

Windows Forms

Ing. Michal Radecký, Ph.D.

.NET framework, .NET 5, .Net Core

- .NET framework

- Běhový framework primárně na MS Windows
- Obsahuje m.j. Common Language Infrastructure (CLI), Framework Class Library (FCL), Common Language Runtime (CLR), podpora vícejazyčnosti
- Poslední verze 4.8 (2019)

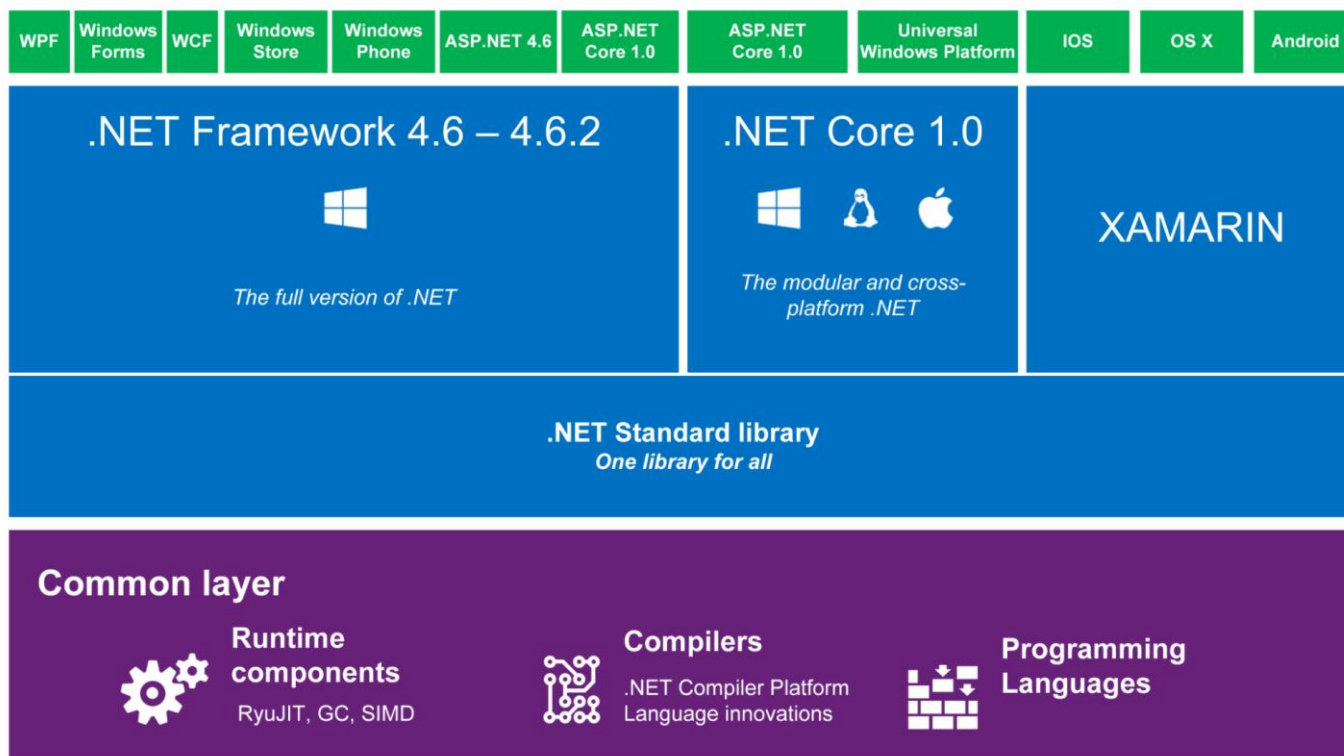
- .NET Core

- Oproti .NET frameworku je z principu multiplatformní
- Free a open-source (MIT), modernější a výkonnější přístup
- Poslední verze 3.1 (2019)

