StackPanel

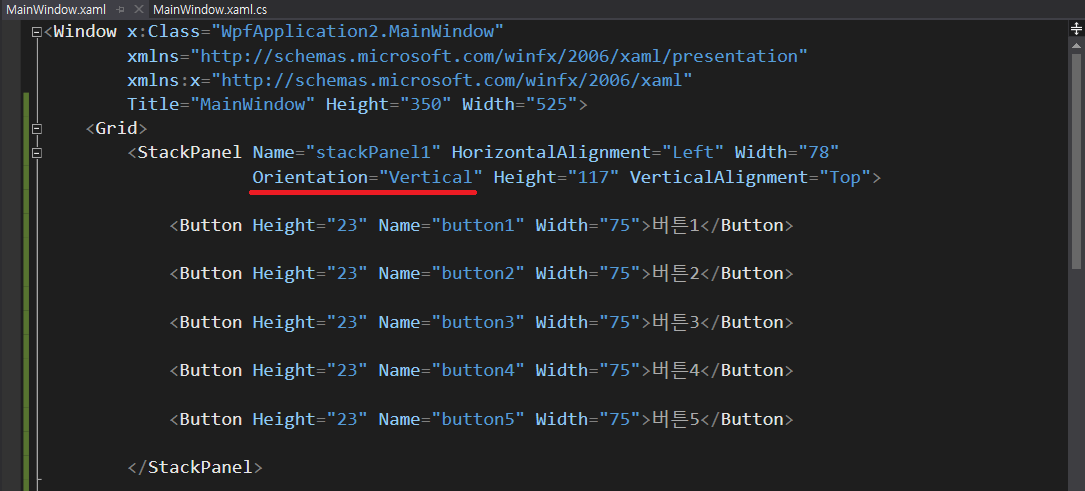
**StackPanel이란?**

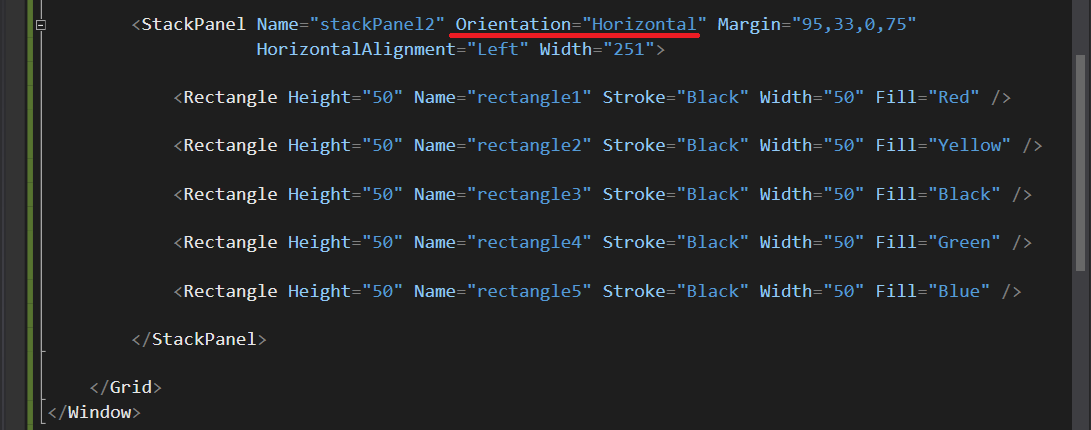
StackPanel을 사용하면 요소를 할당된 방향으로 “쌓을” 수 있다. 보통 다른 패널과 결합되어 많이 사용된다.

