

# WPF Validacija

Katedra za informatiku  
Fakultet tehničkih nauka  
Univerzitet u Novom Sadu

# Validacija

- Provera ispravnosti podataka pri unosu
- Ukoliko unesena vrednost nije u skladu sa određenim kriterijumom, korisnik u grafičkom interfejsu dobija obaveštenje da je vrednost neispravna

E-mail

markogmail.com

**Neispravan format e-mail adrese**

- WPF omogućuje različite tehnike validacije:
  - Validacija u slučaju izuzetka
  - Validacija putem validacionih pravila
  - Validacija korišćenjem `IDataErrorInfo` interfejsa
- Sve tehnike validacije se koriste u sklopu Data Binding mehanizma
- Trenutak kada se validacija vrši: `Binding.UpdateSourceTrigger`

# Validacija u slučaju izuzetka

- Najjednostavniji oblik validacije
- Ako se pri prebacivanju podatka iz izvornog objekta u ciljni pojavi izuzetak, korisnik dobija obaveštenje o neispravnoj vrednosti
- Primer
  - unos godine upisa studenta (u klasi **Student** godina upisa je **int**)
  - ako se unese tekst koji nije moguće konvertovati u broj, desiće se izuzetak, što ukazuje na nevalidnu vrednost

```
<TextBox Name="tbGodinaUpisa"  
    Text="{Binding Path=GodinaUpisa,  
        ValidatesOnExceptions=True}" />
```

# Validacija putem validacionih pravila

- Koristi se u slučaju kada neispravan podatak ne izaziva izuzetak, već je potrebno u skladu sa određenim kriterijumom utvrditi da li je podatak ispravan
- Primer
  - Unos E-mail adrese studenta
  - E-mail je u klasi **Student** tipa **string**, pa bilo koja unesena vrednost ne izaziva izuzetak
  - Vrednost je validna ukoliko je u skladu sa formatom e-mail adrese

# Validacija putem validacionih pravila

- Kreira se klasa naslednica klase **ValidationRule**
- U ovoj klasi se definiše programski kod koji utvrđuje ispravnost podatka
- Rezultat validacije je tipa **ValidationResult**
- **ValidationResult** sadrži svojstva
  - ❑ **ErrorContent** – sadrži detalje o grešci
  - ❑ **IsValid** – podatak da li je validirani podatak ispravan

---

# Validacija putem validacionih pravila

```
public class EmailValidationRule: ValidationRule
{
    public override ValidationResult Validate(
        object value,
        System.Globalization.CultureInfo cultureInfo)
    {
        //utvrdjivanje da li je objekat value validan
    }
}
```

---

# Validacija putem validacionih pravila

- Pri definisanju ulazne komponente definiše se klasa koja vrši validaciju

```
<TextBox Name="tbEmail">  
  <TextBox.Text>  
    <Binding Path="Email">  
      <Binding.ValidationRules>  
        <local:EmailValidationRule/>  
      </Binding.ValidationRules>  
    </Binding>  
  </TextBox.Text>  
</TextBox>
```

# Validacija putem IDataErrorInfo interfejsa

- Koristi se kada validnost podatka zavisi od vrednosti drugih podataka
- Primer
  - ❑ ako je za studenta unesen podatak o datumu odbrane diplomskog rada, tada je obavezno uneti i naslov teme rada
  - ❑ Ovo nije moguće realizovati putem validacionog pravila, jer pravilo samostalno validira samo jedan objekat



# Validacija putem IDataErrorInfo interfejsa

- Klasa implementira ovaj interfejs i definiše programski kod za validaciju svojih atributa

```
class Student: IDataErrorInfo
```

- Svojstvo **Error** definiše grešku na nivou celog objekta (npr. za prikaz u tabeli kao indikaciju da je ceo red neispravan)

```
public string Error  
{  
    get  
    {  
        return "Neispravni podaci o  
            studentu";  
    }  
}
```

# Validacija putem IDataErrorInfo interfejsa

- Indekser za validaciju konkretnog atributa klase

```
public string this[string propertyName]
{
    get
    {
        switch (propertyName)
        {
            case "Email":
                bool valid = EMailValidacija();
                if (!valid)
                    return "Neispravan e-mail";
                ...
            ...
        }
    }
}
```

# Validacija putem IDataErrorInfo interfejsa

- Ulazna kontrola automatski poziva validaciju definisanu u klasi koja implementira **IDataErrorInfo** interfejs

```
<TextBox Name="tbEmail">  
    <Binding Path="Email"  
        ValidatesOnDataErrors="True">  
    </Binding>  
</TextBox>
```

# Prikaz greške pri validaciji

- Default – crveni okvir oko kontrole
- Prilagođeni prikaz
  - U posebnoj labeli poruka o grešci – direktan prikaz poruke o grešci

```
<TextBlock Text="{Binding ElementName=tbEmail,  
    Path=(Validation.Errors).CurrentItem.  
    ErrorContent}" />
```

# Prikaz greške pri validaciji

## ■ Prilagođeni prikaz

- U posebnoj labeli poruka o grešci – konverzija greške preko konvertera
- Konverter

```
class ValidatorConverter: IValueConverter
{
    public object Convert(object value, Type
        targetType, object parameter,
        System.Globalization.CultureInfo culture)
    {
        //konverzija objekta value
        //(tipa ReadOnlyCollection<ValidationError>)
        //u tekst za prikaz
    }
}
```

# Prikaz greške pri validaciji

## ■ Prilagođeni prikaz

- ❑ U posebnoj labeli poruka o grešci – konverzija greške preko konvertera
- ❑ Definisanje i poziv konvertera

```
<Window.Resources>
```

```
    <local:ValidatorConverter x:Key="conv" />
```

```
</Window.Resources>
```

```
...
```

```
<TextBlock Text="{Binding ElementName=tbEmail,  
    Path=(Validation.Errors),  
    Converter={StaticResource ResourceKey=conv}}"  
</>
```

# Prikaz greške pri validaciji

- Prilagođeni prikaz
  - Definisanjem šablona za prikaz greške pri validaciji

```
<ControlTemplate x:Key="TextBoxErrorTemplate">
  <DockPanel LastChildFill="True">
    <TextBlock DockPanel.Dock="Right"
      Foreground="Orange" FontSize="12pt">
      !!!
    </TextBlock>
    <Border BorderBrush="Green"
      BorderThickness="1">
      <AdornedElementPlaceholder />
    </Border>
  </DockPanel>
</ControlTemplate>
```

# Prikaz greške pri validaciji

## ■ Prilagođeni prikaz

- Definisanjem šablona za prikaz greške pri validaciji

```
<TextBox Validation.ErrorTemplate=
    "{StaticResource ResourceKey=
        TextBoxErrorTemplate}"
    Name="tbEmail" >
    <TextBox.Text>
        <Binding Path="Email">
            <Binding.ValidationRules>
                <local:EmailValidationRule/>
            </Binding.ValidationRules>
        </Binding>
    </TextBox.Text>
</TextBox>
```