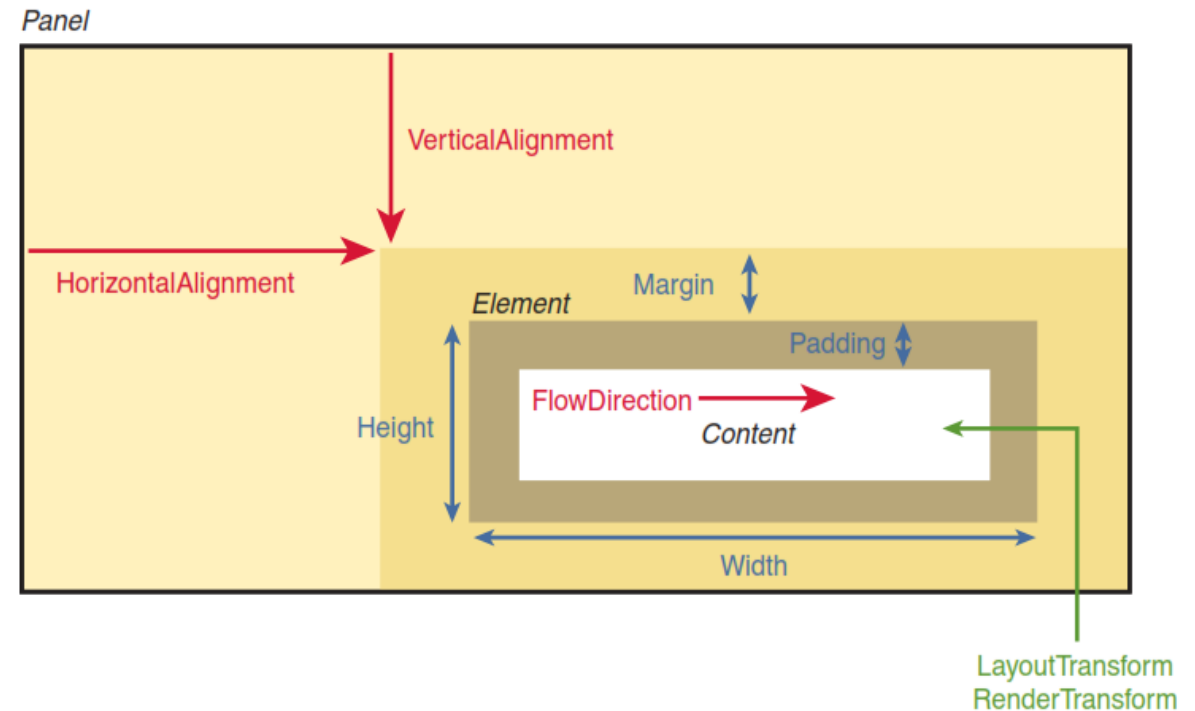


# Objektno orijentirano programiranje - praktikum u .NET okolini

Predavanje 06

# Razmještaj

- Podrazumijevamo dimenzioniranje i pozicioniranje kontrola
- Razlikujemo tri vrste upravljanja:
  - Kontrola veličine
  - Kontrola pozicije
  - Primjena transformacija



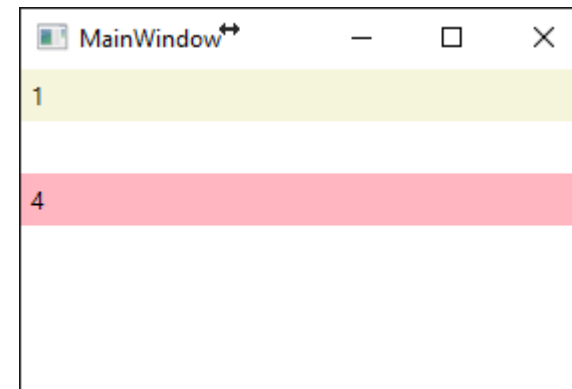
# Kontrola veličine

- WPF elementi uglavnom nastoje prilagoditi veličinu svom sadržaju
- Na određivanje veličine elementa utječu različita svojstva:
  - **Width, Height** – eksplicitna širina i visina („jače” od Stretch postavke)
    - Mora biti između Min... i Max...
  - **MinWidth, MinHeight** – minimalne dimenzije elementa
  - **MaxWidth, MaxHeight** – maksimalne dimenzije elementa
    - Ako postoji više mjesta u roditelju, element se neće povećati, čak i ako je postavka Stretch
  - Svojstva **Margin** i **Padding** – određuju dodatni prostor izvan, odnosno unutar elementa (vrijednost je strukture **Thickness**)