**StackPanel 의 장점과 단점**

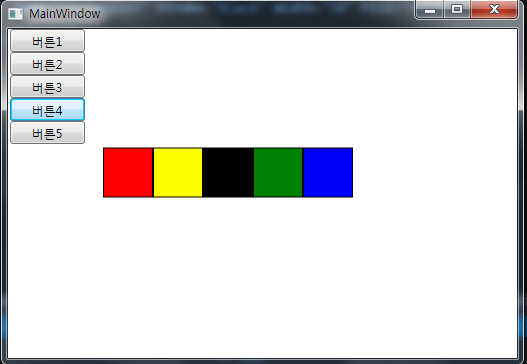
- **Grid**만큼 유연하지 않지만(원래 **StackPanel**은 작게 구분된 영역을 정렬하는데 유용하다.)  Style을 설정함으로써 Grid와 비슷한 유연함을 가질 수 있다.

- 기존 UI 엘리먼트의 변경, 새로운 엘리먼트 추가 시 번호를 매기는 번거로움이 없다.





위의 코드는 **StackPanel**의 요소로 각각 버튼 다섯 개 사각형 다섯 개를 넣어 StackPanel안의 요소들의 나열을 보기 위해 만든 샘플로 첫 번째 **StackPanel**의 버튼들은 “Oriantaion” 속성이 “Vertical”로 수직 나열된 것을 볼 수 있고, 두 번째 StackPanel안의 사각형 요소들은 “Oriantaion” 속성이 “Horizontal”로 수평 나열된 것을 볼 수 있다.

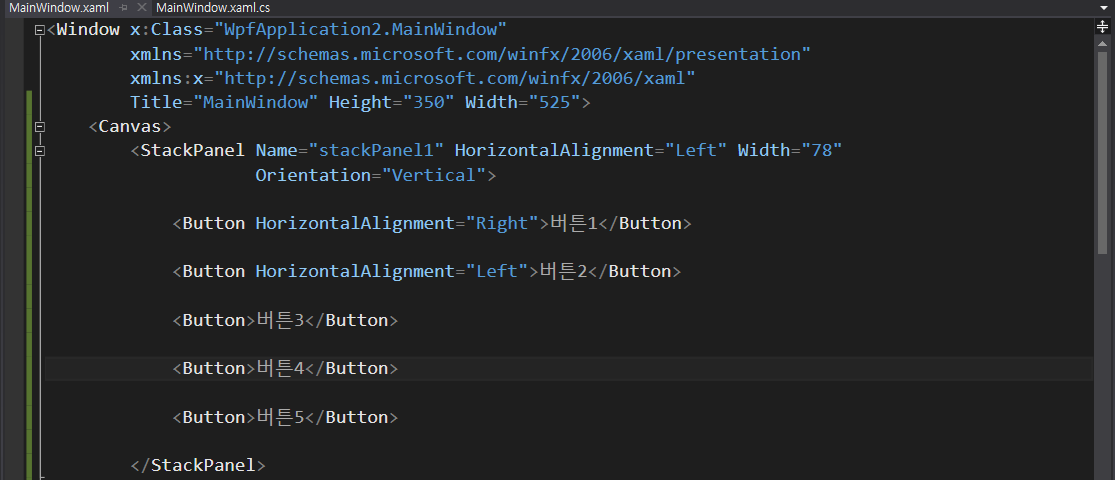


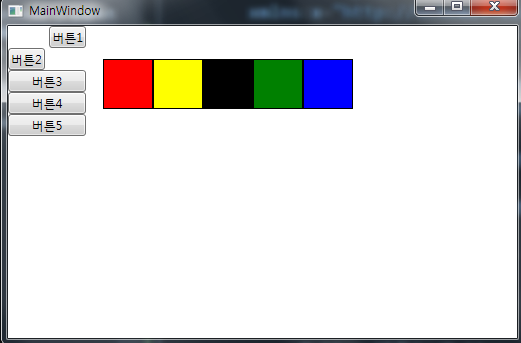
<결과>

StackPanel은 자식요소를 행이나 열로 나열하는데 매우 간단한 컨트롤로 StackPanel에 정의된 속성을 사용하여 세로로 흐르거나(기본 설정) 가로로 흐를 수 있다. StackPanel의 기본 스택 방향은 수직으로 StackPanel의 흐름을 제어하려면 Orientation속성을 사용하여 Orientation="Horizontal" 또는 Orientation="Vertical" 로 설정을 해준다.

