

# Window Presentation Foundation

- Dipl.-Inf., Dipl.-Ing. (FH) Michael Wilhelm
- Hochschule Harz
- FB Automatisierung und Informatik
- [mwilhelm@hs-harz.de](mailto:mwilhelm@hs-harz.de)
- <http://www.miwilhelm.de>
- Raum 2.202
- Tel. 03943 / 659 338

# ·Inhalt

- Einführung WPF
- **Layouts**
- C# Sprache
- Dialog-Elemente, Menüs
- 2D- / 3D-Grafik, Audio, Video, Animation
- Routet Events, Dependency Properties, Command
- Textdarstellung (Flow-FixedDocuments)
- Datenbanken
- Navigation / Browser
- Eigene Komponenten

# Layout-Elemente (kein Anchor mehr vorhanden)

## ■ Elemente:

- Canvas Position mittels x/y, setLayout(null);
- DockPanel andocken an Elemente, letzte mit Fill
- Grid GridBagLayout in Java
- UniformGrid Grid in Java
- StackPanel Orientation="Horizontal"  
Orientation= "**Vertical**"
- VBox Ein Element, aber mit Zoom
- WrapPanel FlowLayout
- VirtualizingStackPanel Scrolling sichtbarer Elemente (ListBox)
- DataGridCellsPanel Layout für DataGrid-Element

# Layout-Elemente

## Hilfselemente:

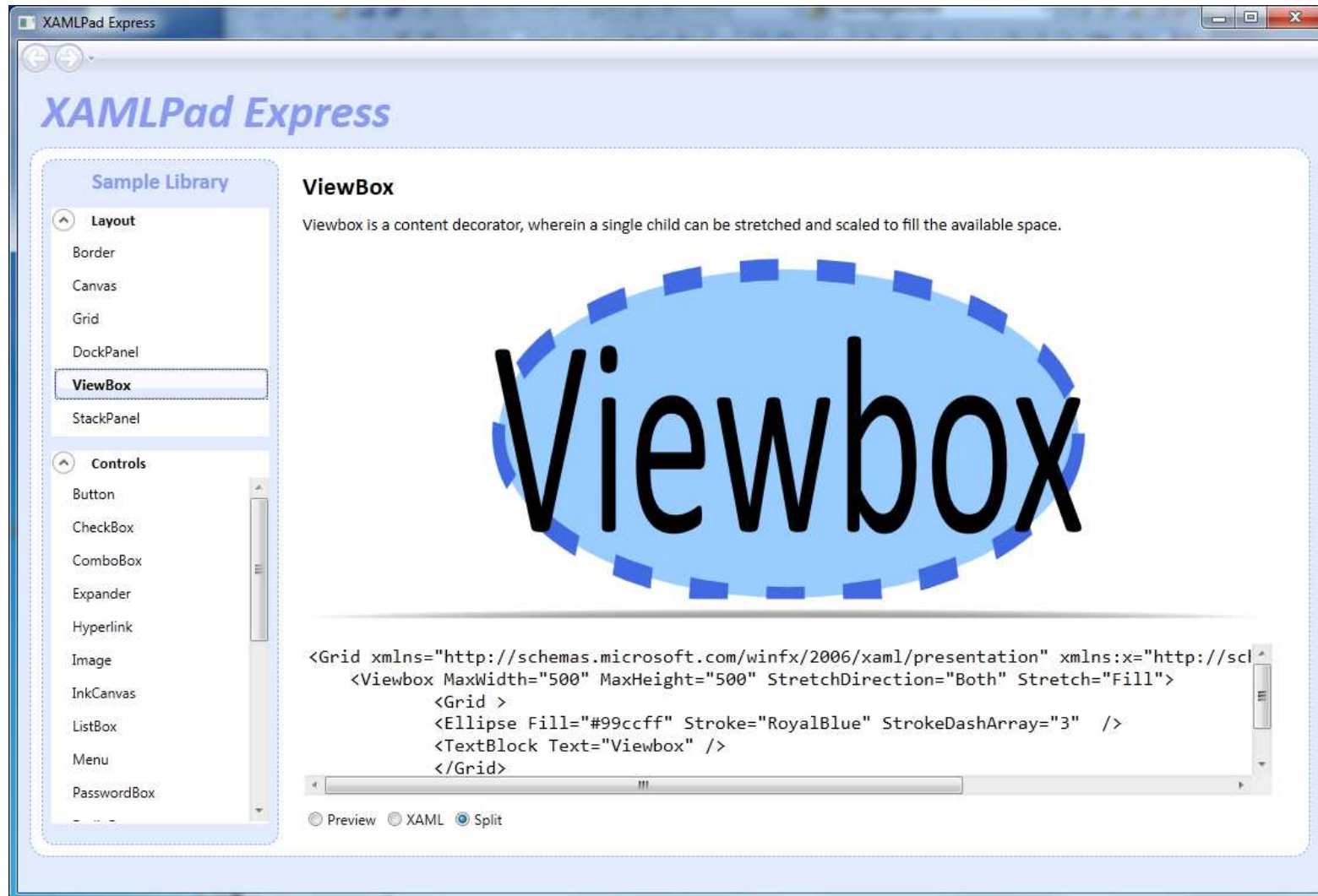
- Border

Rand

- GridSplitter

Ändern der Breite einer Zelle

# XAMLPad Express: Link auf meiner Homepage



Doppelklick: [SdkXamlBrowser.csproj](#)

# WPF und XAML

## ■ Winforms:

- Zwei partial Klassen: form1.cs, form.Designer.cs

## ■ WPF:

- Eine Klasse: form1.cs
- Eine XAML-Datei form.xaml (optional)
- XAML als deklarative Programmiersprache
- Definieren, *was* gemacht wird. Nicht *wie*

## ■ XAML

- Alle Elemente werden in einem Baum dargestellt (DOM)
- à la XHTML
- Mit Attributen

# XAML: 1. Beispiel, Visual Tree

```
<Window x:Class="bsp1.MainWindow"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  Title=",,1. Beispiel" Height="224" Width="525" Margin="1,25,1,1" >

  <StackPanel Height="175" HorizontalAlignment="Left" Margin="10,10,0,0"
    Name="stackPanel1" VerticalAlignment="Top" Width="465">

    <Label Content="Label" Height="31" Name="label1" Width="400" />
    <TextBox Height="24" Name="textBox1" Width="400" />
    <Button Content="Button" Height="37" Name="button1" Width="400"
      Margin="0,50,0,0" Click="button1_Click" />

  </StackPanel>

</Window>
```

Margin: Left, Top, Right, Bottom





# XAML: 1. Beispiel, Visual Tree

## Leerer Rumpf:

```
<Window x:Class="bsp1.MainWindow"
```

```
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"  
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"  
  Title="MainWindow" Height="350" Width="525">
```

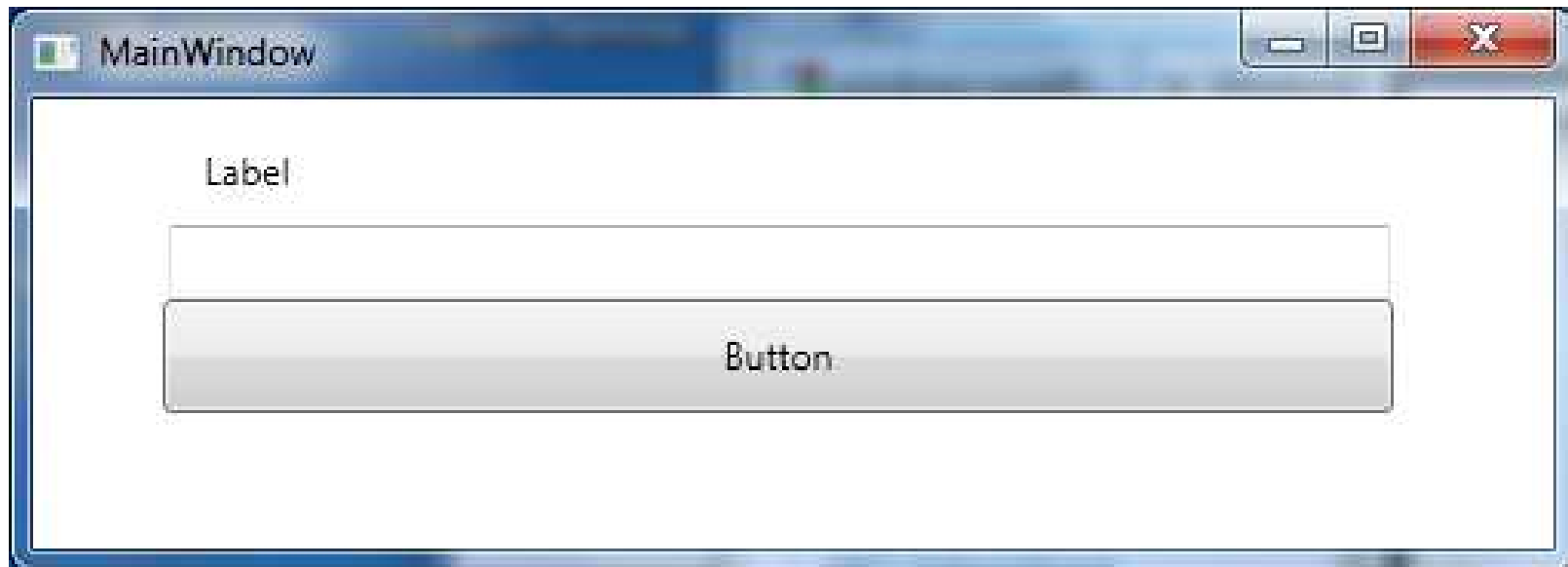
```
<Grid>
```

```
</Grid>
```

```
</Window>
```

