

# 1 - Úvod do platformy .NET

IW5 - Programování v .NET a C#

## Obsah přednášky

- Objektivě orientované paradigma
- .NET Framework
- Základní rysy jazyka C#

# Objektová orientace

## **C# implementuje základní koncepty OOP**

- Zapouzdření
- Dědičnost
- Polymorfismus

## **Charakteristické rysy C# oproti OOP**

- Unifikovaný typový systém
- Třídy a rozhraní
- Properties, Metody a Eventy

## Rysy C# - Unifikovaný typový systém

- Typ – zapouzdřuje data a funkce
- Sdílení základní functionality
- Převod instance na `string` – metoda `ToString()`

```
namespace System
{
    public class Object
    {
        public virtual string ToString() {}
        public virtual bool Equals(object obj) {}
        public virtual int GetHashCode() {}
    }
}
```

# Rysy C# - Třídy a rozhraní

## Třída = typ

- Data (členy)
- Operace (metody)

## Rozhraní (interface)

- Popisuje pouze členy třídy
- Chování „definuje“ třída, které jej implementuje
- Vícenásobná dědičnost tříd - **NE**
- Vícenásobná implementace rozhraní - **ANO**

```
public interface IBoy  
{  
    string Name {get;}  
}
```

```
public class Boy : IBoy  
{  
    public string Name  
    {  
        ...  
    }  
}
```

# Rysy C# – členy třídy

## Properties

- Zapouzdřují část stavu objektu
- Např. Color

## Metoda

- Implementuje chování objektu
- Obdoba funkce
- Např. SetButtonColor

## Event

- Změnu stavu objektu
- Např. ColorChanged

```
public class Button
{
    public event EventHandler ColorChanged;

    public Color Color { get; set; }

    public void SetButtonColor(Color color)
    {
        Color = color;

        if (ColorChanged != null)
        {
            ColorChanged(this, EventArgs.Empty);
        }
    }
}
```

## C# - Typová bezpečnost

- Silně typovaný jazyk = typ musí být znám v době překladu
- Podpora IntelliSense ve Visual Studiu
- **POZN:** klíčové slovo `dynamic` – lze použít dynamický typ

### Výhody

- Eliminace chyb již v době překladu
- Ochrana objektu před narušením jeho stavu – „Sandbox”

```
Button button = new Button();
```

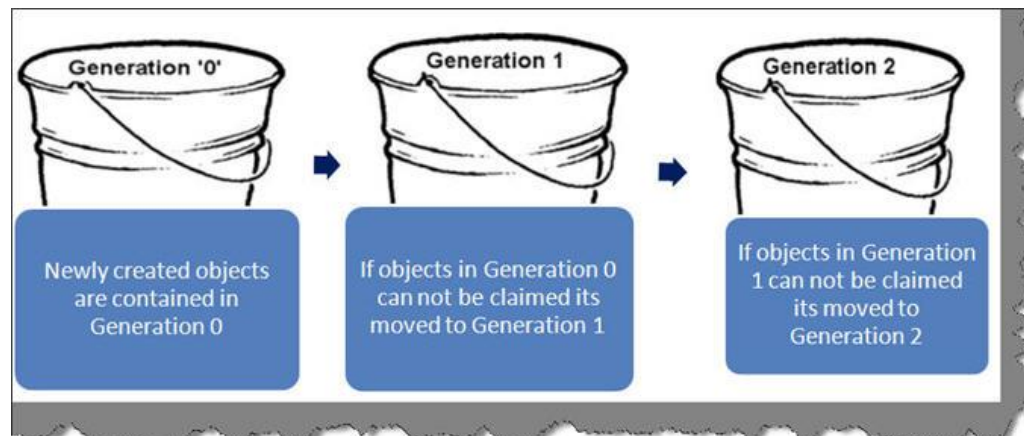
```
var button = new Button();
```

```
Button button = new Color();
```