- .NET

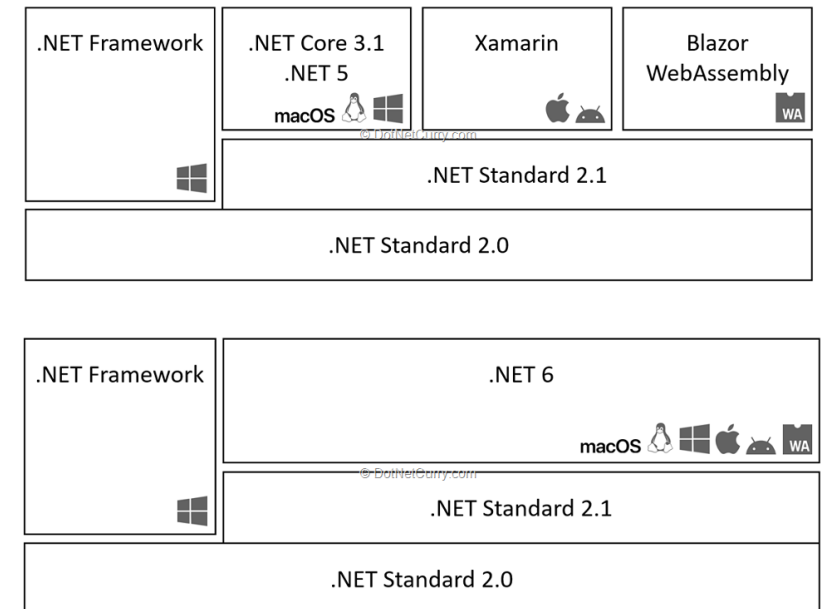
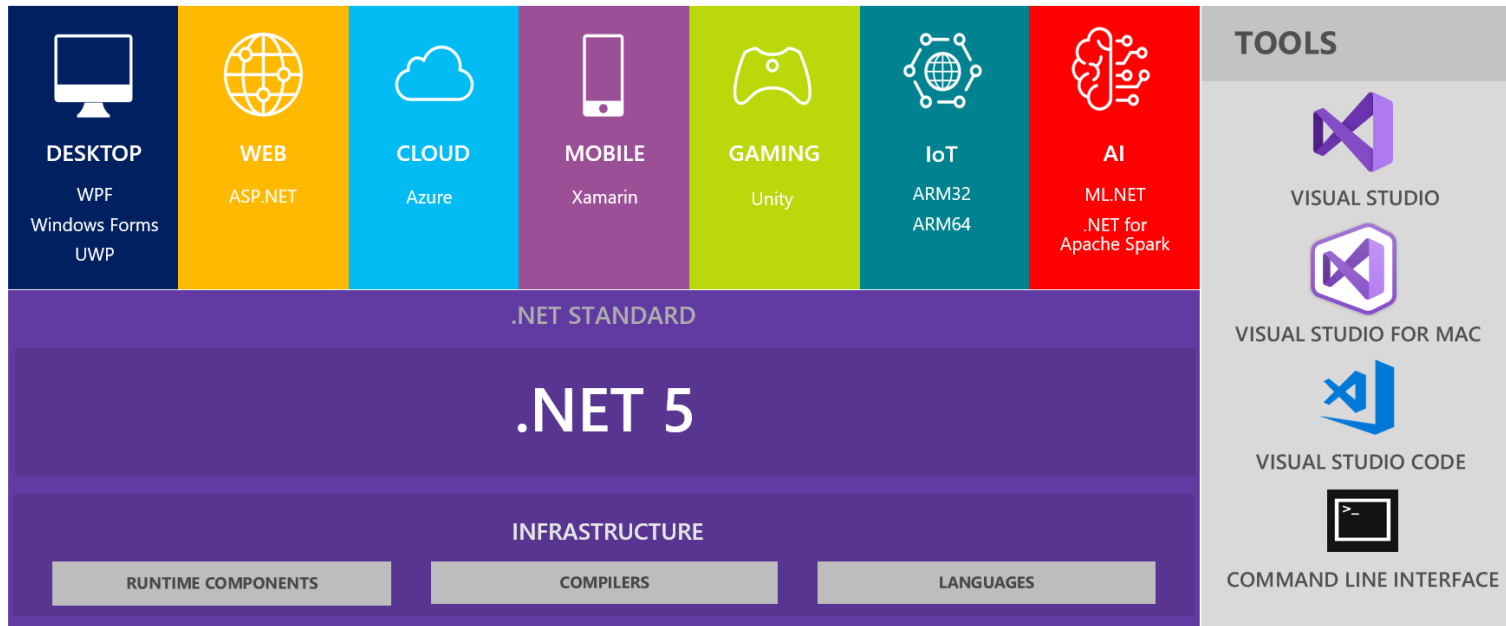
- Pokračování konceptu .NET Core (přejmenování)
- Aktuální verze 6, obsahuje C# 10, F# 6