## XAML:

- Beispiel Eingabe eines Textes
- Ausgabe des Textes



# XAML und Attribute

- **<Typname Attribut="Value" />:**
  - Ersetzt durch
    - <Typname>
      - <Typname.Attribut>
        - Value
      - </Typname.Attribut>
    - </Typname>

# XAML: Attribute mit Typkonvertierung

- `<Grid Margin="10,20,30,40" />`

- Ersetzt durch

- `<Grid>`

  - `<Grid.Margin>`

  - `<Thickness Left="10" Top="20" Right="30" Bottom="40"/>`

  - `</Grid.Margin>`

- `</Grid>`

  - Vereinfachung muss man in eigene Klassen einbauen

  - Get und set-Methoden werden weitergeleitet

# XAML: Markup Extension

## ■ Anbindung an eine Datenbank

- Die Attribute erhalten eine Referenz auf ein beliebiges Objekt
- `<TextBlock Text="{ Binding Path=Name}" ... />`

## ■ Beispiel

`<DataGridTextColumn`

`x:Name="iDColumn" Binding="{ Binding Path=ID}"`

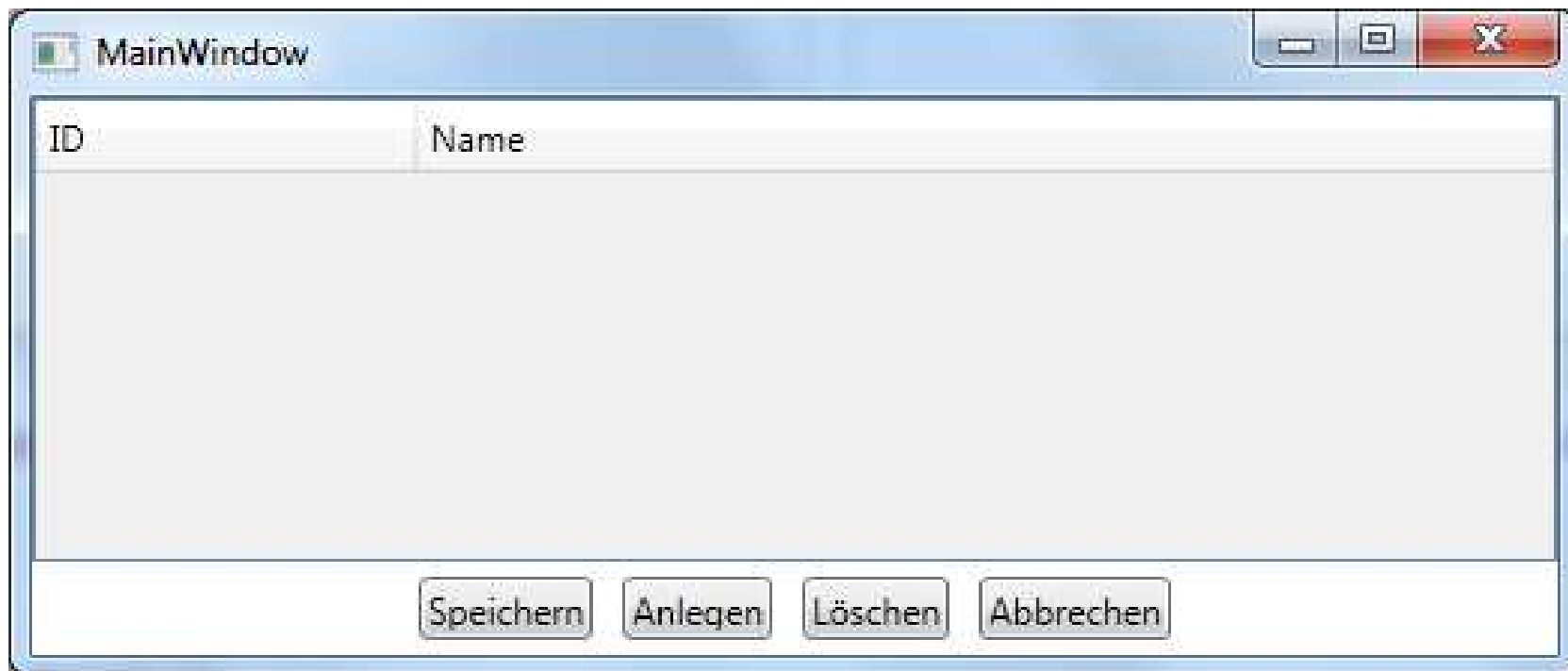
`Header="ID" Width="*" IsReadOnly="True" />`

`<DataGridTextColumn`

`x:Name="nameColumn" Binding="{ Binding Path=Lastname}"`

`Header="Lastname" Width="3*" />`

# XAML: Markup Extension



# Layout-Elemente, abgeleitet von Panel

## ■ Elemente:

- Canvas Position mittels x/y, setLayout(null);
- DockPanel andocken an Elemente, letzte Fill
- Grid GridBagLayout in Java
- UniformGrid Grid in Java
- **StackPanel** **Orientation="Horizontal"**  
**Orientation= "Vertical"**
- ViewBox Ein Element, aber mit Zoom
- WrapPanel FlowLayout
- VirtualizingStackPanel Scrolling sichtbarer Elemente (ListBox)
- DataGridCellsPanel Layout für DataGrid-Element

# StackPanel

- **Wichtiges Layout-Element:**
- Orientation="Horizontal"
- Orientation= "**Vertical**" , Default-Wert
- Kann verschachtelt werden
- Stapelt beliebige Elemente
  - Vertikal von oben
  - Horizontal von links
- Die Elemente in einem vertikalen StackPanel werden horizontal gestreckt
- Die Elemente in einem horizontalen StackPanel werden vertikal gestreckt



# StackPanel: Attribute

- Orientation: Horizontal / Vertical
- HorizontalAlignment: Left / Right / Center / Stretch (Fill)
- VerticalAlignment: Left / Right / Center / Stretch (Fill)
- Cursor
- FlowDirection LeftToRight / RightToLeft
- Höhe: Height / MinHeight / MaxHeight
- Breite: Width / MinWidth / MaxWidth
- Transparenz: Opacity / OpacityMask
- Margin: Left / Top / Right / Bottom
- Hintergrund: Background
- ContextMenu
- Visibility: Visible / Hidden / Collapsed
- Scrollen: CanHorizontallyScroll / CanVerticallyScroll
- Style:
- Hinweis: ToolTip

# StackPanel: Vorl01

- Neues WPF-Projekt anlegen
- ToolBox, Strg+Alt+X
- Anklicken des Symbols „StackPanel“
- Aufziehen in der Form
- Wird von einem Grid-Layout gekapselt (kann man löschen)

<Grid>

```
<StackPanel Height="262"
  HorizontalAlignment="Left"
  Margin="20,26,0,0" Name="stackPanel1"
  VerticalAlignment="Top"
  Width="462" />
```

