

Desktopové aplikace – Úvod

Desktopové aplikace jsou počítačové programy navržené k běhu na stolních počítačích nebo noteboocích. Na rozdíl od konzolových aplikací disponují GUI (grafickým uživatelským rozhraním) pomocí kterého je možné aplikaci ovládat. Na rozdíl od webových aplikací jsou desktopové aplikace nainstalovány přímo na počítači, přes který je aplikace spuštěna.

Při vytváření desktopových aplikací existuje široká škála možností. Byť je možné i tento typ aplikací vytvářet kompletně od základu s využitím pouze základních knihoven velmi často se z důvodu usnadnění práce využívají k jejich tvorbě frameworky. Framework je v kontextu tvorby aplikací nástroj, který zjednoduší práci tím, že předefinuje opakující se procesy, umožňuje opakování využití kódu a usnadňuje organizaci projektu. Frameworky jsou většinou vytvořeny pro specifický typ aplikací – existují frameworky pro tvorbu webových aplikací, mobilních aplikací, her, ale i desktopových aplikací. Mezi frameworky pro tvorbu desktopových aplikací je možné zařadit nástroje jako je například GTK, Electron, AppKit, WinForms, ale také WPF, kterému se budeme dále věnovat.

Co je to WPF?

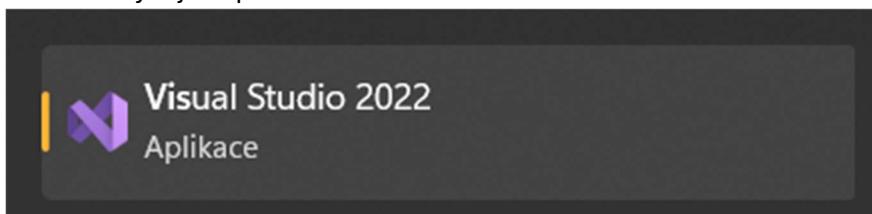
WPF je zkratkou Windows Presentation Foundation a jedná se o nástroj pro tvorbu aplikací pro operační systém Windows s grafickým uživatelským rozhraním pomocí platformy .NET.

V základu je možné WPF rozdělit na dvě varianty, a to na základě použité implantace .NET. Při vytváření aplikací je možné využít .NET Core a .NET Framework. .NET Framework je starší verze, která je nadále podporována, ale již se pro ni nevydávají nové funkce. .NET Core je nejnovější verze založená na open source implementaci .NET, která je nadále vyvíjena. V použití obou verzí jsou určité rozdíly, ale v úplném základu se s oběma pracuje stejně. Hlavními rozdíly jsou nastavení projektu, využití knihoven a nasazení aplikace. Pokud si budete moci mezi jednotlivými verzemi vybrat je většinou výhodnější volit novější verzi .NET Core, nicméně pokud budete z Vaší aplikace potřebovat vytvořit samostatně spustitelný exe soubor je nutné projekt správně nastavit. Pokud ale víte, že si vystačíte se starší verzí .NET Framework, je v pořádku využívat i ji.

