



WPF bevezetés

Répásné Babucs Hajnalka



Mi az a WPF?



- Windows Presentation Foundation rövidítése
- GUI keretrendszer, amely a .NET keretrendszerrel együtt használható
 - GUI: Graphical User Interface rövidítése, azaz grafikus felhasználói felület
- Grafikus alkalmazások fejlesztésének megkönnyítése
 - Label, TextBox, Image, Button, RadioButton, CheckBox, ListBox, ComboBox, dialógus ablakok, stb.
 - Felhasználói interakciók: egér események, billentyűzet események, stb.





WPF vs. WinForms



➤ WinForms:

- Windows vezérlőket használ
- Régebbi, így sokkal több minta
- A tervező felület használata könnyebb

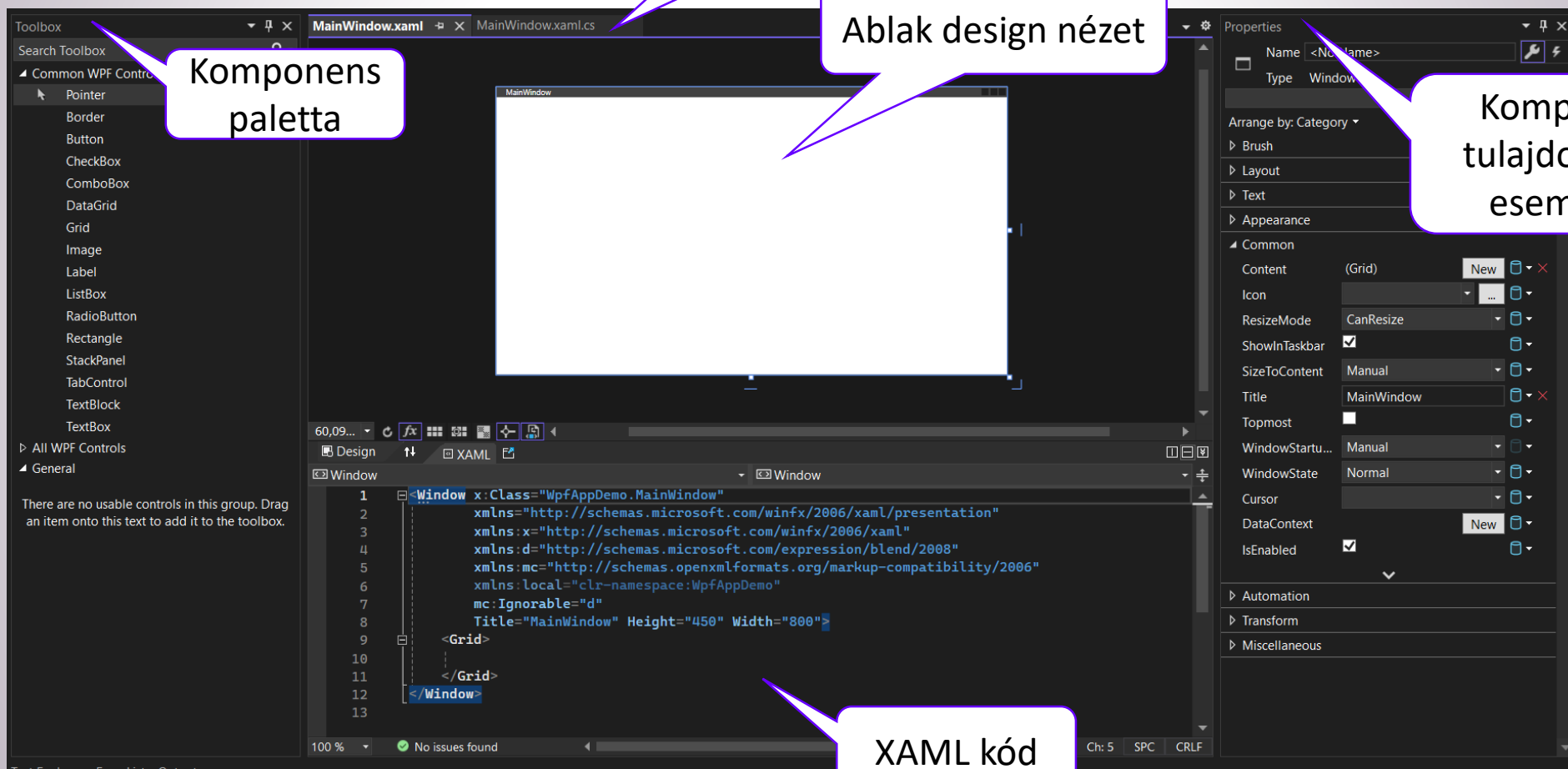
➤ WPF: Windows-tól független vezérlők

- Újabb, rugalmasabb, összhangban van a jelenlegi elvárásokkal
- XAML megkönnyíti a felhasználói felület létrehozását, szerkesztését, lehetővé teszi az UI tervezés (XAML) és az üzleti logika (C#) elkülönítését
- Adatösszekötést biztosít, az adatok tárolása és a megjelenítés szétválasztható
- Hardveres gyorsítás a GUI-ra rajzolás során, teljesítmény növelés
- Windows és web alkalmazásokban is használható a felhasználói felület





WPF alkalmazás





XAML és CodeBehind



- XAML: eXtensible Application Markup Language
 - XML-hez hasonló, a Microsoft által kidolgozott GUI leíró nyelv
- Akár ablakot (Window), akár oldalt (Page) hozol létre, az egy XAML dokumentumból és egy CodeBehind fájlból áll. Ezek együttesen alkotják a Window / Page kódját.
- XAML jellemzi a felületet minden elemével együtt
- A CodeBehind kezeli az eseményeket (event), és hozzáférése van a XAML controlokhoz





XAML felépítése



- Hasonlít a HTML tag-ekhez a komponensek megadása
- Kis- és nagybetű érzékeny!