</Grid>

## StackPanel: Vorl01

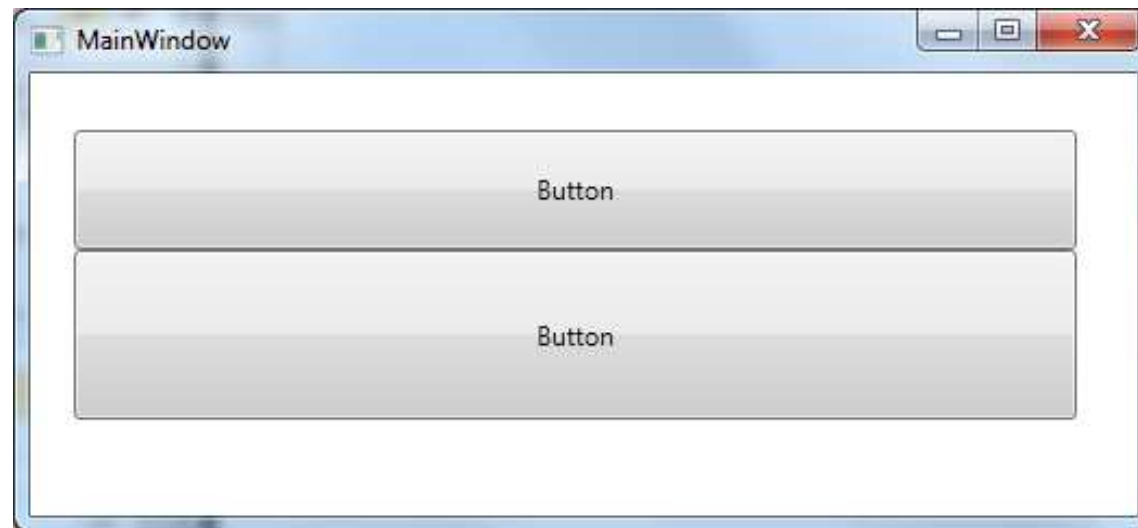
- Aus der ToolBox zwei Button einfügen
- Unterschiedliche Länge
- Schalter werden zentriert, da Width einen Wert hat

```
<StackPanel Height="262" HorizontalAlignment="Left"  
    Margin="20,26,0,0" Name="stackPanel1"  
    VerticalAlignment="Top" Width="462"  
    Orientation="Vertical">
```

```
<Button Content="Button" Height="55" Name="button1"  
    Width="359" />
```

```
<Button Content="Button" Height="78" Name="button2"  
    Width="292" />
```

# StackPanel: Vorl01



# StackPanel: Vorl01

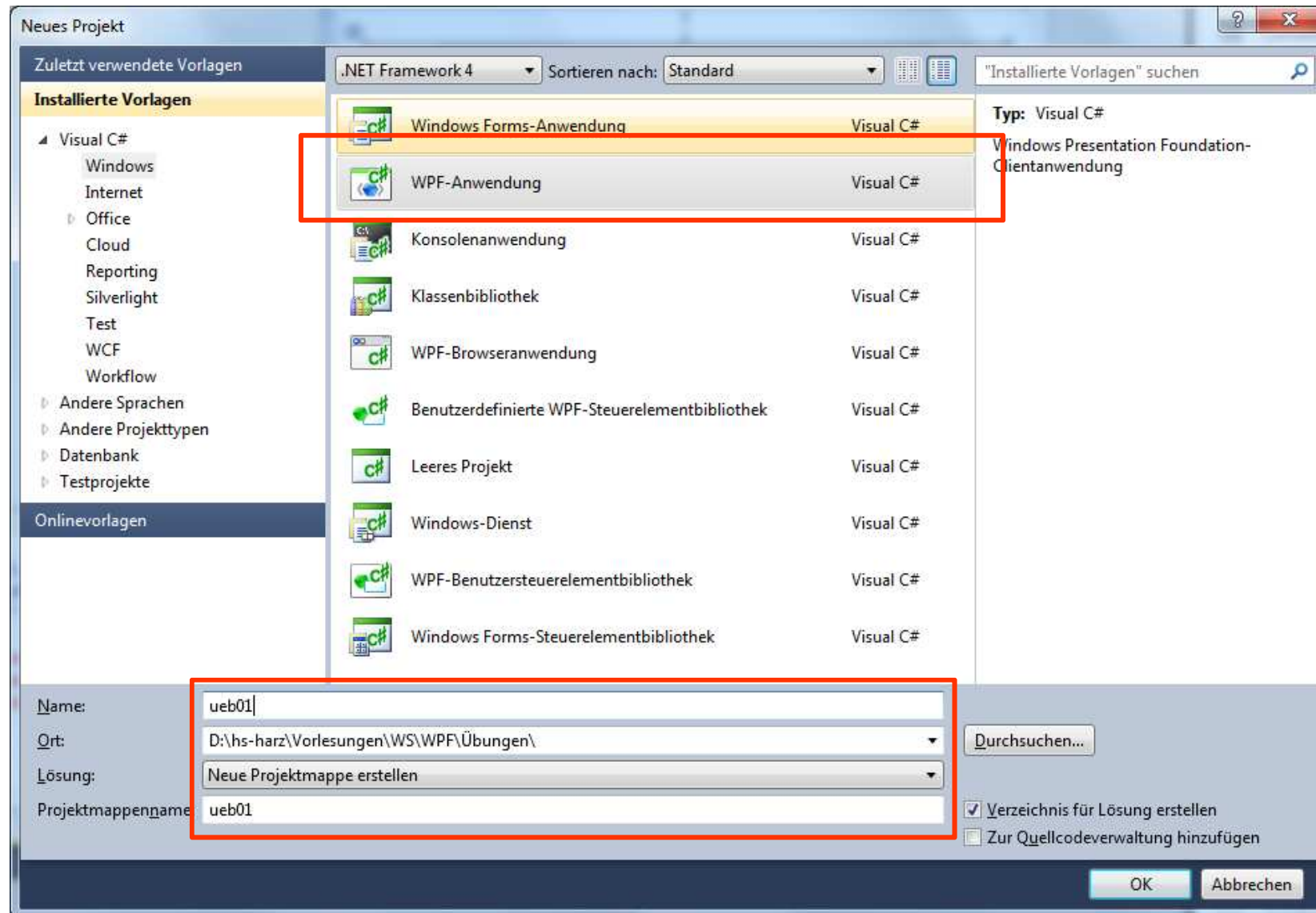


```
<StackPanel Height="262" HorizontalAlignment="Left"  
Margin="20,26,0,0" Name="stackPanel1"  
VerticalAlignment="Top" Width="800"  
Orientation="Horizontal"
```

# StackPanel: 1. Aufgabe: ueb01

- **Erstellen eines WPF-Programms**
- StackPanel, Vertikal
- Ein Label
- Eine TextBox
- Ein Schalter
- Beim Anklicken soll der Text der Eingabe ausgegeben werden
- Falls leer, .Equals, soll eine Meldung erscheinen  
    MessageBox.Show(...)
- **Variablennamen:**
  - tInput
  - bnOk

# StackPanel: 1. Aufgabe: ueb01



# StackPanel: 1. Aufgabe: ueb01

```
<Window x:Class="ueb01.MainWindow"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    Title="MainWindow" Height="200" Width="525" FontSize="24">

    <StackPanel Height="313" HorizontalAlignment="Left" Margin="0"
        Name="stackPanel1" VerticalAlignment="Top" Width="503">

        <Label Content="Eingabe" Height="38" Name="label1" Width="344" />
        <TextBox Height="45" Name="tInput" Width="349" />
        <Button Content="Klick mich an" Height="57" Name="bnOk"
            Width="175" Click="bnOk_Click" />

    </StackPanel>

</Window>
```



# StackPanel: 1. Aufgabe: ueb01

## Quellcode in MainWindow.xaml.cs:

```
private void bnOk_Click(object sender, RoutedEventArgs e)
{
    if ( tInput.Text.Equals("") ) {
        MessageBox.Show("Bitte geben einen Text in die Editorzeile ein",
            "Hinweis",  MessageBoxButton.OKCancel,
            MessageBoxImage.Information);
        tInput.Focus();
    }
    else
        MessageBox.Show(tInput.Text);
}
```

## StackPanel: 2. Aufgabe: ueb02

- **Erstellen eines WPF-Programms**
- StackPanel, 1xVertikal und 1xHorizontal
- Ein Label mit einer TextBox in einer Zeile
- Ein Schalter
- Beim Anklicken soll der Text der Eingabe ausgegeben werden
- Falls leer, .Equals, soll eine Meldung erscheinen
  
