

Programování

Řetězce, sestavení, .NET framework, přetypování, hlavní funkce

Řetězce - string

- ▶ Datový typ pro reprezentaci textu
- ▶ Lze na něj pohlížet jako na dynamické pole znaků (**char**)
- ▶ String vs string
- ▶ Ve většině jazyků je String objektem, ale string je datovým typem
- ▶ V C# však mezi nimi není rozdíl a String je implicitně přetypován na string
- ▶ Spojování řetězců za pomoci operandu +
- ▶ Přestože se jedná o datový typ, má stejně jako běžný objekt své vlastnosti a metody

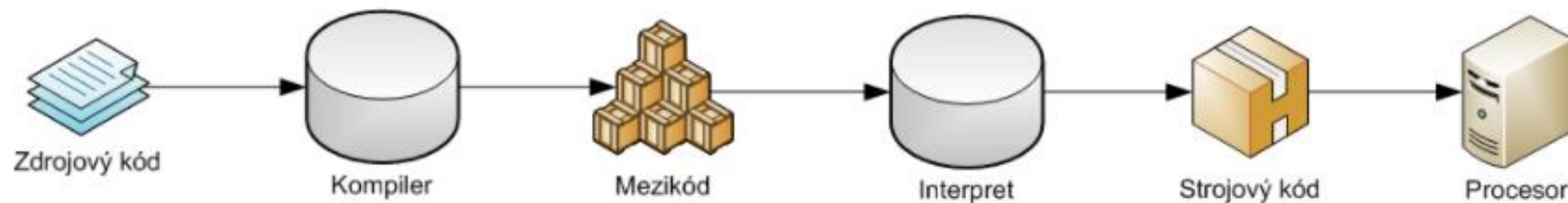
Přetypování

- ▶ Jazyk C# je silně staticky zadáný
- ▶ Do proměnných určitého typu lze tedy vkládat hodnoty pouze příslušného typu
- ▶ `int number = 3,14` -> **NELZE**, **int podporuje pouze celočíselné hodnoty**
- ▶ `double number2 = 1` -> **LZE**, **dochází k implicitnímu přetypování**
 - ▶ v proměnné `number2` se nachází hodnota 1,0
- ▶ **Explicitní přetypování** = přetypování vynucené vývojářem
 - ▶ V proměnné `a` se nachází hodnota 1234
- ▶ **Přetypování s podpůrnými třídami** - např. Parse metody
 - ▶ `int.Parse()` - metoda, která se snaží z řetězce získat celé číslo

```
double x = 1234.7;  
int a;  
// Cast double to int.  
a = (int)x;
```