.

또한 자식요소에 HorizontalAlignment, VerticalAlignment 속성을 설정하면 부모 패널과의 남는 공간에 대한 배치를 다르게 할 수 있다.



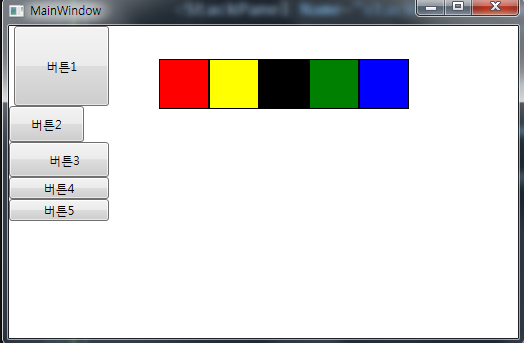


<결과>

버튼1은 오른쪽으로 배치되며 컨텐츠의 크기에 맞추어 Width가 설정되며 버튼2는 왼쪽으로 배치되며 또한 컨텐츠의 크기에 맞추어 Width가 설정된다.

Padding 속성을 사용하면 요소와 컨텐츠 간의 여백(안쪽)을 결정할 수 있다.





<결과>

1번 버튼은 Left, Top, Right, Bottom의 모든 안쪽 여백이 30으로 설정되었고, 2번 버튼은 Left, Right가 20으로 Top, Bottom이 8으로 설정되어 있다. 3번 버튼은 Left, Top, Right, Bottom 을 각각 80, 8, 9, 7으로 설정하였다.

**05\_StackPanel 예제**

**예제1) StackTenButtons**

public StackTenButtons()

{

Title = "Stack Ten Buttons";

StackPanel stack = new StackPanel();

Content = stack;

Random rand = new Random();

for (int i = 0; i < 10; i++)

{

Button btn = new Button();

btn.Name = ((char)('A' + i)).ToString();

btn.FontSize += rand.Next(10);

btn.Content = "Button " + btn.Name + " says 'Click me'";

btn.Click += ButtonOnClick;

stack.Children.Add(btn);

stack.Background = Brushes.Aquamarine;

}

stack.AddHandler(Button.ClickEvent, new RoutedEventHandler(ButtonOnClick));

}

void ButtonOnClick(object sender, RoutedEventArgs args)

{

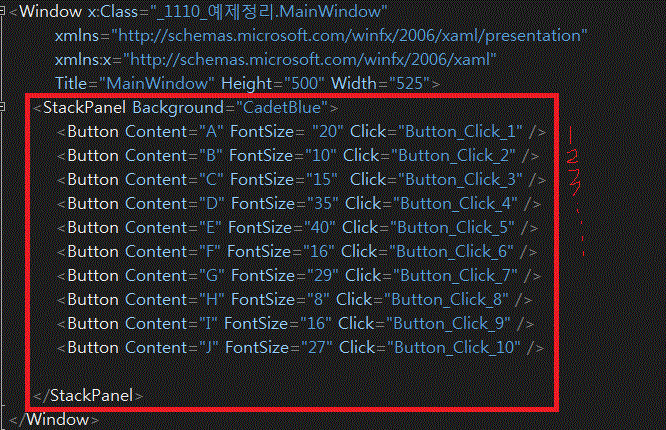
Button btn = args.Source as Button;

MessageBox.Show("Button " + btn.Name + " has been clicked",

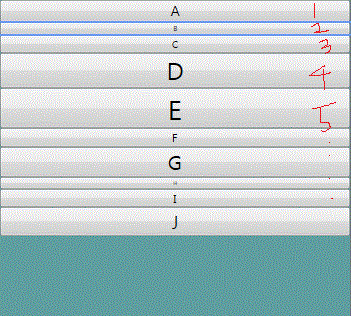
"Button Click");