- **Variablennamen:**
  - tInput
  - bnOk

## StackPanel: 2. Aufgabe: ueb02



- Die Höhe des inneren StackPanels muss verringert werden
- Kein Zentrieren des Schalters
- Der Editor „zoomt“ nicht mit der Größe

```
<Border CornerRadius="10" Margin="5" BorderBrush="Red" BorderThickness="2">  
  <Button Content="Klick mich an" Height="57" Name="bnOk" Width="175"  
    Click="bnOk_Click" />  
</Border>
```

## StackPanel: 3. Aufgabe: ueb03



- Die Schalter arbeiten nun mit einem WrapPanel
- Alignment leider nur left oder right
- Bei right muss man die Reihenfolge des Einfügens ändern

```
<WrapPanel Name="wrapPanel1" Width="433" FlowDirection="RightToLeft">  
    <elemente />  
</WrapPanel>
```

## WrapPanel: Attribute

- HorizontalAlignment: Left / Right / Center / Stretch (Fill)
- VerticalAlignment: Left / Right / Center / Stretch (Fill)
- Cursor
- Höhe: Height / MinHeight / MaxHeight
- Breite: Width / MinWidth / MaxWidth
- Transparenz: Opacity / OpacityMask
- Margin: Left / Top / Right / Bottom
- Hintergrund: Background
- ContextMenu
- Visibility: Visible / Hidden / Collapsed
- Style:
- Hinweis: ToolTip

## StackPanel: 3. Aufgabe: ueb03



- Zentrierte Schalter mit
  - WrapPanel ohne Width
  - HorizontalAligment="Center"
  - Reihenfolge umdrehen

```
<WrapPanel Name="wrapPanel1"  
    HorizontalAligment="Center">  
    <elemente />  
</WrapPanel>
```

# UniformGrid: Entspricht dem Grid à la Java

## ■ Einfaches Layout-Element:

- Tabellenstruktur
- Jede Zelle ist gleich groß, **wenn** Width nicht definiert wurde
- Jede Zelle ist gleich groß, **wenn** Height nicht definiert wurde
- FirstColumn="0": Anzahl der leeren Zellen
- Automatisches Gitter-Anordnung, wenn keine Größe angegeben wurde
- oder

- Columns="2"

- Rows="5"

```
<UniformGrid FirstColumn="0" Columns="2" Rows="5" Cursor="Hand" >
```

```
  <Button Content="Button1" Name="button1" />
```

```
  <Button Content="Button2" Name="button2" />
```

```
  <Button Content="Button3" Name="button3" />
```

```
  ...
```

```
  <Button Content="Button7" Name="button4" />
```

```
</UniformGrid>
```

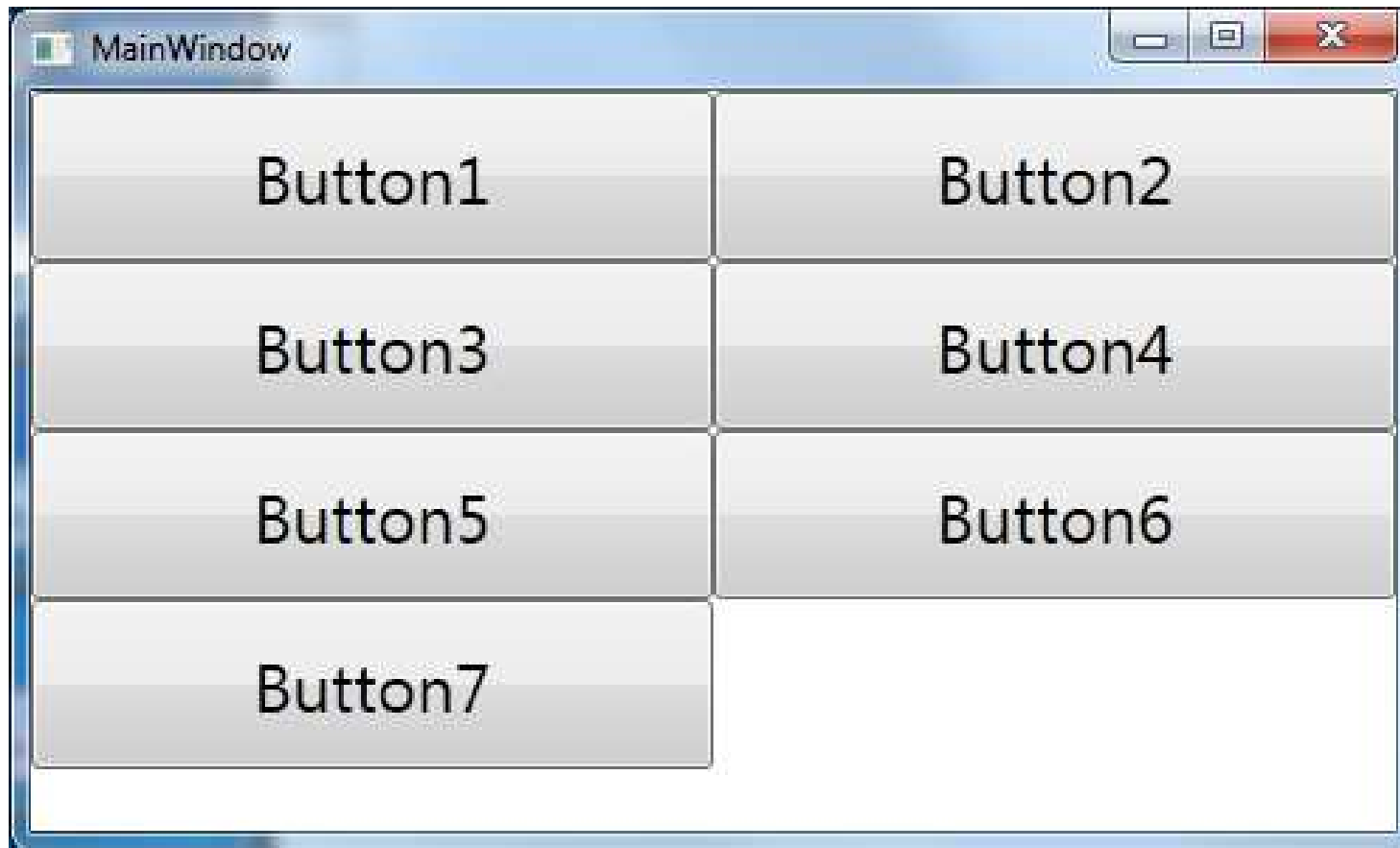
# UniformGrid: Attribute

- HorizontalAlignment: Left / Right / Center / Stretch (Fill)
- VerticalAlignment: Left / Right / Center / Stretch (Fill)
- Cursor
- Höhe: Height / MinHeight / MaxHeight
- Breite: Width / MinWidth / MaxWidth
- Transparenz: Opacity / OpacityMask
- Anzahl der Zellen: Rows / Columns
- Margin: Left / Top / Right / Bottom
- Hintergrund: Background
- ContextMenu
- Visibility: Visible / Hidden / Collapsed
- Style:
- Hinweis: ToolTip



## UniformGrid:

- zwei Spalten, vier Zeilen
- Eingefügt von links nach rechts



# UniformGrid: 5. Aufgabe: ueb05

- Entwicklung einer Taschenrechner
- Stackpanel: Insgesamt **ZWEI** Zeilen

- 1. Zeile: Editor
- 2. Zeile: UniformGrid (5x3)



&#x221a;

&#8730;

Alt 251

"\u221A"

<RepeatButton Content="1" Name="bn1" Delay="100" Click="bn1\_Click" />

# Grid: Entspricht dem GridBagLayout in Java

- **Komplexes Layout-Element:**
- Tabellenstruktur
- Anzahl der Zeilen können am Anfang definiert werden
- Jede Zelle ist gleich groß, wenn Height nicht definiert wurde
- Jedes Element hat eine Positionierung mittels Margin(Top/Left)
- Können auch übereinander liegen
  - `</Grid.RowDefinitions>`
  - `<Grid.ColumnDefinitions>`
- Elemente werden dann direkt positioniert
  - `Grid.Row="1"`                      `Grid.Column="2"`
- RowSpan und ColSpan verbinden Zellen, wie in HTML

# Grid: Entspricht dem GridBagLayout in Java

<Grid>

<Button Content="Button" Height="23" HorizontalAlignment="Left"  
Margin="52,36,0,0" Name="button1" VerticalAlignment="Top" Width="75" />

