

Programovací jazyk

C#

Úvod do programování v C#

Ing. Marek Běhálek
Katedra informatiky FEI VŠB-TUO
A-1018 / 597 324 251
<http://www.cs.vsb.cz/behalek>
marek.behalek@vsb.cz



.NET Framework



- Dramaticky zjednodušit vývoj aplikace.
- Zjednodušený model nasazení a managementu aplikace.
- Vytvoření robustního a bezpečného prostředí pro běh aplikace.
- Multi-jazyková podpora.
- Jednotný programový model.

- Podobný model jaký znáte z Javy

Úvod do programování v C#

2

.NET Framework - Architektura .NET Framework



Úvod do programování v C#

3

.NET Framework - Common Language Runtime



- Aplikace skládající se z komponent v různých jazycích:
 - Common Language Specification (CLS);
 - Common Type System (CTS).
- Robustní prostředí.
- Potenciálně multi-platformní.
- Zjednodušení vývoje a nasazování aplikace.
- Bezpečnost.

Úvod do programování v C#

4

.NET Framework - Common Language Runtime

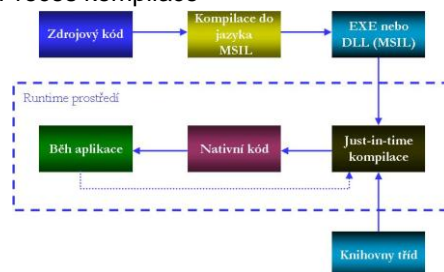
- Automatický management životního cyklu objektu.
 - Využívá „garbage collector“ pro management systémových zdrojů.
- Korektní ošetření chyb
 - Chyby jsou ošetřovány pomocí výjimek.
 - CLR přímo podporuje zpracovávání výjimek. Tento proces je tedy nezávislý na programovacím jazyce.
- Typová bezpečnost
 - Garance že nad definovanými typy nemohou být provedeny nepovolené operace.
 - Odstraňuje chyby pramenící z nekontrované manipulace s proměnnými nebo pamětí.

Úvod do programování v C#

5

.NET Framework - Common Language Runtime

- Proces kompilace



Úvod do programování v C#

6

.NET Framework - Common Language Runtime

- Řízený kód
 - Kód o jehož provádění se stará CLR
 - Jazyk generující jen řízený kód je Visual Basic
- Neřízený kód
 - Aplikace které nejsou napsány v .NET
 - Zřekneme-li se služeb CLR
 - Jazyk schopný generovat neřízený kód je například C++
- Výstupem kompilátoru jazyka schopného generovat řízený kód je MS intermediate language.
 - "Vyspělejší assembler" – objekty, výjimky, zásobníkové instrukce, ... (bytecode Javy)
 - Důvodem zavedení je snaha o jednoduché přenášení mezi různými hardwarovými platformami.
 - Rychlost zajištěna Just-in time kompilací.

Úvod do programování v C#

7

.NET Framework - Common Language Runtime

- Poskytuje „čistý“ objektově orientovaný přístup.
 - Třídy a Rozhraní.
 - Konstruktory, vlastnosti, metody, události, ...
 - Umožňuje rozšiřovat funkcionalitu třídy pomocí dědičnosti v různých jazycích.
- Vestavěna zpětná kompatibilita:
 - s aplikacemi postavenými na technologii COM;
 - s nativními DLL ve stylu Win32®.

