid

- Id 01
- Name 홍길동
- Age 1,111
- Address 서울
- Married 기혼

StackPanel

- Behavior.xaml
- CurrentPerson 바인딩
- TextBoxBehavior 추가
- 데이터를 출력하는 TextBlock를 TextBox로 변경
- 각 TextBox에 Behavior 연결

IMultiValueConverter

- 2개 이상의 Binding이 필요한 경우 사용

특가 이벤트

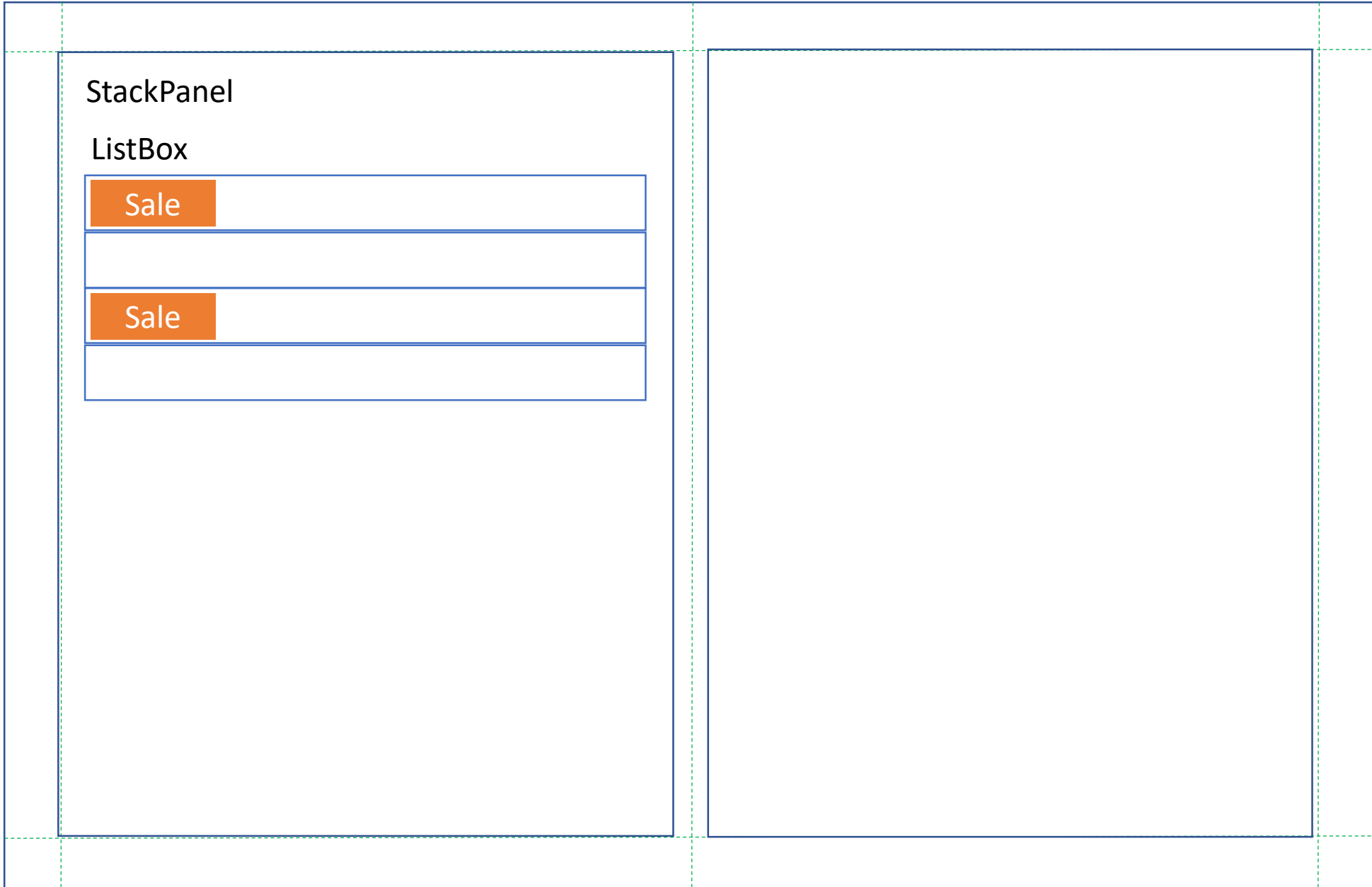
- 보이기 여부(Or)
 - 출시일 182일 지난 상품
 - 단가가 5,000원 이하