# C# - Správa paměti

## Garbage collector (GC)

- Automatická správa paměti bez assistance programátora
- Součást CLR, princip počítání referencí na daný objekt
- **Výhody**
  - Odpadá manuální uvolňování paměti
  - Eliminace problému s ukazateli (časté v C++)



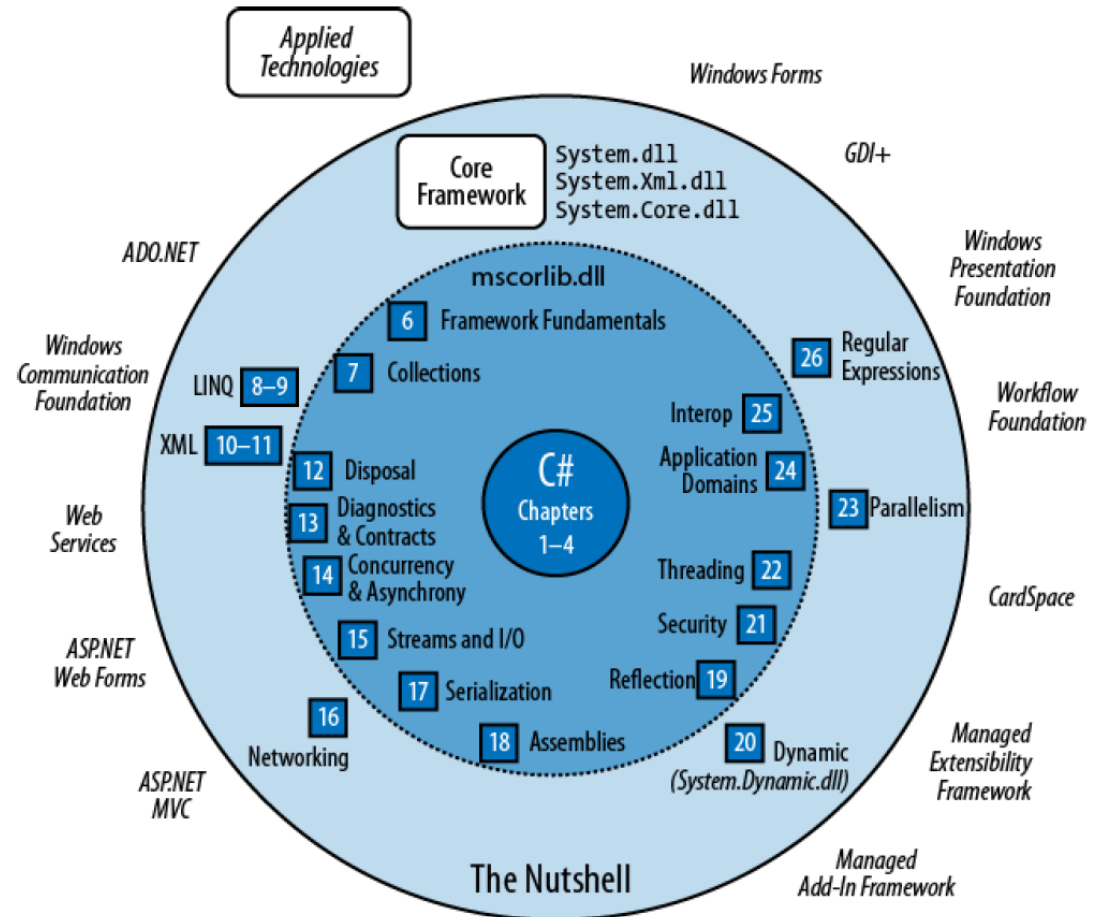


# .NET Framework

- Tvořen Common language runtime (CLR) a velkým množstvím knihoven

## Typy knihoven

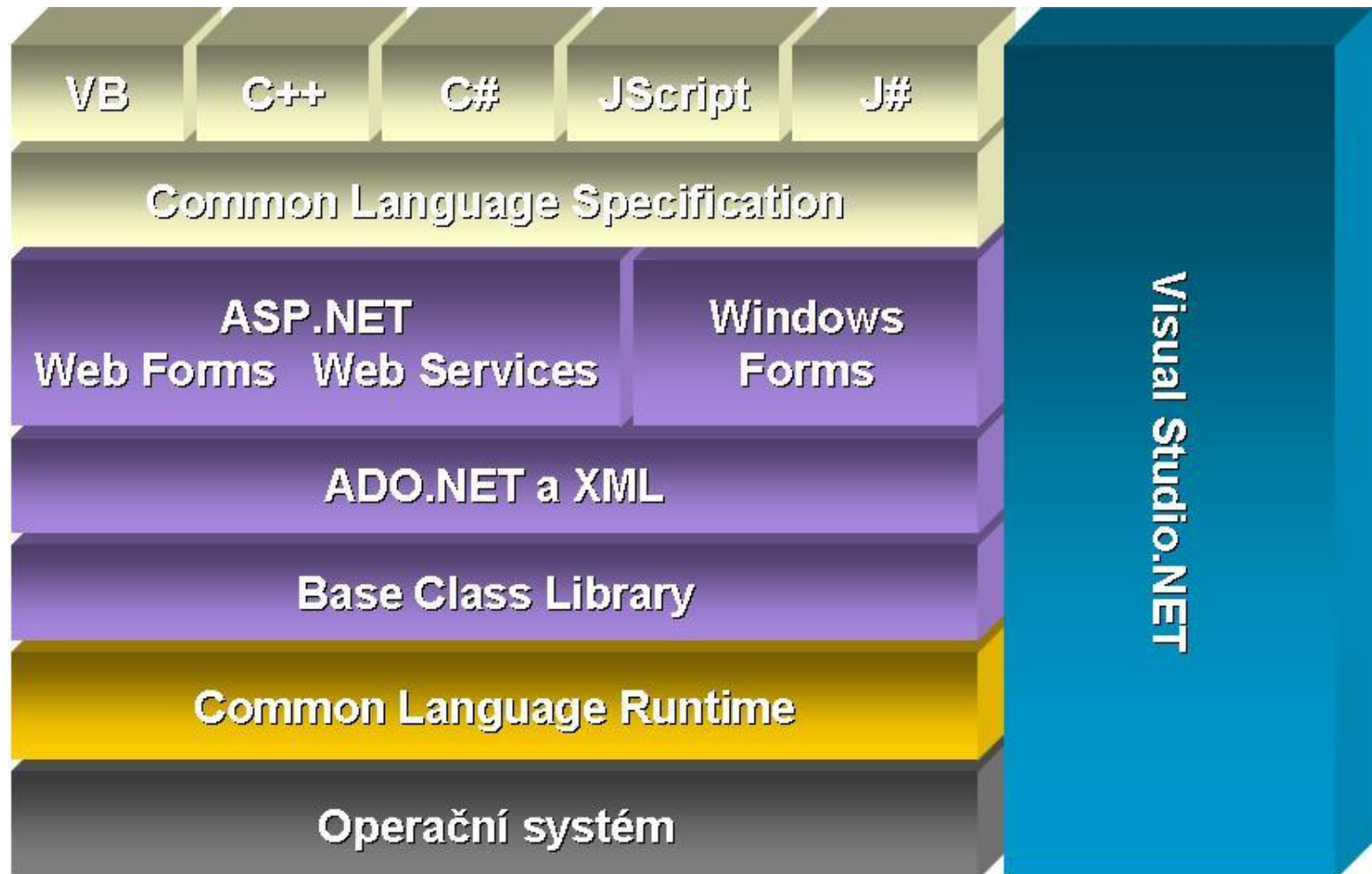
- Core framework
- Aplikační technologie