.NET Framework tomorrow



.NET 5

- .NET Standard
 - Vrstva architektury, která umožnila vývoj .NET aktuálním směrem (eliminace .NET Frameworku)
 - Formální specifikace rozhraní .NET API napříč platformami, jednotná knihovna pro další navazující technologie
- .NET 5 „nahrazuje“ vrstvu .NET Standard a vystupuje jako jednotný produkt (vč. .NET standard) sjednocující všechny .NET běhové prostředí/aplikační modely do jedné platformy.



.NET 6

- Ještě více zaměřeno na unifikaci a univerzálnost platformy jako takové
- Novinky
 - Snadnější vývoj – nové možnosti C# 10
 - Vyšší výkon a nasazení (hot reload)
 - Blazor, .NET MAUI



- <https://docs.microsoft.com/cs-cz/dotnet/core/whats-new/dotnet-6>
- <https://devblogs.microsoft.com/dotnet/whats-new-in-windows-forms-in-net-6-0/>

Windows Forms

- Uživatelské rozhraní pro systémy Windows (primárně)
- Následník MFC (Microsoft Foundation Class Library) v C++
- Dnes i jako součást .NET 5+ (to neznamená, že WF jsou multiplatformní)
Pozor na možné odlišnosti v rámci vývoje .NET a .NET Framework
- Využívá principu „event-driven“ paradigmatu
- Poskytuje napojení ovládacích prvků na systémové Windows API, a to v rámci tzv. managed kódu (CLR), což oproti MFC (kde je vše na programátorovi) přináší snadnější a bezpečnější vývoj
- GDI+ - Graphics Design Interface je základní součást Windows a je zodpovědná za 2D grafiku (prvky UI, obrázky, atd.) ve vazbě na HW. GDI+ je knihovna (vyšší úroveň abstrakce) pro snadnou manipulaci s grafikou.

Hierarchie ovládacích prvků

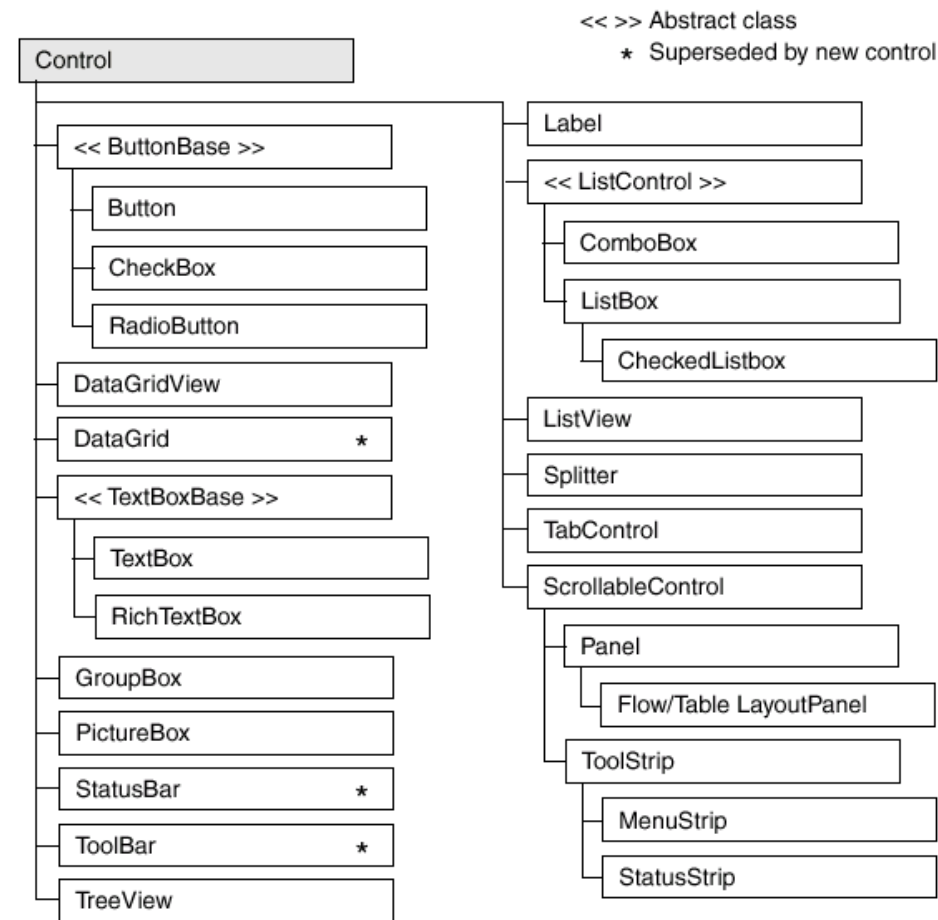
- Veškeré ovládací prvky UI jsou děděny z třídy Control
- Další důležitou třídou je kontejner pro okna/dialogy – Form

