

GUI u C# - Termin 1

Objektno orijentisantisane tehnologije

Sadržaj

1. GUI
2. WPF
3. Kreiranje projekta u VS
4. Primer - Hello world program
5. XAML
6. App xaml
7. Događaji
8. Primer - Primer događaja
9. WPF paneli
10. Kalkulator
11. Osnovni kontrolni elementi
12. Zadaci

GUI u C#

- GUI je skraćenica od Graphical User Interface, što znači grafičko korisničko okruženje
- GUI okruženje omogućava kreiranje aplikacija primenjivanjem širokog spektra GUI elemenata, poput oznaka, polja za tekst, dugmića itd
- Bez GUI okruženja bi kreiranje ovih elemenata bilo ručno (crtanjem) što bi oduzimalo mnogo više vremena

WPF

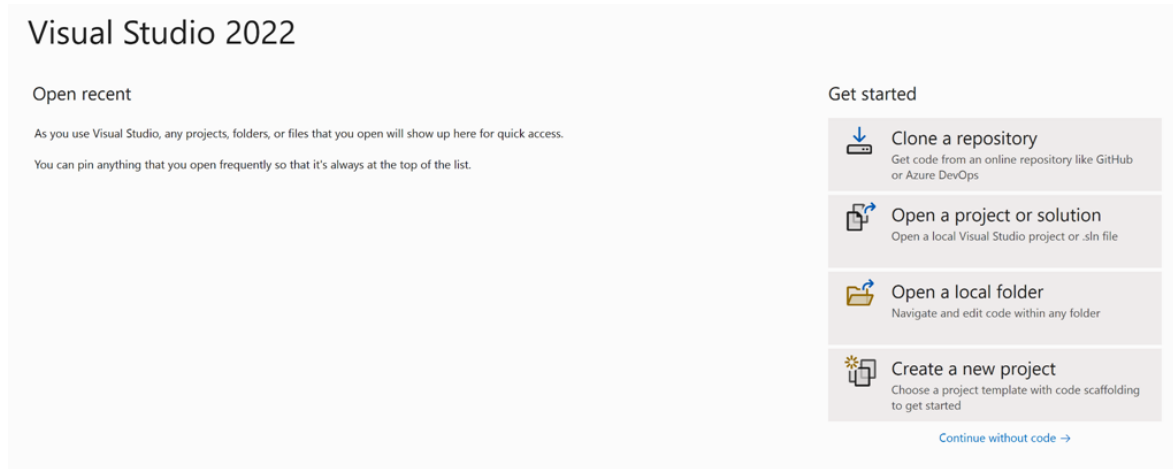
- WPF je skraćenica od Windows Presentation Foundation
- WPF je GUI okruženje koje se koristi u C# sa .NET radnim okruženjem
- Ima puno različitih GUI okruženja, ali za .NET programere trenutno su najzanimljiviji:
 - WinForms
 - WPF

WPF

- WPF je kombinacija XAML (markup jezika) i C#/VB.NET/bilo kog drugog .NET jezika
- Svaki od njih se može menjati putem bilo kog tekstualnog editora, čak i pomoću Notepad-a i zatim kompajlirati preko komandne linije
- Ipak, uglavnom se koristi i neki IDE (Integrated Development Environment - razvojno okruženje), jer mnogo olakšava sve procese od pisanja koda do dizajniranja interfejsa i kompajliranja

