

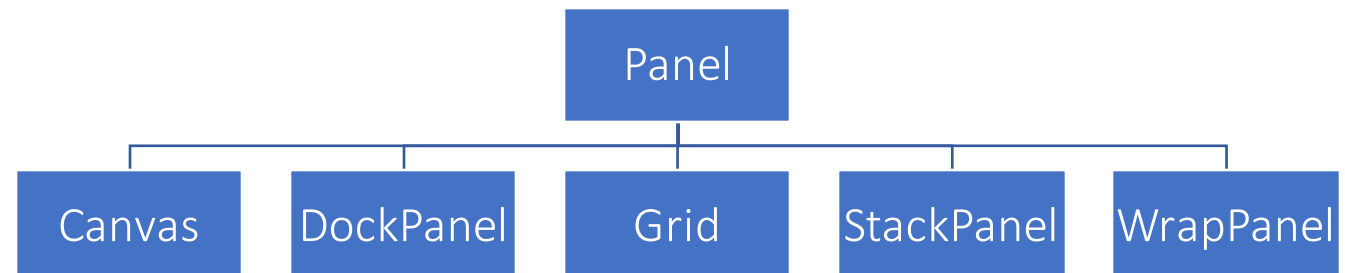
Panels

Introduzione ai diversi pannelli

Panel

I panels agiscono come **contenitori** per altri controlli e ne definiscono il layout.

Poiché una finestra può contenere **un solo** controllo figlio, viene spesso utilizzato un pannello per suddividere lo spazio in aree, dove ogni area può contenere un controllo o un altro pannello.



Proprietà comuni

Di seguito sono elencate alcune proprietà comuni a tutti i container

- Background
- Width
- Height
- MinWidth / MaxWidth
- MinWeight / MaxWeight
- Margin
- Visibility

Canvas

Il canvas è un semplice container, molto simile al comportamento di WinForms.

Consente di assegnare coordinate specifiche a ciascuno dei controlli figlio, dando il controllo totale del layout.

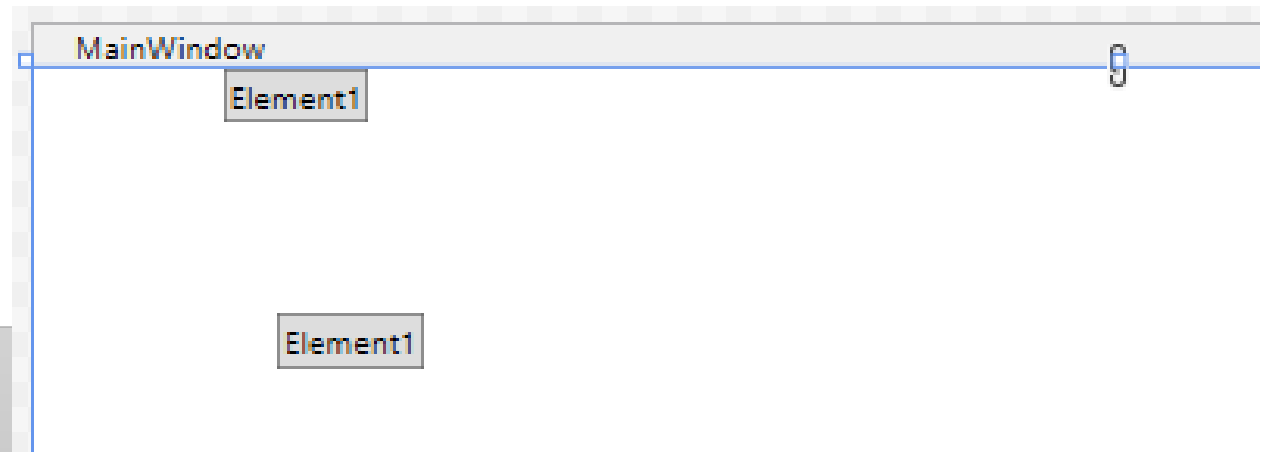
Tuttavia, questo non è molto flessibile, perché bisogna spostare manualmente i controlli figlio e assicurarsi che siano allineati nel modo desiderato.

Inoltre non garantisce un comportamento responsive della UI.

Ideale quando si vuole il controllo completo delle posizioni di controllo figlio.

Esempio 1 Canvas

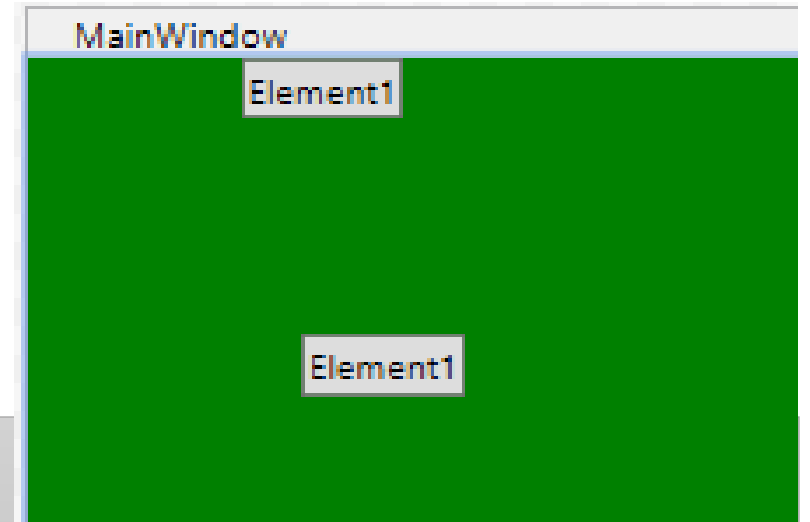
Nella canvas gli elementi **non si "adattano"** al ridimensionamento della finestra, questo **non permette** un comportamento responsive della UI.



```
...
    Title="MainWindow" Height="450" Width="800">
<Canvas>
    <Button x:Name="bElement1" Content="Element1" Canvas.Left="70"></Button>
    <Button x:Name="bElement2" Content="Element1" Canvas.Left="90" Canvas.Top="90"></Button>
</Canvas>
</Window>
```

Esempio 2 Canvas

L'attributo background permette di impostare un colore di sfondo alla canvas



```
...
    Title="MainWindow" Height="450" Width="800">
<Canvas Background="green">
    <Button x:Name="bElement1" Content="Element1" Canvas.Left="70"></Button>
    <Button x:Name="bElement2" Content="Element1" Canvas.Left="90" Canvas.Top="90"></Button>
</Canvas>
</Window>
```

Esercizio

Svolgere l'esercizio 1.3 Canvas

Grid

In XAML un elemento Grid è costituito da una serie di righe e colonne.

Le righe e le colonne vengono definite con gli elementi **RowDefinition** e **ColumnDefinition**.

ShowGridline



The image displays two screenshots of a WPF application window titled "MainWindow". Both windows contain a 3x3 grid of cells, each labeled with its row and column index (e.g., "Row0 Column0", "Row0 Column1", "Row0 Column2" in the first row).