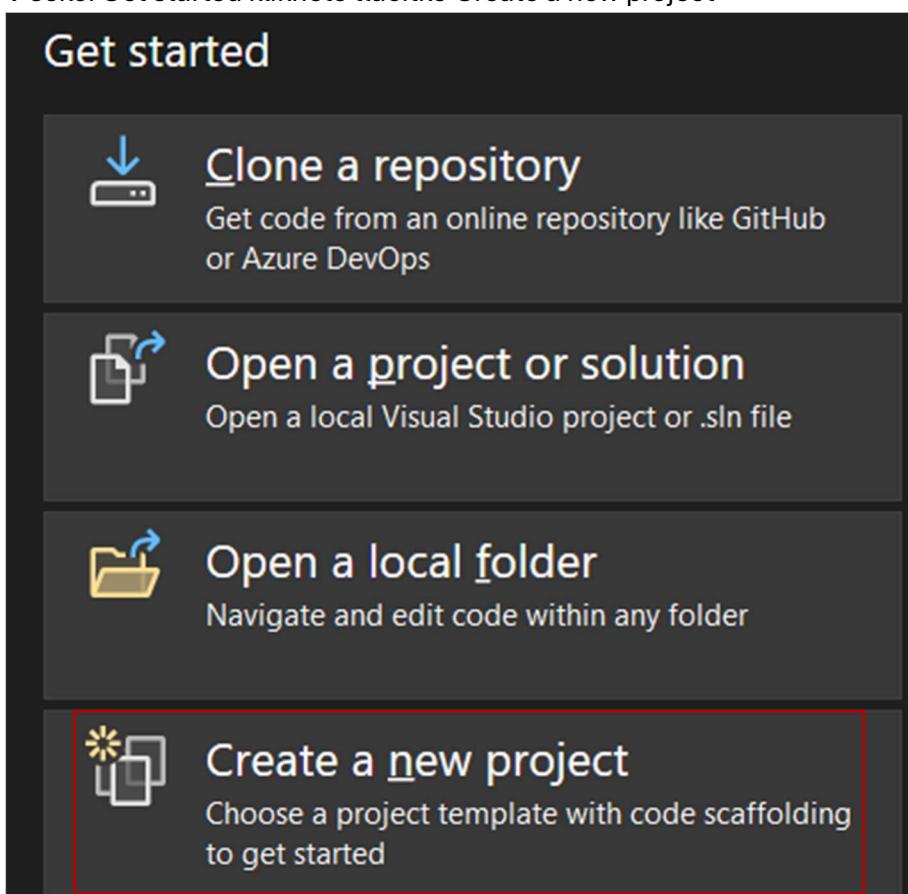
Postup při vytváření projektu

Před vytvořením projektu je nejprve nutné vybrat vývojové prostředí ve kterém budete pracovat. WPF aplikace je možné vytvářet pomocí široké škály vývojových prostředí a postup při vytváření projektu se bude u každého trochu lišit. Následující postup při vytváření aplikace se bude týkat vývojového prostředí Visual Studio. Visual Studio mám přenastavené do angličtiny (doporučuji udělat totéž), takže pokud budete mít nastavenou češtinu budou názvy sekcí a funkcí jiné.

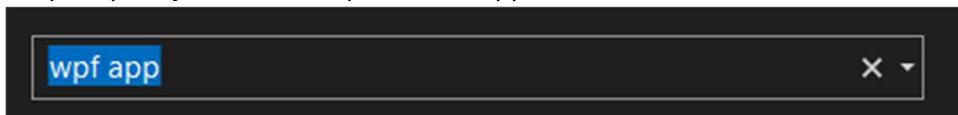
1. Otevřete vývojové prostředí Visual Studio.



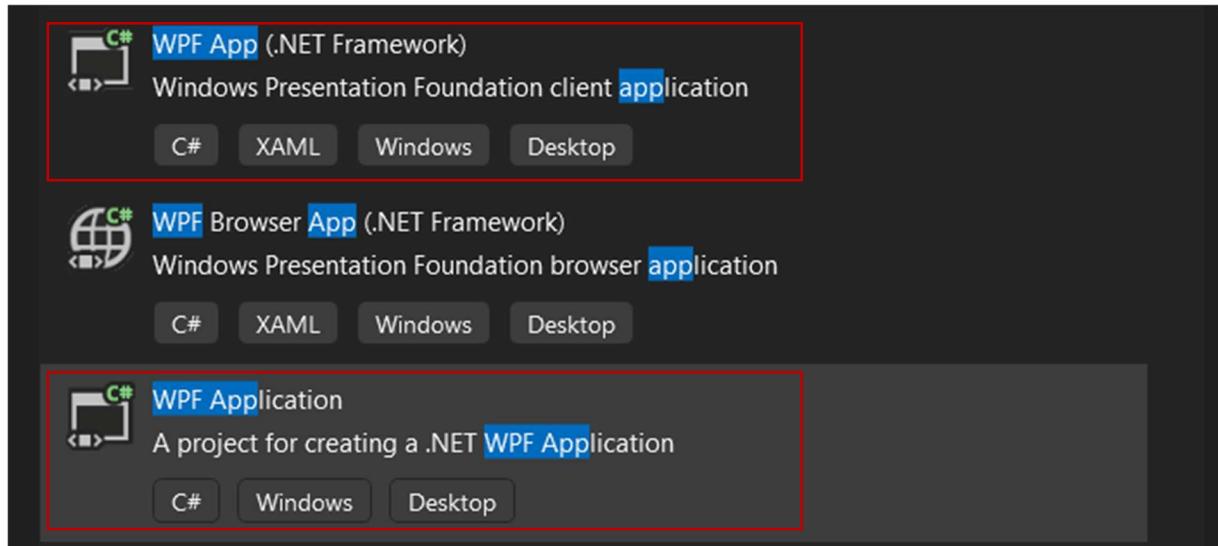
2. V sekci Get started klikněte tlačítko Create a new project