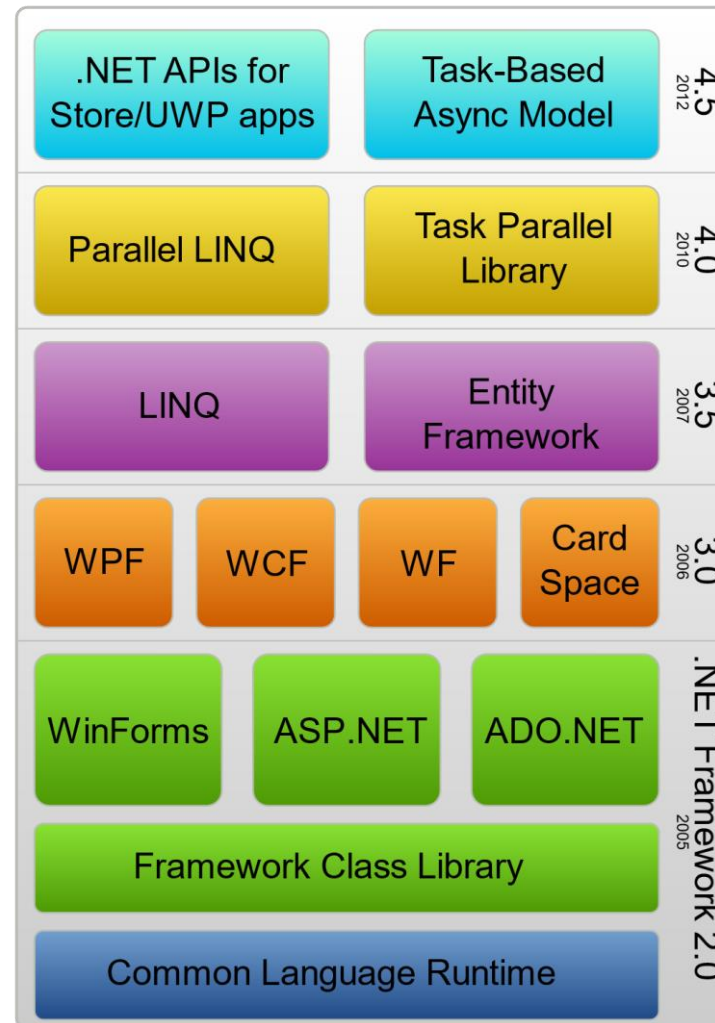
Sestavení programu

- ▶ Zdrojový kód je přeložen pomocí kompilátoru do mezikódu
 - ▶ CIL (Common Intermediate Language)
- ▶ Kód v mezikódu je následně snadno interpretován interpretem
 - ▶ CLR (Common Language Runtime)
- ▶ Tím získáme výslednou aplikaci ve formátu .exe nebo .dll



.NET framework

- ▶ Jazyk, VisualStudio, Virtuální stroj (CLR) a knihovny
- ▶ Knihovny .NET poskytují nepřehledné množství již předpřipravených metod a funkcí - práce s databázemi, formuláři, konzolí, ...
- ▶ **.NET Core** - implementace určená převážně pro webové aplikace, která není vázaná pouze na OS Windows



Hlavní funkce aplikace

- ▶ Main ()
- ▶ Vstupní bod aplikace nejen v C#, ale i v dalších programovacích jazycích
- ▶ Metoda, která se při spuštění aplikace volá jako první
 - ▶ Hlavní vlákno aplikace
- ▶ Definována uvnitř třídy nebo struktury
- ▶ Vždy musí být statická
- ▶ Může být definována s parametrem `string[] args`
- ▶ Pole obsahující argumenty pro spuštění aplikace s příkazového řádku
- ▶ `static void Main(string[] args)`

Metody / funkce v C#

- ▶ Metoda = blok kódu, který obsahuje řadu příkazů
- ▶ Program zajistí provedení metody jejím voláním a zadáním příslušných argumentů
- ▶ Každá metoda musí obsahovat:
 - ▶ Viditelnost
 - ▶ Název
 - ▶ Návratový typ
 - ▶ Seznam parametrů
- ▶ `public bool IsValid(string email) { }`

Práce se standardním proudem

- ▶ Při spuštění hlavního vlákna aplikace se automaticky spouští i tři hlavní proudy (stream)
 - ▶ Standardní vstup - **System.Console.In**
 - ▶ Standardní výstup - **System.Console.Out**
 - ▶ Standardní chybový proud - **System.Console.Error**
- ▶ Pro čtení ze standardního vstupu využíváme **Console.Read()**
Console.ReadLine()
- ▶ Pro výpis na standardní výstup využíváme **Console.Write()**
Console.WriteLine()
- ▶ Pro výpis na standardní chybový proud používáme **Console.Error.WriteLine()**

Práce se standardním proudem - přehled některých metod

- ▶ `public static void WriteLine (string value);`
 - ▶ Výpis textu do konzole ukončené novým řádkem
- ▶ `public static void Write (string value);`
 - ▶ Výpis textu do konzole bez odřádkování
- ▶ `public static int Read ();`
 - ▶ Načtení znaku z konzole
- ▶ `public static int ReadLine ();`
 - ▶ Přečtení celého řádku z konzole – načítání ukončujeme pomocí Enter
- ▶ `public static ConsoleKeyInfo ReadKey ();`
- ▶ `public static void Clear ();`
 - ▶ Smazání veškerého textu v konzoli
- ▶ `public static void Beep ();`