System.Windows.Controls.Control

<https://docs.microsoft.com/en-us/dotnet/api/system.windows.controls.control?view=net-5.0>

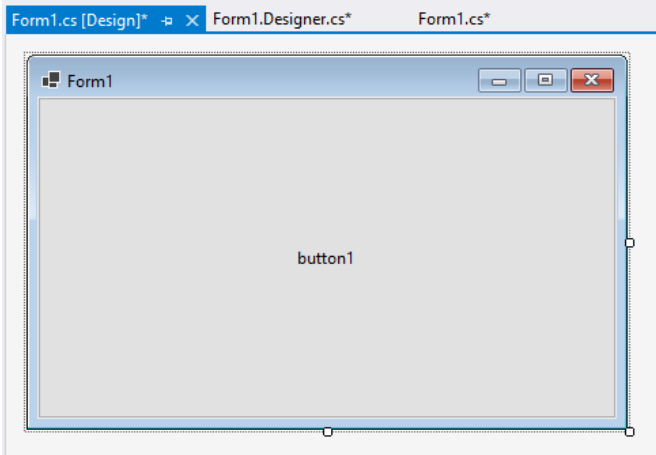
System.Windows.Forms.Form

<https://docs.microsoft.com/en-us/dotnet/api/system.windows.forms.form?view=net-5.0>



