

Pokročilé nástroje a knihovny pro Windows (WPF, .NET)

Jan Pluskal

Credits: Michal Zachariáš, Roman Jašek, Martin Dybal

Vysoké učení technické v Brně, Fakulta informačních technologií

Božetěchova 1/2, 612 66 Brno - Královo Pole

pluskal@vut.cz

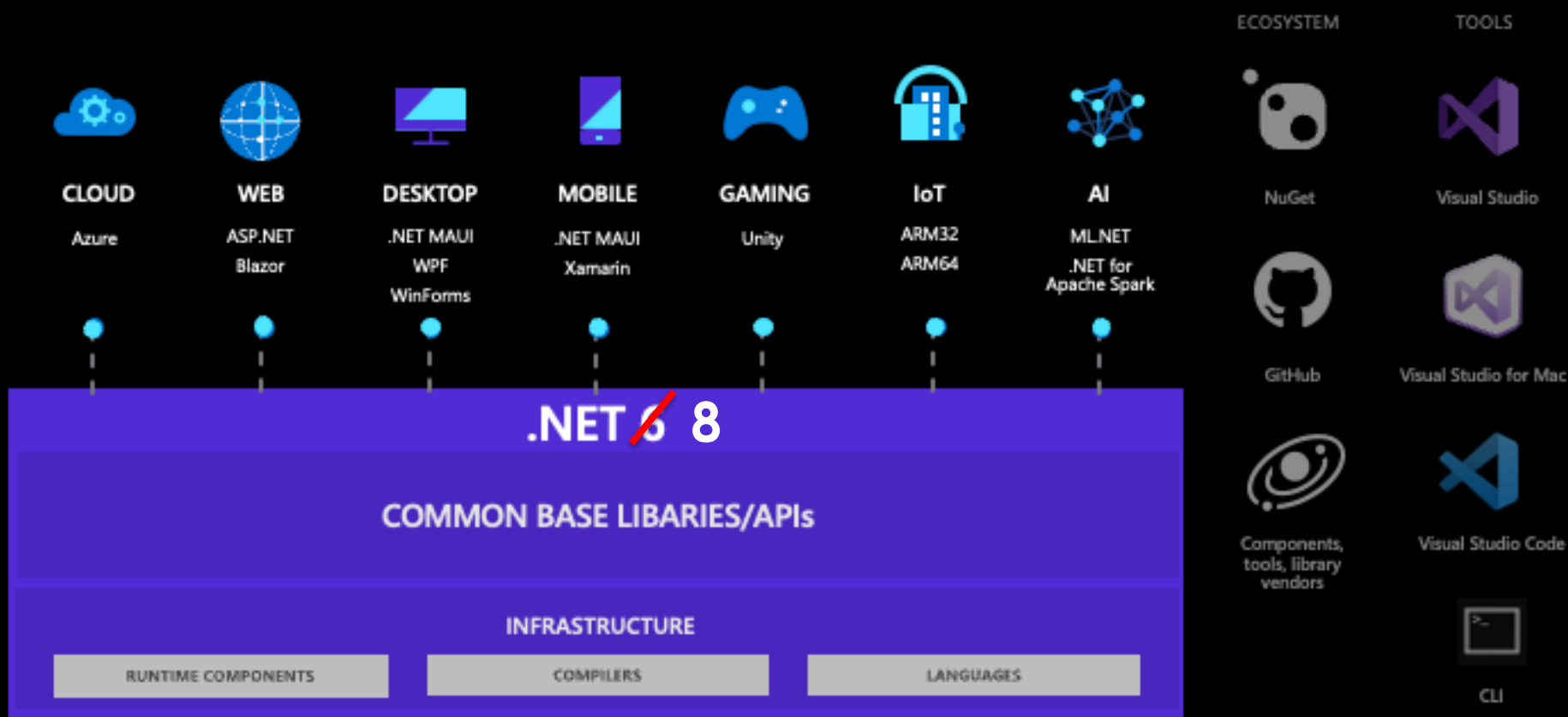


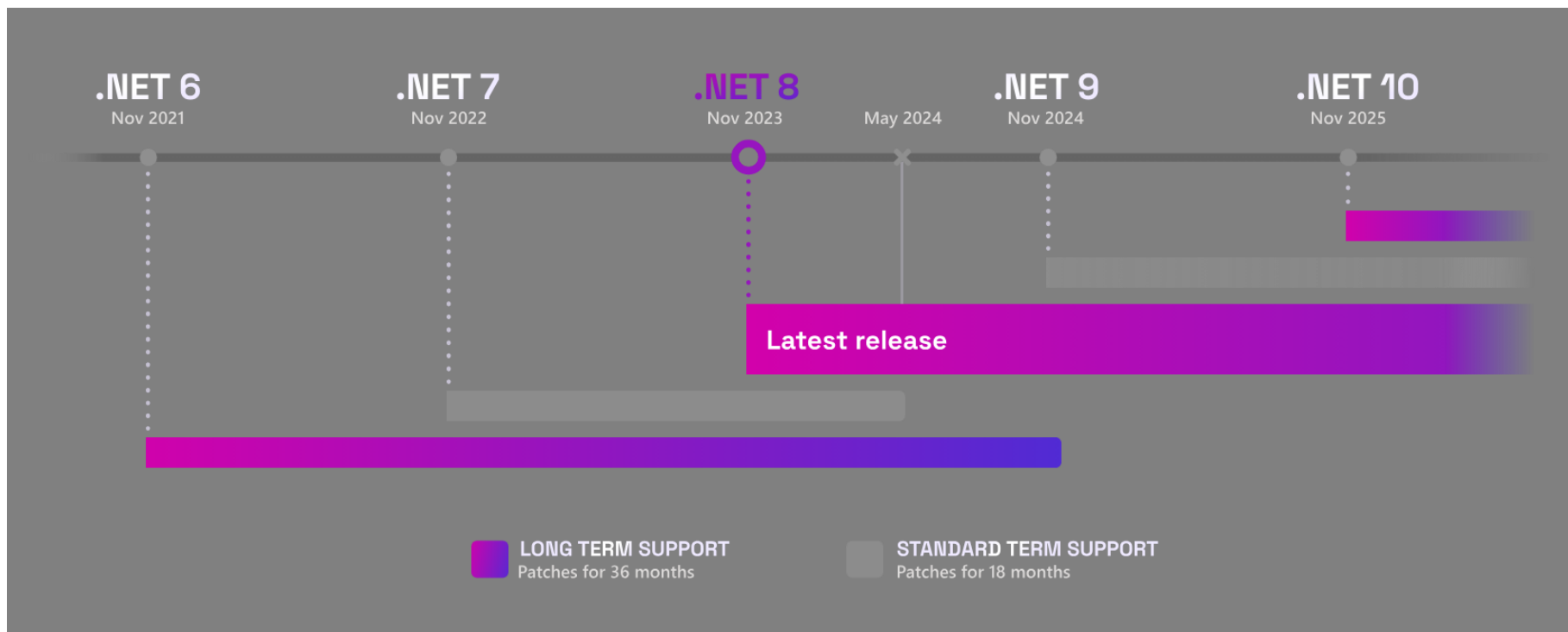
22.10.2024

- Co je to **OOP**?
- Co je to **Class Library**?
- Co je to **Framework**?
- Co je to **SDK**?
- Co je to **.NET**?
- Co je to **WPF**?



.NET – A unified development platform





Feature	.NET MAUI	Blazor Hybrid	React Native (RNW)	UWP XAML (Windows.UI.Xaml)	Win32 (MFC or ATL)	Windows Forms	WinUI 3	WPF
Language	C#	C#	JavaScript, TypeScript	C#, C++, Visual Basic	C++, Rust	C#, Visual Basic	C#, C++	C#, Visual Basic
UI language	XAML/Code	Razor	JSX	XAML	Code	Code	XAML	XAML
UI designer (drag & drop)	✗	✗	✗	✓	✗	✓	✗	✓
UI debugging	Hot Reload	Hot Reload	Fast Refresh	Hot Reload	-	Hot Reload	Hot Reload	Hot Reload
Fluent Design	✓	✓	✓	✓ (via WinUI 2)	✗	✗	✓	✗
.NET	.NET	.NET	N/A	.NET Core & .NET Native	N/A	.NET & .NET Framework	.NET	.NET & .NET Framework
Windows App SDK	✓ (more info)	✓ via MAUI	✓ (more info)	✗	✓	✓ (more info)	✓	✓ (more info)