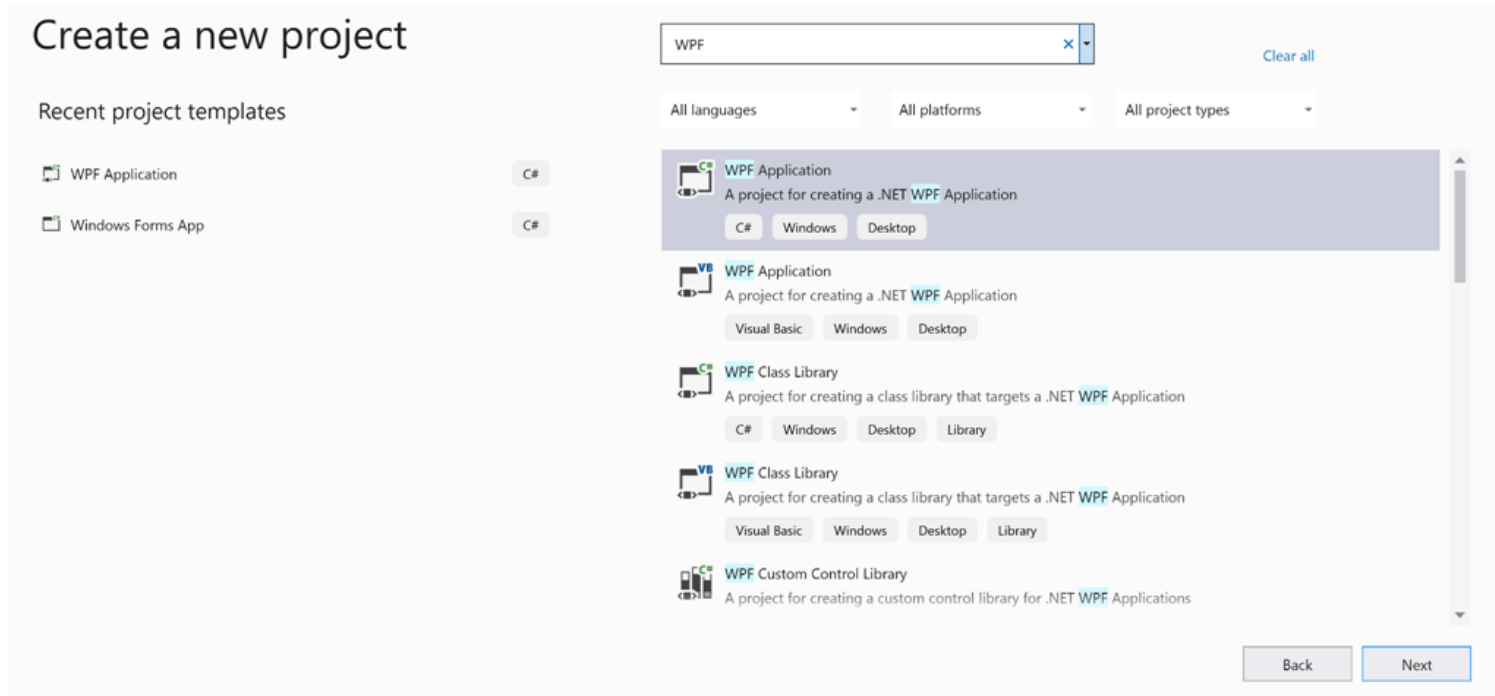
Kreiranje projekta u VS

Unutar Visual Studio okruženja, potrebno je odabrati **Create new project** (kao na slici 1). Nakon toga potrebno je odabrati **WPF Application** (kao na slici 2), dati naziv projektu i odabrati dodatne opcije (slike 3 i 4)



Slika 1

Kreiranje projekta u VS



Slika 2

Kreiranje projekta u VS

Configure your new project

WPF Application C# Windows Desktop

Project name

WpApp

Location

C:\Users\User\source\repos ...

Solution name ⓘ

WpApp

☐ Place solution and project in the same directory

Back Next

Slika 3

Additional information

WPF Application C# Windows Desktop

Framework ⓘ

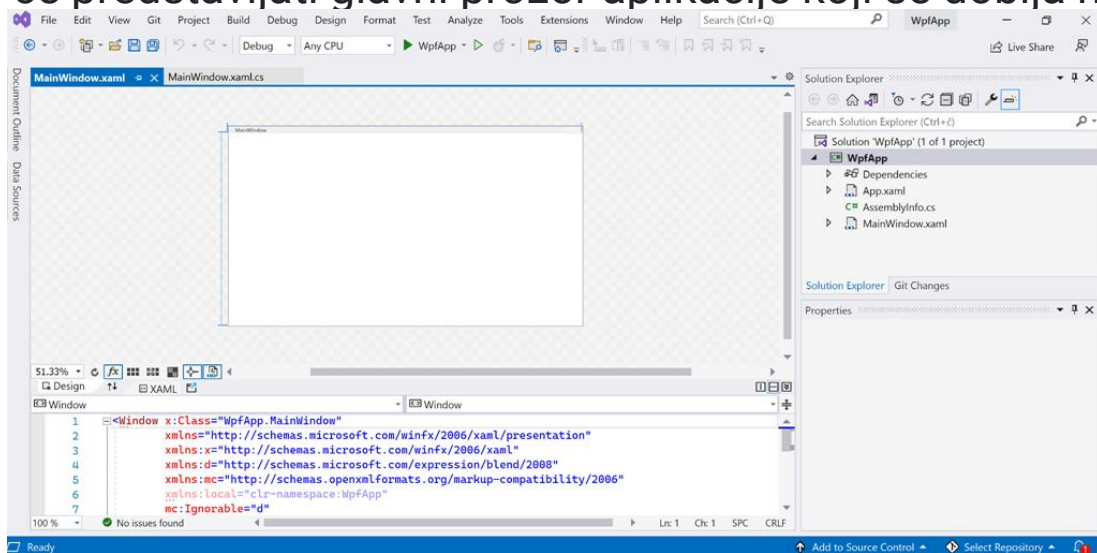
.NET 6.0 (Long-term support)

