



Objektno orijentirano programiranje - praktikum u .NET okolini

Predavanje 07

Teme

- Resursi
- Animacije
- Stilovi
- *Lokalizacija
- *Korisničke kontrole

Resursi

- Daju nam mogućnost spremanja podataka za pojedinu kontrolu, prozor ili aplikaciju
- Mogu se spremiti bilo koji .NET objekti

```
<Window x:Class="Predavanje6.MainWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:local="clr-namespace:Predavanje6"
        xmlns:clr="clr-namespace:System;assembly=mscorlib"
        Title="MainWindow" Height="350" Width="525">
    <Window.Resources>
        <clr:String x:Key="cs">Server=.;Database=AdventureWorksOBP;Uid=sa;Pwd=SQL</clr:String>
    </Window.Resources>
    <Grid>
        <TextBlock Text="{StaticResource cs}" />
    </Grid>
</Window>
```

Resursi

- Možemo ih definirati na razini elementa, prozora ili aplikacije
- Prilikom definiranja postavljamo atribut **x:Key**
 - U XAML-u:
 - Kao vrijednost atributa - koristimo proširenje `StaticResource`
 - Vrijednost se učitava nakon učitavanja XAML-a, a promjena vrijednosti za vrijeme trajanja aplikacije neće se manifestirati u aplikaciji
 - `DynamicResource`
 - Vrijednost se učitava kada ju se pokuša dohvatiti, a promjena vrijednosti za vrijeme trajanja aplikacije manifestirat će se u aplikaciji
 - Kao novi element - `StaticResourceExtension`
 - U kôdu koristimo metodu `FindResource()` definiranu na svakom elementu

Primjeri korištenja resursa

```
<Window x:Class="Predavanje6.MainWindow"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  xmlns:local="clr-namespace:Predavanje6"
  xmlns:clr="clr-namespace:System;assembly=mscorlib"
  Background="{DynamicResource AppBackground}"
  Title="MainWindow" Height="350" Width="525">
  <Window.Resources>
    <clr:String x:Key="cs">Server=.;Database=AdventureWorksOBP;Uid=sa;Pwd=SQL</clr:String>

    <x:Array x:Key="cbOpcije" Type="clr:String">
      <clr:String>Opcija 1</clr:String>
      <clr:String>Opcija 2</clr:String>
      <clr:String>Opcija 3</clr:String>
    </x:Array>

    <LinearGradientBrush x:Key="AppBackground" StartPoint="0.5,0" EndPoint="0.5, 1">
      <GradientStop Color="DarkBlue" Offset="0"/>
      <GradientStop Color="LightBlue" Offset="1"/>
    </LinearGradientBrush>
  </Window.Resources>
  <Grid>
    <TextBlock Foreground="White" Margin="10">
      <TextBlock.Text>
        <StaticResource ResourceKey="cs"/>
      </TextBlock.Text>
    </TextBlock>
    <ComboBox ItemsSource="{StaticResource cbOpcije}" Width="100" Height="20"/>
  </Grid>
</Window>
```

Animacije

- Animirati možemo svako ovisno svojstvo za koje postoji odgovarajući tip animacije (postoji preko 40 ugrađenih tipova animacije)
- WPF ima ugrađena tri tipa animacija:
 - Interpolacijske
 - Definiramo početnu i krajnju vrijednost ovisnog svojstva te vrijeme trajanja animacije
 - Animacije ključnog okvira (engl. *key frame*)
 - Definiramo ključne točke koje zatim WPF povezuje u zadanom trajanju
 - Animacije putanje (engl. *path-based*)
 - Služe najviše za pomicanje elementa duž zadane putanje (*napredne animacije)

Ugrađene animacije

- Prema tipu podataka ovisnog svojstva kojeg želimo animirati moramo odabrati odgovarajuću animaciju
- WPF dolazi s nizom ugrađenih animacija:
 - DoubleAnimation
 - ColorAnimation
 - Int32Animation
 - PointAnimation
 - ThicknessAnimation
 - ...