## .NET Framework – vybrané knihovny

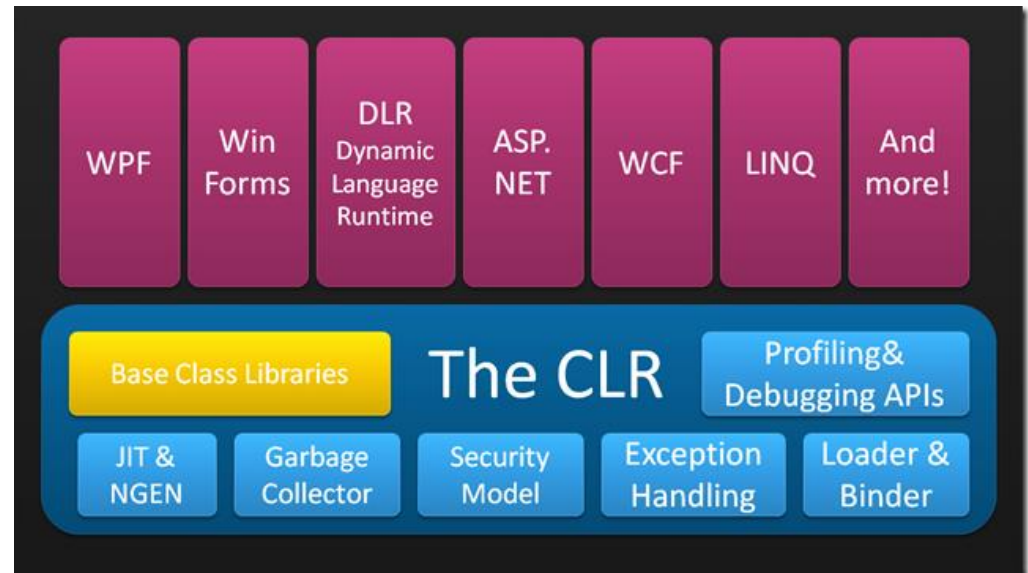
- **WinForms**
- **ASP.NET**
- **WPF** – Windows Presentation Foundation
- **WCF** – Windows Communication Foundation
- **WF** – Windows Workflow Foundation
- **LINQ** – Language Integrated Query