Back Create

Slika 4

Kreiranje projekta u VS

Nakon kreiranja projekta dobija se sledeći prikaz koji sadrži *MainWindow.xaml* fajl koji će predstavljati glavni prozor aplikacije koji se dobija nakon pokretanja programa

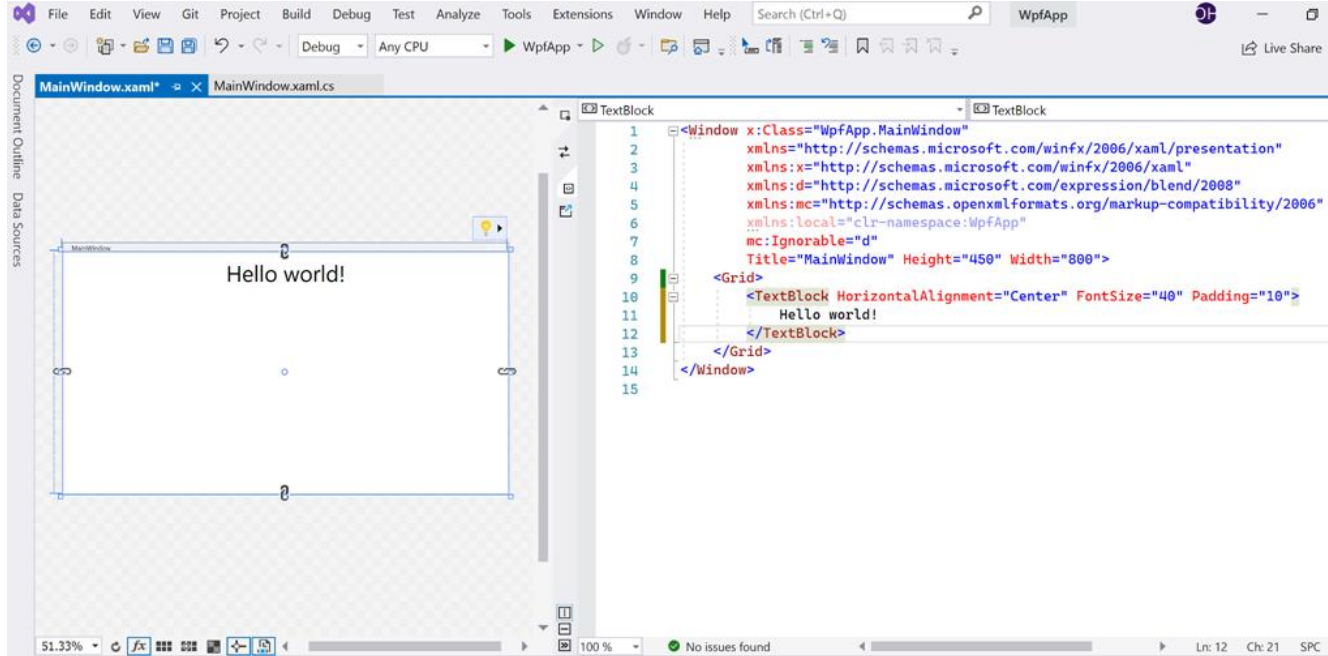


Slika 5

Hello world program

- Ukoliko kada pokrenemo program (CTRL+F5) želimo da dodamo i poruku na ekranu u XAML fajl ćemo dodati TextBlock kontrolu unutar Grid panela, sa sadržajem poruke
- U primeru smo, unutar TextBlock kontrolnog elementa koristili tri različita atributa kako bismo dobili željeno centriranje (centar prozora), željenu veličinu fonta putem FontSize atributa i odvajanje od gornje granice prozora
- Pored TextBlock-a koji smo mi dodali, Main prozor automatski sadrži Window i Grid kontrolni element