The top screenshot shows the grid with visible dashed blue grid lines. To its right, the XAML code is:

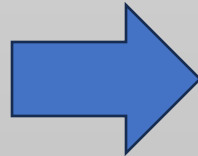
```
<Grid ShowGridLines="True">
```

The bottom screenshot shows the same grid but without visible grid lines. To its right, the XAML code is:

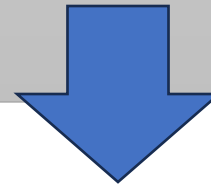
```
<Grid ShowGridLines="False">
```

Grid

```
...  
<Grid ShowGridLines="True">  
  <Grid.RowDefinitions>  
    <RowDefinition Height="*" />  
    <RowDefinition Height="*" />  
  </Grid.RowDefinitions>  
  <Grid.ColumnDefinitions>  
    <ColumnDefinition Width="*" />  
    <ColumnDefinition Width="*" />  
  </Grid.ColumnDefinitions>  
  <Button x:Name="bElement1" Content="Row0 Column0" Grid.Row="0" Grid.Column="0" Width="100" Height="100" />  
</Grid>  
...
```



Definiscono le colonne e le righe della Grid.

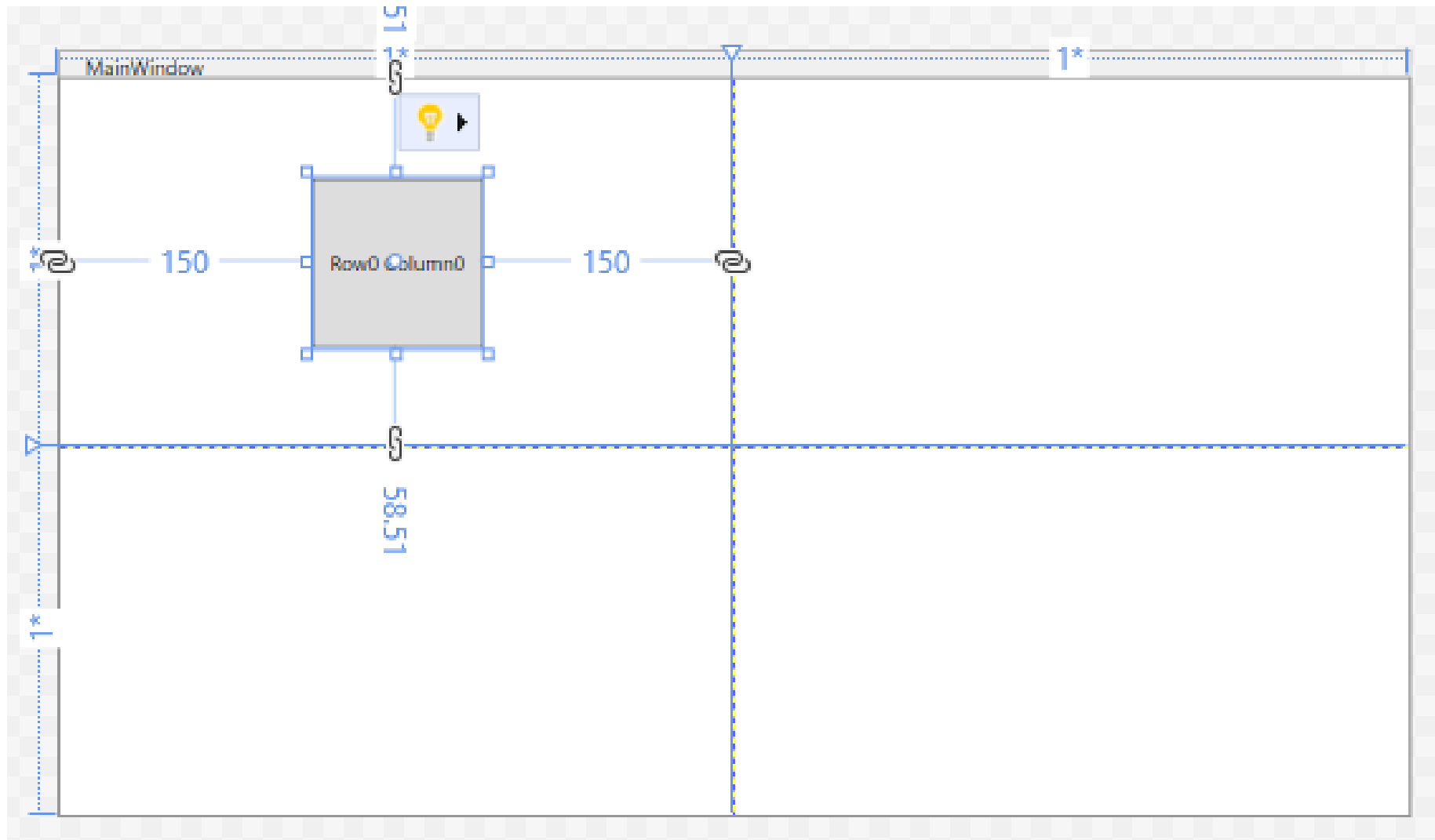


Definiscono la posizione
dell'elemento nella Grid

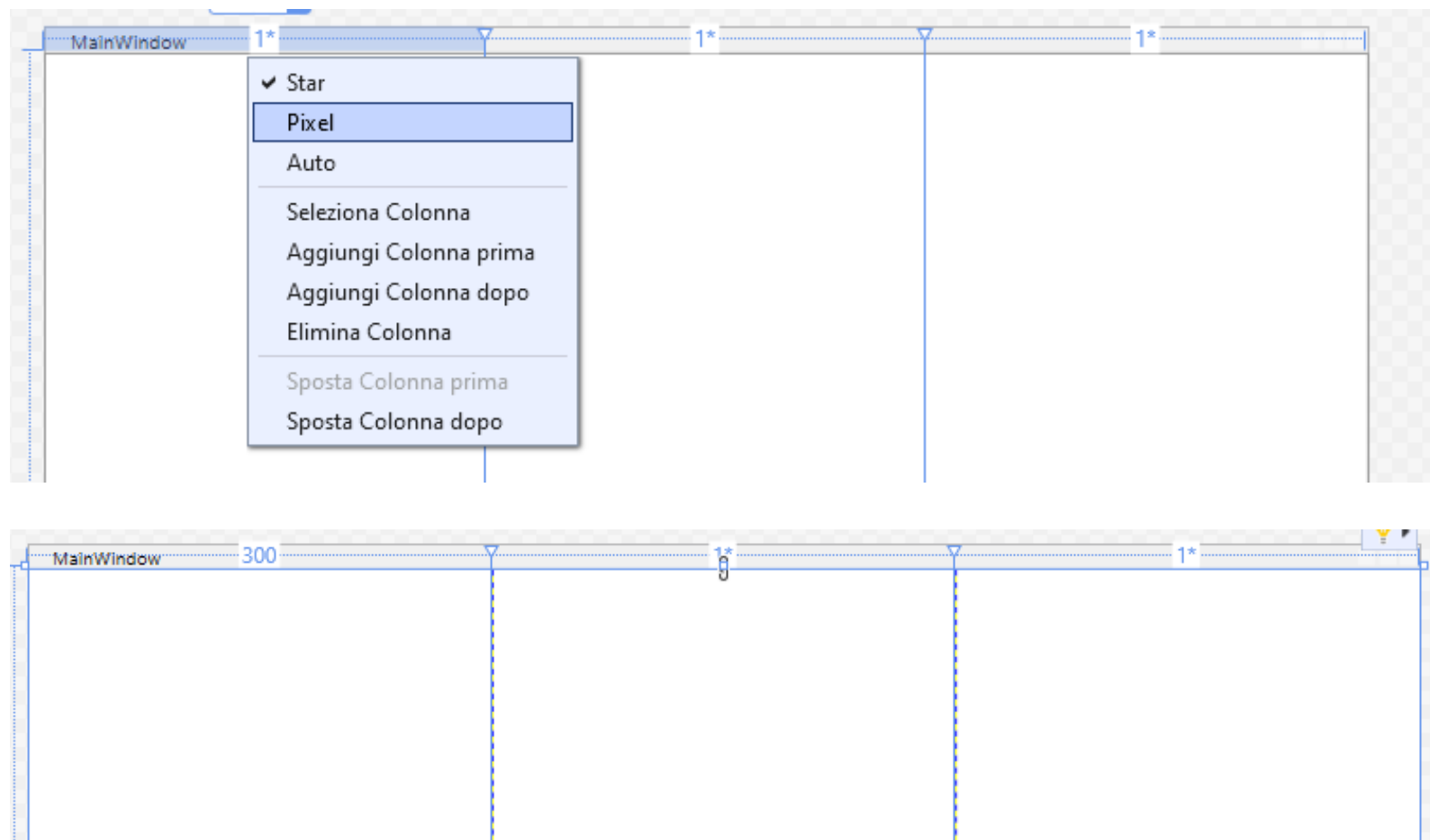


Definiscono l'altezza e
la larghezza del
componente

Grid



Unità di misura

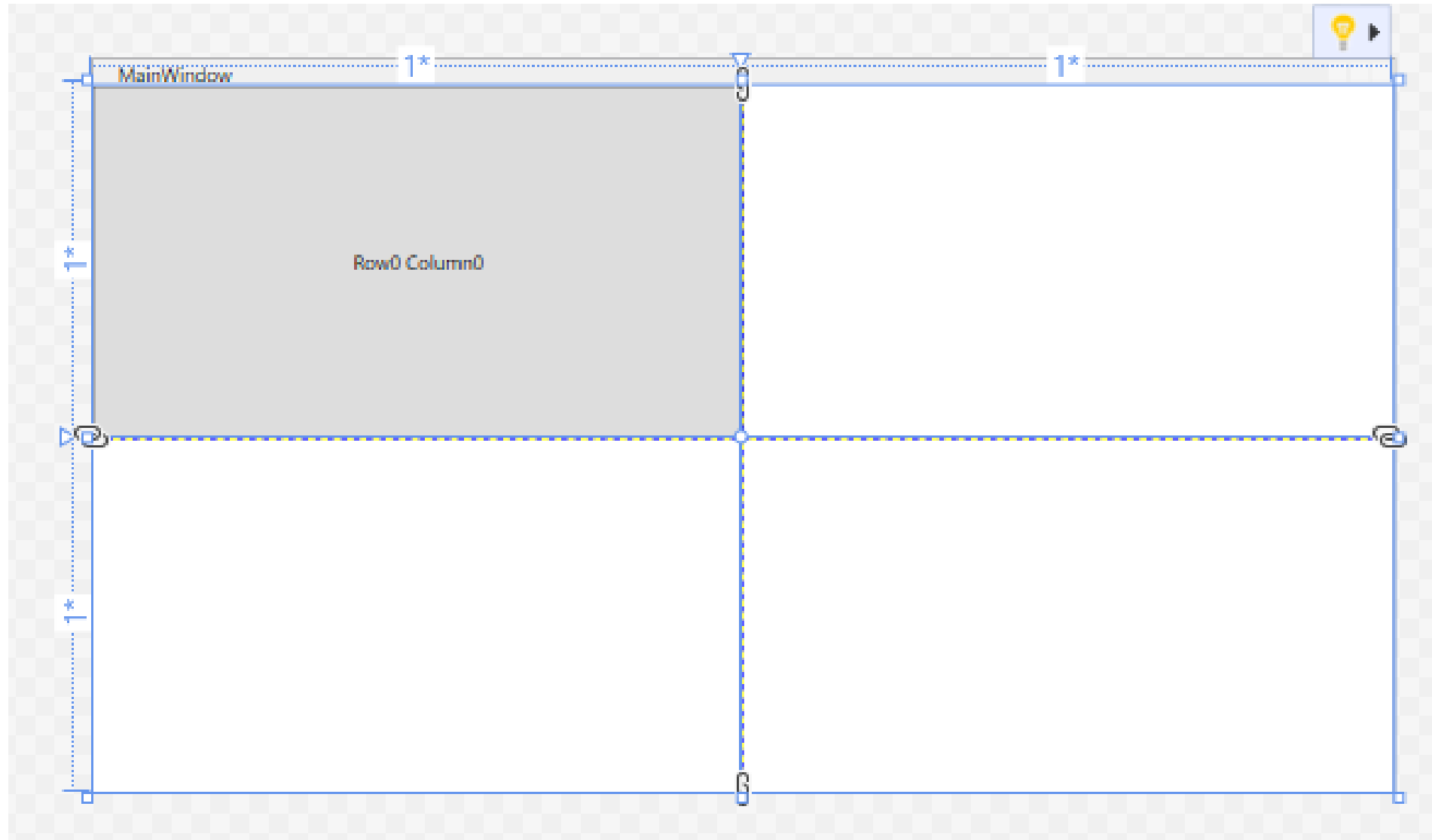


Grid in modo proporzionale

```
...
  Title="MainWindow" Height="450" Width="900">
<Grid ShowGridLines="True">
  <Grid.RowDefinitions>
    <RowDefinition Height="*" />
    <RowDefinition Height="*" />
  </Grid.RowDefinitions>
  <Grid.ColumnDefinitions>
    <ColumnDefinition Width="*" />
    <ColumnDefinition Width="*" />
  </Grid.ColumnDefinitions>
  <Button x:Name="bElement1" Content="Row0 Column0" Grid.Row="0" Grid.Column="0" />
</Grid>
...
```

Se nelle proprietà del controllo inserito nel container, non viene definito height e Width, l'elemento si adatterà alla dimensione della cella