}

**\*바꾸기전 예제 – C# code**

****

**\*코드 작성 – Xaml**

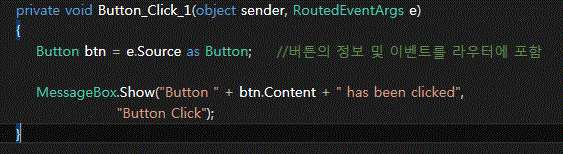
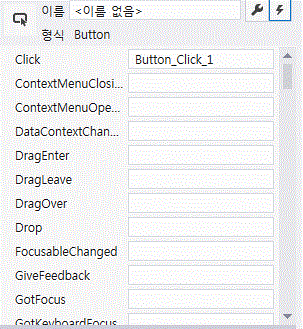
****

**\*Xaml코드 작성후 디자인뷰**

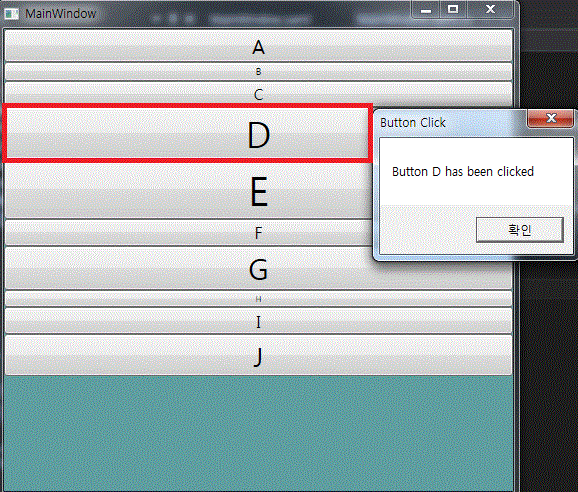
**-StackPanel 은 자식요소를 수직으로 자동배치 한다.**

**-StackPanel 안에 만들어진 컨트롤들은 만들어진 순서대로 위쪽에 생성되고**

**그 다음 생성된 컨트롤들은 그 밑에 순차적으로 나열되게 된다.**

****

**\*이벤트 설정**

****

**\*버튼 클릭 이벤트**

**원하는 이벤트를 클릭해 그 이벤트 호출시 발생코드를 작성할수 있다. 본 예제에서는 버튼 클릭시 메시지박스를 출력해준다.**

**예제2) StackthirtyButtons**

public StackThirtyButtons()

{

Title = "Stack Thirty Buttons";

SizeToContent = SizeToContent.WidthAndHeight;

ResizeMode = ResizeMode.CanMinimize;

AddHandler(Button.ClickEvent, new RoutedEventHandler(ButtonOnClick)); // 라우터를 통해 모든 버튼은 같은 Click 이벤트핸들러를 공유

StackPanel stackMain = new StackPanel(); // MainStackPanel을 하나 지정해 놓고

stackMain.Orientation = Orientation.Horizontal;

stackMain.Margin = new Thickness(5);

Content = stackMain;

for (int i = 0; i < 3; i++)

{

StackPanel stackChild = new StackPanel();

stackMain.Children.Add(stackChild); // 여기서 3개의 stackChild 패널을 Main에 담는다.

for (int j = 0; j < 10; j++)

{

Button btn = new Button();

btn.Content = "Button No. " + (10 \* i + j + 1);

btn.Margin = new Thickness(5);

stackChild.Children.Add(btn);

}

}

}

void ButtonOnClick(object sender, RoutedEventArgs args)

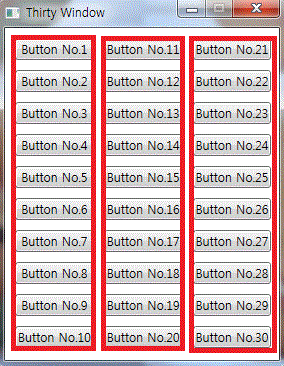
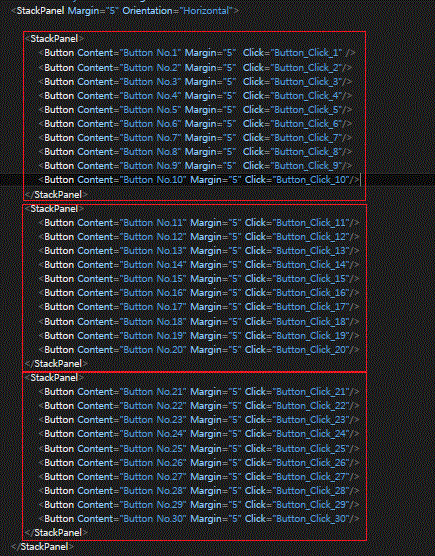
{