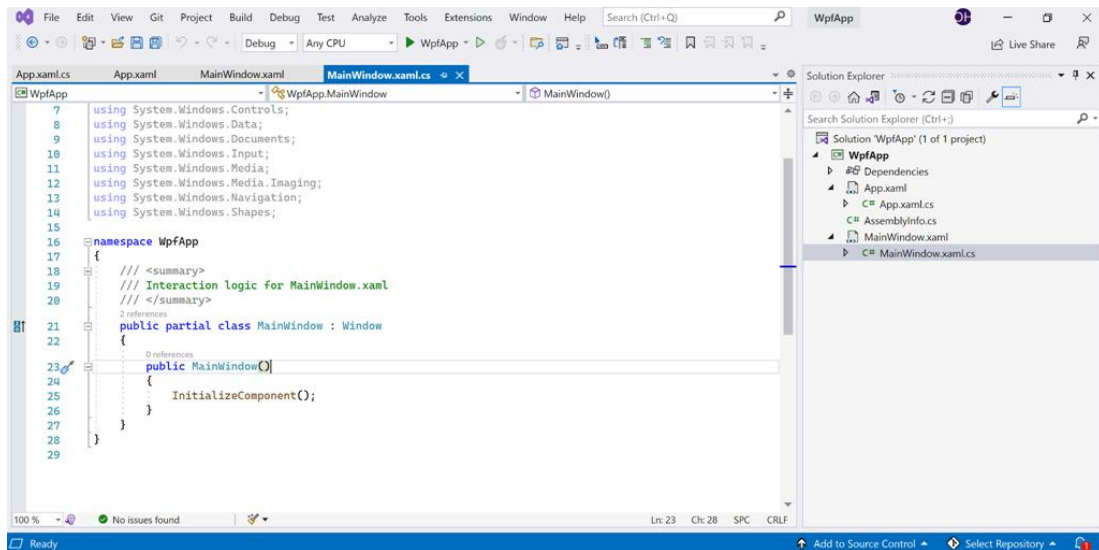
Hello world program



Slika 6

Hello world program

Kod koji se nalazi iza *MainWindow.xaml* fajla; u x:Class delu se nalazi naziv klase



Slika 7

XAML

- XAML, što je skraćenica od eXtensible Application Markup Language, je Mikrosoftova varijanta XML-a koja služi za opis GUI-a
- Svaki Window/Page se sastoji od XAML dokumenta i fajla sa kodom koji je povezan sa XAML-om. XAML fajl opisuje korisnički interfejs sa svim njegovim elementima, dok se fajl sa kodom brine o svim događajima i ima pristup manipulaciji XAML kontrolama

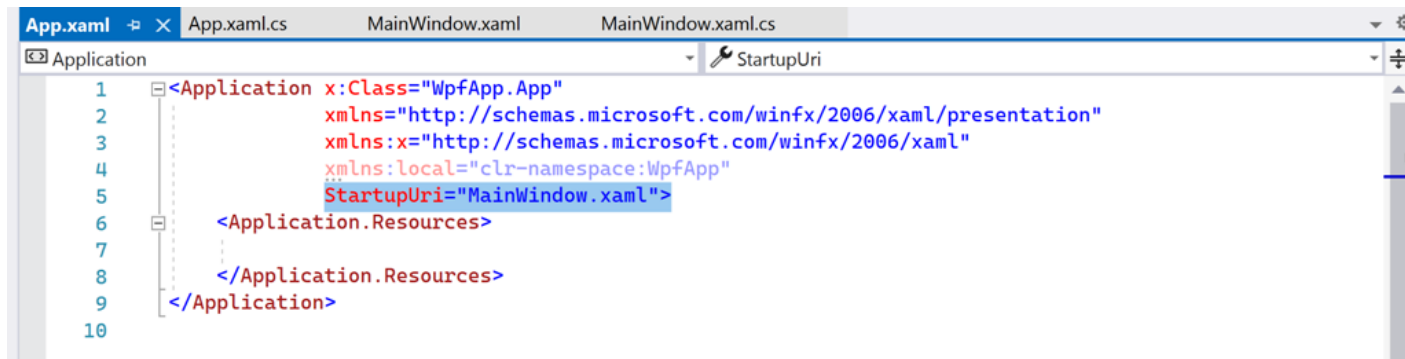
XAML

- XAML kontrolni elementi moraju biti zatvoreni, bilo pisanjem zatvorenog taga ili kose crte na kraju otvorenog taga
- Grafički kontrolni element dugme (Button) izleda ovako: `<Button></Button>`
- Većina kontrolnih elemenata dozvoljava umetanje sadržaja između otvorenog i zatvorenog taga, što čini sadržaj kontrolnog elementa. Na primer, grafički kontrolni element dugme nam dozvoljava da između otvorenog i zatvorenog taga napišemo tekst koji se prikazuje na njemu: `<Button>Klikni me</Button>`
- Kontrolni elementi imaju attribute koji predstavljaju njihova svojstva. Npr:
- `<Button FontWeight="Bold" Content="Klikni me" />`

App xaml

- *App.xaml* je automatski kreiran fajl prilikom kreiranja VS projekta i to je ulazna tačka svakog projekta. Kreira se zajedno sa fajlom koji sadrži kod (*App.xaml.cs*). U *App.xaml.cs* fajlu će biti kreirana parcijalna klasa kao i u *MainWindow*-u. Dva fajla rade zajedno
- Klasa u *App.xaml.cs* nasleđuje *Application* klasu
- Izvršavanje programa počinje od Aplikacije u kojoj je naznačeno u koji prozor ili stranicu se prvo ide. Ovde se takođe navode važni događaji u aplikaciji kao i neobrađeni izuzeci
- U *App.xaml* fajlu se definišu globalni resursi koji se koriste u čitavoj aplikaciji