Great for touch	✓	✓	✓	✓	✗	✗	✓	✗
Cross-platform	✓	✓	✓	✗	✗	✗	✗	✗
Xbox/HoloLens apps	✗	✗	✓	✓	✗	✗	✗	✗
Sandboxing (AppContainer)	✗	✗	✓	✓	✗	✗	✗	✗
Currently supported	✓	✓	✓	✓	✓	✓	✓	✓
Receiving updates	✓	✓	✓	✓ (security & bugfix)	✓	✓	✓	✓
Roadmap	GitHub	GitHub	GitHub	n/a	n/a	GitHub	GitHub	GitHub

- <https://learn.microsoft.com/en-us/windows/apps/get-started/?tabs=cpp-win32%2Cnet-maui>
- <https://visualstudiomagazine.com/Articles/2024/02/13/desktop-dev.aspx>

- **WinUI 3:** Nejnovější UI framework, který vám umožní vytvářet krásné, rychlé a plynulé aplikace pro Windows.
- **UWP:** UI framework, který vám umožní vytvářet univerzální aplikace pro Windows, které běží na různých typech zařízení s Windows.
- **WPF:** UI framework, který vám umožní vytvářet bohaté a výkonné desktopové aplikace pro Windows pomocí .NET a XAML.
- **Windows Forms:** UI framework, který vám umožní rychle vytvářet tradiční desktopové aplikace pro Windows pomocí .NET.
- **.NET MAUI:** UI framework, který vám umožní vytvářet multiplatformní desktopové a mobilní aplikace pomocí .NET a XAML.
- **Uno Platform:** Využijte své znalosti UWP a WinUI k vytváření aplikací pro jakoukoliv platformu.