Definiranje animacije

- Svaka animacija podržava sljedeća svojstva:
 - **Storyboard.TargetProperty**: ovisno svojstvo koje animiramo
 - **From**: početna vrijednost ovisnog svojstva koje animiramo
 - **To**: završna vrijednost ovisnog svojstva koje animiramo
 - **Duration**: trajanje animacije u formatu h:m:s.ms
 - **AutoReverse**: animacija se odvija i unatrag ili ne
 - **RepeatBehavior**: koliko puta treba ponoviti animaciju (1x, 2x, ..., Forever)

Smještaj Storyboarda

- Animacije se uvijek definiraju unutar Storyboard objekta
- Storyboard definira ovisno svojstvo koje se animira
- BeginStoryboard definira kada animacija počinje
 - Obično se pokreće nakon nekog događaja
 - Stavljamo ga kao sadržaj EventTriggera
 - EventTrigger može se postaviti unutar Triggers kolekcije bilo kojeg elementa

Primjer jednostavne animacije

```
<Button HorizontalAlignment="Center" VerticalAlignment="Top"
    Height="50">
    <Button.Triggers>
        <EventTrigger RoutedEvent="Window.Loaded">
            <EventTrigger.Actions>
                <BeginStoryboard>
                    <Storyboard TargetProperty="Width">
                        <DoubleAnimation
                            From="100"
                            To="200"
                            Duration="0:0:1"
                            AutoReverse="True"
                            RepeatBehavior="Forever"/>
                    </Storyboard>
                </BeginStoryboard>
            </EventTrigger.Actions>
        </EventTrigger>
    </Button.Triggers>
</Button>
```

Primjeri složenijih animacija

```
<Button HorizontalAlignment="Center" VerticalAlignment="Top" Content="Klikni me" Width="100"
Height="50">
  <Button.Background>
    <LinearGradientBrush StartPoint="0, 0.5" EndPoint="1, 0.5">
      <GradientStop Offset="0" Color="DarkBlue"/>
      <GradientStop Offset="0" Color="White"/>
      <GradientStop Offset="1" Color="DarkBlue"/>
    </LinearGradientBrush>
  </Button.Background>
  <Button.Triggers>
    <EventTrigger RoutedEvent="Button.Loaded">
      <EventTrigger.Actions>
        <BeginStoryboard>
          <Storyboard TargetProperty="Background.GradientStops[1].Color">
            <ColorAnimation
              From="White"
              To="Yellow"
              Duration="0:0:1"
              AutoReverse="True" RepeatBehavior="Forever"/>
          </Storyboard>
        </BeginStoryboard>
        <BeginStoryboard>
          <Storyboard TargetProperty="Background.GradientStops[1].Offset">
            <DoubleAnimation
              To="1"
              Duration="0:0:1"
              AutoReverse="True" RepeatBehavior="Forever"/>
          </Storyboard>
        </BeginStoryboard>
      </EventTrigger.Actions>
    </EventTrigger>
  </Button.Triggers>
</Button>
```

Stilovi

- Stil sadrži niz postavki koje se primjenjuju na željene elemente
 - Cilj je ujednačiti izgled elemenata i omogućiti jednostavne izmjene
 - Mogu se definirati na razini kontrole, prozora ili aplikacije
- Stil se obično sastoji od:
 - Postavljača (engl. *setters*) koji postavljaju vrijednosti ovisnih svojstava
 - Okidača koji reagiraju na događaj i pokreću animacije (`EventTrigger`) ili postavljaju svojstva (`Trigger`)

Primjena stila na *child* elemente

- Primjenjuje se na sve *child* elemente (ne samo direktne)

```
<StackPanel Margin="10">
    <StackPanel.Resources>
        <Style TargetType="TextBlock">
            <Setter Property="Foreground" Value="Blue" />
            <Setter Property="FontSize" Value="21" />
        </Style>
    </StackPanel.Resources>
    <TextBlock>Prva linija teksta</TextBlock>
    <TextBlock>Druga linija teksta</TextBlock>
    <TextBlock Foreground="Red">
        Svojstvo elementa ima veći prioritet od primijenjenog stila
    </TextBlock>
</StackPanel>
```