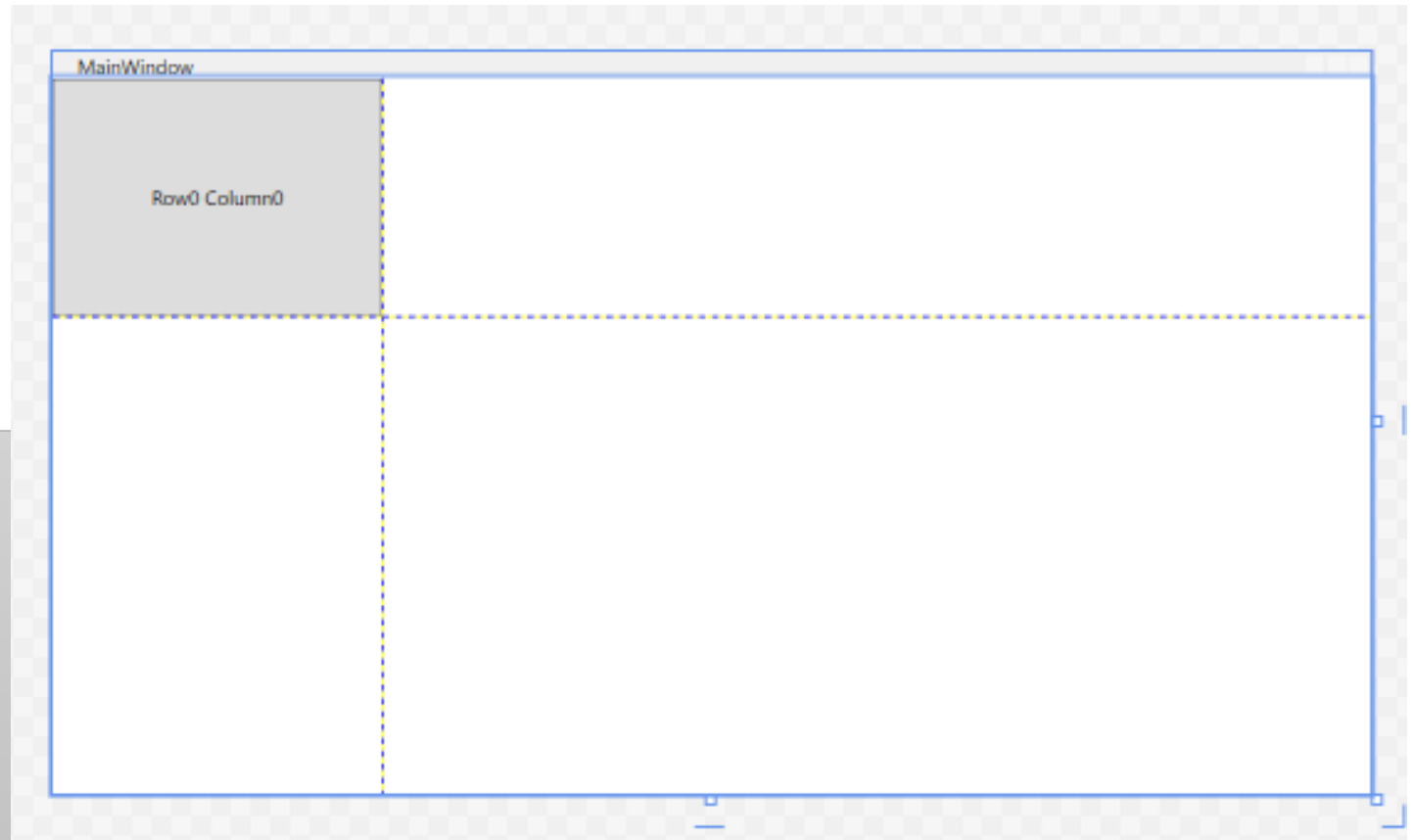
Esempio elemento che si adatta alla dimensione



Proportional Sizing

Nel dimensionamento proporzionale, lo spazio disponibile è diviso proporzionalmente tra colonne e righe.

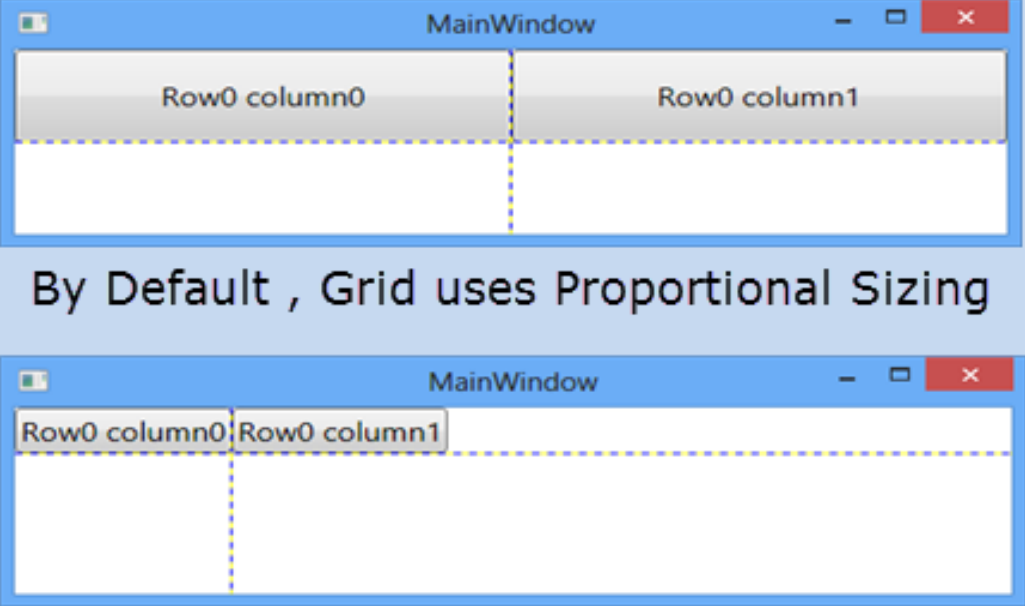
```
<Grid ShowGridLines="True">
  <Grid.RowDefinitions>
    <RowDefinition Height="*" />
    <RowDefinition Height="2*" />
  </Grid.RowDefinitions>
  <Grid.ColumnDefinitions>
    <ColumnDefinition Width="*" />
    <ColumnDefinition Width="3*" />
  </Grid.ColumnDefinitions>
  <Button x:Name="bElement1" Content="Row0 Column0" Grid.Row="0" Grid.Column="0" />
</Grid>
```



Automatic Sizing

Le righe e le colonne vengono ridimensionate automaticamente per adattarsi all'elemento figlio.

```
<Grid ShowGridLines ="True">  
  <Grid.RowDefinitions>  
    <RowDefinition Height="Auto"/>  
    <RowDefinition Height="Auto"/>  
  </Grid.RowDefinitions>  
  <Grid.ColumnDefinitions>  
    <ColumnDefinition Width="Auto"/>  
    <ColumnDefinition Width="Auto"/>  
  </Grid.ColumnDefinitions>  
  <Button x:Name="bElement1" Content="Row0 Column0" Grid.Row ="0" Grid.Column ="0" Width="100" Height="100"/>  
</Grid>
```



