

# Module 2

## WPF – základní návrh UI

MGR. TOMÁŠ HAVETTA - MCT

# Module Overview

- ▶ Nástroje pro vývoj WPF aplikace
- ▶ Základní třídy WPF
- ▶ Logický a vizuální strom
- ▶ Umístění prvků v okně
- ▶ Panely
- ▶ Controls



# Blok 1: Nástroje pro vývoj WPF aplikace

- ▶ Visual Studio
- ▶ Blend
- ▶ XAML
- ▶ WPF vytvořené kódem

# Visual Studio

- ▶ Silná podpora zápisu XAML (Intellisense)
- ▶ Vizuální náhled pro tvorbu UI negeneruje ideální kód
- ▶ VS 2019 a .NET 5
  - ▶ Hot Reload
  - ▶ Upozornění na chyby Bindingu



# Blend

- ▶ Speciální nástroj pro grafickou tvorbu UI
- ▶ Mnohem silnější možnosti tvorby XAML
- ▶ Podpora pro design time data
- ▶ Podpora pro tvorbu Resources
- ▶ Ideální alespoň jeden grafik ve firmě, který zvládne Blend
- ▶ Blend sdílí s VS WPF projekt, dokáže udělat build a spustit aplikaci.
- ▶ Lze psát kód

# XAML

- ▶ XML s extenzemi pro zjednodušený zápis vazebních hodnot
- ▶ Slouží pro konstruktor okna jako návod pro vytváření instancí objektů a nastavení jejich vlastností
- ▶ XAML umožňuje oddělit práci programátora a designéra UI



# WPF vytvořené kódem

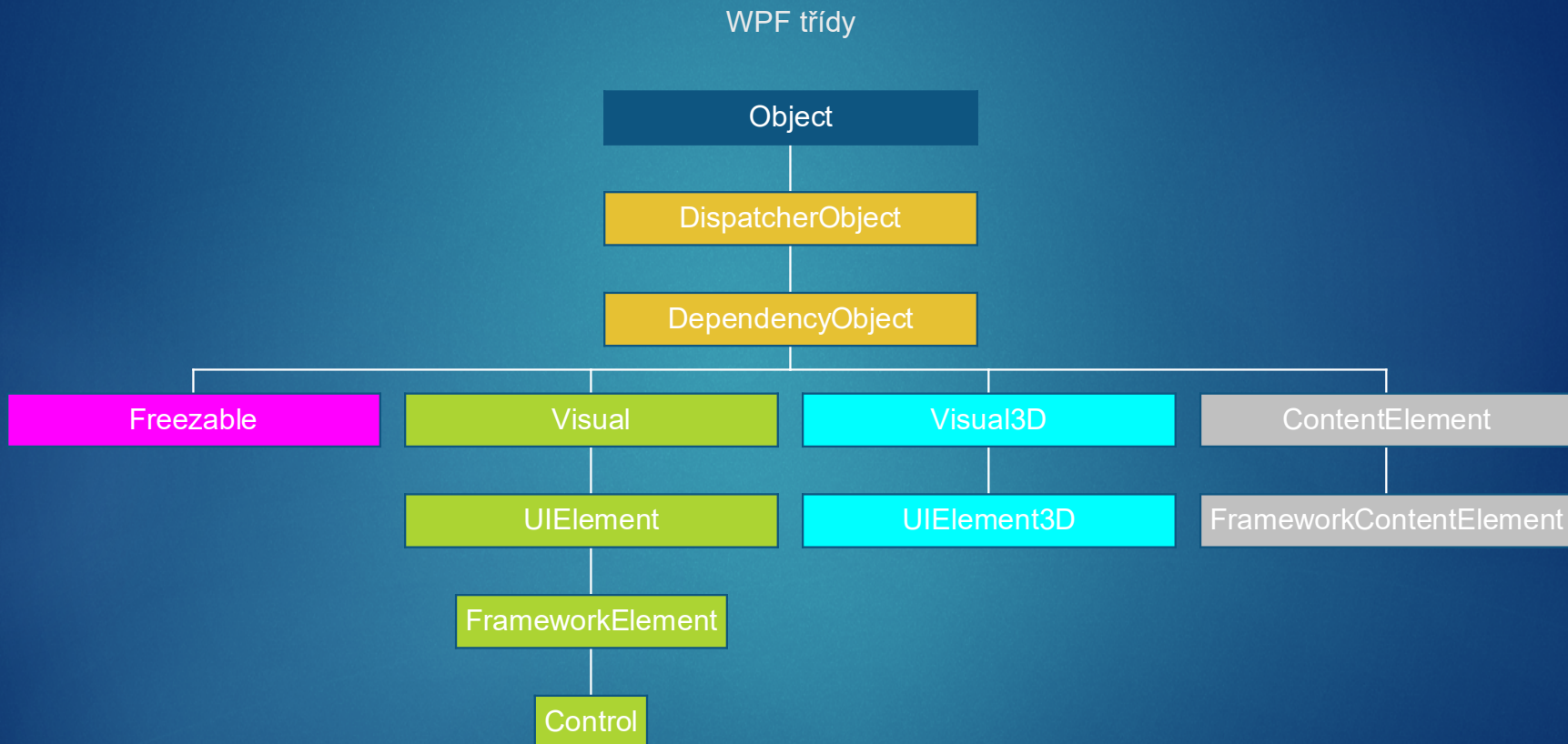
- ▶ WPF aplikace nemusí obsahovat XAML
- ▶ Všechny třídy lze vytvořit v kódu
- ▶ Všechny vazby lze nastavit z kódu
- ▶ Ztrácí se jednoduchá možnost měnit vzhled aplikace
- ▶ Ideální pro aplikace s generovaným UI na základě metadat nebo XML uloženého v DB