The screenshot shows a Visual Studio editor window with XAML code. The code defines a `Window` element with various attributes and a `Grid` child element. Three callouts provide explanations:

- Az ablak beállításához szükséges kódok** (The code needed for window settings) points to the opening `<Window` tag.
- Tulajdonságok megadása** (Setting properties) points to the attribute list within the `<Window>` tag, specifically `Height="450"` and `Width="800"`.
- Konténer típusú elem, amibe a komponensek kerülhetnek (alapesetben Grid)** (Container type element where components can be placed (by default Grid)) points to the `<Grid>` element.

```
<Window x:Class="WpfAppDemo.MainWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:d="http://schemas.microsoft.com/winfx/2006/xaml/data"
        xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
        mc:Ignorable="d"
        Title="MainWindow" Height="450" Width="800">
    <Grid>
    </Grid>
</Window>
```



XAML alapok



➤ Vezérlőelemek létrehozása: `<>` karakterek között

➤ Pl.: gomb létrehozása:

➤ `<Button>` `</Button>`

➤ `<Button />`

➤ A tartalmát megadhatjuk a 2 tag között, vagy a Content tulajdonság beállításával:

➤ `<Button>Ez egy gomb</Button>`

➤ `<Button Content="Ez egy gomb" />`

➤ A Content tulajdonság csak egyetlen elemet foglalhat magába!

➤ Ha több elemet kell megadni, akkor konténer elembe tesszük





XAML alapok, tulajdonság tag-ek



- Az egyes tulajdonságokat is megadhatjuk tag-ekkel (Control.Property), vagy belső tulajdonság értékkel

<Button>

<Button.FontWeight>Bold</Button.FontWeight>

<Button.Content>Ez egy gomb</Button.Content>

</Button>



- A tulajdonságok C# kódból is állíthatók szükség esetén
 - Az alap tulajdonságokat XAML-ból érdemes állítani, gyorsabb, rövidebb
 - Kódból akkor állítunk tulajdonságot, ha egy esemény hatására a komponens valamelyik tulajdonsága megváltozik