3. Do pole pro vyhledávání napište WPF app



4. Vyberte WPF App(.NET Framework) anebo WPF Application (pokud se Vám tyto možnosti neukazují, pak nemáte ve Visual Studiu nainstalované potřebné nástroje a je potřeba je doinstalovat – doinstalování je možné provést pomocí nástroje Visual Studio Installer)



5. Zvolte název projektu a složku do které bude projekt vložen

Configure your new project

WPF Application

C# Windows Desktop

Project name

MyDesktopApp

Location

C:\Users\Martin\source\repos

Solution name ⓘ

MyDesktopApp

Place solution and project in the same directory

Project will be created in "C:\Users\Martin\source\repos\MyDesktopApp\"

6. Vyberte verzi frameworku (nejlépe nejnovější LTS verzi) a klikněte na tlačítko Create

Additional information

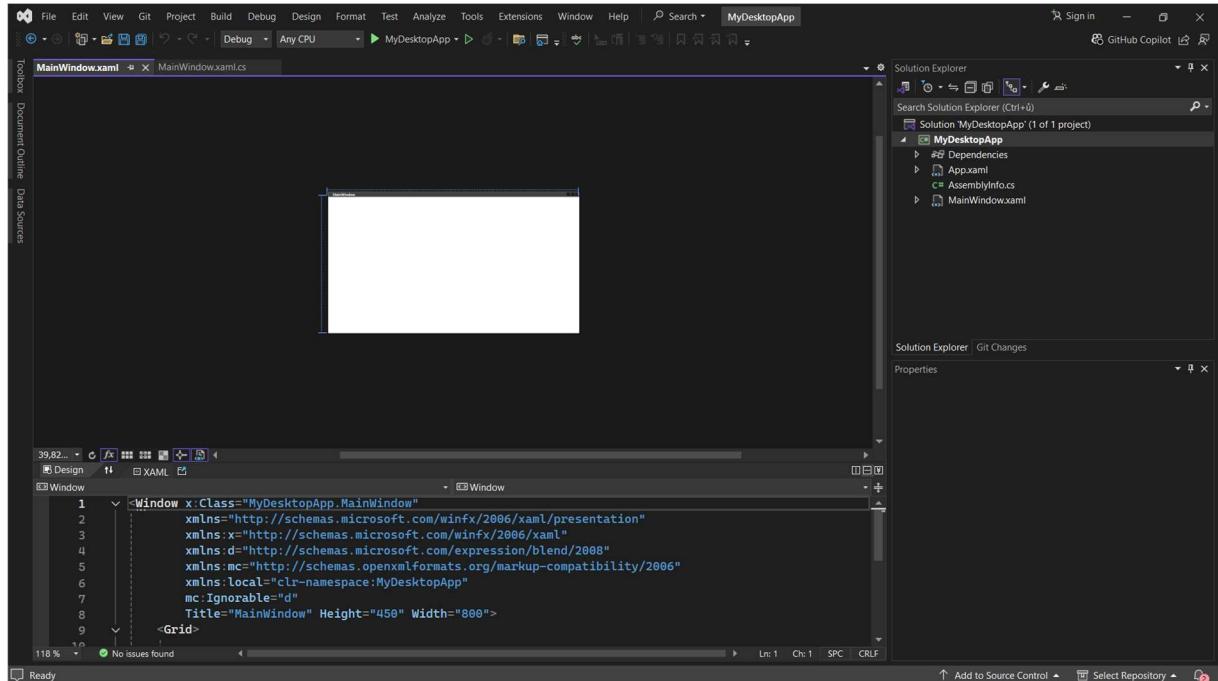
WPF Application

C# Windows Desktop

Framework ⓘ

.NET 8.0 (Long Term Support)