<Button Content="Button" Height="23" HorizontalAlignment="Left"  
Margin="268,98,0,0" Name="button2" VerticalAlignment="Top" Width="75" />

<Button Content="Button" Name="button3" Height="23" HorizontalAlignment="Left"  
Margin="108,96,0,0" VerticalAlignment="Top" Width="75" />

</Grid>

<Grid ShowGridLines="True">

<Grid.RowDefinitions>

<RowDefinition/>

<RowDefinition/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition/>

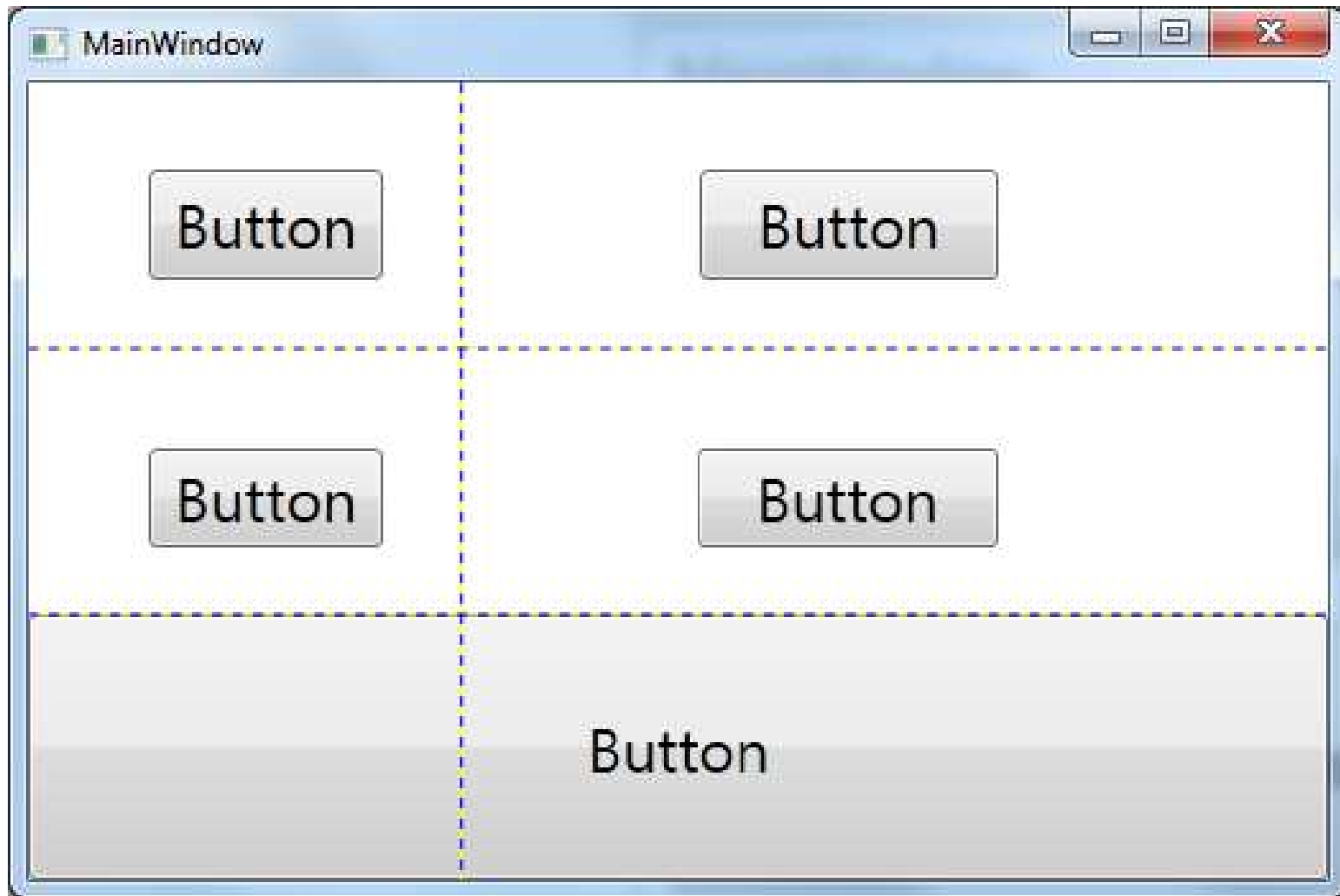
<ColumnDefinition/>

</Grid.ColumnDefinitions>

</Grid>

**Fill:** <Button Content="Button" Grid.Column="1" Grid.Row="2" Name="button6" />

## Gridbeispiel:ueb06



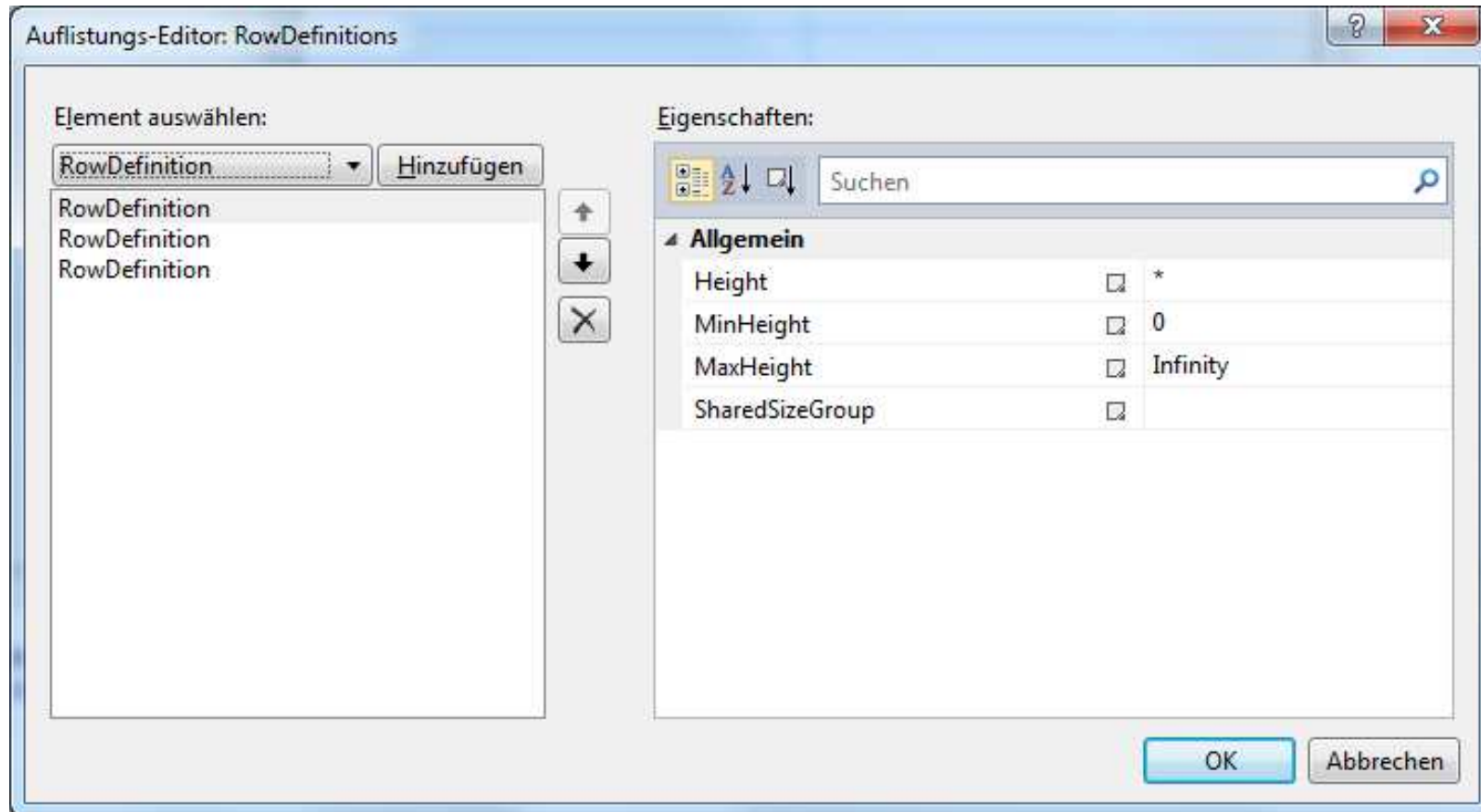
## Gridbeispiel:ueb06

```
<Grid ShowGridLines="True">
  <Grid.RowDefinitions>
    <RowDefinition/>          <RowDefinition/>          <RowDefinition/>
  </Grid.RowDefinitions>
  <Grid.ColumnDefinitions>
    <ColumnDefinition Width="1*"/>
    <ColumnDefinition Width="2*"/>
  </Grid.ColumnDefinitions>
  <Button Content="Button" Height="43" HorizontalAlignment="Left"
    Margin="47,34,0,0" Name="button1" VerticalAlignment="Top" Width="92" />
  <Button Content="Button" Grid.Column="1" HorizontalAlignment="Left"
    Height="43" Margin="93,34,0,0" Name="button2" VerticalAlignment="Top"
    Width="117" />
  <Button Content="Button" Grid.Row="1" Height="39" HorizontalAlignment="Left"
    Margin="47,39,0,0" Name="button3" VerticalAlignment="Top" Width="92" />
  <Button Content="Button" Grid.Column="1" Grid.Row="1" Height="39"
    HorizontalAlignment="Left" Margin="92,39,0,0" Name="button4"
    VerticalAlignment="Top" Width="118" />
  <Button Content="Button" Grid.Column="0" Grid.Row="2" Grid.ColumnSpan="2"
    Name="button6" />
</Grid>
```