App xaml



Slika 8

U StartupUri atributu (property) je navedeno od kog prozora ili stranice se počinje. U ovom slučaju, počinje se od *MainWindow.xaml*

Ovaj deo može biti urađen i direktno iz koda, a tada se koriste događaji i Startup properti

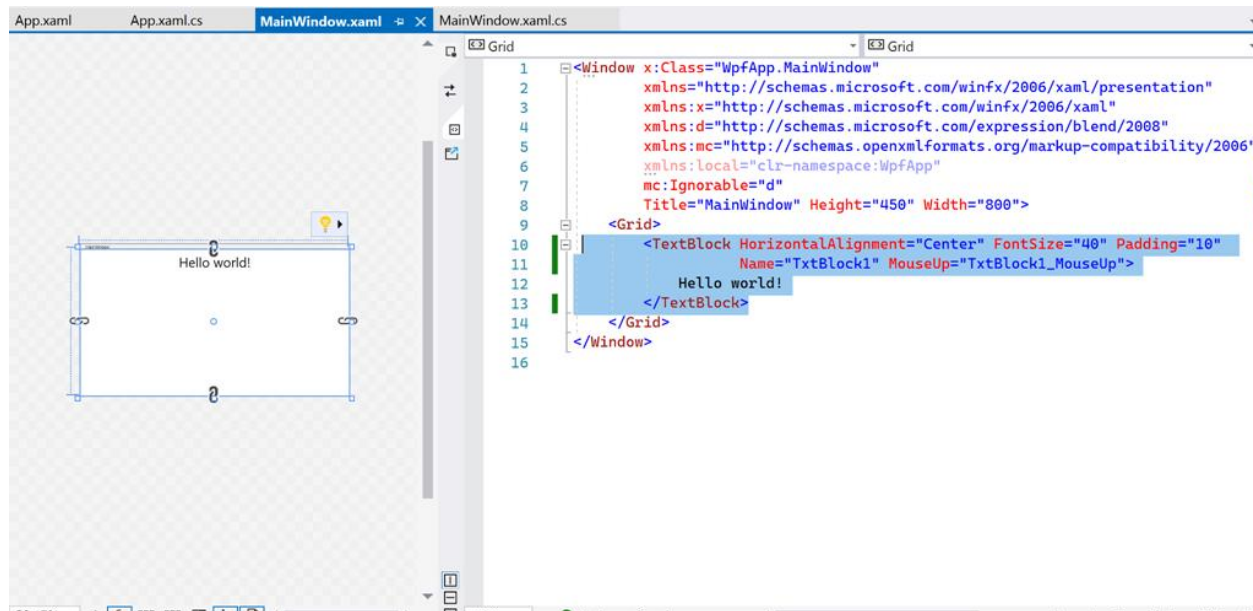
Događaji

- WPF je kontrolisan događajima. Sve kontrole, uključujući i Window imaju niz događaja koje možemo da slušamo. Ako se “pretplatimo” na te događaje, to znači da ih slušamo, što znači da će našoj aplikaciji biti javljeno kada se oni dogode i onda možemo da odgovorimo na njih
- Primeri događaja su oni koji odgovaraju na korisnikovu interakciju sa aplikacijom koristeći miša ili tastaturu. Npr "Dugme dole" (dugme je pritisnuto), "Dugme gore" (dugme je otpušteno), "Miš dole", "Miš ulazi" (kurzor ulazi na kontrolu), "Miš izlazi" (miš izlazi sa kontrole), "Miš gore" itd