# Kontrola veličine

- Svojstvo **Visibility**:
  - **Visible** – element se iscrtava i sudjeluje u razmještaju
  - **Collapsed** – element je nevidljiv i ne sudjeluje u razmještaju
  - **Hidden** – element je nevidljiv, ali sudjeluje u razmještaju

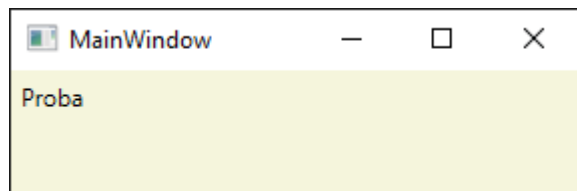
```
<StackPanel>
  <Label Background="Beige"
    Visibility="Visible">1</Label>
  <Label Background="Blue"
    Visibility="Hidden">2</Label>
  <Label Background="Orange"
    Visibility="Collapsed">3</Label>
  <Label Background="LightPink"
    Visibility="Visible">4</Label>
</StackPanel>
```



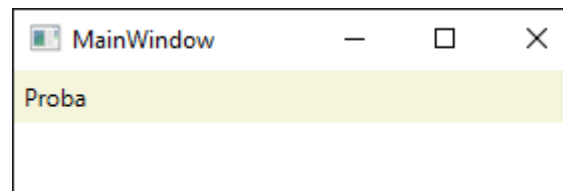
# Kontrola pozicije

- Svojstva **HorizontalAlignment** i **VerticalAlignment** određuju vodoravno i okomito poravnanje kontrole unutar svog kontejnera
  - Vodoravno: Left, Right, Center, Stretch
  - Okomito: Top, Bottom, Center, Stretch
  - Podrazumijevana vrijednost oba svojstva najčešće je Stretch

```
<Grid>
  <Label Background="Beige">
    Proba
  </Label>
</Grid>
```



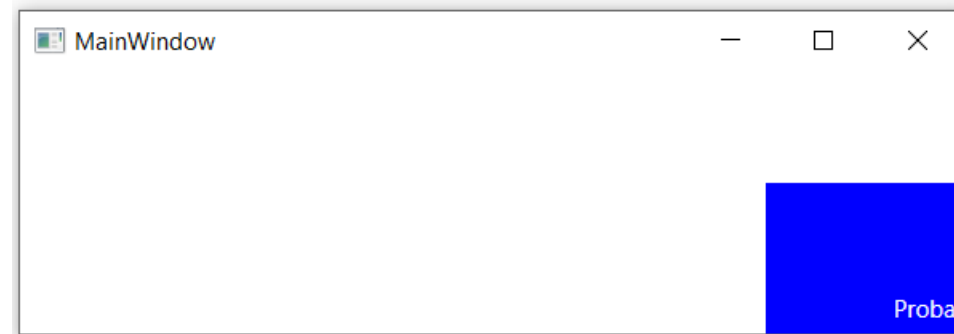
```
<StackPanel>
  <Label Background="Beige">
    Proba
  </Label>
</StackPanel>
```



# Kontrola pozicije

- Svojstva **HorizontalAlignment** i **VerticalContentAlignment** – određuju poravnanje sadržaja unutar kontrole
  - Podrazumijevano su **Left** i **Top**