Úvod do programování v C#

8

.NET Framework - Common Language Runtime



- Common Type System
 - Vše je objekt. Základem každého typu je třída System.Object. Tato třída definuje tyto metody:
 - Equals;
 - GetHashCode;
 - GetType;
 - ToString.
 - Základní dělení
 - Hodnotové typy
 - Referenční typy

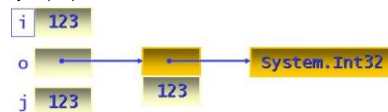
Úvod do programování v C#

9

.NET Framework - Common Language Runtime



- Common Type System
 - Automatická konverze mezi hodnotovými a referenčními typy (Boxing / Unboxing).
 - `int i = 123;`
 - `Object o = i;`
 - `int j = (int)o;`



Úvod do programování v C#

10

.NET Framework - Common Language Runtime



- Požadavky vedoucí k zjednodušení nasazení aplikace
 - Aplikace musí být samostatné – logicky nezávislé na registry, ...
 - Aplikace musí obsahovat čísla verzí a musí být na ně vázána.
 - Čísla verzí komponent
 - Musí podporovat Side-by-side komponenty.
 - Musí umožňovat izolaci aplikace.
 - Musí zajistit bezpečný přístup ke kódu.
 - Komponenty musí obsahovat informace o „veřejných typech“.

Úvod do programování v C#

11

.NET Framework - Common Language Runtime



- Základní jednotkou distribuce v .NET je assembly.
- Assembly je:
 - logická kolekce jednoho nebo více .exe, .dll nebo .module souborů a zdrojů doplněná Manifestem;
 - programová jednotka určená k nasazení. Umožňuje opakované použití, řízení verzí a podporuje bezpečnost.

Úvod do programování v C#

12

.NET Framework - Common Language Runtime

- Manifest je blok metadat obsahující informace
 - Identitu – jméno, verze a kultura;
 - Seznam souborů + kryptografické zabezpečení;
 - Odkaz na další použité assembly + jejich verze;
 - Exportované (veřejně viditelné) typy a zdroje;
 - Bezpečnostní požadavky:
 - Nutné pro spuštění assembly;
 - Doporučené pro běh;
 - Ty, které by neměly být nikdy přiděleny.

Úvod do programování v C#

13

.NET Framework - Common Language Runtime

- Instalace .NET aplikací a komponent
 - Soukromé assembly
 - Instalace typu XCopy.
 - Manifest obsahuje všechny potřebné údaje.
 - Sdílené assembly
 - Nejčastěji instalovány do GAC – Global Assembly Cache.
 - Složitější instalace i odinstalování aplikace.
 - Definuje Sdílené jméno.

Úvod do programování v C#

14

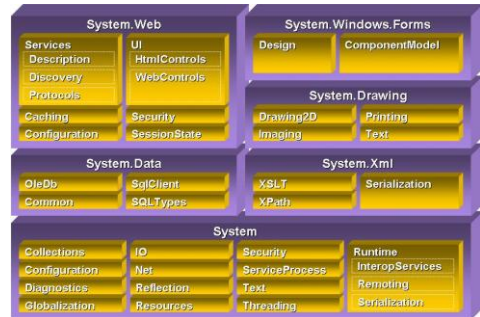
.NET Framework – Základní knihovny



Úvod do programování v C#

15

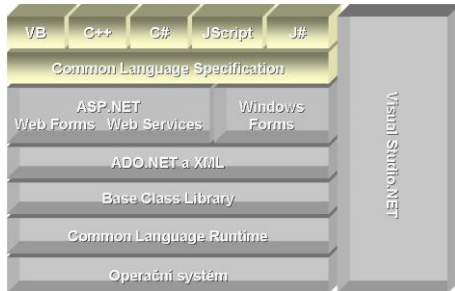
.NET Framework – Základní knihovny



Úvod do programování v C#

16

.NET Framework – Programovací jazyky



Úvod do programování v C#

17

.NET Framework – Programovací jazyky

- Platforma .NET je jazykově nezávislá.
 - Všechny jazyky si jsou principiálně v rámci .NET platformy rovny.
 - Preferovány jsou ovšem jazyky C# a Visual Basic
- Common Language Specification
 - Definuje základní vlastnosti jenž jsou očekávány v každém programovacím jazyce na platformě .NET.
- Microsoft implementoval podporu jazyků (ve verzi 1.0):
 - Visual Basic®, C++, C#, J#, JScript®.
- Jazyky integrované(?) třetí stranou:
 - APL, COBOL, Pascal, Eiffel, Haskell, ML, Oberon, Perl, Python, Scheme, Smalltalk, F#, ...