Primer događaja

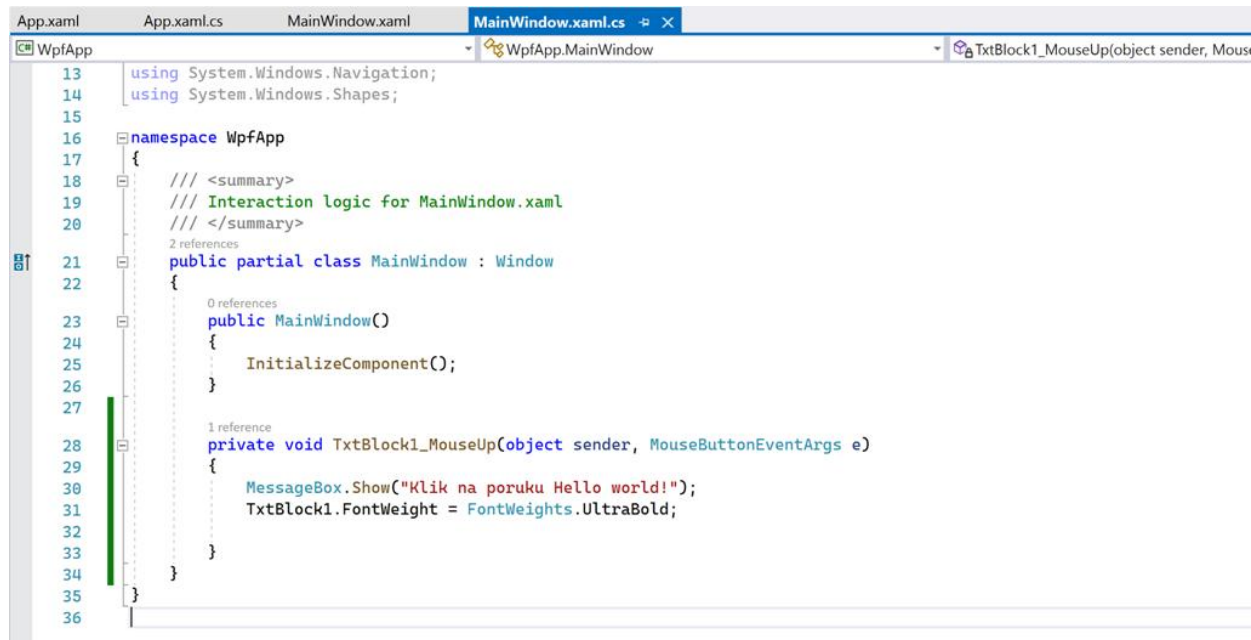
- Najpre je potrebno dati naziv kontrolnom elementu koji je uključen u događaj (*Name="TxtBlock1"*) - slika 9
- Potom je potrebno kreirati događaj (*MouseUp="TxtBlock1_MouseUp"*) - slika 9
- *TxtBlock1_MouseUp* će biti izgenerisan u kodu i tada je potrebno napisati deo koda koji sadrži reakciju na događaj koji se dogodio - slika 10
- Svaki kontrolni element sadrži attribute koji omogućavaju manipulaciju izgleda i ponašanja datog kontrolnog elementa

Primer događaja



Slika 9

Primer događaja



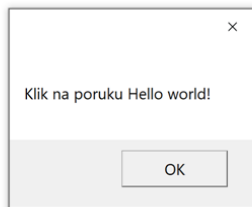
```
13 using System.Windows.Navigation;
14 using System.Windows.Shapes;
15
16 namespace WpfApp
17 {
18     /// <summary>
19     /// Interaction logic for MainWindow.xaml
20     /// </summary>
21     public partial class MainWindow : Window
22     {
23         public MainWindow()
24         {
25             InitializeComponent();
26         }
27
28         private void TxtBlock1_MouseUp(object sender, MouseButtonEventArgs e)
29         {
30             MessageBox.Show("Klik na poruku Hello world!");
31             TxtBlock1.FontWeight = FontWeights.UltraBold;
32         }
33     }
34 }
35
36
```

Slika 10

Primer događaja

MainWindow

Hello world!



Pogledajte šta se dešava kada se klikne OK.

Neki od atributa za TextBlock

- Background
- Cursor
- DataContext
- DesiredSize
- FontFamily
- FontSize
- FontWeight
- Name
- Padding

Neki od osnovnih kontrolnih elemenata

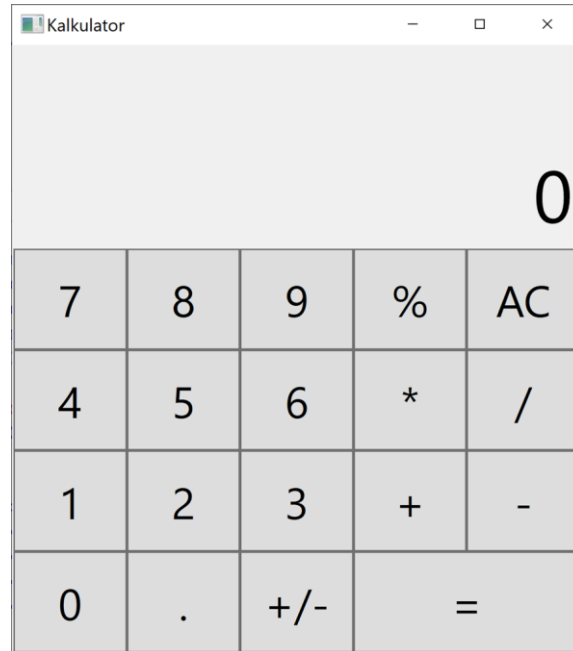
- Window
- Grid
- StackPanel
- Border
- ComboBox
- TextBlock
- Label
- TextBox
- Button

WPF paneli

- Paneli su veoma važni kontrolni elementi u WFP-u. Oni služe kao kontejneri za druge elemente i kontrolišu izgled prozora/stranica
- Neki od panela su: Canvas, WrapPanel, StackPanel, DockPanel, Grid i UniformGrid
- Najvažnije vrste panela:
 - StackPanel - Koristi se kada želimo da zauzmemo ceo prostor ređanjem kontrolnih elemenata jednog do drugog
 - Grid - Može da sadrži više kolona i redova i svaki od njih možemo da definišemo. Najkompleksnija je vrsta panela jer imamo manipulaciju sa svakim delom prozora/stranice pojedinačno. Navodimo koliko prostora zauzima svaki red/ kolona (u pikselima ili u procentualnom delu sadržaja)

Zadatak 1 - Kalkulator

Kreirati pojednostavljen kalkulator izgleda kao na slici.