특가 이벤트

ShowEventMultiConverter

ViewModel





- BindingCollection.xml
- 상품 데이터 바인딩
- Product
 - Id, Int
 - Name, string
 - ReleaseDate
 - UnitPrice
- SaleMultiValueConverter를 이용해서 Sale 표시 출력 여부 조정

Style

- Html CSS
- 컨트롤의 모양을 정의
- TargetType
- 기본 Style
- Key Style
- BaseOn
- Page or ResourceDictionary
- Blend를 사용하면 쉬움

```
<!--기본 스타일-->
<Style TargetType="TextBlock">
    <Setter Property="FontSize" Value="12"/>
    <Setter Property="FontWeight" Value="SemiBold"/>
</Style>
```

```
<!--Key 스타일-->
<Style x:Key="boldTextBlockStyle"
    TargetType="TextBlock">
    <Setter Property="FontSize" Value="20"/>
    <Setter Property="FontWeight" Value="Bold"/>
</Style>
```

```
<!--BaseOn 스타일-->
<Style x:Key="italicTextBlockStyle"
    TargetType="TextBlock"
    BaseOn="{StaticResource boldTextBlockStyle}">
    <Setter Property="FontStyle" Value="Italic"/>
</Style>
```


StackPanel

TextBlock1

TextBlock2(**boldTextBlockStyle**)

TextBlock3(*italicTextBlockStyle*)

- StylePage.xaml 추가
- StackPanel 추가
- TextBlock 3개 추가
- Style 3개 추가
- TextBlock에 Style 적용

Template

- UI
 - ControlTemplate
 - ItemTemplate
 - ItemsPanelTemplate
 - HeaderTemplate
 - ...
- Template
 - DataTemplate
 - HierarchicalDataTemplate

```
<Style x:Key="ListBoxStyle1" TargetType="{x:Type ListBox}">
  <Setter Property="Background" Value="{StaticResource ListBox.Static.Background}"/>
  <Setter Property="BorderBrush" Value="{StaticResource ListBox.Static.Border}"/>
  <Setter Property="BorderThickness" Value="1"/>
  <Setter Property="Foreground" Value="{DynamicResource {x:Static
SystemColors.ControlTextBrushKey}}"/>
  <Setter Property="ScrollViewer.HorizontalScrollBarVisibility" Value="Auto"/>
  <Setter Property="ScrollViewer.VerticalScrollBarVisibility" Value="Auto"/>
  <Setter Property="ScrollViewer.CanContentScroll" Value="true"/>
  <Setter Property="ScrollViewer.PanningMode" Value="Both"/>
  <Setter Property="Stylus.IsFlicksEnabled" Value="False"/>
  <Setter Property="VerticalContentAlignment" Value="Center"/>
  <Setter Property="Template">
    <Setter.Value>
      <ControlTemplate TargetType="{x:Type ListBox}">
        <Border x:Name="Bd" BorderBrush="{TemplateBinding BorderBrush}"
BorderThickness="{TemplateBinding BorderThickness}" Background="{TemplateBinding Background}"
Padding="1" SnapsToDevicePixels="true">
          <ScrollViewer Focusable="false" Padding="{TemplateBinding Padding}">
            <ItemsPresenter SnapsToDevicePixels="{TemplateBinding
SnapsToDevicePixels}"/>
          </ScrollViewer>
        </Border>
        <ControlTemplate.Triggers>
          <Trigger Property="IsEnabled" Value="false">
            <Setter Property="Background" TargetName="Bd" Value="{StaticResource
ListBox.Disabled.Background}"/>
            <Setter Property="BorderBrush" TargetName="Bd" Value="{StaticResource
ListBox.Disabled.Border}"/>
          </Trigger>
          <MultiTrigger>
            <MultiTrigger.Conditions>
              <Condition Property="IsGrouping" Value="true"/>
              <Condition Property="VirtualizingPanel.IsVirtualizingWhenGrouping"
Value="false"/>
            </MultiTrigger.Conditions>
            <Setter Property="ScrollViewer.CanContentScroll" Value="false"/>
          </MultiTrigger>
        </ControlTemplate.Triggers>
      </ControlTemplate>
    </Setter.Value>
  </Setter>
</Style>
```