## .NET Framework - architektura



# CLR – Common Language Runtime

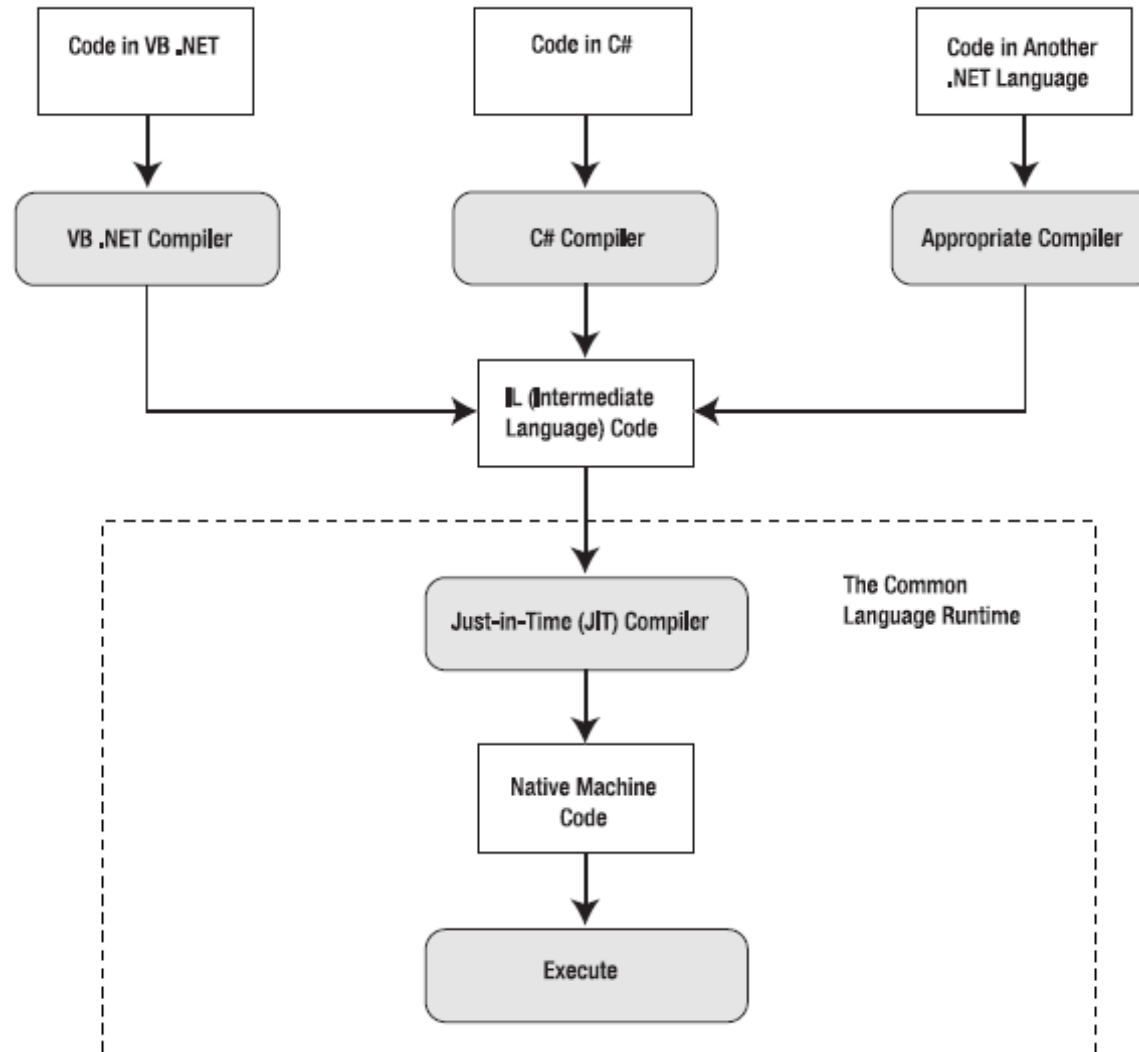
- Běhové prostředí pro vykonávání managed kódu
- Obdoba Java VM pro .NET
- **Poskytované služby**
  - Správa paměti
  - Načítání knihoven
  - Bezpečnostní služby
  - Zachytávání výjimek
  - ...
- **Jazykově neutrální**
  - Podpora vývoje ve více jazycích
  - C#, VB, Managed C++, Delphi .NET, F#, ...



## CLR - detaily

- C# je kompilován do tzv. **managed kódu**
- Managed kód je zabalen do assembly, která je dvou typů
  - Spustitelný kód (\*.exe)
  - Knihovna (\*.dll)
- **IL - Intermediate language**
  - Reprezentace managed kódu
  - Při čtení CLR je kód z assembly konvertován do nativního kódu stroje (x86)
- **JIT – Just-In-Time kompilátor**
  - Realizuje proces převodu IL do nativního kódu
  - Dynamické generování kódu

# CLR - detaily



## Novinky ve verzích C#

### C# 2.0

- Generika
- Nullable typy
- Anonymní metody
- Iterator blocks
- Properties – getter and setter
- Partial typy

```
List<T> list = new List<T>();
```

```
Nullable<int> pocet = null;
```

```
p = delegate(string j) {Console.WriteLine(j); };
```

```
yield return;
```

```
public Color Color { get{...} set{...} }
```

```
public partial class TasksWindow { public int x = 1; }
```

```
public partial class TasksWindow { public int y = 1; }
```

```
public partial class TasksWindow { public TasksWindow() {
```

```
Console.WriteLine(x+y);}
```

```
}
```