# Blok 2: Základní třídy WPF

- ▶ Základní typy WPF tříd
- ▶ WPF 2D třídy



# Základní typy WPF tříd



# WPF 2-D třídy

- ▶ Visual
  - ▶ Základní prvek schopný 2D zobrazení
- ▶ UIElement
  - ▶ Základní třída s podporou routed eventů, command binding, umístění prvku, focus
- ▶ FrameworkElement
  - ▶ Přidává podporu pro styly, data binding, resources, tooltip, content menu
- ▶ Control
  - ▶ Základ pro většinu 2D controls. Přidává některé vlastnosti jako Background, Foreground, FontSize, ...



# Blok 3: Logický a vizuální strom

- ▶ Logický strom
- ▶ Vizuální strom

# Logický strom

- ▶ XAML je z principu stromová struktura, kde na vrcholu je root element (většinou Window)
- ▶ Strom odpovídající XAML zápisu nazýváme Logický strom
- ▶ Možnost výpisu pomocí třídy

`System.Windows.LogicalTreeHelper`



# Vizuální strom

- ▶ Každý control je reprezentován prvky, které ho tvoří (rámeček, content viewer, ...)
- ▶ Strom obsahující všechny prvky pomocí kterých vzniká finální obraz okna nazýváme Vizuální strom
- ▶ Možnost výpisu pomocí třídy