## Grid: Attribute

■ Spalten, Zeilen	ColumnDefinition / RowDefinition ColSpan / RowSpan
■ Linien	ShowGridLines
■ HorizontalAlignment:	Left / Right / Center / Stretch (Fill)
■ VerticalAlignment:	Left / Right / Center / Stretch (Fill)
■ Cursor	
■ Höhe:	Height / MinHeight / MaxHeight
■ Breite:	Width / MinWidth / MaxWidth
■ Transparenz:	Opacity / OpacityMask
■ Anzahl der Zellen:	Rows / Columns
■ Margin:	Left / Top / Right / Bottom
■ Hintergrund:	Background
■ ContextMenu	
■ Visibility:	Visible / Hidden / Collapsed
■ Style:	
■ Hinweis:	ToolTip

# Grid: Column- und RowDefinitionen

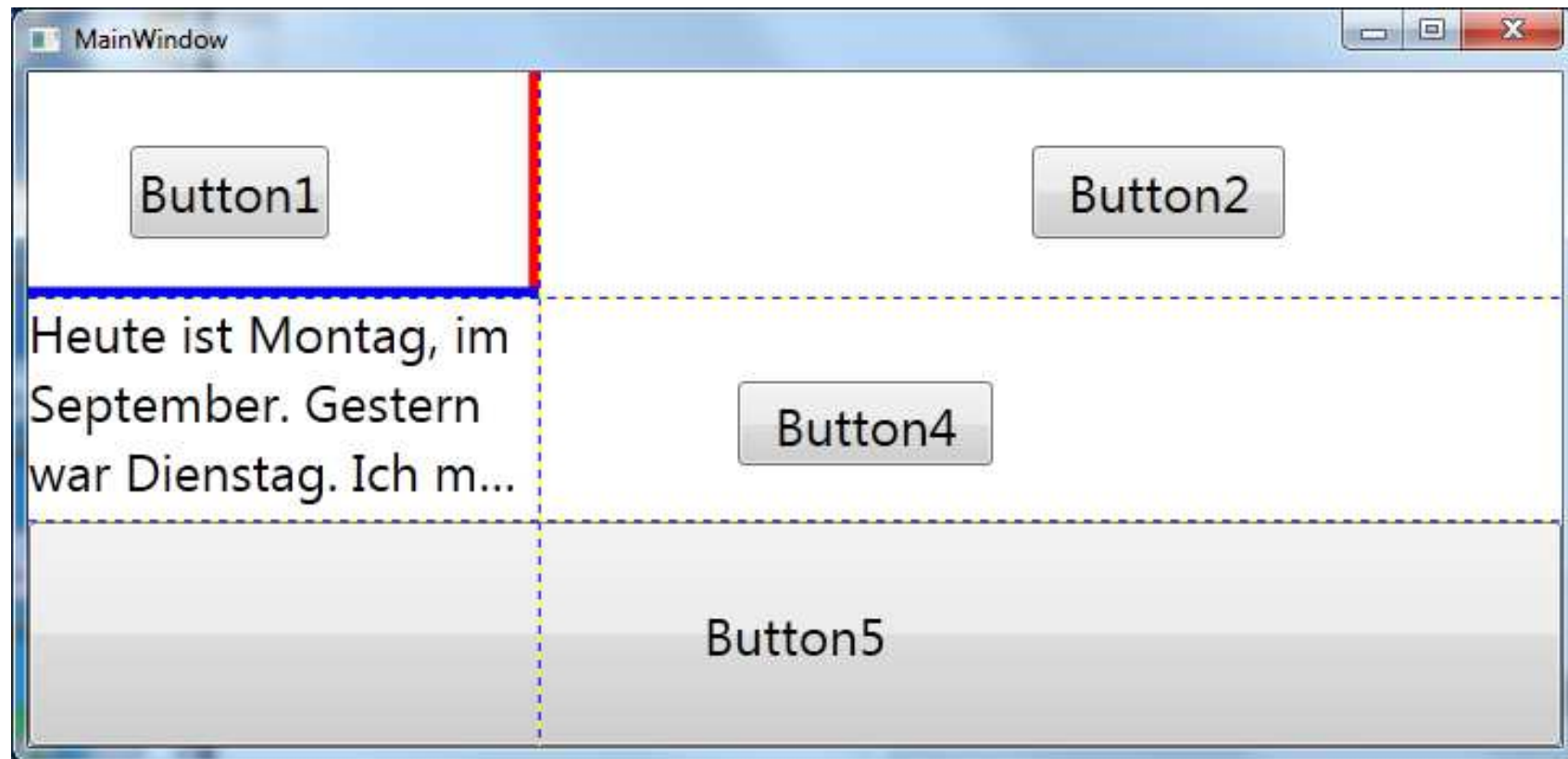


## Splitter:

<GridSplitter Background="Red" Width="5" Grid.Column="0,, HorizontalAlignment="Right" />



## Grid: Beispiel ueb06



Mit den beiden Splitter kann man die Spaltenbreite und Zeilenhöhe dynamisch

# Grid: Beispiel ueb06

```
<Window x:Class="ueb06.MainWindow"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  Title="MainWindow" Height="350" Width="725" FontSize="24">

  <Grid ShowGridLines="True">
    <Grid.RowDefinitions>
      <RowDefinition/>
      <RowDefinition/>
      <RowDefinition/>
    </Grid.RowDefinitions>
    <Grid.ColumnDefinitions>
      <ColumnDefinition Width="1*"/>
      <ColumnDefinition Width="2*"/>
    </Grid.ColumnDefinitions>

    ...
  </Grid>
</Window>
```

# Grid: Beispiel ueb06

```
<Grid ShowGridLines="True">
```

```
...
```

```
<Button Content="Button1" Grid.Column="0" Grid.Row="0" Height="43"  
  HorizontalAlignment="Left" Margin="47,34,0,0" Name="button1"  
  VerticalAlignment="Top" Width="92" />
```

```
<Button Content="Button2" Grid.Column="1" Grid.Row="0" Height="43"  
  HorizontalAlignment="Left" Margin="228,34,0,0" Name="button2"  
  VerticalAlignment="Top" Width="117" />
```

```
<TextBlock Grid.Column="0" Grid.Row="1" Name="tBlock" Text="Heute ist ::"  
  TextWrapping="Wrap" TextTrimming="CharacterEllipsis" />
```

```
<Button Content="Button4" Grid.Column="1" Grid.Row="1" Height="39"  
  HorizontalAlignment="Left" Margin="92,39,0,0" Name="button4"  
  VerticalAlignment="Top" Width="118" />
```

```
<Button Content="Button5" Grid.Column="0" Grid.Row="2" Grid.ColumnSpan="2"  
  Name="button6" />
```