- **Windows Presentation Foundation**
- pro tvorbu bohatého uživatelského rozhraní
- **vektorová** a rastrová grafika
- velmi silná podpora pro animace - **storyboardy**
- oddělení **vzhledu** od **chování**
- podpora **templatování** a **stylování**
- mocný **databinding**
- „nástupce“ WinForms
- akcelerováno/vykreslováno pomocí **Direct3D**

- Oddělení definice vzhledu od chování (a zdroje dat)
- **Markup** (vzhled)
 - Extensible Application Markup Language (**XAML**)
- **Code-Behind** (chování)
 - Nějaký .NET jazyk (**C#**, Visual Basic...)

- Co je to WPF?
- V jakých jazycích na jakých platformách jej můžeme použít?



- Vývoj a údržba **levnější a rychlejší**, protože vzhled a chování nejsou pevně provázány
 - Ke konci vývoje se GUI dost mění ← feedback od zákazníka
- **Designer a vývojář** mohou pracovat **odděleně**
- Specializované programy pro designera a vývojáře
 - Programátor – **Visual Studio 2019/2022**, Rider, VS Code ...
 - Designer – **Blend for Visual Studio** (formerly Microsoft Expression Blend)

- Deklarativní markup jazyk
- XML soubor s příponou .xaml

```
<Window
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  Title="Window with Button"
  Width="250" Height="100">

  <!-- Add button to window -->
  <Button Name="button">Click Me!</Button>

</Window>
```



- Svázáno s XAML, ale ne pevně
- Akce spojené s GUI
 - Reakce na události (key pressed, mouse moved...)
 - Bahaviors (EventToCommand, InvokeCommandAction)
- Ideálně prázdné – používat MVVM! (... ale nepředbíhejme)

```
<Window
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  Title="Window with Button"
  Width="250" Height="100">

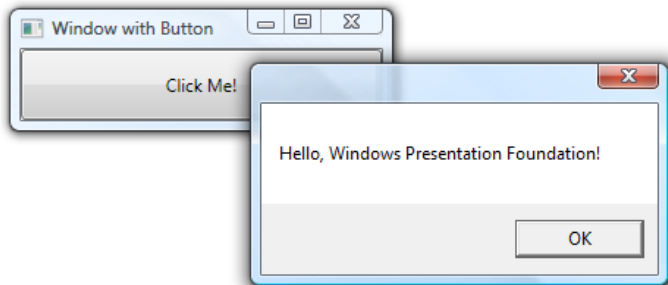
  <!-- Add button to window -->
  <Button Name="button">Click Me!</Button>

</Window>
```

```
using System.Windows; // Window, RoutedEventArgs, MessageBox
```

```
namespace SDKSample
{
    public partial class AWindow : Window
    {
        public AWindow()
        {
            // InitializeComponent call is required to merge the UI
            // that is defined in markup with this class, including
            // setting properties and registering event handlers
            InitializeComponent();
        }

        void button_Click(object sender, RoutedEventArgs e)
        {
            // Show message box when button is clicked
            MessageBox.Show("Hello, Windows Presentation Foundation!");
        }
    }
}
```



- XAML letem světem, aneb co budete potřebovat na cvikách
- Nenahrazuje referenční příručku a ani se o to nepokouší...



- Windows Presentation Foundation
<https://docs.microsoft.com/en-us/dotnet/desktop/wpf/>
- Introduction to WPF
<https://docs.microsoft.com/en-us/dotnet/desktop/wpf/getting-started/introduction-to-wpf-in-vs>
- XAML Overview
<https://learn.microsoft.com/en-us/dotnet/desktop/wpf/xaml/?view=netdesktop-8.0>
- WPF Tutorial
<https://wpf-tutorial.com/>

**PŘEČÍST nebo alespoň proletět!
MŮŽE SE OBJEVIT NA ZKOUŠCE!**

HANDS-ON VISUAL STUDIO


```
$ winget install Git.Git
```

```
$ winget install Microsoft.DotNet.SDK.8
```

```
$ winget install
```