MessageBox.Show("You clicked the button labeled " +

(args.Source as Button).Content);

}

**\*바꾸기전 예제 – C# Code**

****

**\*코드 작성 – Xaml \*Xaml코드 작성후 디자인뷰**

**Main StackPanel**

**Stack Child 1**

**버 튼 10개**

**Stack Child 2**

**Stack Child 3**

**버 튼 10개**

**버 튼 10개**

* **간략 정리**

**본 예제는 StackPanel 내에 StackPanel-Child(자식)을 만들어 패널을 중첩해서 사용할수 있는 것을 알려주고 Orientation 속성을 통해 각 패널들을 수평으로 배치하는 방법도 알수 있는 예제이다**

**예제3) StackFiftyButtons**

public ScrollFiftyButtons()

{

Title = "Scroll Fifty Buttons";

SizeToContent = SizeToContent.Width;

AddHandler(Button.ClickEvent, new RoutedEventHandler(ButtonOnClick));

//ScrollViewer scroll = new ScrollViewer(); //스크롤 생성해서 컨텐츠에 넣음

//Content = scroll;

Viewbox view = new Viewbox(); //이 방법을 사용하면 창에 크기에 맞춰 버튼도 줄어 들기 떄문에 효율적이지는 않다.

Content = view;

StackPanel stack = new StackPanel();

stack.Margin = new Thickness(5);

//scroll.Content = stack; //스크롤 컨텐츠에 페널을 넣음

view.Child = stack;

//stack.AddHandler(Button.ClickEvent, new RoutedEventHandler(ButtonOnClick)); //버튼 이벤트 핸들러 따로 등록

for (int i = 0; i < 50; i++)

{

Button btn = new Button();

btn.Name = "Button" + (i + 1);

btn.Content = btn.Name + " says 'Click me'";

btn.Margin = new Thickness(5);

stack.Children.Add(btn);

}

}

void ButtonOnClick(object sender, RoutedEventArgs args)

{

Button btn = args.Source as Button; //args의 Source를 버튼으로 형변환 시키고

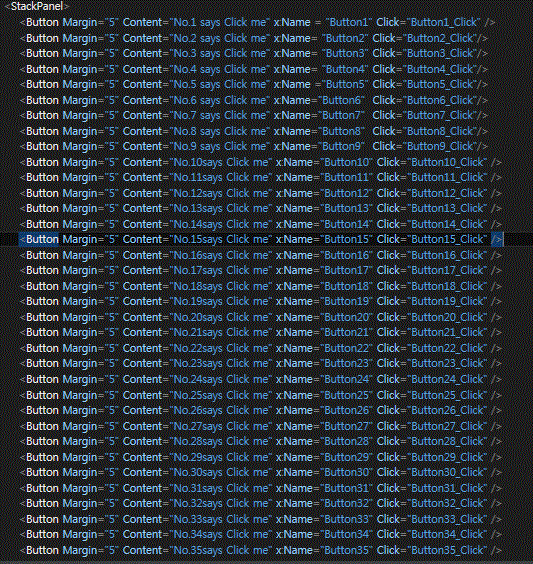
if (btn != null) //널을 지울경우

MessageBox.Show(btn.Name + " has been clicked",

"Button Click");