ListBox

Image	Name
	Memo

ListBox

RadioButtom1 RadioButtom2 RadioButtom3

- TemplatePage.xaml 추가
- 왼쪽 ListBox 추가
- People 연결
- ItemTempalte 추가
- 오른쪽 ListBox 추가
- RadioButton 3개 추가
- ItemsPanel 수정해서 가로 방향으로 변경

Selector

- 사용자가 자식 요소 중 특정 조건으로 선택
- ComboBox, ListBox, ListView
- ItemTemplateSelector
- ItemContainerStyleSelector
- StyleSelector
- ...

Id	Name	Age	Address	HasMarried
1	지수	23	주소1	False
2	제니	23	주소2	True
3	로제	21	주소3	False
4	리사	22	주소4	True

Id	Name	Age	Address	HasMarried
1	지수	23	주소1	False
2	제니	23	주소2	True
3	<i>로제</i>	21	주소3	<i>False</i>
4	리사	22	주소4	True

나이가 23이상은 주홍색 배경에 굵은 글씨
나이가 21이하는 노란색 배경에 이탤릭 글씨

ListBox

Id	Name	Age	Address	HasMarried
1	지수	23	주소1	False
2	제니	23	주소2	True
3	로제	21	주소3	False
4	리사	22	주소4	True

- SelectorPage.xaml 추가
- ListBox 추가
- People 연결
- ItemTemplateSelector 추가
- 23살 이상이면 주홍색 배경색과 굵은 글씨
- 21살 이하이면 노란색 배경색과 이탤릭 글씨

VisualState

- 상태에 따른 UI 형태 제어
- Blend를 이용해서 작업

```
<vsm:VisualStateManager.VisualStateGroups>
  <vsm:VisualStateGroup x:Name="CommonStates">
    <vsm:VisualState x:Name="Normal" >
      <Storyboard>
        <ObjectAnimationUsingKeyFrames Storyboard.TargetName="BackgroundEllipse"
Storyboard.TargetProperty="Stroke">
          <DiscreteObjectKeyFrame KeyTime="0" Value="{DynamicResource HpGray13Brush}" />
        </ObjectAnimationUsingKeyFrames>
      </Storyboard>
    </vsm:VisualState>
    <vsm:VisualState x:Name="MouseOver">
      <Storyboard>
        <ObjectAnimationUsingKeyFrames Storyboard.TargetName="BackgroundEllipse"
Storyboard.TargetProperty="Stroke">
          <DiscreteObjectKeyFrame KeyTime="0" Value="{DynamicResource HpGray15Brush}" />
        </ObjectAnimationUsingKeyFrames>
      </Storyboard>
    </vsm:VisualState>
    <vsm:VisualState x:Name="Pressed">
      <Storyboard>
        <ObjectAnimationUsingKeyFrames Storyboard.TargetName="BackgroundEllipse"
Storyboard.TargetProperty="Stroke">
          <DiscreteObjectKeyFrame KeyTime="0" Value="{DynamicResource GreenBrush}" />
        </ObjectAnimationUsingKeyFrames>
      </Storyboard>
    </vsm:VisualState>
    <vsm:VisualState x:Name="Disabled">
      <Storyboard>
        <ObjectAnimationUsingKeyFrames Storyboard.TargetName="BackgroundEllipse"
Storyboard.TargetProperty="Stroke">
          <DiscreteObjectKeyFrame KeyTime="0" Value="{DynamicResource HpGray1Brush}" />
        </ObjectAnimationUsingKeyFrames>
        <ObjectAnimationUsingKeyFrames Storyboard.TargetName="CheckGlyph" Storyboard.TargetProperty="Fill">
          <DiscreteObjectKeyFrame KeyTime="0" Value="{DynamicResource Gray1Brush}" />
        </ObjectAnimationUsingKeyFrames>
      </Storyboard>
    </vsm:VisualState>
  </vsm:VisualStateGroup>
```

Day 2 정리