Úvod do programování v C#

18

.NET Framework – Ukázkový příklad

- Program vypíše text „Hello world!“. Je napsán v jazyce C#.

- Soubor: HelloWorld.cs

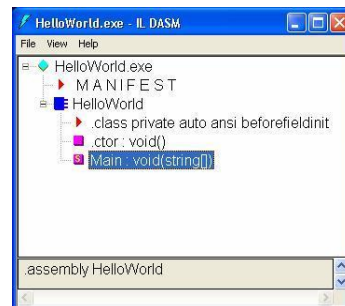
```
using System;
class HelloWorld {
    public static void Main() {
        Console.WriteLine("Hello world!");
    }
}
```

Po překladači vznikne soubor HelloWorld.exe který je možno spustit.

Úvod do programování v C#

19

.NET Framework – Ukázkový příklad



Úvod do programování v C#

20

.NET Framework – Ukázkový příklad

```
HelloWorld::Main : void(string[])
.method private hidebySig static void Main(string[] args) cil managed
{
  .entrypoint
  // Code size 11 (0xb)
  .maxstack 1
  IL_0000: ldstr    "Hello world!"
  IL_0005: call     void [mscorlib]System.Console::WriteLine(string)
  IL_000a: ret
} // end of method HelloWorld::Main
```

Úvod do programování v C#

21

.NET Framework – Ukázkový příklad

```
MANIFEST
assembly extern mscorlib
{
  assemblyName = (0F 7A 5C 00 19 34 ED 89 ) # zřV A...
  version = 1.0.2000.0
}
assembly HelloWorld
{
  custom instance void [mscorlib]System.Reflection.AssemblyKeyNameAttribute::ctor(string) = ( 01 00 00 00 00 )
  custom instance void [mscorlib]System.Reflection.AssemblyNameAttribute::ctor(string) = ( 01 00 00 00 00 )
  custom instance void [mscorlib]System.Reflection.AssemblyFileVersionAttribute::ctor(string) = ( 01 00 00 00 00 )
  custom instance void [mscorlib]System.Reflection.AssemblyTrademarkAttribute::ctor(string) = ( 01 00 00 00 00 )
  custom instance void [mscorlib]System.Reflection.AssemblyCopyrightAttribute::ctor(string) = ( 01 00 00 00 00 )
  custom instance void [mscorlib]System.Reflection.AssemblyProductAttribute::ctor(string) = ( 01 00 00 00 00 )
  custom instance void [mscorlib]System.Reflection.AssemblyCompanyAttribute::ctor(string) = ( 01 00 00 00 00 )
  custom instance void [mscorlib]System.Reflection.AssemblyDescriptionAttribute::ctor(string) = ( 01 00 00 00 00 )
  custom instance void [mscorlib]System.Reflection.AssemblyTitleAttribute::ctor(string) = ( 01 00 00 00 00 )
  // -- The following custom attribute is added automatically, do not uncomment. ---
  // custom instance void [mscorlib]System.Diagnostics.DebuggableAttribute::ctor( bool ) = ( 01 00 01 01 00 00 )
}
hash algorithm 0x00000004
hash 1 011653695
module HelloWorld.exe
#MD5: 6D5E8A9B-00F3-4C5E-A73E-ECEAB818FCA2)
imagebase 0x00000000
subsystem 0x00000003
file alignment 512
corflags 0x00000001
image base 0x02000000
```

Úvod do programování v C#

22

.NET Framework – Ukázkový příklad

- .NET Framework
 - Zjednodušuje vývoj a nasazení aplikace.
 - Poskytuje robustní a bezpečné prostředí pro běh aplikace.
 - Multi-jazyková podpora.
 - Rozsáhlé knihovny funkcí.
- Vše integrováno v jednom vývojovém nástroji Visual Studio.NET.
 - To je v rámci MSDN licence na katedře k dispozici studentům!