}

**\*바뀌기전 예제 – C# Code**

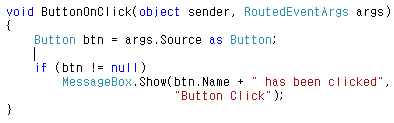
** **

**\*코드 작성 – Xaml \*Xaml코드 작성후 디자인뷰**

**코드 작성시 Xaml에서 버튼을 추가하고 스크롤바를 생성하는 코드를 써주게 되면**

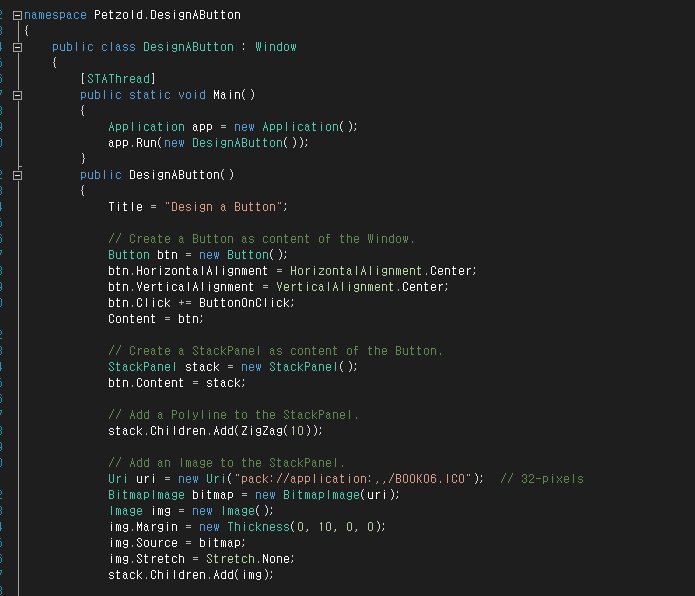
**스크롤 바가 필요하지 않을 경우에는 비활성화되어 있다가 한 화면에 모든 컨트롤들을**

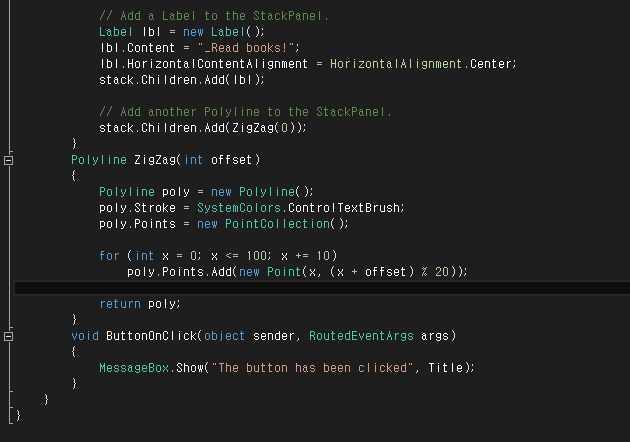
**표현할수 없을 경우 스크롤바를 통해 표현할수 있게 만들어주는 예제이다**

****

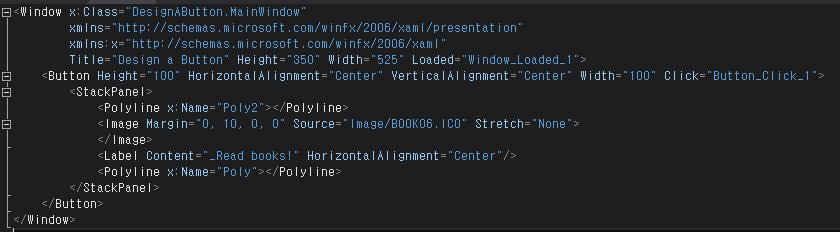
**이벤트 코드 작성하는 곳에 버튼이 != null일 경우 스크롤바를 생성하게끔 만들어준다**