`System.Windows.Media.VisualTreeHelper`

# Blok 4: Umístění prvků v okně

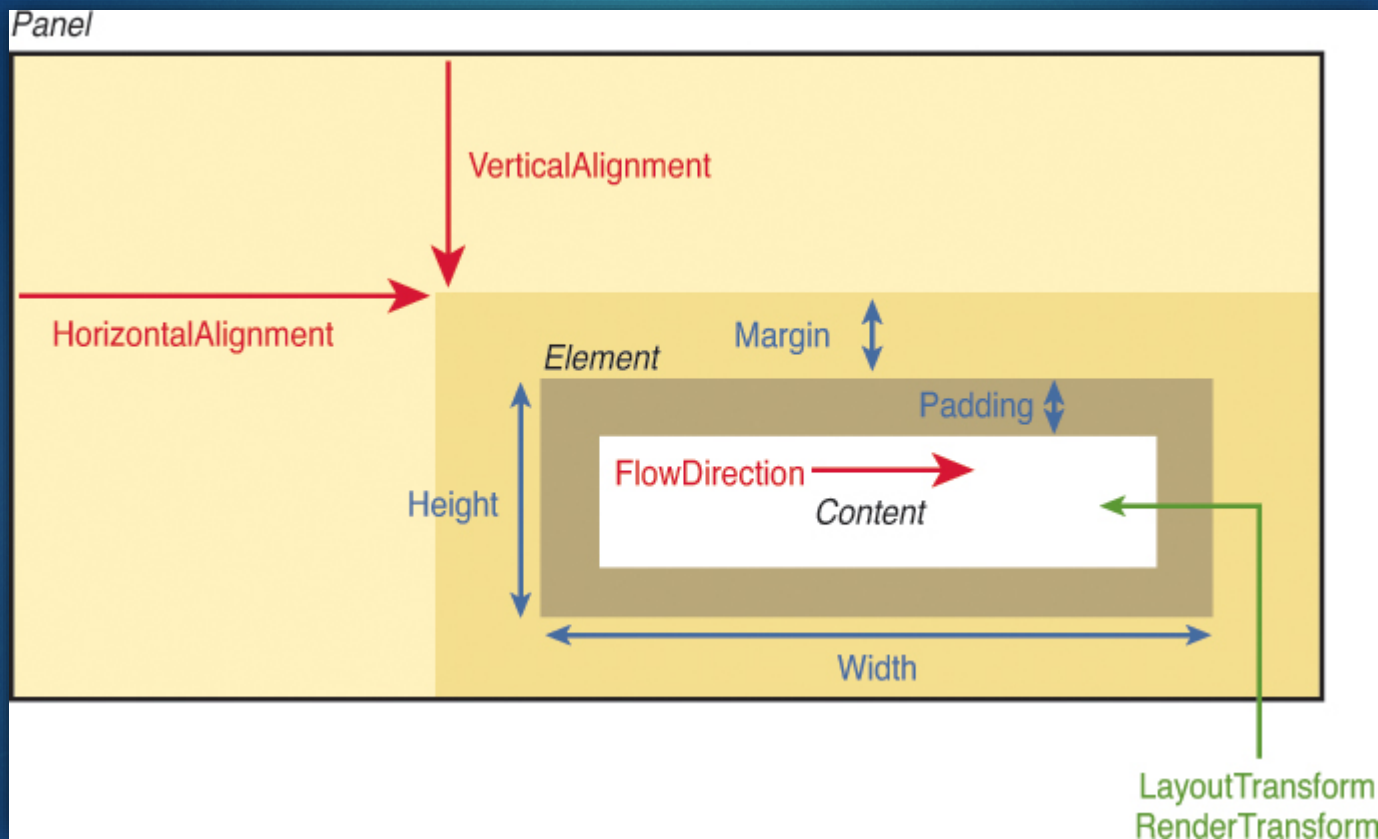
- ▶ Jednotky pro práci s UI
- ▶ Základy rozměry a umístění
- ▶ Transformace
- ▶ Viditelnost



# Jednotky pro práci s UI

- ▶ Device-independent pixel – základní jednotka
- ▶ Za číslo je možno přidat
  - ▶ px – default, není potřeba psát (1/96 inch)
  - ▶ cm
  - ▶ in
  - ▶ pt – starý point (1/72 inch)
- ▶ Pozor, pracujete s device-independent zobrazením, takže skutečná velikost bude záviset na použitém zařízení. Zadáváte vzájemné poměry prvků.

# Základy rozměry a umístění





# Transformace

- ▶ RotateTransform - rotace
- ▶ ScaleTransform – zmenšení, zvětšení podle os x,y
- ▶ SkewTransform – zkosení podle os x,y
- ▶ TranslateTransform – posun o x,y
- ▶ MatrixTransform
- ▶ LayoutTransform
- ▶ RenderTransform
- ▶ TransformGroup

# Viditelnost - Visibility

- ▶ Visible
  - ▶ viditelný, defaultní stav
- ▶ Collapsed
  - ▶ neviditelný a nezabírá místo
- ▶ Hidden
  - ▶ neviditelný, ale zabírá místo pro své zobrazení



# Blok 5: Panely

- ▶ StackPanel
- ▶ WrapPanel
- ▶ DockPanel
- ▶ Canvas
- ▶ Grid

# StackPanel

- ▶ Nejjednodušší, umísťuje prvky pod sebe nebo vedle sebe
- ▶ Vlastnost Orientation (default Vertical)
- ▶ VirtualizingStackPanel
  - ▶ Zajímavý při zobrazení velkého množství prvků pomocí data bindingu. Prvky mimo aktuální zobrazení dokáže uvolnit z RAM.