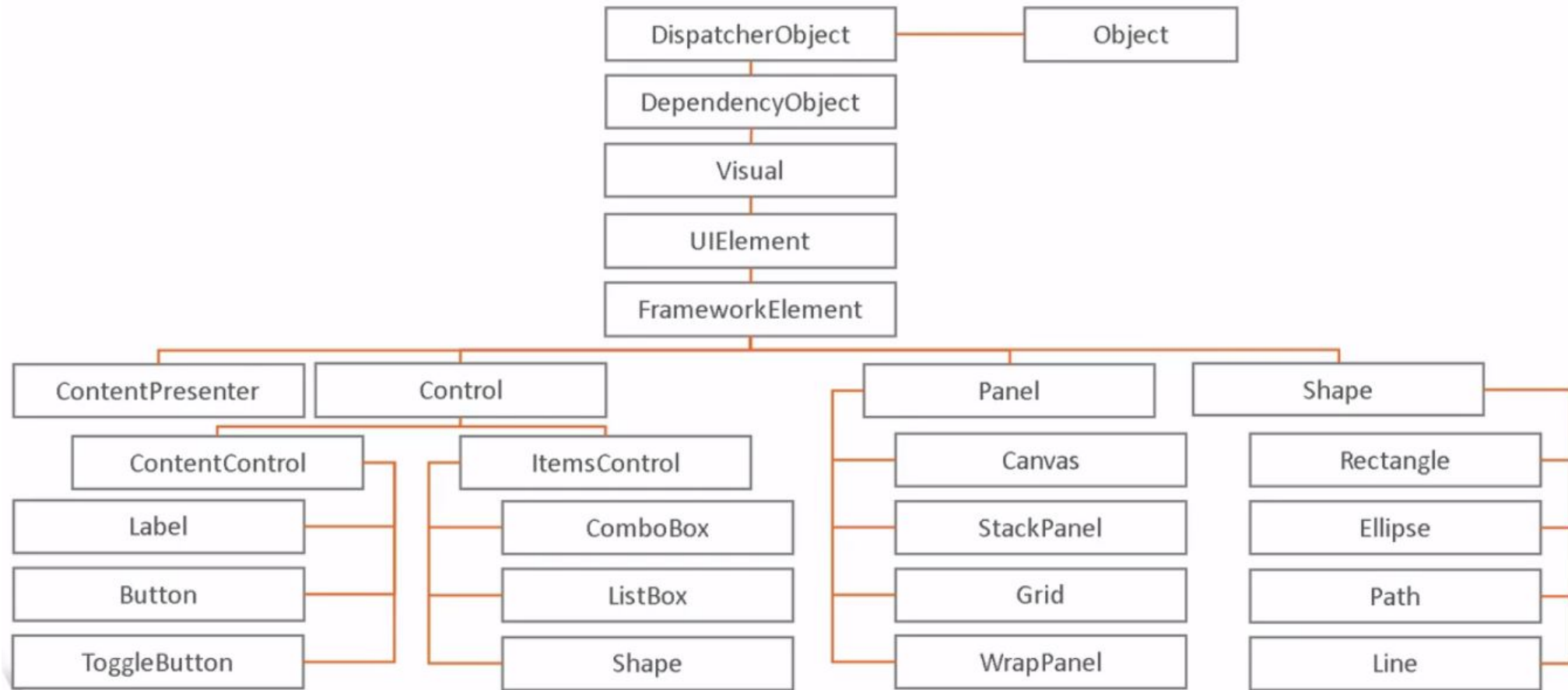
```
<Grid>  
    <Label Background=■"Blue"  
           Foreground=□"White"  
           Content="Proba"  
           Width="100"  
           Height="75"  
           HorizontalAlignment="Right"  
           VerticalAlignment="Bottom"  
           HorizontalContentAlignment="Right"  
           VerticalContentAlignment="Bottom"/>  
</Grid>
```



# WPF principi

- WPF preporuča sljedeće:
  - Elementima ne treba stavljati eksplicitnu veličinu
    - Trebaju rasti sa svojim sadržajem
    - Prihvatljive granice se određuju s Min... i Max... postavkama
  - Elementi ne trebaju definirati svoju poziciju koordinatama
    - Treba ih slagati njihov roditelj prema njihovoj veličini, redoslijedu i drugim informacijama specifičnim za roditelja
    - Za dodatni prostor treba koristiti Margin
  - Paneli dijele raspoloživi prostor djeci
    - Svakome pokušavaju dati željenu veličinu
  - Paneli se ugnježđuju: najčešće prvo Grid, StackPanel pa ostali...

# Hijerarhija klasa





# Razmještaj pomoću panela

- Najvažniji paneli u WPF-u su:
  - Grid
  - StackPanel
  - WrapPanel
  - DockPanel
  - Canvas

# Grid

- Najkorisniji i najčešće korišteni panel koji omogućuje razmještaj svojih elemenata u više redaka i stupaca
  - Koristi se za dijeljenje prozora u više regija, a u pojedinim regijama često se koriste StackPaneli
  - **RowDefinitions** svojstvo sadrži **RowDefinition** elemente koji definiraju retke
  - **ColumnDefinitions** svojstvo sadrži **ColumnDefinition** elemente koji definiraju stupce
  - Pripojena svojstva **Grid.RowSpan** i **Grid.ColumnSpan** omogućuju sadržaju da se prostire kroz više ćelija

# Grid

- Veličina redaka i stupaca izražava se vrijednošću tipa **System.Windows.GridLength** koja može biti:
  - **Apsolutna** – kao visina i širina postavlja se numerička vrijednost
  - **Auto** – veličina se postavlja automatski prema sadržaju
  - **Proporcionalna (\* - zvjezdica)** – dostupan prostor dijeli se proporcionalno (podrazumijevana vrijednost)
- Kontrole se postavljaju u odgovarajuću ćeliju Grida pomoću pripojenih svojstava `Grid.Column` i `Grid.Row`

# StackPanel

- Koristi se za raspored manjeg broja kontrola i često se postavlja u ćeliju Grida
- Ovisno o svojstvu **Orientation** slaže redom elemente okomito (podrazumijevano) ili vodoravno