Upotreba TextBlock-a

- TextBlock kontrolni element je jedan od najosnovnijih kontrolnih elemenata u WPF-u, ali je veoma često korišćen i veoma koristan
- On nam dopušta stavljanje teksta na ekran i radi dobro sa višelinijskim stringovima
- Ima properti Text u koji se piše sadržaj stringa koji se ispisuje

```
<TextBlock>  
    TextBlock with <Bold>bold</Bold>, <Italic>italic</Italic> and <Underline>underlined</Underline> text.  
</TextBlock>
```

```
<TextBlock Text="Ovo je textBlock"/>
```

Upotreba Label-a

- Label kontrolni element je sličan TextBlock-u, služi za ispis sadržaja na ekran
 - sadržaj može biti npr i slika, a ne samo string
- Ima properti Content u koji se piše sadržaj koji se prikazuje

```
<Label Content="Ovo je Label"/>
```

Upotreba TextBox-a

- TextBox kontrolni element je jedan od najčešće korišćenih elemenata
- Služi za unošenje odnosno pisanje teksta (jednolinijskog i multilinijskog)

```
<TextBox Text="Hello, world!" />
```

```
<TextBox AcceptsReturn="True" TextWrapping="Wrap" />
```

Ovo je tekst u vise
linija
...

Upotreba CheckBox-a i ComboBox-a

- CheckBox kontrolni element omogućava korisniku da označi određenu opciju što u kodu rezultira bool vrednošću

```
<CheckBox>Oznaci osobinu 1</CheckBox>  
<CheckBox IsChecked="True">Oznaci osobinu 2</CheckBox>  
<CheckBox>Oznaci osobinu 3</CheckBox>
```

<input type="checkbox"/>	Oznaci osobinu 1
<input checked="" type="checkbox"/>	Oznaci osobinu 2
<input type="checkbox"/>	Oznaci osobinu 3

- ComboBox kontrolni element omogućava korisniku da označi jednu od ponuđenih opciju iz padajuće liste

```
<ComboBox>  
  <ComboBoxItem>Opcija 1</ComboBoxItem>  
  <ComboBoxItem>Opcija 2</ComboBoxItem>  
  <ComboBoxItem>Opcija 3</ComboBoxItem>  
</ComboBox>
```

Opcija 2	▼
Opcija 1	
Opcija 2	
Opcija 3	

Upotreba RadioButton-a

- RadioButton kontrolni element daje korisniku izbor nekoliko različitih opcija od kojih samo jedna može da se odabere

```
<StackPanel Margin="10">  
    <Label FontWeight="Bold">WPF je zabavan?</Label>  
    <RadioButton>Da</RadioButton>  
    <RadioButton>Ne</RadioButton>  
    <RadioButton IsChecked="True">Onako</RadioButton>  
</StackPanel>
```

WPF je zabavan?

- ☐ Da
☐ Ne
☒ Onako

Upotreba PasswordBox-a

- PasswordBox je kontrolni element koji radi slično kao i TextBox, ali se koristi za lozinke
- Karakteri koji se unose su sakriveni prilikom unosa

```
<StackPanel Margin="0 10">  
    <Label>Korisnicko ime:</Label>  
    <TextBox Padding="2" />  
    <Label>Lozinka:</Label>  
    <PasswordBox Padding="2" />  
</StackPanel>
```

Korisnicko ime:

Lozinka:

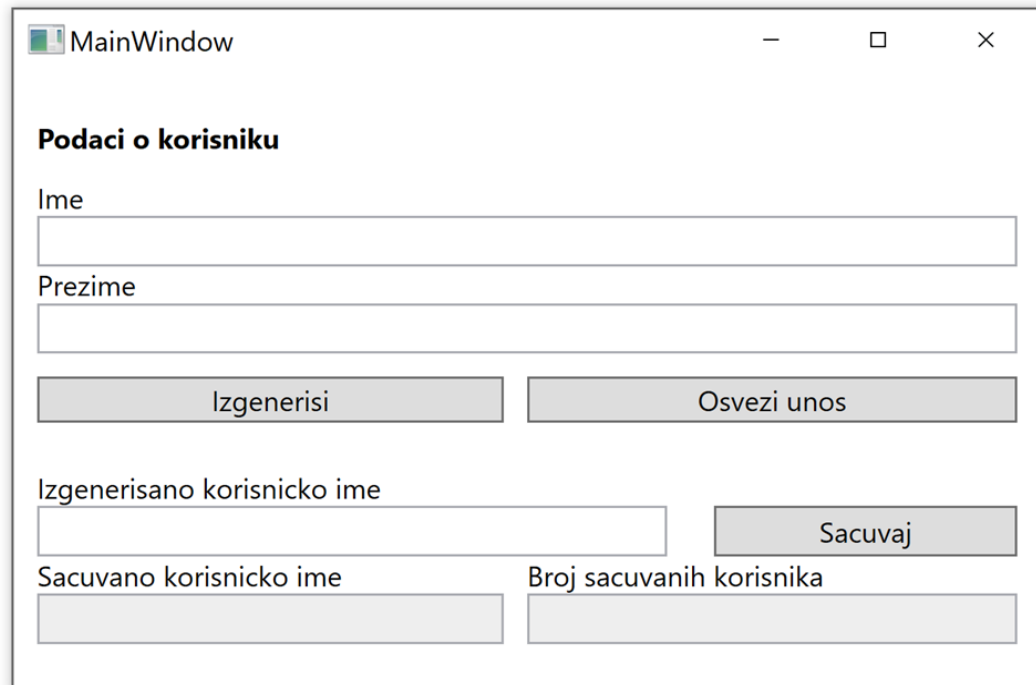
Upotreba MessageBox-a

- MessageBox je najjednostavniji način da komuniciramo sa korisnikom putem dijaloga
- MessageBox se koristi iz koda pozivom metode Show() koja ima različite parametre uz pomoć kojih kreiramo različite vrste dijaloga
 - MessageBox sa jednim parametrom koji je tekst koji treba ispisati, ispisuje datu poruku uz dugme OK
 - MessageBox sa dva parametra ima pored poruke i tekst naslova u dijalogu
 - MessageBox sa tri parametra ima pored poruke i naslova i raspored dugmića koje želimo da prikažemo
 - MessageBox sa četiri parametra ima dodatno i ikonicu koja prikazuje da li je dati dijalog informativnog tipa, upozorenje, pitanje, greška...

Zadatak 2 - Generisanje korisničkog imena

- Kreirati formu za unos imena i prezimena i generisanje korisničkog imena na osnovu unetih podataka. Korisničko ime se kreira tako što se uneti podaci za ime i prezime konkatenuiraju, a na to se kontaktenira slučajno generisan broj. Da bi se generisalo, mora da ima bar ime ili prezime. Ukoliko nije moguće generisati lozinku, ispisati poruku o tome u vidu MessageBox-a. Generisanje vrši dugme "Izgenerisi".
- Kada se izgeneriše, korisničko ime treba da se prikaže u TextBoxu namenjenom za to. Dugme "Osvezi unos" služi za osvežavanje (brisanje) tekstualnih polja za ime, prezime i generisano korisničko ime. Dugme "Sacuvaj" čuva informaciju o izgenerisanom korisničkom imenu i upisuje ga u polje namenjeno za to. Potrebno je brojati koliko ima sačuvanih korisnika. U slučaju da je nastala neka greška prikazati poruku o tome u vidu MessageBox-a. (Pogledati sliku u nastavku)

Zadatak 2 - Generisanje korisničkog imena



The screenshot shows a Windows application window titled "MainWindow". The window contains a form with the following elements:

- Podaci o korisniku** (User Data):
 - Ime** (Name): A text input field.
 - Prezime** (Surname): A text input field.
 - Izgenerisi** (Generate): A button.
 - Osvezi unos** (Refresh input): A button.
- Izgenerisano korisnicko ime** (Generated username): A text input field.
- Sacuvaj** (Save): A button.
- Sacuvano korisnicko ime** (Saved username): A text input field.
- Broj sacuvanih korisnika** (Number of saved users): A text input field.

Zadatak 3 - Generisanje korisničkog imena i lozinke

- Dodati funkcionalnost za unos lozinke za poslednjeg sačuvanog korisnika. Unos je moguć samo ako je korisničko ime korisnika uspešno sačuvano. U slučaju da lozinka sadrži korisničko ime, ona nije validna i treba ispisati poruku korisniku kao informaciju o tome. U slučaju da je lozinka ispravna, poruka korisniku treba da bude da je lozinka uspešno kreirana i polje za kreiranje lozinke treba da bude zaključano. Otključavanje polja i brisanje trenutnog sačuvanog korisnika i podataka o njemu uraditi samo kada se pritisne novokreirano dugme "Resetuj sve". Ispisivati poruke u vidu MessageBox prozora.