# WrapPanel

- ▶ Panel podporuje „zalamování“ obsahu podle šířky nebo výšky panelu (Explorer zobrazující soubory)
- ▶ Vlastnost Orientation určuje režim „zalamování“. Default je Horizontal.
- ▶ ItemWidth a ItemHeight můžou předepsat velikost pro prvky

# DockPanel

- ▶ Panel umožňuje „přilepit“ vnitřní elementy k stranám panelu.
- ▶ Elementy uvnitř DockPanelu definují vlastnost **DockPanel.Dock** pro určení umístění
- ▶ Poslední prvek bez DockPanel.Dock vyplní celý zbývající prostor
- ▶ Záleží na pořadí zápisu elementů!!!!



# Canvas

- ▶ Umožňuje přesné pozicování podle souřadnic
- ▶ Pozice se zadává pomocí vlastnosti
  - ▶ `Canvas.Top`
  - ▶ `Canvas.Left`
  - ▶ `Canvas.Right`
  - ▶ `Canvas.Bottom`
- ▶ Nelze zadat 3 vzdálenosti, ani dvě stejného rozměru (nezadáte např. `Top` a `Bottom`)
- ▶ Canvas má nejlepší performance, hodí se pro kreslení či inteligentní grafiku

# Grid

- ▶ Nemá nic společného s DataGridem!
- ▶ Rozděluje UI na řádky a sloupce pro lepší pozicování prvků
- ▶ Řádky a sloupce podporují zadání svého rozměru:
  - ▶ Auto – podle potřeby elementů v řádku/sloupci
  - ▶ 300 – přesné určení šířky
  - ▶ 2\* - proporcionální určení zbývajícího prostoru
- ▶ Elementy pomocí vlastností Grid.Row a Grid.Column určí své umístění



# Blok 6: Controls

- ▶ Základní Controls
- ▶ Content Controls
- ▶ Items Controls

# Základní Controls

- ▶ Label, TextBlock
- ▶ TextBox, RichTextBox
- ▶ ComboBox, ListBox
- ▶ GroupBox, RadioButton
- ▶ CheckBox, Button
- ▶ TabControl
- ▶ Image, MediaElement
- ▶ Slider, ProgressBar
- ▶ DataGrid
- ▶ ...



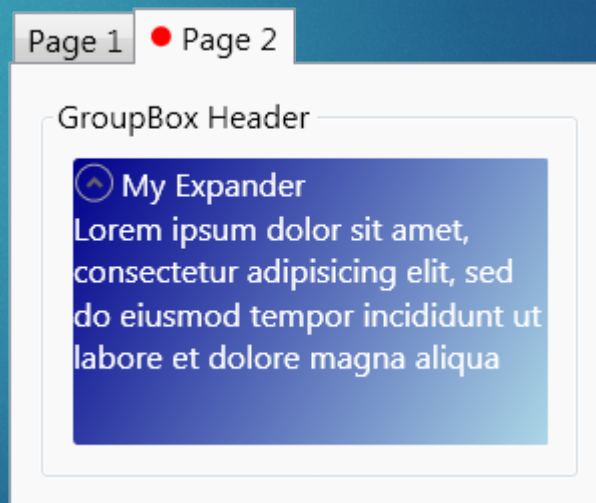
# Content Controls

- ▶ Element obsahující jednu hodnotu
- ▶ Vlastnost Content
- ▶ Bývá text, ale může to být jakýkoliv UIElement

```
<Button Margin="5" Click="RedAkce">  
    <Button.Content>  
        <StackPanel>  
            <Ellipse Fill="Red" Width="20" Height="20" />  
            <TextBlock>Pokus</TextBlock>  
        </StackPanel>  
    </Button.Content>  
</Button>
```