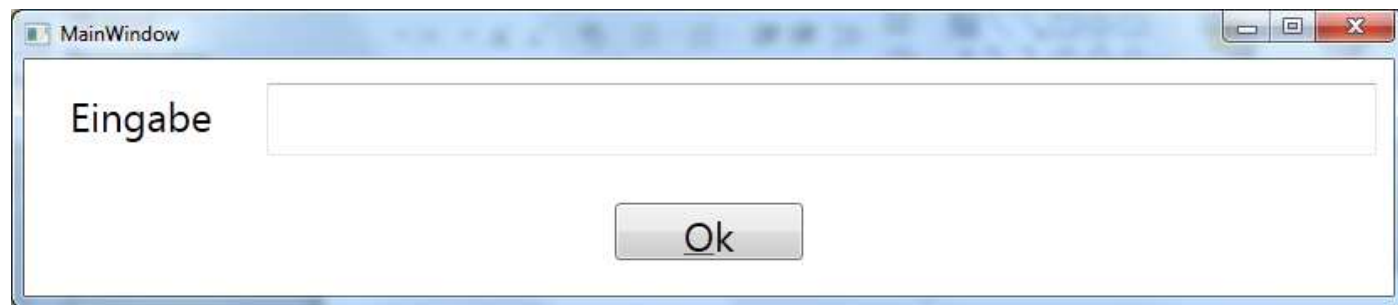
```
<GridSplitter Background="Red" Width="5" Grid.Column="0"  
  HorizontalAlignment="Right"/>
```

```
<GridSplitter Background="Blue" Height="5" Grid.Column="0"  
  VerticalAlignment="Bottom" HorizontalAlignment="Stretch" />
```

```
</Grid>
```

## Grid: 7. Aufgabe: ueb07

- Entwicklung eines Eingabedialogs
- Panel mit einem Grid: Insgesamt **ZWEI** Zeilen und **zwei** Spalten
- Der Editor „**zoomt**“ mit der Größe des Fensters
- Der Schalter ist **immer** in der Mitte, ColSpan
- Verwenden Sie für alle Elemente:
  - Grid.Row="?"
  - Grid.Column="?"



## Grid: 8. Aufgabe: ueb08

- Entwicklung eines Eingabedialogs
- Panel mit einem Grid: Insgesamt **DREI** Zeilen und **drei** Spalten
- Die Editoren „**zoomen**“ mit der Größe des Fensters
- Die Schalter sind **immer** in der Mitte, ColSpan
- Verwenden Sie für alle Elemente:
  - Grid.Row="?"
  - Grid.Column="?"

The screenshot shows a Windows dialog box titled "Übung 08". It contains a grid of input fields. The first row has one input field labeled "1. Eingabe". The second row has three input fields labeled "2. Eingabe" containing the values 7, 4, and 2. The third row has two input fields labeled "3. Eingabe" containing the values 70 and 30. At the bottom are "Ok" and "Abbrechen" buttons.

# DockPanel:

- **Einfaches Layout-Element:**
- Ähnlich wie ein Stackpanel
- Aber das letzte Element kann den restlichen Platz ausfüllen (Fill)
- Interessant dabei:
  - Andocken an Left, Top, Right oder Bottom
- Man damit auch Elemente am unteren Rand stapeln
- Das StackPanel erlaubt das nur von oben
- **Die Reihenfolge des Einfügen ist wichtig!!**

- |                |   |
|----------------|---|
| ■ Letztes Fill | <code>LastChildFill="False" / "True"</code> |
| ■ Andocken     | <code>DockPanel.Dock="Left"</code>          |

## DockPanel mit C#:

```
Label label1 = new Label();  
Label label2 = new Label();  
TextBox editor = new TextBox();
```

```
WrapPanel wrapPanel = new WrapPanel();  
wrapPanel.HorizontalAlignment = HorizontalAlignment.Center;  
wrapPanel.Children.Add( label1 );  
wrapPanel.Children.Add( label2 );
```

```
DockPanel dockpanel = new DockPanel();  
dockpanel.LastChildFill = true;  
DockPanel.SetDock(wrapPanel, Dock.Bottom);  
DockPanel.SetDock(editor, Dock.Top);  
dockpanel.Children.Add(flowLayoutPanel1);  
dockpanel.Children.Add(editor);
```

# DockPanel: Attribute

- Letztes Fill                      LastChildFill="False" / "True"
- Andocken                        DockPanel.Dock="Left" / Top / Right / Bottom
  
- HorizontalAlignment:    Left / Right / Center / Stretch (Fill)
- VerticalAlignment:        Left / Right / Center / Stretch (Fill)
- Cursor
- Höhe:                        Height / MinHeight / MaxHeight
- Breite:                      Width / MinWidth / MaxWidth
- Transparenz:                Opacity / OpacityMask
- Anzahl der Zellen:        Rows / Columns
- Margin:                     Left / Top / Right / Bottom
- Hintergrund:                Background
- ContextMenu
- Visibility:                  Visible / Hidden / Collapsed
- Style:
- Hinweis:                     ToolTip

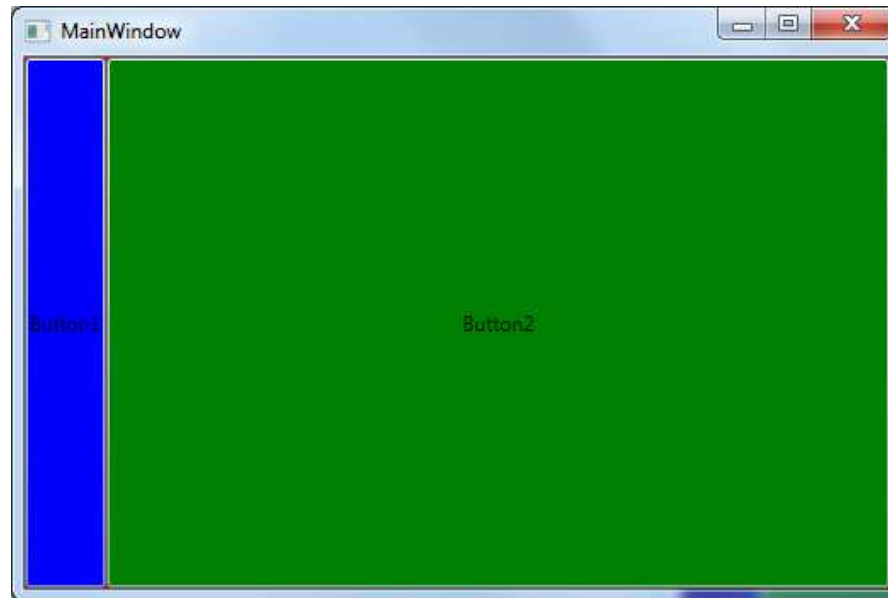


## DockPanel: Beispiel ueb09



```
<DockPanel Background="Red" LastChildFill="False" >  
  <Button DockPanel.Dock="Left" Background="Blue"  
    Content="Button1" Name="button1" />  
  <Button DockPanel.Dock="Left" Background="Green"  
    Content="Button2" Name="button2" />  
  
</DockPanel>
```

# DockPanel: Beispiel ueb09



```
<DockPanel Background="Red" LastChildFill="True" >  
    <Button DockPanel.Dock="Left" Background="Blue"  
        Content="Button1" Name="button1" />  
    <Button DockPanel.Dock="Left" Background="Green"  
        Content="Button2" Name="button2" />  
  
</DockPanel>
```

## DockPanel: Beispiel ueb09



```
<Button DockPanel.Dock="Left" Background="Blue" Content="Bn1" Name="bn1" />
```

```
<Button DockPanel.Dock="Left" Background="Green" Content="Bn2" Name="bn2" />
```