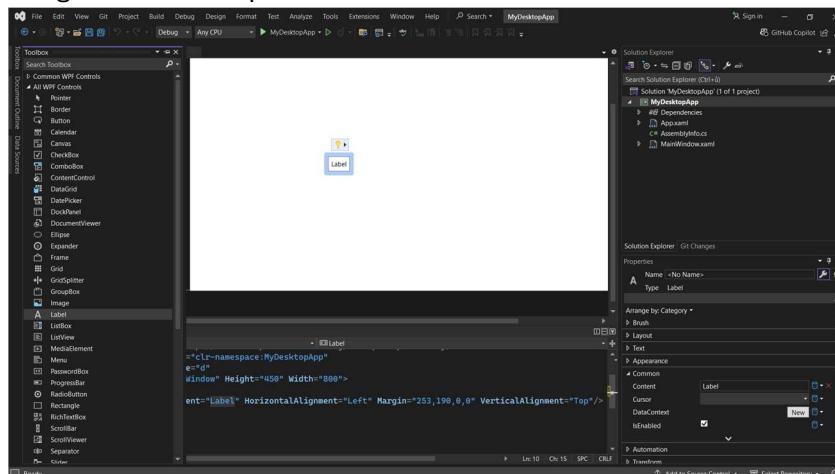
7. Po vytvoření projektu by se Vám mělo zobrazit takováto, či podobná obrazovka



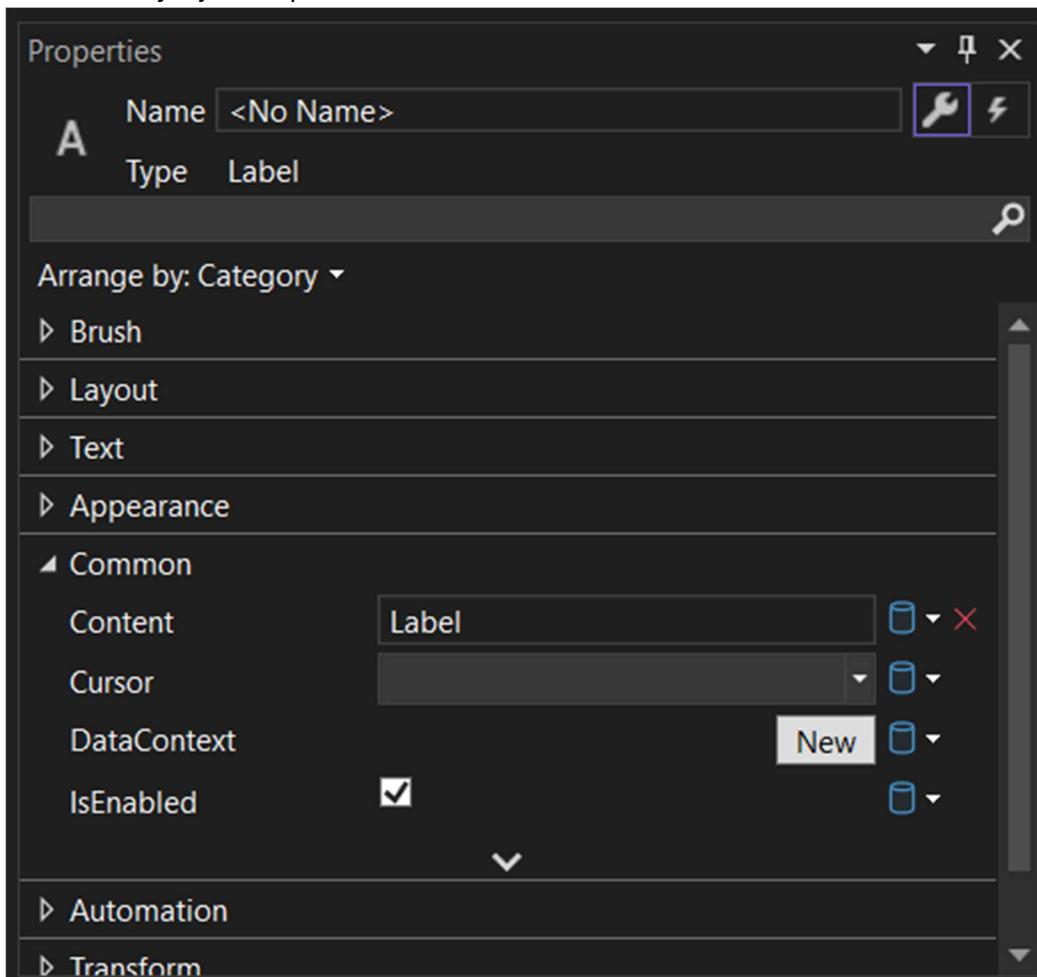
Uprostřed okna je možné vidět design vaší aplikace, pod designem je vidět XAML editor nalevo pak průzkumník řešení Vašeho projektu. Při vytváření WPF aplikací se primárně používají dva jazyky – XAML a C#. XAML slouží pro definici obsahu a grafické stránky aplikace (něco jako HTML a CSS ve webových aplikacích), C# slouží pro definici programové části aplikace. Po vytvoření aplikace automaticky vzniknou dva soubory, ve kterých budete určovat obsah a funkčnost aplikace. Jedná se o soubor MainWindow.xaml (XAML) a MainWindow.xaml.cs (C#) se kterými budeme nadále pracovat.