Primjena stila prema nazivu

- Stil se obično definira kao resurs
 - **x:Key** – definiranjem ključa, stil se mora eksplicitno primijeniti na pojedinu kontrolu
 - Najčešće se definira i **TargetType** na odgovarajući tip elementa

```
<Window.Resources>
  <Style x:Key="stil1" TargetType="{x:Type Label}">
    <Style.Setters>
      <Setter Property="Background"
        Value="BlanchedAlmond" />
      <Setter Property="Foreground" Value="DarkCyan" />
      <Setter Property="Padding" Value="3" />
      <Setter Property="Margin" Value="3" />
    </Style.Setters>
  </Style>
</Window.Resources>
<StackPanel>
  <Label Style="{StaticResource ResourceKey=stil1}"
    Content="Jedan"/>
  <Label Style="{StaticResource ResourceKey=stil1}"
    Content="Dva"/>
</StackPanel>
```

Primjena stila prema tipu

- Ako pri definiranju stila izostavimo **x:Key**, odnosi se na sve elemente definiranog tipa

```
<Window.Resources>
  <Style TargetType="{x:Type Label}">
    <Style.Setters>
      <Setter Property="Background"
              Value="BlanchedAlmond" />
      <Setter Property="Foreground" Value="DarkCyan" />
      <Setter Property="Padding"   Value="3" />
      <Setter Property="Margin"   Value="3" />
    </Style.Setters>
  </Style>
</Window.Resources>
<StackPanel>
  <Label Content="Jedan"/>
  <Label Content="Dva"/>
</StackPanel>
```

Primjena više stilova

- Jedan stil možemo temeljiti na drugom stilu i tako postići efekt nasljeđivanja

```
<Window.Resources>
  <Style x:Key="s1" TargetType="{x:Type Label}">
    <Setter Property="Background"
      Value="BlanchedAlmond" />
    <Setter Property="Foreground" Value="DarkCyan" />
    <Setter Property="Padding" Value="3" />
    <Setter Property="Margin" Value="3" />
  </Style>
  <Style x:Key="s2"
    TargetType="{x:Type Label}"
    BasedOn="{StaticResource s1}">
    <Setter Property="FontSize" Value="25" />
  </Style>
</Window.Resources>
<StackPanel>
  <Label Style="{StaticResource ResourceKey=s1}"
    Content="Jedan"/>
  <Label Style="{StaticResource ResourceKey=s2}"
    Content="Dva"/>
</StackPanel>
```


Primjena okidača u stilu

- Stil često sadrži i okidače (kolekcija Triggers)
 - Objekt **Trigger** omogućuje postavljanje niza vrijednosti kada zadano svojstvo popravi zadanu vrijednost

```
<StackPanel>
  <Button>
    <Button.Style>
      <Style TargetType="Button">
        <Setter Property="Foreground" Value="Blue"/>
        <Setter Property="Content" Value="Ja sam gumb"/>
        <Style.Triggers>
          <Trigger Property="IsMouseOver" Value="True">
            <Setter Property="Foreground" Value="Green"/>
            <Setter Property="Content" Value="Klikni me" />
          </Trigger>
        </Style.Triggers>
      </Style>
    </Button.Style>
  </Button>
</StackPanel>
```

Primjena okidača u stilu

- **EventTrigger** omogućuje korištenje animacija kada se dogodi zadani događaj

```
<Style TargetType="{x:Type Label}">
  <Style.Setters>
    <Setter Property="Background" Value="BlanchedAlmond" />
  </Style.Setters>
  <Style.Triggers>
    <Trigger Property="IsMouseOver" Value="True">
      <Setter Property="Background" Value="BurlyWood" />
    </Trigger>
    <EventTrigger RoutedEvent="MouseLeave">
      <BeginStoryboard>
        <Storyboard>
          <DoubleAnimation To="5" AutoReverse="True"
                           Storyboard.TargetProperty="FontSize"
                           Duration="0:0:0.3" />
        </Storyboard>
      </BeginStoryboard>
    </EventTrigger>
  </Style.Triggers>
</Style>
```