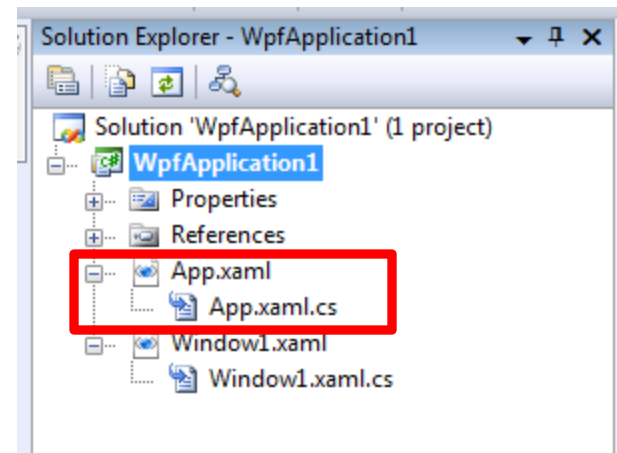
```
Microsoft.VisualStudio.2022.Community --  
override --add
```

```
Microsoft.VisualStudio.Workload.ManagedDesktop  
Microsoft.VisualStudio.ComponentGroup.WindowsA  
ppSDK.Cs"
```

```
$ winget install JetBrains.ReSharper
```

```
#https://www.jetbrains.com/community/education  
/#students
```

- Application-scoped
 - Vlastnosti (properties)
 - Služby (services)
 - Zdroje (resources)
 - Event Handlers
- Parametry příkazové řádky
- Life-time management (IHostApplicationLifetime)
- Splashscreen
- StartupUri
 - které okno se má při spuštění aplikace otevřít

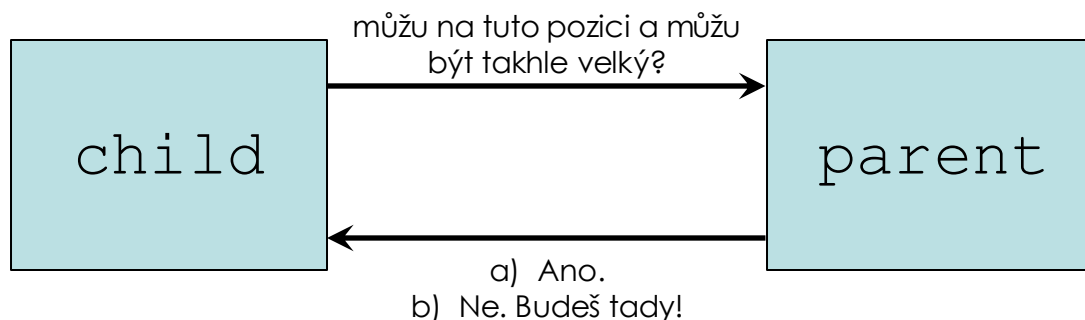


- **XAML**
 - rozložení
 - UI prvky
 - animace
 - ...
- **C#**
 - Logika GUI okna
 - Ideálně prázdné – používat MVVM
 - Injekce **DataContext** (IoC, ViewModelLocator)

- Určování pozice a velikosti prvkům UI
- Kladen velký důraz na **přenositelnost mezi různými displeji** (rozlišeními) a **různými velikostmi oken**
 - relativní pozicování
 - device independent pixel (1/96 place)

<https://learn.microsoft.com/en-us/windows/win32/learnwin32/dpi-and-device-independent-pixels>

- „vyjednávání“ mezi UI prvky, které je dvou stupňové



PANEL/LAYOUT

- **Grid:** Komponenty se pozicují zadáním souřadnice řádku (Row) a sloupce (Column)
- **StackPanel:** Komponenty jsou umístěny buď horizontálně nebo vertikálně.
- **Canvas:** Komponenty jsou pozicovány absolutně vzhledem k levému verchnímu rohu.
- **DockPanel:** Komponenty jsou zarovnány na hrany.
- **WrapPanel:** Komponenty jsou umísťovány buď horizontálně nebo vertikálně (zleva do prave či zvrchu dolů). V okamžiku kdy by došlo k přetečení jsou zalomeny vytváří další řádek či sloupec – Pexesso.

Layout with Absolute and Dynamic Positioning

[https://docs.microsoft.com/en-us/previous-versions/visualstudio/visual-studio-2010/bb514707\(v=vs.100\)](https://docs.microsoft.com/en-us/previous-versions/visualstudio/visual-studio-2010/bb514707(v=vs.100))

Grid

Kontrolky uvnitř jsou
pozicovány do řádků a
sloupců - **rows/columns**.