By Default , Grid uses Proportional Sizing

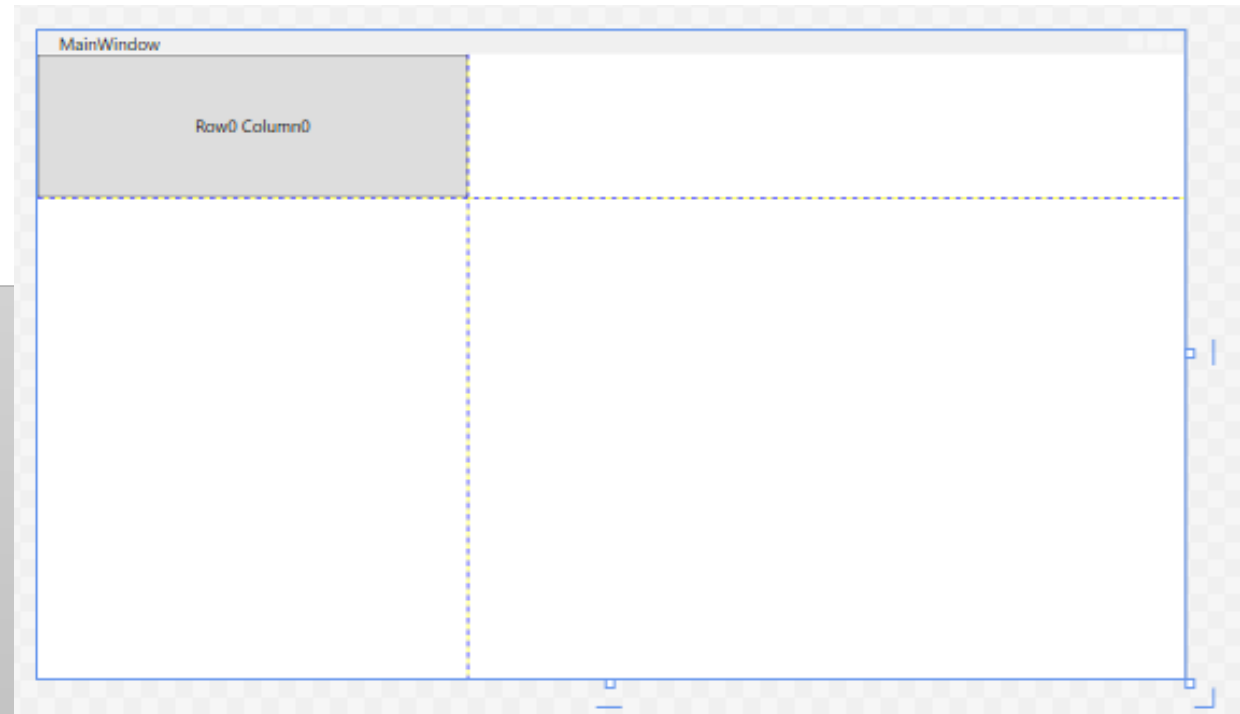
Automatic Sizing reduce the size of object to the size of content

Dimensionamento assoluta

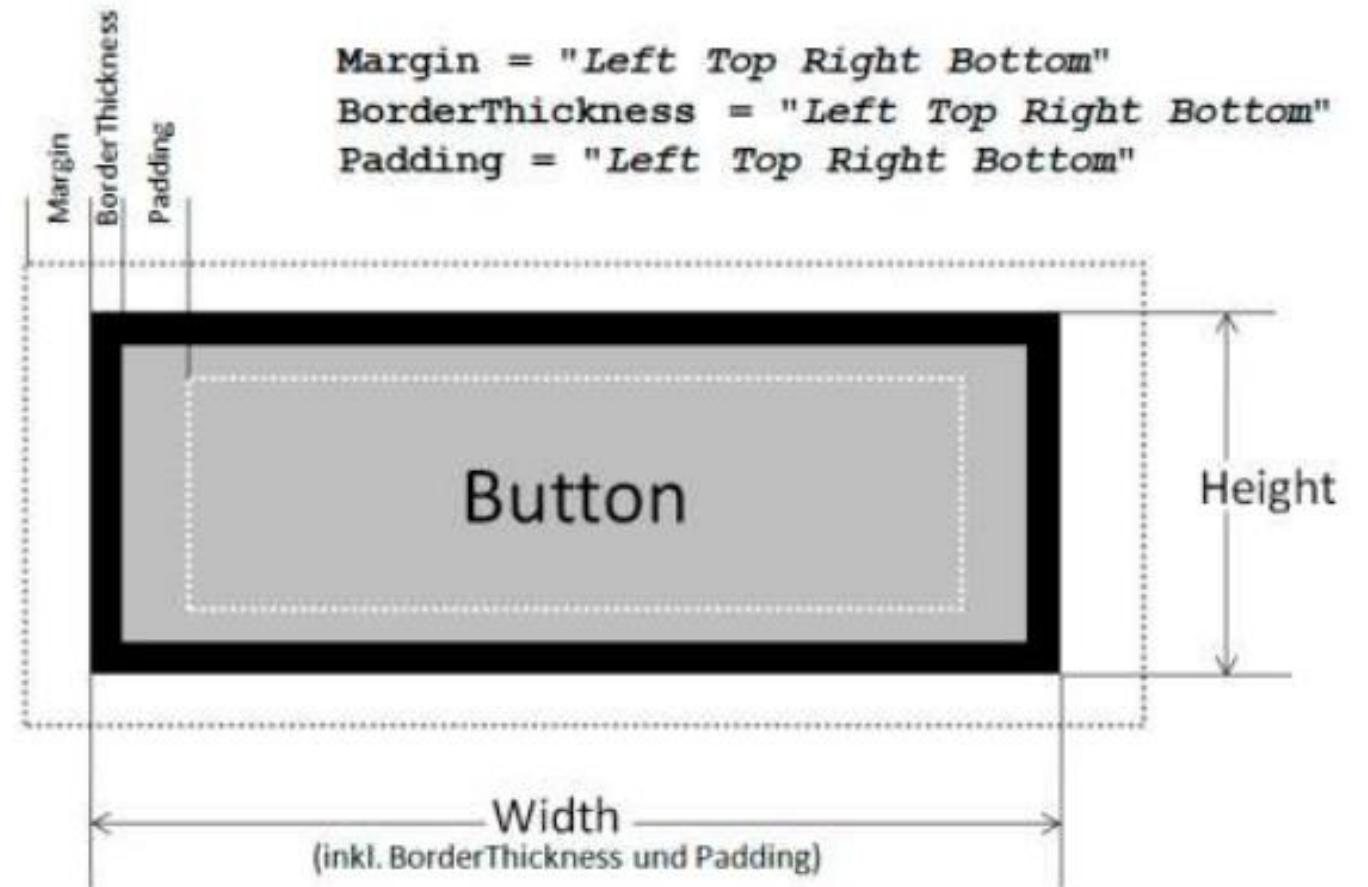
In Absolute Sizing, le dimensioni esatte dell'altezza e della larghezza sono specificate in RowDefinition e ColumnDefinition.