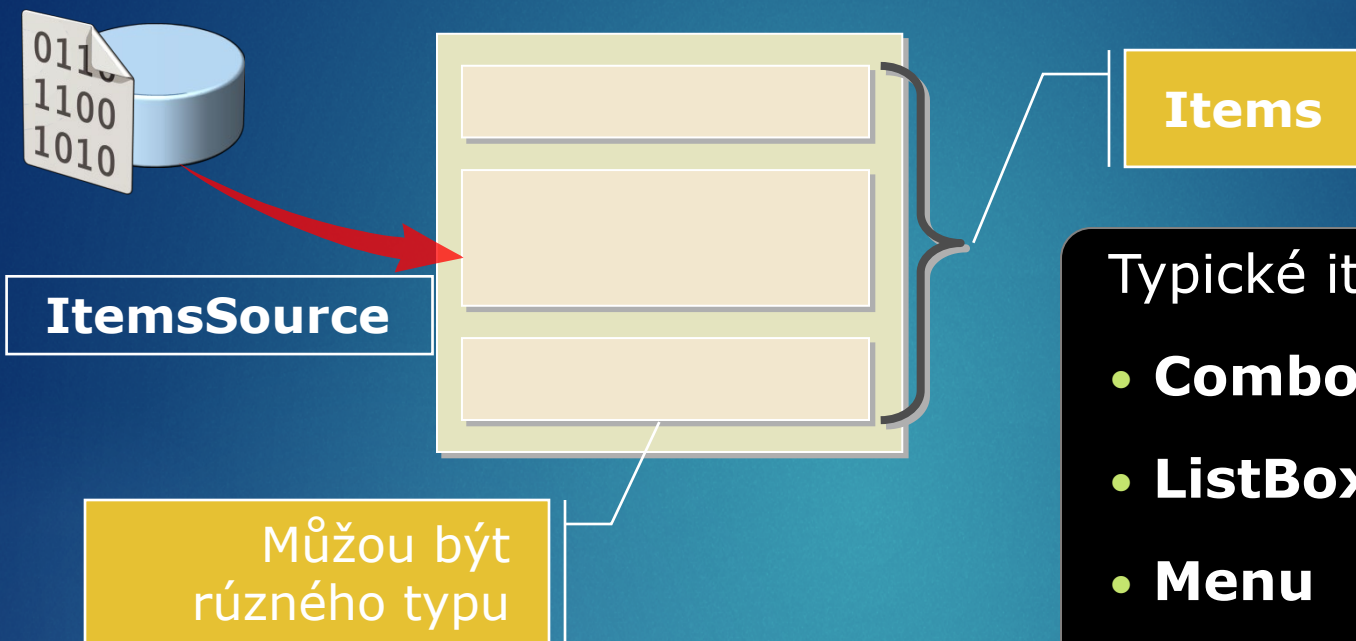
# Prvky s nadpisem

- ▶ GroupBox, TabItem, Expander
- ▶ Vlastnost Header
  - ▶ Dá se nastavit stejně jako Content





# Items Control



- Obsahuje několik objektů
- Má vlastnost **Items**
- Má vlastnost **ItemsSource**

Typické items controls:

- **ComboBox**
- **ListBox**
- **Menu**
- **StatusBar**
- **TabControl**
- **ToolBar**
- **TreeView**

# Lab: WPF – tvorba UI

- Vytvořte podle vzoru aplikaci na převod jednotek

The screenshot shows a WPF application window titled "Prevodnik". The window has a standard Windows title bar with minimize, maximize, and close buttons. Below the title bar is a toolbar with icons for file operations and a "Hot Reload" button. The main content area is divided into four rows, each representing a different unit conversion. Each row contains an input field on the left, a conversion button in the center, and the resulting value on the right.

| Input | Conversion     | Output             |
|-------|----------------|--------------------|
| 100   | C na F         | 212                |
| 50    | F na C         | 10                 |
| 100   | metry na stopy | 328,08398950000003 |
| 1000  | stopy na metry | 304,8000000012192  |