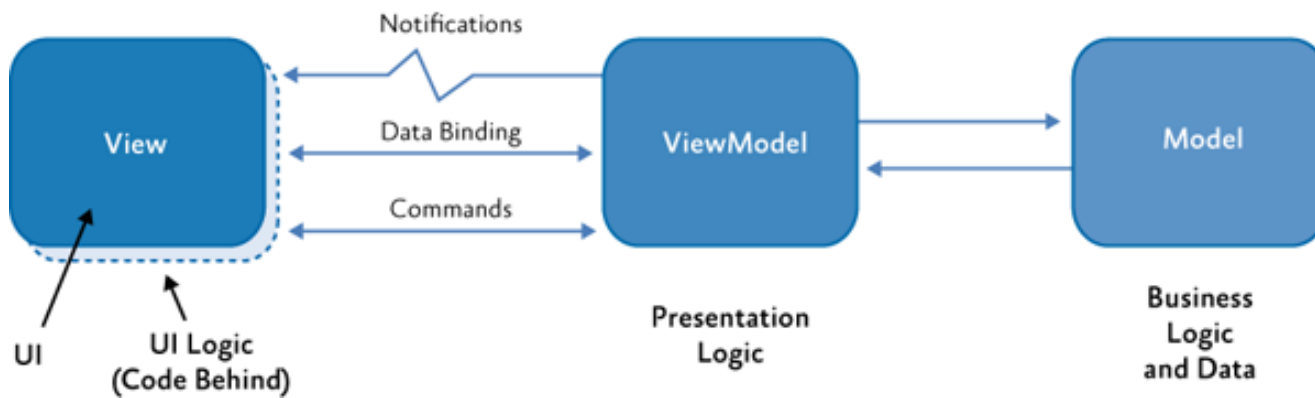
- **Absolutní číslo** – přesná velikost v dip
- **Hvězdička**
 - vyplní všechno zbylé místo
 - navíc je místo rozděleno v poměru počtu hvězdiček u každého řádku/sloupce
- **Auto** – řádek/sloupec se přizpůsobuje obsahu

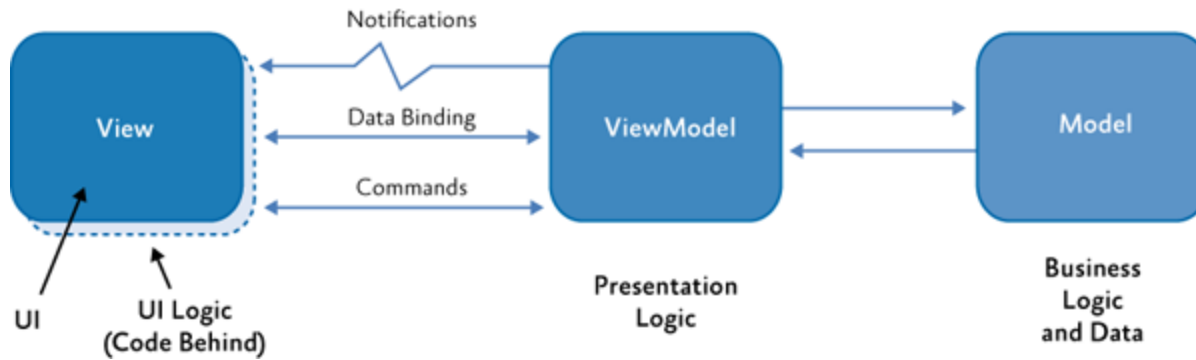
- `<Grid ShowGridLines="True">`
 - zobrazování vodících čar v Gridu
- **Live Visual Tree**
 - zobrazování struktury za běhu
 - od Visual Studio 2015
- **Properties**
 - Okno zobrazující všechny vlastnosti elementu, který je označen v XAML
- **Hot Reload**
 - Úprava XAML kódu za běhu

StackPanel

Kontrolky uvnitř jsou pozicovány buď pod sebe, nebo vedle sebe - **stacked vertically or horizontally**.

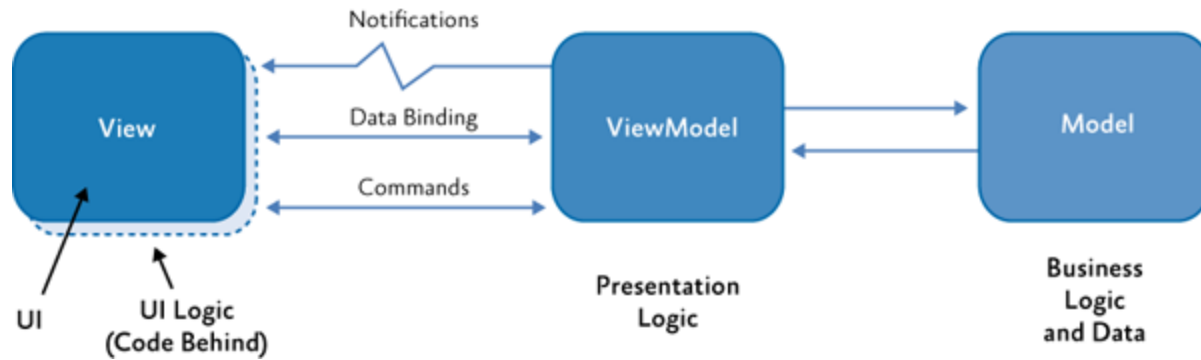
MODEL-VIEW-VIEWMODEL (MVVM)