# WrapPanel

- Elemente slaže kao **StackPanel**, osim što u ih nedostatku mjesta prebacuje u dodatne retke ili stupce ovisno o orijentaciji
- Najvažnija svojstva:
  - **Orientation** – kao kod `StackPanela`, osim što se podrazumijeva vodoravna orijentacija

# DockPanel

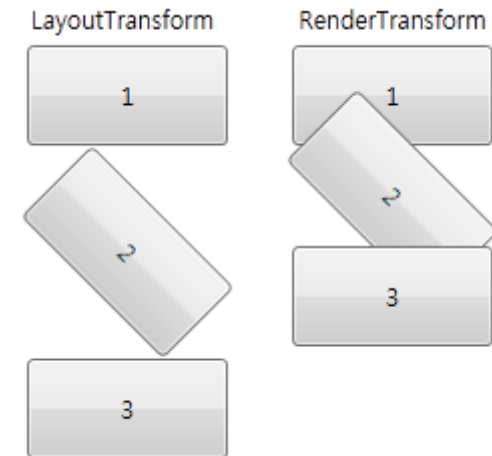
- Omogućuje prislanjanje elemenata duž cijele strane panela rastežući ih do ispunjenja cijele njegove širine, odnosno visine
- Priloženo svojstvo **Dock** može poprimiti četiri vrijednosti:
  - **Left**
  - **Top**
  - **Right**
  - **Bottom**
- Nedostatak vrijednosti **Fill** može se nadomjestiti konfiguracijom svojstva **LastChildFill**

# Canvas

- Podržava razmještaj elemenata u klasičnom smislu – pozicioniranje pomoću eksplicitnih koordinata
  - Koordinate su jedinice neovisne o uređaju
  - Osim u odnosu na gornji lijevi kut, koordinate se mogu specificirati i u odnosu na bilo koji drugi kut
  - Svojstva: Left, Top, Right, Bottom
- Predodređeni Z-redosljed određen je prema redosljedu dodavanja elemenata
  - Priloženo svojstvo **Panel.ZIndex** može promijeniti redosljed
- Rijetko se koristi

# Primjena transformacija

- Svi elementi imaju dva svojstva tipa **Transform** koja se koriste za transformacije elemenata:
  - **LayoutTransform** – primjenjuje se u fazi razmještaja, prije iscrtavanja
  - **RenderTransform** – primjenjuje se nakon iscrtavanja
- **RenderTransformOrigin** predstavlja početnu točku
- Metode transformacije:
  - **RotateTransform** – rotacija
  - **ScaleTransform** – povećanje ili smanjenje elementa
  - **SkewTransform** – iskrivljenje elementa
  - **TranslateTransform** – premještanje elementa





# Događaji

- **Usmjereni događaji** (engl. *routed events*) su događaji dizajnirani za rad sa stablom elemenata
  - Skoro svi WPF događaji su usmjereni
- Svaki usmjereni događaj koristi jednu od tri strategije :
  - **Tunneling** – događaj se prvo podiže na korijenu, a zatim na svakom elementu niz stablo dok ne stigne do elementa koji je stvarni izvor događaja (obično imaju prefiks `Preview`)
    - npr. `PreviewMouseDown()`
  - **Bubbling** – događaj se najprije podiže na izvornom elementu, a zatim redom na svakom elementu iznad u hijerarhiji dok ne stigne do korijena
    - npr. `MouseDown()`
  - **Direct** – događaj se podiže samo na izvornom elementu
    - npr. `MouseEnter()`

# Tip RoutedEventArgs

- Sve metode za obradu imaju prvi parametar tipa **object**, a drugi tipa **RoutedEventArgs** (nasljeđuje **EventArgs**)
- Klasa **RoutedEventArgs** ima sljedeća korisna svojstva:
  - **Source** – element logičkog stabla koji je podigao događaj
  - **OriginalSource** – element vizualnog stabla koji je podigao događaj
- Ako kliknemo na rub **Label** kontrole:
  - **Source**: `System.Windows.Controls.Label`
  - **OriginalSource**: `System.Windows.Controls.Border`

# Primjer usmjerenih događaja

- Prijenos usmjerenih događaja uz i niz stablo događaja se čak i ako neki elementi ne podržavaju određeni događaj
  - Zbog toga se usmjereni događaji nazivaju i **priloženi događaji** (engl. *attached events*)
- Primjerice, StackPanel ne podržava događaj Click, ali ga može obraditi:

```
<StackPanel Button.Click="StackPanel_Click">  
    <Label Content="Ja sam labela" />  
</StackPanel>
```