Použití grafických uživatelských prvků

Při definici uživatelských prvků vaší aplikace je možné v zásadě tři postupy – vkládání prvků pomocí panelu nástrojů (toolbox), pomocí jazyka XAML anebo pomocí kódu v C#. Pro vložení prvku pomocí panelu nástrojů stačí při otevřeném souboru MainWindow.xaml kliknout na tlačítko na levé straně obrazovky s názvem Toolbox. Následně je možné vybrat uživatelský prvek z otevřené nabídky a přetáhnout ho do designu okna vaší aplikace.



Po vložení prvku do aplikace je možné prvek nastavit pomocí nabídky zobrazené v pravé dolní části vývojového prostředí.



Zároveň je ale automaticky vytvořen v XAMLu kód, který definuje vlastnosti vytvořeného prvku a stejně nastavení jako pomocí výše zobrazené postranní nabídky je možné definovat pomocí XAMLu.

```
<Window x:Class="MyDesktopApp.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/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:local="clr-namespace:MyDesktopApp"
    mc:Ignorable="d"
    Title="MainWindow" Height="450" Width="800">
<Grid>
    <Label Content="Label" HorizontalAlignment="Left" Margin="253,190,0,0"
VerticalAlignment="Top"/>
</Grid>
</Window>
```

Pokud se na XAML podíváme blíže zjistíme, že se vněm nyní nachází celkem tři elementy – kořenový element Window, element Grid a element Label (námi vložený uživatelský prvek). U všech těchto elementů můžeme nastavovat jejich atributy a tím určit jejich chování. Element Window bude mít už spoustu atributů nastavených. Některé z nich budeme moci upravit a tím změnit nastavení aplikace, jiné budeme muset nechat, aby aplikace fungovala správně.

x:Class - Spojuje XAML soubor s odpovídajícím souborem kódu na pozadí (např. MainWindow.xaml.cs).

xmlns - Odkazuje na prvky WPF.

xmlns:x - Odkazuje na třídy a funkce z System.Xaml.

xmlns:d - Používá se pro data a atributy, které mají vliv pouze v návrhovém prostředí, nikoliv v době běhu aplikace.

xmlns:mc – mc (Markup Compatibility). Umožňuje kontrolovat, jak jsou neznámé nebo nepodporované atributy a prvky zpracovávány.

xmlns:local - Používá se pro přístup k vlastním třídám a ovládacím prvkům v aplikaci.

mc:Ignorable - Označuje, že návrhové nástroje mohou ignorovat specifické obory názvů nebo atributy, které nejsou potřeba pro spuštění aplikace, ale jsou použity při designu aplikace.

Title – titulek aplikace

Height – automaticky nastavená výška aplikace

Width – automaticky nastavená šířka aplikace

Uvnitř elementu Window se nachází element Grid (to co tento element dělá bude popsáno příště) a nakonec uvnitř elementu Grid je vložen námi definovaný element Label.