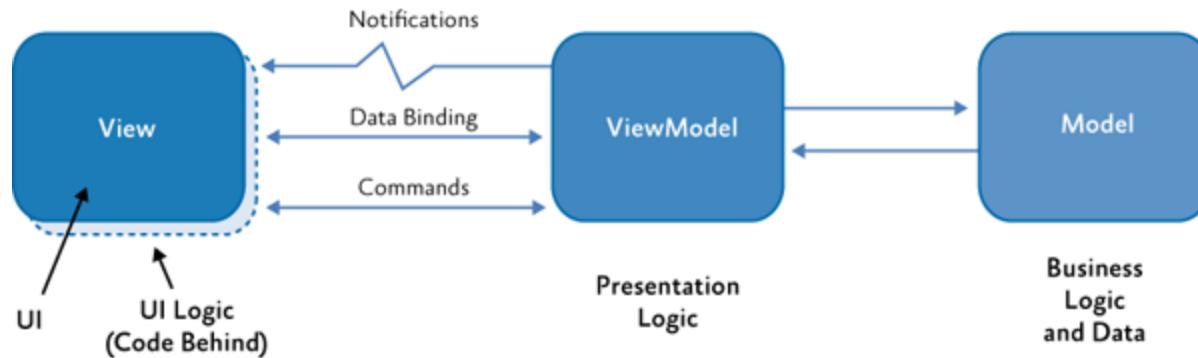
- **Model**

- „business logic“, „domain model“
- popisuje datové objekty
- SQL databáze, XML, JSON



- **View**

- GUI a GUI logika
- Buttons, labels, textboxes...

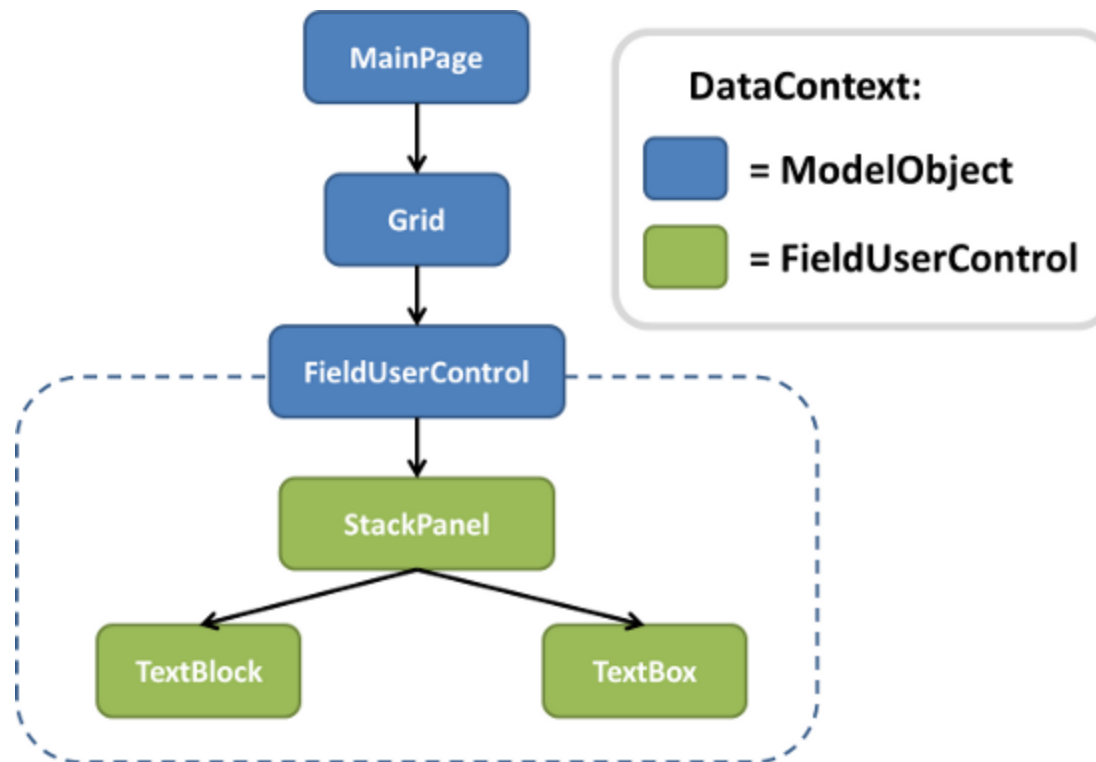


• ViewModel

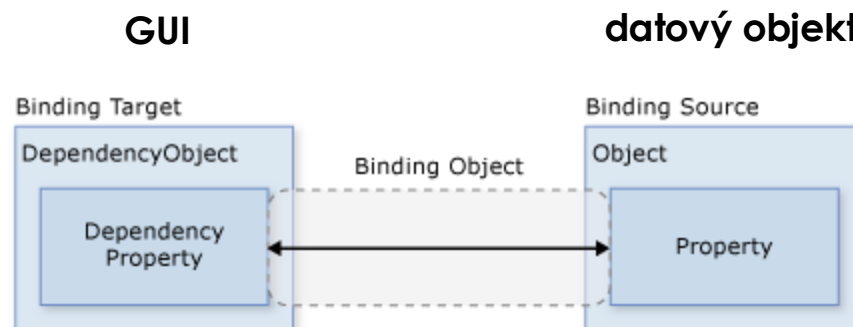
- „presentation logic“
- „data converter“
- zpracování dat z *Modelu* pro zobrazení ve *View*
- provádění akcí – commandy

- Co je to View, ViewModel a Model?