La dimensione della riga e della colonna non si restringe o si espande quando la dimensione della griglia/finestra viene modificata.

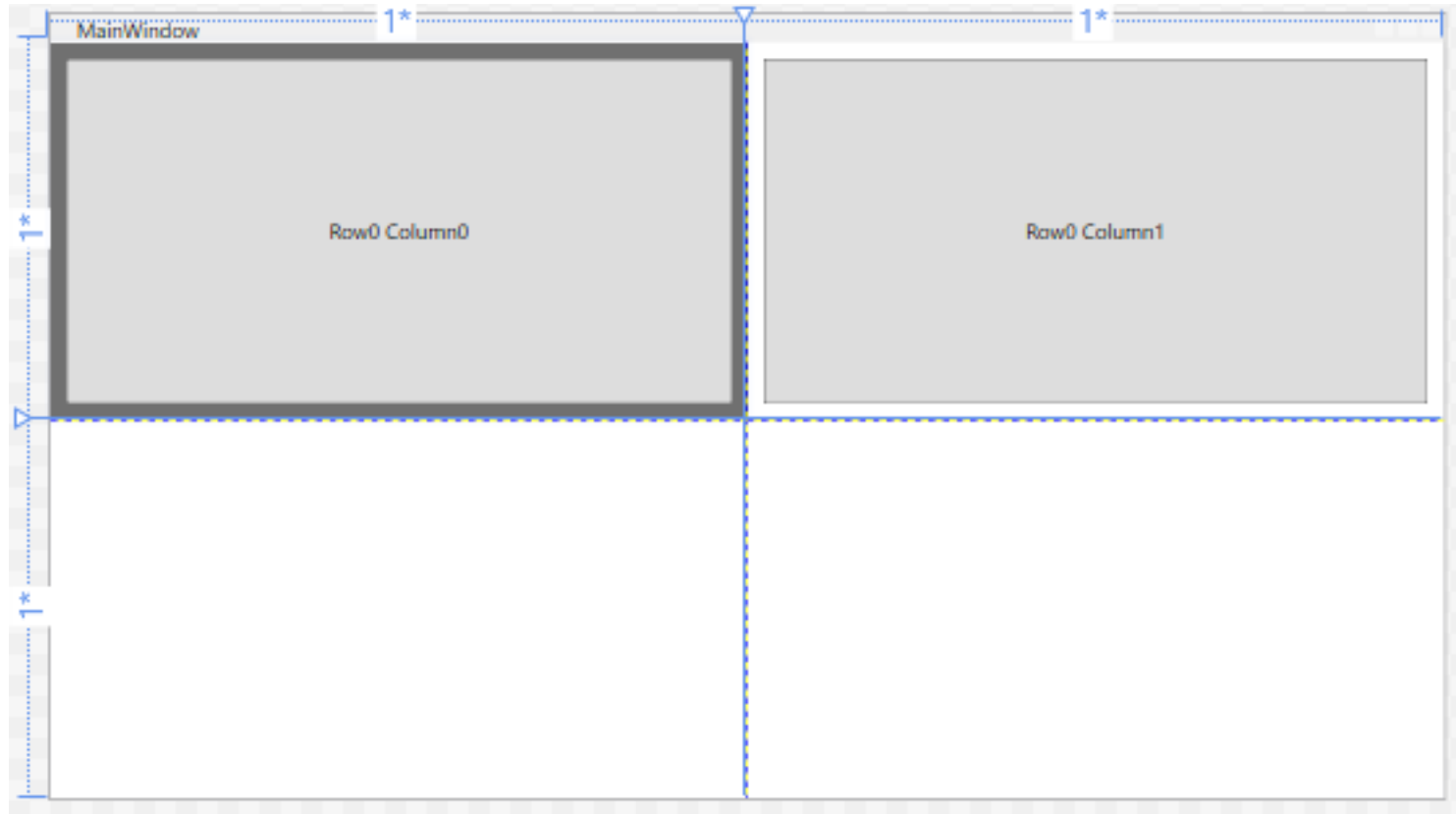
```
<Grid ShowGridLines="True">
  <Grid.RowDefinitions>
    <RowDefinition Height="100"/>
    <RowDefinition Height="350"/>
  </Grid.RowDefinitions>
  <Grid.ColumnDefinitions>
    <ColumnDefinition Width="300"/>
    <ColumnDefinition Width="500"/>
  </Grid.ColumnDefinitions>
  <Button x:Name="bElement1" Content="Row0 Column0" Grid.Row="0" Grid.Column="0"/>
</Grid>
```



Margin, Border e Padding



Esempio



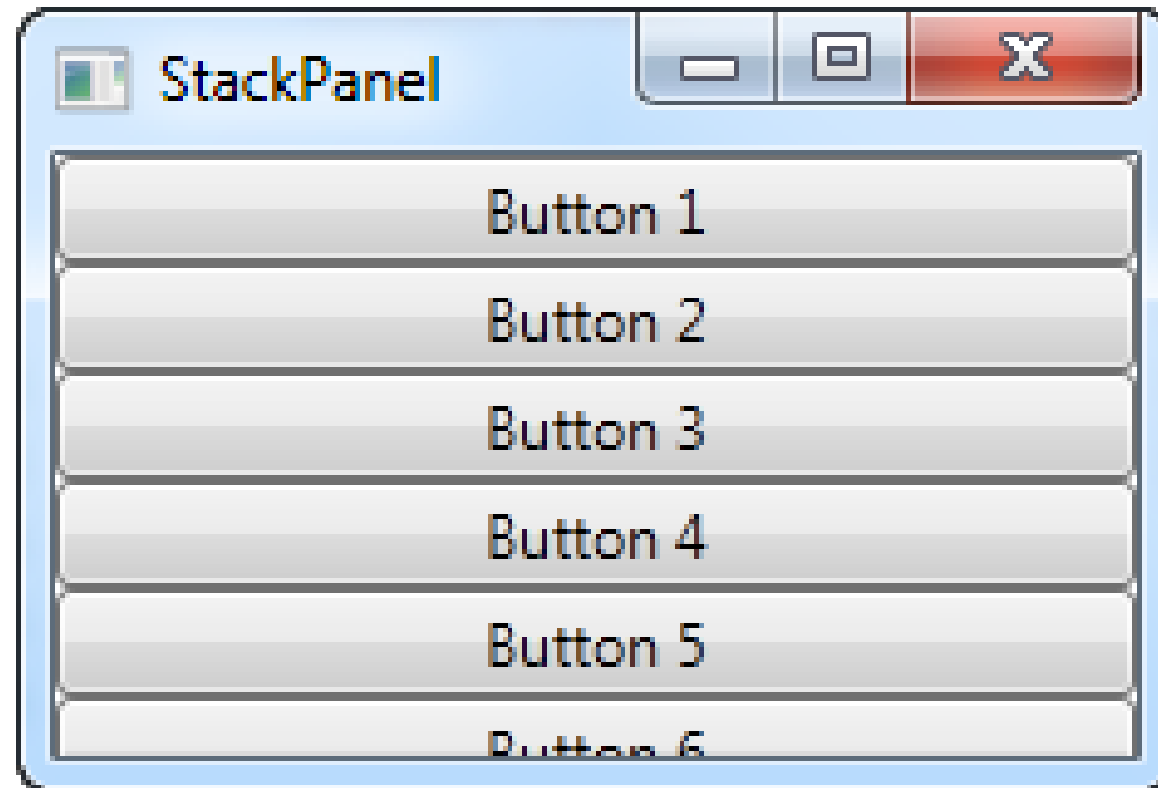
```
<Button x:Name="bElement1" Content="Row0 Column0" Grid.Row="0" Grid.Column="0" BorderThickness="10"/>  
<Button x:Name="bElement2" Content="Row0 Column1" Grid.Row="0" Grid.Column="1" Margin="10"/>
```