Element Label se nejčastěji používá při vytváření popisků jednotlivých částí Vaší aplikace. Atributy, které mu byly automaticky vytvořeny jsou

Content – Obsah popisku. Nejčastěji text, ale obecně vzato se může jednat o libovolný další uživatelský prvek. Obsah popisku je možné definovat pomocí atributu Content anebo pomocí vložení obshau mezi otevírající a uzavírající tag. Následující zápis v XAMLu je soubě ekvivalentní

1.

```
<Label Content="Label" HorizontalAlignment="Left" Margin="253,190,0,0"
VerticalAlignment="Top"/></Label>
```

2.

```
<Label HorizontalAlignment="Left" Margin="253,190,0,0"
VerticalAlignment="Top">Label</Label>
```

HorizontalAlignment – vodorovné zarovnání

VerticalAlignment – svislé zarovnání

Margin – odsazení – chování tohoto atributu je závislé na rodičovském prvku – v tomto případě Grid

Dalšími základními atributy, které můžeme u tohoto elementu nastavit, jsou například

FontFamily – font textu

FontSize – velikost fontu v pixelech

FontStyle – kurzíva textu

FontWeight – tučnost textu

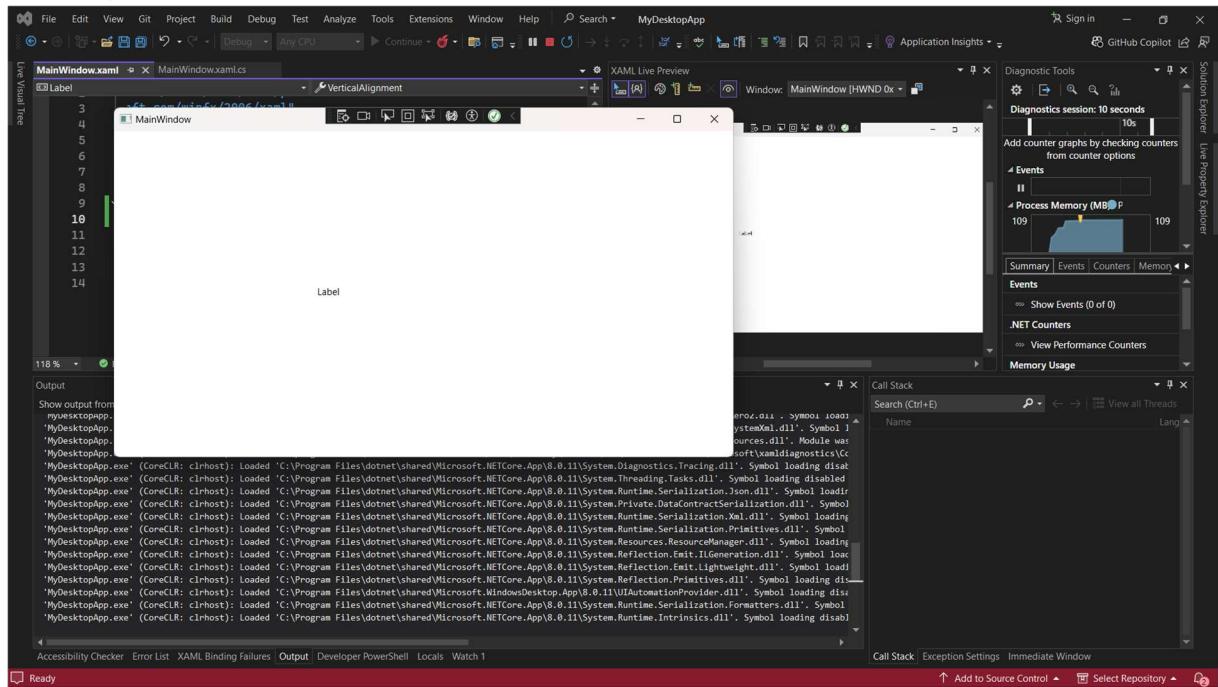
Background – barva pozadí

Foreground – barva písma

Width – šířka elementu

Height – výška elementu

Nyní, když budeme aplikaci potřebovat spustit, stačí kliknout na zelený trojúhelník v horní části obrazovky. Po jeho stisknutí byste měli vidět podobné okno jako je na následujícím obrázku.



Příště se podíváme na další uživatelský prvek, který budeme velmi často používat – tlačítko. Také si ukážeme, jakým způsobem určit, jak budou jednotlivé prvky v aplikaci umístěny.