- Výchozí zdroj pro všechny data binding v daném prvku, ale také pro všechny potomky.



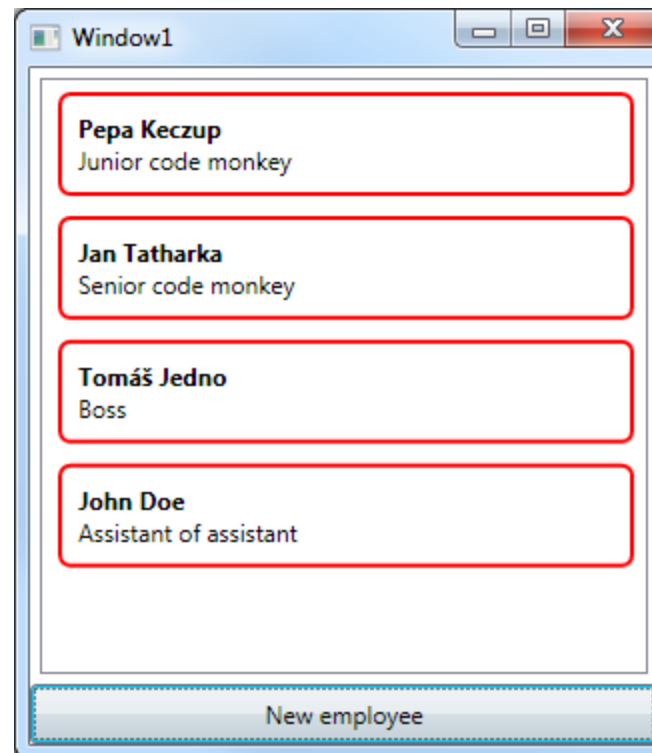
- Potřebujeme **zobrazit** a **editovat** uložená data
 - zobrazit a neřešit aktualizace ve zdroji – **OneTime**
 - zobrazit – **OneWay** binding
 - zobrazit a editovat – **TwoWay** binding
- Data jsou např. uložena v SQL databázi
- Po načtení dat musí aplikace:
 - Zkopírovat data do UI prvků
 - Zajistit, aby se provedené změny uložily zpět



- **INotifyPropertyChanged**
 - data oznamují, že se změnila
- **ObservableCollection**
 - implementuje *INotifyCollectionChanged*
 - dává vědět Binding Targetu, že se změnila data
- **Command** – odstranění logiky z code-behind
- **Tip**
 - Nuget balíček PropertyChanged.Fody – vyřeší implementaci notifikací za vás

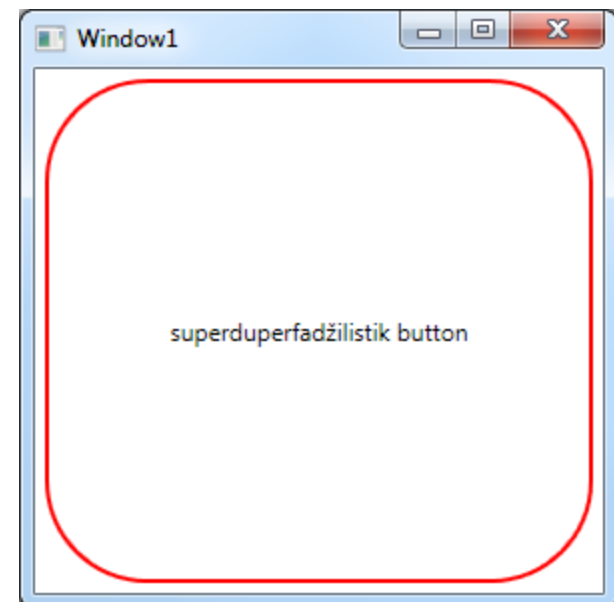
- Co je to data binding? Jaké rozhraní musí cíl bindingu implementovat, aby se data dynamicky překreslila ve View?

- Definice vzhledu **obsahu UI prvku**
- Např. záznamy v ListBoxu



- Definice vzhledu **celého UI prvku**

```
<Button Content="superduperfadžilistik button">
  <Button.Template>
    <ControlTemplate TargetType="{x:Type Button}">
      <Grid Margin="5">
        <Border BorderBrush="Red" BorderThickness="2" CornerRadius="50" Background="Transparent" />
        <ContentPresenter HorizontalAlignment="Center" VerticalAlignment="Center" />
      </Grid>
    </ControlTemplate>
  </Button.Template>
</Button>
```



- Ne všechny WPF komponenty jsou plně MVVM kompatibilní
- ***RelayCommands*** and ***InvokeCommandAction***
- Microsoft.Toolkit.Mvvm/Microsoft.Xaml.Behaviors.Wpf