Esercizio

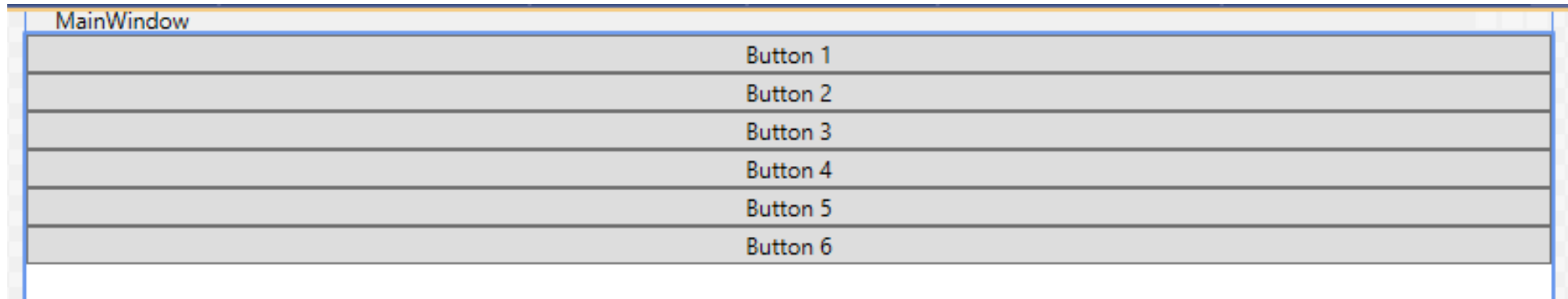
Svolgere l'esercizio 1.4 Buttons

StackPanel

Lo StackPanel espande il contenuto in una direzione, permettendo di impilare gli elementi uno sull'altro.

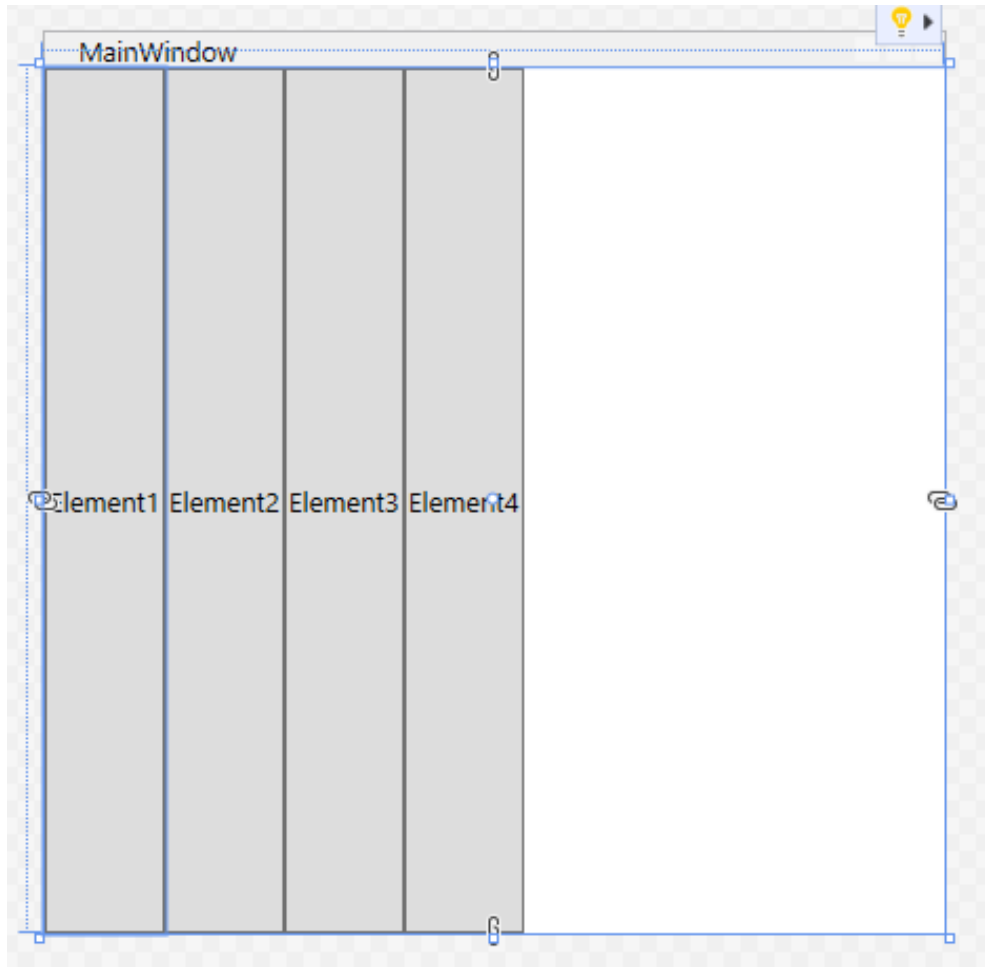


Esempio - Vertical



```
<Grid>
  <StackPanel>
    <Button x:Name="bElement1" Content = "Button1"/>
    <Button x:Name="bElement2" Content = "Button2"/>
    <Button x:Name="bElement3" Content = "Button3"/>
    <Button x:Name="bElement4" Content = "Button4"/>
  </StackPanel>
</Grid>
```