## Novinky ve verzích C#

### C# 3.0

- Expression trees
- Implicitní lokální typ – `var`
- Lambda výrazy
- Extension metody
- Auto property
- LINQ

```
var cars = new List<Car>();  
(param)=>{Console.WriteLine(param);}
```

```
public Color Color { get; set; }
```

```
List<Car> cars = new List<Car>();  
var redCars = cars.Where(c => c.Color == Color.Red)  
                    .Select(r => r.Name);
```



## Novinky ve verzích C#

### C# 4.0

- Dynamický binding
- Volitelné parametry a jména argumentů
- Typová variance – generické interface a delegáty
- COM interoperabilita

### C# 5.0

- Podpora pro asynchronní funkce – `async` a `await`

### C# 6.0

- Nový kompilátor Roslyn
- Součástí VS 2015

## Podpora platforem u jazyka C#

- Primárně navržen pro běh na platformě Windows, ale existují výjimky

### ASP.NET

- C# kód spouštěn na serveru
- Překlad do HTML – podporováno na všech platformách
- ASP.NET 5.0 Linux, Mac OS X

### Mono projekt

- Běh na jiném runtime než CLR, má vlastní kompilátor
- Linux, Solaris, MAC OS X, Windows

### Silverlight

- Host podporující aplikaci psanou v C#, obdoba Adobe Flash
- Windows, MAC OS X

## .NET Core

- Open source, podpora pro Windows, Linux and Mac OsX
- Multiplatformní implementace základních knihoven .NET

# .NET 2015

## .NET Framework



ASP.NET 5  
ASP.NET 4.6  
WPF  
Windows Forms

## .NET Core



ASP.NET 5  
.NET Native



ASP.NET 5 for Mac and Linux

## Common



**Runtime**  
Next gen JIT  
SIMD



**Compilers**  
.NET Compiler Platform  
Languages innovation



## NuGet packages

.NET Core 5 Libraries  
.NET Framework 4.6 Libraries

## Reference

- <http://www.amazon.com/5-0-Nutshell-The-Definitive-Reference/dp/1449320104>
- [http://2.bp.blogspot.com/\\_QZT9kid9l2Q/TJha8ti9JjI/AAAAAAAAAB0/xnJCLlpz-iU/s1600/8.jpg](http://2.bp.blogspot.com/_QZT9kid9l2Q/TJha8ti9JjI/AAAAAAAAAB0/xnJCLlpz-iU/s1600/8.jpg)
- <http://www.cs.vsb.cz/behalek/vyuka/pcsharp/text/resources/1.jpg>
- [http://gwb.blob.core.windows.net/sdorman/WindowsLiveWriter/CLR4.0InProcessSidebySideCLRHosting\\_93B4/image\\_thumb.png](http://gwb.blob.core.windows.net/sdorman/WindowsLiveWriter/CLR4.0InProcessSidebySideCLRHosting_93B4/image_thumb.png)
- [http://www.c-sharpcorner.com/uploadfile/prvn\\_131971/chapter-1-introducing-AspNet/Images/1.gif](http://www.c-sharpcorner.com/uploadfile/prvn_131971/chapter-1-introducing-AspNet/Images/1.gif)
- [http://blogs.msdn.com/cfs-filessystemfile.ashx/\\_key/communityserver-blogs-components-weblogfiles/00-00-01-12-34/5488.Pic2.png](http://blogs.msdn.com/cfs-filessystemfile.ashx/_key/communityserver-blogs-components-weblogfiles/00-00-01-12-34/5488.Pic2.png)

## Poděkování

- **Autoři původních materiálů k IW5, XMW5 2015**
- Ing. Radek Gajdušek
- Ing. Zdeněk Jurka
- Ing. Martin Minařík, Ph.D.
- Ing. Jaroslav Nečas
- Bc. Václav Pachta
- Mgr. Martin Procháska