<https://docs.microsoft.com/en-us/archive/msdn-magazine/2013/may/mvvm-commands-relaycommands-and-eventtocommand>

```
<ComboBox x:Name="CitySelectComboBox" SelectedValuePath="Content" IsEditable="True">
  <b:Interaction.Triggers>
    <b:EventTrigger EventName="Loaded">
      <b:InvokeCommandAction Command="{Binding DownloadWeatherCommand}"
        CommandParameter="{Binding ElementName=CitySelectComboBox, Path=SelectedValue}" />
    </b:EventTrigger>
    <b:EventTrigger EventName="SelectionChanged">
      <b:InvokeCommandAction Command="{Binding DownloadWeatherCommand}"
        CommandParameter="{Binding ElementName=CitySelectComboBox, Path=SelectedValue}" />
    </b:EventTrigger>
  </b:Interaction.Triggers>
</ComboBox>
```

```
public WeatherViewModel()
{
    DownloadWeatherCommand
    = new AsyncRelayCommand<string>(async city => await UpdateWeatherAsync(city));
}
```

```
private async Task UpdateWeatherAsync(string city)
{
    ...
}
```

- Je vhodné, aby jednotlivé ViewModely držely vzájemné reference? Pokud není, jaký návrhový vzor byste použili k vyřešení jejich vzájemné komunikace.

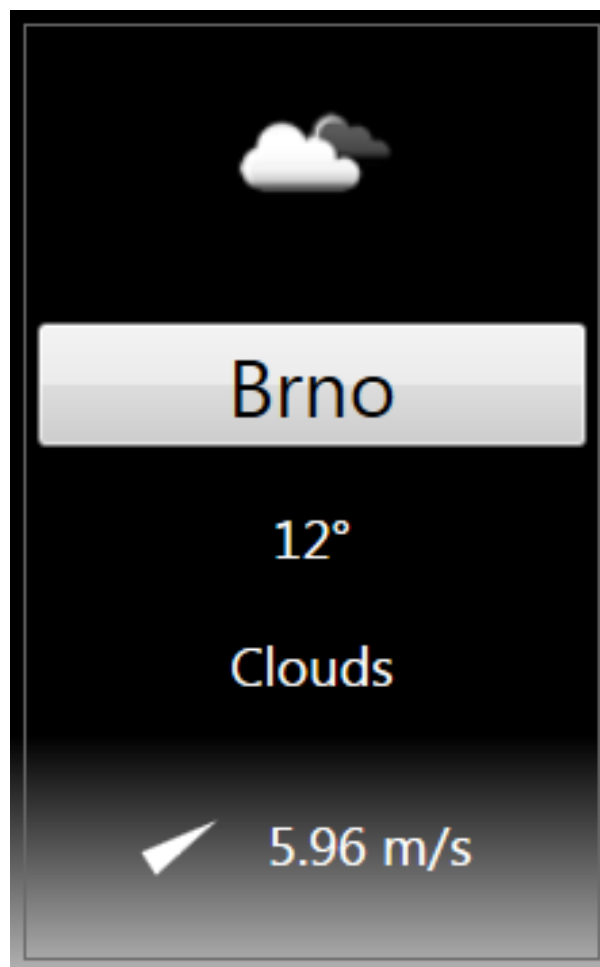
- K čemu slouží návrhový vzor Command? Vysvětlete jeho použití v MVVM.

- Animace
- Přechody mezi okny/obrazovkami

```
<Storyboard>
  <DoubleAnimation
    Storyboard.TargetName="MyRectangle"
    Storyboard.TargetProperty="Width"
    From="100" To="200" Duration="0:0:1" />

  <ColorAnimation
    Storyboard.TargetName="MySolidColorBrush"
    Storyboard.TargetProperty="Color"
    From="Blue" To="Red" Duration="0:0:1" />
</Storyboard>
```

- Animace – Storyboard
- Přehrávání videa, audia – MediaElement
- Stylování prvků/aplikace
- Efekty (stíny, rozmazání, možnost psaní vlastních...)
- Stylování textu
- ...



- XAML +.Net
 - Desktop
 - Universal Applications
 - Xamarin Forms – mobilní aplikace -> MAUI
 - MAUI
 - WinUI 3
 - Avalonia
- .Net
 - Desktop
 - Mobil
 - Web

- **.NET (Framework)**
 - base class library (BCL)
 - nezávislé na jazyku (pokud je to .NET jazyk)
- **Windows Presentation Foundation (WPF)**
 - bohaté uživatelské rozhraní
 - oddělení designu od funkční části
 - relativní pozicování, resolution-independent pixels, control templates, data templates, data binding
- **MVVM**
 - architektura

- MVVM tutoriál
 - <http://www.dotnetportal.cz/clanek/4994/MVVM-Model-View-ViewModel>
- www.pluralsight.com (aktivovat přes Azure Dev Tools for Education – 3 měsíce zdarma)
- Předměty:
 - ICS – letní semestr
 - IW5 – zimní semestr
- WUG