Partial třídy

- Klíčové slovo umožňující rozdělení specifikace jedné třídy (class, struct, interface) do více fyzických souborů
- Musí být dodržen namespace i stejné deklaráce (např. private)
- Při kompilaci dochází ke spojení deklarácí
- Klíčové slovo `partial` lze používat i u metod – deklaráce vs. definice/implementace
- Důležité pro případy, kdy třída je jednak generována automaticky (designerem) a zároveň obsahuje vlastní vytvořený kód



```

Form1.cs [Design]  Form1.Designer.cs  Form1.cs
BasicApp
1
2 namespace BasicApp
3 {
4     2 references
5     partial class Form1
6     {
7         /// <summary>
8         /// Required designer variable.
9         /// </summary>
10        private System.ComponentModel.IContainer components = null;
11
12        /// <summary>
13        /// Clean up any resources being used.
14        /// </summary>
15        /// <param name="disposing">true if managed resources should be disposed; otherwise, false.</param>
16        protected override void Dispose(bool disposing)
17        {
18            if (disposing && (components != null))
19            {
20                components.Dispose();
21            }
22            base.Dispose(disposing);
23        }
24
25        #region Windows Form Designer generated code
26
27        /// <summary>
28        /// Required method for Designer support - do not modify
29        /// the contents of this method with the code editor.
30        /// </summary>
31        private void InitializeComponent()
32        {
33            this.button1 = new System.Windows.Forms.Button();
34            this.SuspendLayout();
35            //
36            // button1
37            //
38            this.button1.Dock = System.Windows.Forms.DockStyle.Bottom;
39            this.button1.Location = new System.Drawing.Point(0, 427);
40            this.button1.Name = "button1";
41            this.button1.Size = new System.Drawing.Size(800, 23);
42            this.button1.TabIndex = 0;
43            this.button1.Text = "button1";
44            this.button1.UseVisualStyleBackColor = true;
45            this.button1.Click += new System.EventHandler(this.button1_Click);
46            //
47            // Form1
48            //
49            this.AutoScaleDimensions = new System.Drawing.SizeF(7F, 15F);
50            this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
51            this.ClientSize = new System.Drawing.Size(800, 450);
52            this.Controls.Add(this.button1);
53            this.Name = "Form1";
54            this.Text = "Form1";
55            this.ResumeLayout(false);
56        }
57
58        #endregion
59
60        private System.Windows.Forms.Button button1;
61    }
62 }
63
    
```

```

Form1.cs [Design]  Form1.Designer.cs  Form1.cs
BasicApp
1 using System;
2 using System.Collections.Generic;
3 using System.ComponentModel;
4 using System.Data;
5 using System.Drawing;
6 using System.Linq;
7 using System.Text;
8 using System.Threading.Tasks;
9 using System.Windows.Forms;
10
11 namespace BasicApp
12 {
13     2 references
14     public partial class Form1 : Form
15     {
16         1 reference
17         public Form1()
18         {
19             InitializeComponent();
20         }
21
22         1 reference
23         private void button1_Click(object sender, EventArgs e)
24         {
25             MessageBox.Show("TEST");
26         }
27     }
28 }
    
```


Designer

- Toolbox
 - Databáze komponent, které je možné využít v rámci návrhu UI
- Properties
 - Nastavování veškerých vlastností daných prvků
 - Vazba na kód v souboru `Form1.Designer.cs`
- Events
 - Definice metod pro zpracování událostí
 - Vazba na kód v souboru `Form1.cs`

The image shows three windows from the Visual Studio IDE:

- Toolbox:** A list of Windows Forms controls. The 'Menus & Toolbars' category is selected, showing controls like Pointer, ContextMenuStrip, MenuStrip, StatusStrip, and ToolStrip.
- Properties window:** Shows the properties for a `button1` of type `System.Windows.Forms.Button`. Properties include `Tag`, `Design` (Name: `button1`), `GenerateMember` (True), `Locked` (False), `Modifiers` (Private), `Fokus` (CausesValidation: True), `Chování` (AllowDrop: False, AutoEllipsis: False, ContextMenuStrip: (none), DialogResult: None, Enabled: True, TabIndex: 0, TabStop: True, UseCompatibleTextRenderin: False, Visible: True), `Rozložení` (Anchor: Top, Left; AutoSize: False; AutoSizeMode: GrowOnly; Dock: Bottom; Location: 0; 427; Margin: 3; 3; 3; 3; MaximumSize: 0; 0; MinimumSize: 0; 0; Padding: 0; 0; 0; 0; Size: 800; 23), `Usnadnění` (AccessibleDescription, AccessibleName, AccessibleRole: Default), `Vzhled` (BackColor: Control, BackgroundImage: (none), BackgroundImageLayout: Tile, Cursor: Default), `FlatAppearance` (FlatStyle: Standard), `Font` (Font: Segoe UI; 9pt; ForeColor: ControlText), `Image` (Image: (none), ImageAlign: MiddleCenter, ImageIndex: (žádný), ImageKey: (žádný), ImageList: (none), RightToLeft: No), `Text` (Text: `button1`, TextAlign: MiddleCenter).
- Events window:** Shows the event subscription list for `button1`. The `Click` event is selected, with the handler `button1_Click` assigned. Other events like `MouseDown`, `MouseEnter`, `MouseLeave`, `MouseUp`, `DragDrop`, `DragEnter`, `DragLeave`, `DragOver`, `GiveFeedback`, `QueryContinueDrag`, `DpiChangedAfterParent`, `DpiChangedBeforeParent`, `Layout`, `MarginChanged`, `Move`, `PaddingChanged`, `Resize`, `Paint`, `AutoSizeChanged`, and `BackColorChanged` are also listed.