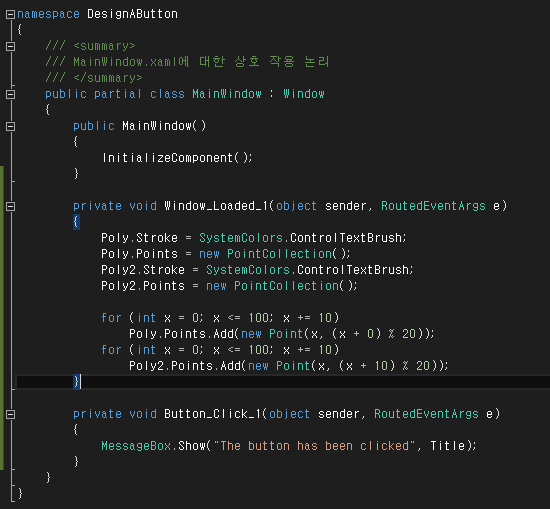
StackPanel을 사용한 예제 4



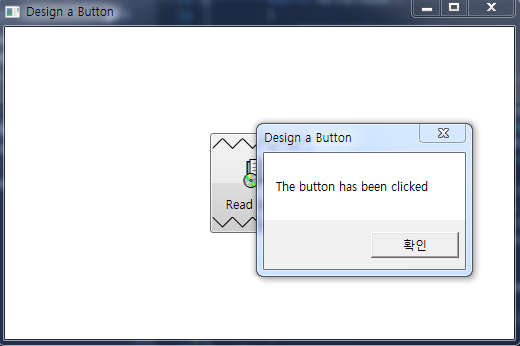
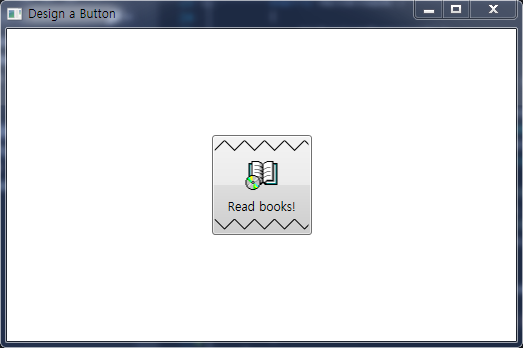