Úvod do programování v C#

23

.NET Framework – Základní charakteristika jazyka C#

- Jazyk C# vyvinula firma Microsoft.
- Vychází z jazyka C++, ale v mnoha ohledech se více podobá Javě.
- Jak tvrdí Microsoft: C# je nový jazyk s jednoduchostí Visual Basicu a možnostmi C++.
- Většina vlastností vychází přímo z vlastností .NET Framework.
- Jazyk C# je integrován ve vývojovém prostředí Visual Studio.

Úvod do programování v C#

24

.NET Framework – Základní charakteristika jazyka C#



- Jazyk C# je čistě objektově orientovaný.
- Obsahuje nativní podporu komponentního programování.
- Podobně jako Java obsahuje pouze jednoduchou dědičnost s možností násobné implementace rozhraní.
- Vedle členských dat a metod přidává vlastnosti a události.
- Správa paměti je automatická. O korektní uvolňování zdrojů aplikace se stará garbage collector.
- Podporuje zpracování chyb pomocí výjimek.
- Zajišťuje typovou bezpečnost a podporuje řízení verzí
- Podporuje atributové programování.
- Zajišťuje zpětnou kompatibilitu se stávajícím kódem jak na binární tak na zdrojové úrovni.

Úvod do programování v C#

25

.NET Framework – Používané konvence pro psaní programů (1)



- Jazyk C# je *case sensitive*.
- Používá tři typy notací
 - Pascal case - Název je složen z několika slov a první písmeno každého takového slova je velké (SomeLongName).
 - Camel case - Název je složen z několika slov a první písmeno všech krom prvního slova je velké (someLongName).
 - Uppercase - Všechna písmena jsou velká. Používá se pro zkratky a krátké (dokumentace uvádí dvouznakové) názvy (System.IO).
- Doporučené konvence lze najít v dokumentaci k .NET Framework (heslo: Naming Guidelines).

Úvod do programování v C#

26

.NET Framework – Používané konvence pro psaní programů (2)



- Třída – *SomeBigClass*
- Výčtový typ – *ErrorLevel*
- Hodnota ve výčtovém – *FatalError*
- Událost – *MouseMoved*
- Třída rozšiřující výjimku – *ParseException*
 - Jméno výjimky by mělo končit slovem *Exception*.
- Pouze pro čtení a statické položky – *SomeValue*
- Rozhraní – *IComparable*
 - Název rozhraní by měl začínat velkým písmenem I.
- Metoda – *SomeNiceMethod*
- Jmenný prostor - *System.Drawing*
- Parametr nebo lokální proměnná – *someName*
- Vlastnosti – *BackColor*
- Privátní nebo chráněné instanční položky třídy – *redValue*
- Veřejné instanční položky - *RedValue*

Úvod do programování v C#

27

.NET Framework – Komentáře (1)



- Poznámky vycházejí z jazyka C
 - jednořádkové komentáře jsou uvozeny dvěma lomítky - //
 - víceřádkový komentář začíná /* a končí */
- Speciální význam má značka TODO. Komentář který po ní následuje se zobrazí v panelu aplikace Visual Studio s názvem Task List.
- Jednořádkové komentáře uvozené třemi lomítky budou obsaženy v dokumentaci, která je standardně generována ze zdrojového kódu.
 - Generovaná dokumentace využívá XML.

Úvod do programování v C#

28

.NET Framework – Kostra programu



```
using System;

namespace Namespace
{
    /// <summary>
    /// Summary description for Class1.
    /// </summary>
    class Class1
    {
        /// <summary>
        /// The main entry point for the application.
        /// </summary>
        [STAThread]
        static void Main(string[] args)
        {
            //
            // TODO: Add code to start application here
            //
        }
    }
}
```