Esempio - Horizontal



```
<Grid>
```

```
  <StackPanel Orientation="Horizontal">
```

```
    <Button x:Name="bElement1" Content = "Element1"/>
```

```
    <Button x:Name="bElement2" Content = "Element2"/>
```

```
    <Button x:Name="bElement3" Content = "Element3"/>
```

```
    <Button x:Name="bElement4" Content = "Element4"/>
```

```
  </StackPanel>
```

```
</Grid>
```

Esercizio

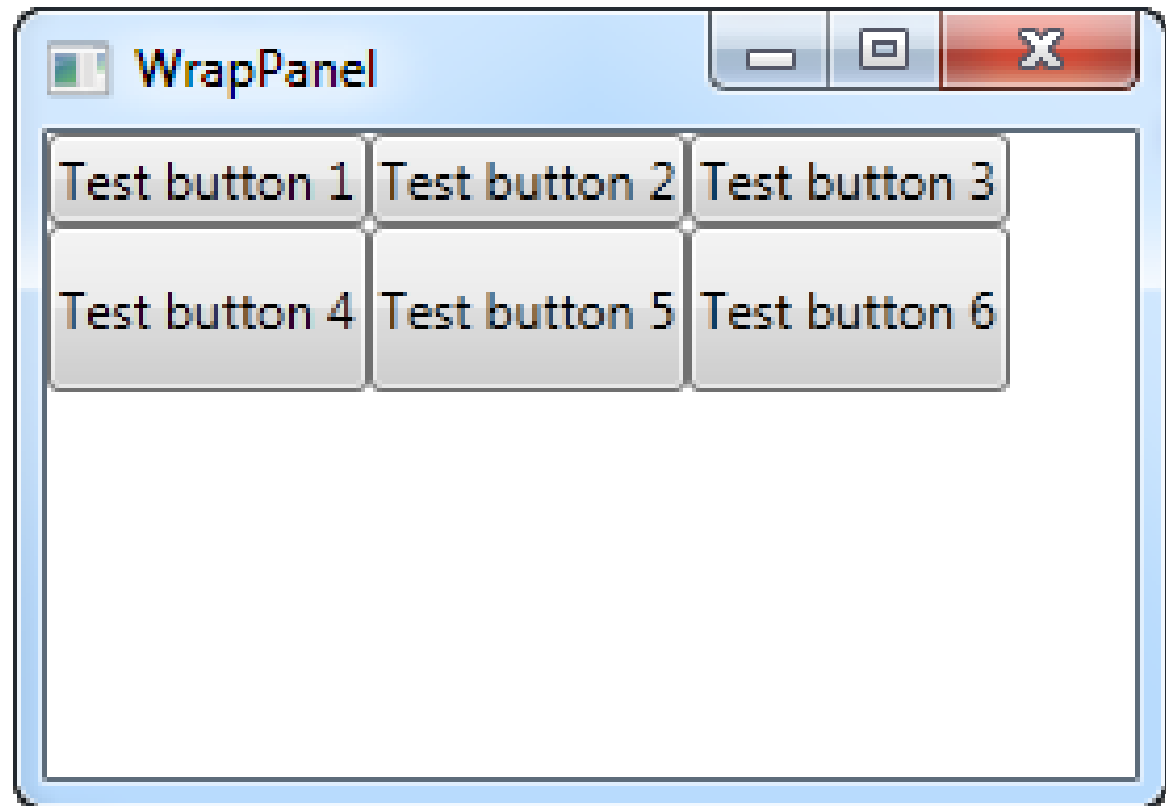
Svolgere l'esercizio 1.5 Pila

Svolgere l'esercizio 1.6 Pila2

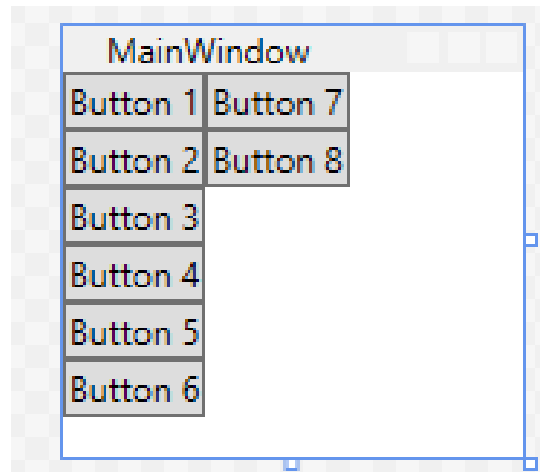
WrapPanel

Il WrapPanel posizionerà i suoi controlli figli uno di fianco all'altro, orizzontalmente (default) o verticalmente, fino a che non finisce lo spazio, per poi spostarsi alla riga successiva e continuare.

Ideale per una lista di controlli orizzontale o verticale che si adatta automaticamente quando non c'è più spazio.

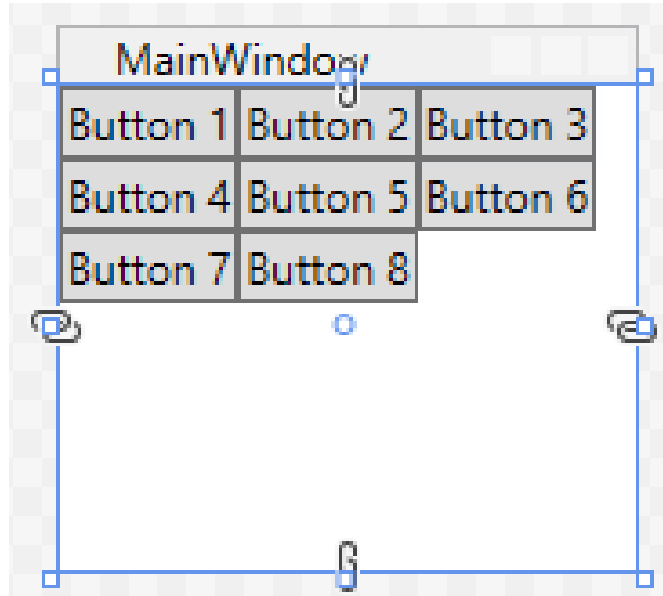


Esempio - Vertical



```
<WrapPanel Orientation="Vertical">  
  <Button Content="Button 1"/>  
  <Button Content="Button 2"/>  
  <Button Content="Button 3"/>  
  <Button Content="Button 4"/>  
  <Button Content="Button 5"/>  
  <Button Content="Button 6"/>  
  <Button Content="Button 7"/>  
  <Button Content="Button 8"/>  
</WrapPanel>
```

Esempio - Horizontal



```
<WrapPanel Orientation="Horizontal">  
  <Button Content="Button 1"/>  
  <Button Content="Button 2"/>  
  <Button Content="Button 3"/>  
  <Button Content="Button 4"/>  
  <Button Content="Button 5"/>  
  <Button Content="Button 6"/>  
  <Button Content="Button 7"/>  
  <Button Content="Button 8"/>  
</WrapPanel>
```

Esercizio

Svolgere l'esercizio 1.7 WrapPanel