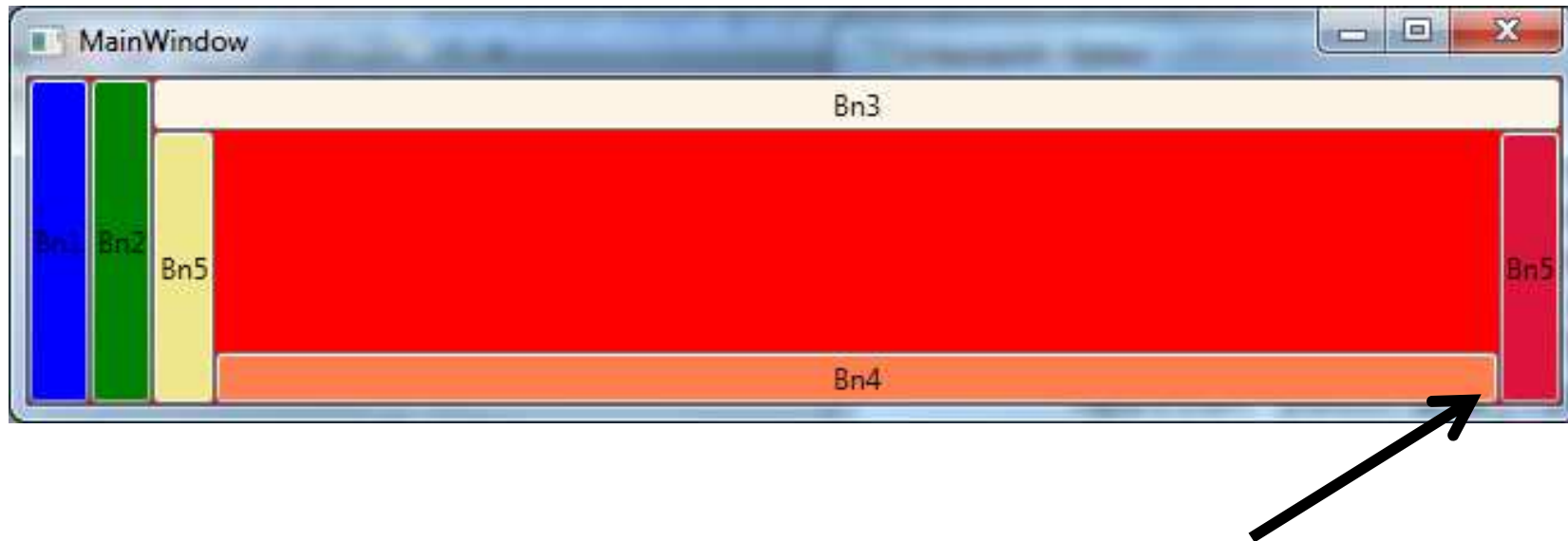
```
<Button DockPanel.Dock="Top" Background="OldLace" Content="Bn3" Name="bn3" />
```

```
<Button DockPanel.Dock="Bottom" Background="Coral" Content="Bn4" Name="bn4" />
```

```
<Button DockPanel.Dock="Right" Background="Crimson" Content="Bn5" Name="bn5" />
```

```
<Button DockPanel.Dock="Left" Background="Khaki" Content="Bn5" Name="bn6" />
```

## DockPanel: Beispiel ueb09



```
<Button DockPanel.Dock="Left" Background="Blue" Content="Bn1" Name="bn1" />
```

```
<Button DockPanel.Dock="Left" Background="Green" Content="Bn2" Name="bn2" />
```

```
<Button DockPanel.Dock="Top" Background="OldLace" Content="Bn3" Name="bn3" />
```

```
<Button DockPanel.Dock="Right" Background="Crimson" Content="Bn5" Name="bn5" />
```

```
<Button DockPanel.Dock="Left" Background="Khaki" Content="Bn5" Name="bn6" />
```

```
<Button DockPanel.Dock="Bottom" Background="Coral" Content="Bn4" Name="bn4" />
```

# Viewbox:

- **Einfaches Layout-Element:**

- Eigentlich kein Layout-Element
- Wird aber im Zusammenhang mit den Layouts verwendet
- Man kann ein oder mehrere Elemente eintragen
- Zoom-Eigenschaft: Stretch

- |                 |  |
|-----------------|--|
| – None          | Originalgröße  |
| – Fill          | Komplett ausgefüllt, Verzerrt                            |
| – Uniform       | Komplett ausgefüllt, Nicht verzerrt                      |
| – UniformToFill | Komplett für Höhe oder Breite ausgefüllt, Nicht verzerrt |

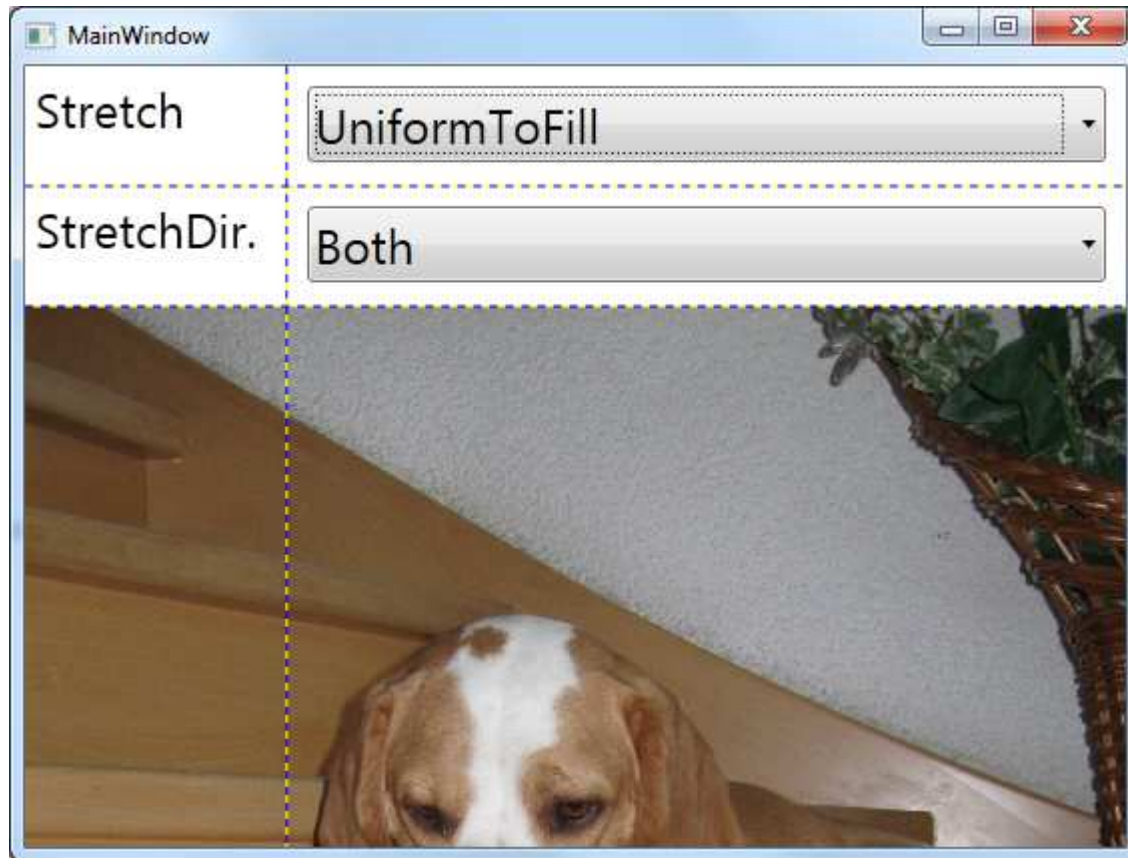
- StretchDirection-Eigenschaft:

- |            |                            |
|------------|----------------------------|
| – UpOnly   | Zoom nur beim Vergrößern   |
| – DownOnly | Zoom nur beim Verkleinern  |
| – Both     | Zoom wird immer ausgeführt |

# Viewbox: Attribute

- Stretch                      None, Fill, Uniform, UniformToFill
- StretchDirection           UpOnly, DownOnly, Both
  
- HorizontalAlignment:      Left / Right / Center / Stretch (Fill)
- VerticalAlignment:        Left / Right / Center / Stretch (Fill)
- Cursor
- Höhe:                      Height / MinHeight / MaxHeight
- Breite:                     Width / MinWidth / MaxWidth
- Transparenz:               Opacity / OpacityMask
- Anzahl der Zellen:        Rows / Columns
- Margin:                    Left / Top / Right / Bottom
- Hintergrund:               Background
- ContextMenu
- Visibility:                Visible / Hidden / Collapsed
- Style:
- Hinweis:                    ToolTip

## Viewbox: Beispiel ueb10



# ScrollView:

- **Hilfselement à la JScrollPane**
- Wenn der Platz nicht ausreicht

<ScrollView> Beispiel [ueb12](#)

<StackPanel>

<Button Background="Yellow" Content="Button1" Name="bn1"/>

<Button Background="Orange" Content="Button2" Name="bn2"/>

<Button Background="Red" Content="Button3" Name="bn3"/>

<Button Background="Lime" Content="Button4" Name="bn4"/>

<Button Background="DarkGray" Content="Button4" Name="bn5"/>

<Button Background="DarkTurquoise" Content="Button5" Name="bn62"/>

<Button Background="IndianRed" Content="Button5" Name="bn7"/>

<Button Background="MediumAquamarine" Content="Button6" Name="bn8"/>

</StackPanel>

</ScrollView>



# ScrollView: Beispiel ueb12