Zpracování událostí

- Využívá se **principu delegátů** pro propojení událostí s metodami pro jejich zpracování (EventHandler, EventHandler<EventArgs>)
- K dané události se registrují metody, které se v případě vyvolání události spustí
 - One-to-many notifikace – delegát spustí postupně všechny registrované metody
 - Many-to-one notifikace – delegát umožňuje jednu metodu registrovat/spustit pro více událostí
 - Vše je dynamické a může se měnit v průběhu provádění kódu
- Události se zpracovávají/vyvolávají v určitém pořadí
 - S ohledem na jejich vznik
 - S ohledem na systémové pořadí, např. při zobrazení okna (<https://docs.microsoft.com/en-us/dotnet/desktop/winforms/order-of-events-in-windows-forms?view=netframeworkdesktop-4.8>)
- Události vznikají nezávisle, ale zpracovávají se v **hlavním vláknu aplikace!**
- Možnost implementace vlastních událostí (event, delegate) [<https://www.tutorialsteacher.com/csharp/csharp-event>]

```
this.button1.Click += new System.EventHandler(this.button1_Click);  
private void button1_Click(object sender, EventArgs e)
```

DataBinding

- Princip, pomocí kterého je možné automaticky jakoukoliv, za běhu přístupnou, vlastnost (Property) ovládacích prvků změnit či nastavit.
- Simple data binding
 - Základní možnost propojení jediné vlastnosti a hodnoty, např. text v TextBoxu

```
Binding binding1 = new Binding("Text", textBox1, "Text", true,
DataSourceUpdateMode.OnPropertyChanged);
textBox2.DataBindings.Add(binding1);
```

- Complex data binding
 - Rozšířená možnost propojení, kdy se s ovládacím prvkem propojuje celá kolekce, např. DataGridView, ListBox, atd. – využití DataSource

```
var bindingList = new BindingList<Person>(people.Members);
var source = new BindingSource(bindingList, null);
dataGridView1.DataSource = source;
```

Vlastní uživatelské prvky

- **UserControl**

- Snadné vytváření vlastních uživatelských komponent
- Vše je již „připraveno“ a pouze rozšiřujeme základní chování
- Např. znovupoužitelný formulář
- **System.Windows.Forms.UserControl**

<https://docs.microsoft.com/en-us/dotnet/api/system.windows.forms.usercontrol?view=net-5.0>

- **Control**

- Nejnižší úroveň tvorby vlastních ovládacích prvků
- Je třeba implementovat chování, vnitřní logiku i grafickou reprezentaci
- Např. analogové hodiny
- **System.Windows.Forms.Control**

<https://docs.microsoft.com/en-us/dotnet/api/system.windows.forms.control?view=net-5.0>

- Rozšíření stávajících prvků

- Pokud potřebujeme pouze upravenou funkcionalitu již existujícího prvku
- Zdědíme a rozšíříme již existující třídu
- Např. upravena vizualizace prvku (OnPaint)

MDI aplikace

- Multiple Document Interface
- Umožňuje snadno vytvářet vnořená okna a jejich řízení v rámci jedné aplikace
- Obvykle se využívají různé správci oken (externí knihovny)

```
this.IsMdiContainer = true;
```

```
Form2 f = new Form2();
```

```
f.MdiParent = this;
```

```
f.Show();
```

Další zdroje

- <https://docs.microsoft.com/en-us/dotnet/desktop/winforms/?view=netdesktop-5.0>
- <https://docs.microsoft.com/en-us/dotnet/desktop/winforms/?view=netframeworkdesktop-4.8>
- <https://docs.microsoft.com/en-us/visualstudio/designers/windows-forms-designer-overview?view=vs-2019>

- <https://flylib.com/books/en/3.61.1/>
- <https://flylib.com/books/en/4.253.1.69/1/>

- https://www.youtube.com/watch?v=W6vJ_c9Mt6A
- https://www.youtube.com/watch?v=s0_luQgVEfw

- **Obecně zajímavá výuková videa:**
<https://www.youtube.com/channel/UCJ3AxeCHGPZkMi3kRfCuiHw>