\*CS 코드



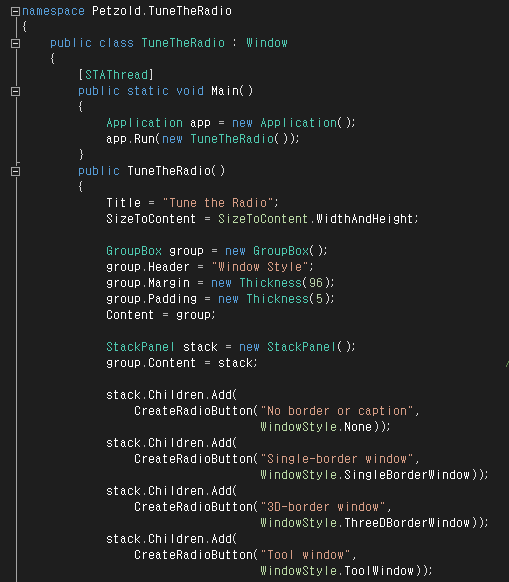


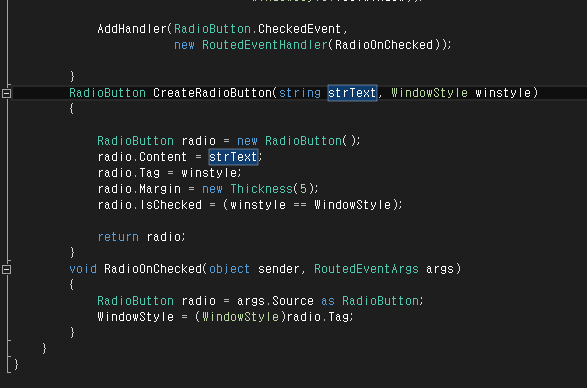
\*XAML 코드로 변경



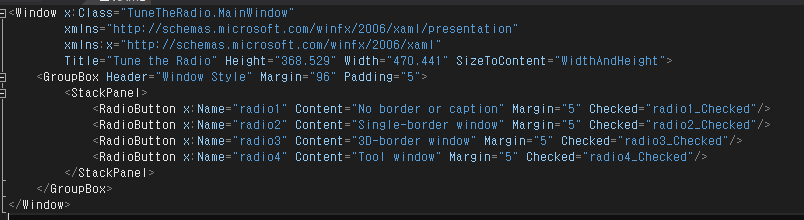
\*실행 화면

StackPanel을 사용한 예제 5



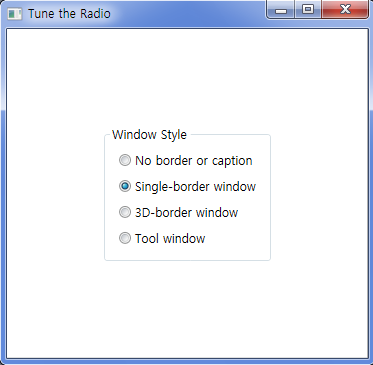
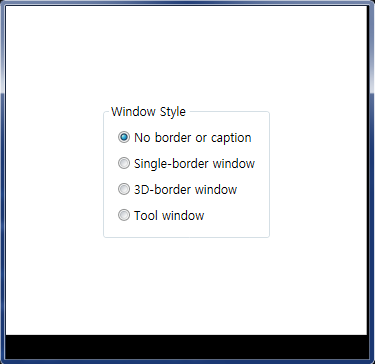


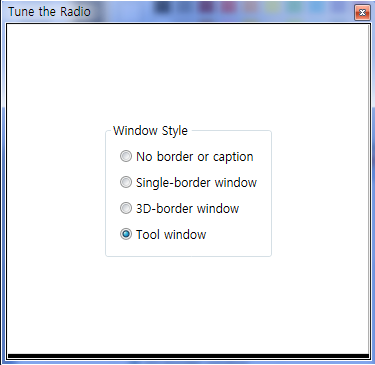
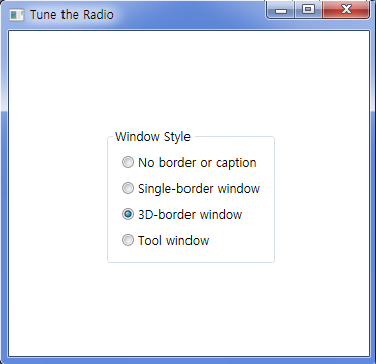
\*CS 코드





\*XAML코드로 변경





\*실행 화면

1.WrapPanel

* StackPanel과 비슷 (다른점:개행시 짤림의 여부)
* 자식 요소를 왼쪽에서 오른쪽으로 순차적으로 배치
* WrapPanel 안에 보이지 않는 행과 열에 일정하게 요소들을 배치시킴
* 해당 부모 컨테이너의 가장자리에 도달할 경우 내용을 개행하는 곳에 사용

2.WrapPanel 사용예제

|  |  |  |
| --- | --- | --- |
| **Wrappanel 사용하지 않는 경우▼** | **사용코드▼** | **Wrappanel 사용 ▼** |
|  |  |  |

2.ScrollViewer사용

자동 개행하여도 전체 창을 초과된 부분은 짤리는데 **ScrollViewer** 을 사용하면 해결 할 수 있음

|  |  |  |
| --- | --- | --- |
| **▼ ScrollViewer를 사용안함** | **▼ ScrollViewer사용 코드** | **▼ ScrollViewer를 사용** |
| http://postfiles10.naver.net/20130227_137/curlicu_1361891436878c93MI_PNG/022613_1510_WPF3.png?type=w2 |  |  |

수평으로 배열되어 있는 버튼을 수직으로 바꾸고 싶으면 WrapPanel Orientation의 속성은 vertical로 바꾸어 주면 버튼의 정렬이 바뀌게 된다.

|  |  |  |
| --- | --- | --- |
| **▼Orientation 사용 안함** | **▼ Orientation 코드** | **▼ Orientation=”vertical” 사용** |
|  |  |  |