

## Zpravodaj moderní Programování 2/2016: C# v interaktivním režimu

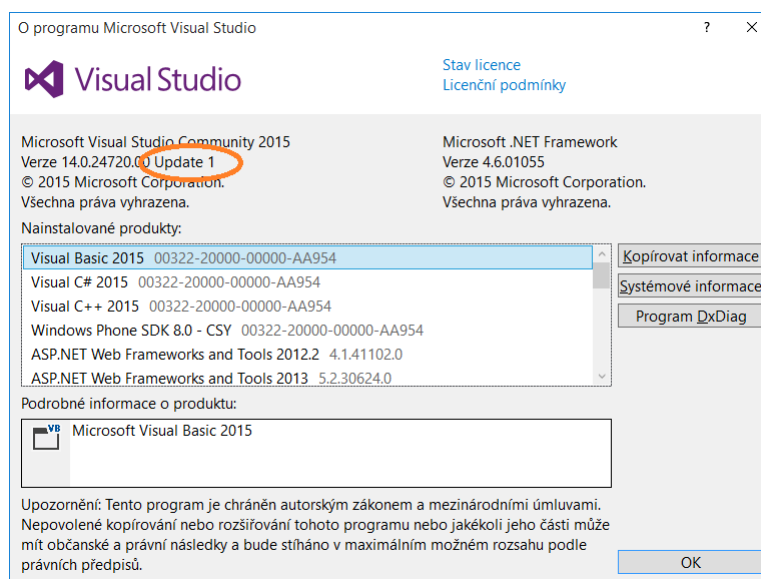
Poměrně nedávno Microsoft uvedl aktualizaci Update 1 vývojového prostředí Visual Studio 2015. V této aktualizaci se nachází, řekl bych zásadní, novinka, a tou je možnost programovat v C# v *interaktivním režimu*.

### Interaktivní režim

O co jde? Je to **možnost zapisovat příkazy jazyka C# a ihned je spouštět**. To znamená, že je můžete studovat, můžete si s nimi hrát, a to **s okamžitou odezvou**. Nemusíte vytvářet nový projekt, ten sestavovat a spouštět, znovu zadávat data apod. Příkaz, co zapíšete, nedělá to, co jste si mysleli? Hned můžete zapsat a spustit jiný... Pěkná věc pro školní výuku i samostudium.

### Update 1

Abyste mohli interaktivní režim využívat, potřebujete Visual Studio Community 2015 Update 1. Možná již Update 1 máte - podívejte se do nabídky *Nápověda > O programu Microsoft Visual Studio*.



Pokud nemáte, přes nabídku *Nástroje > Rozšíření a aktualizace* si jej můžete nainstalovat.

### Kam zmizela čeština?

Nevím, jak u vás, ale u mne po instalaci Update 1 zmizela z Visual Studia čeština. Nainstaloval jsem tedy znovu český jazykový modul a vše bylo zase v pořádku.

Pro jistotu, instalaci českého jazykového modulu popisují na

<http://moderniprogramovani.cz/knihy/moderni-programovani/ucebnice-pro-zacatecniky/vyvojove-prostredi.aspx>.

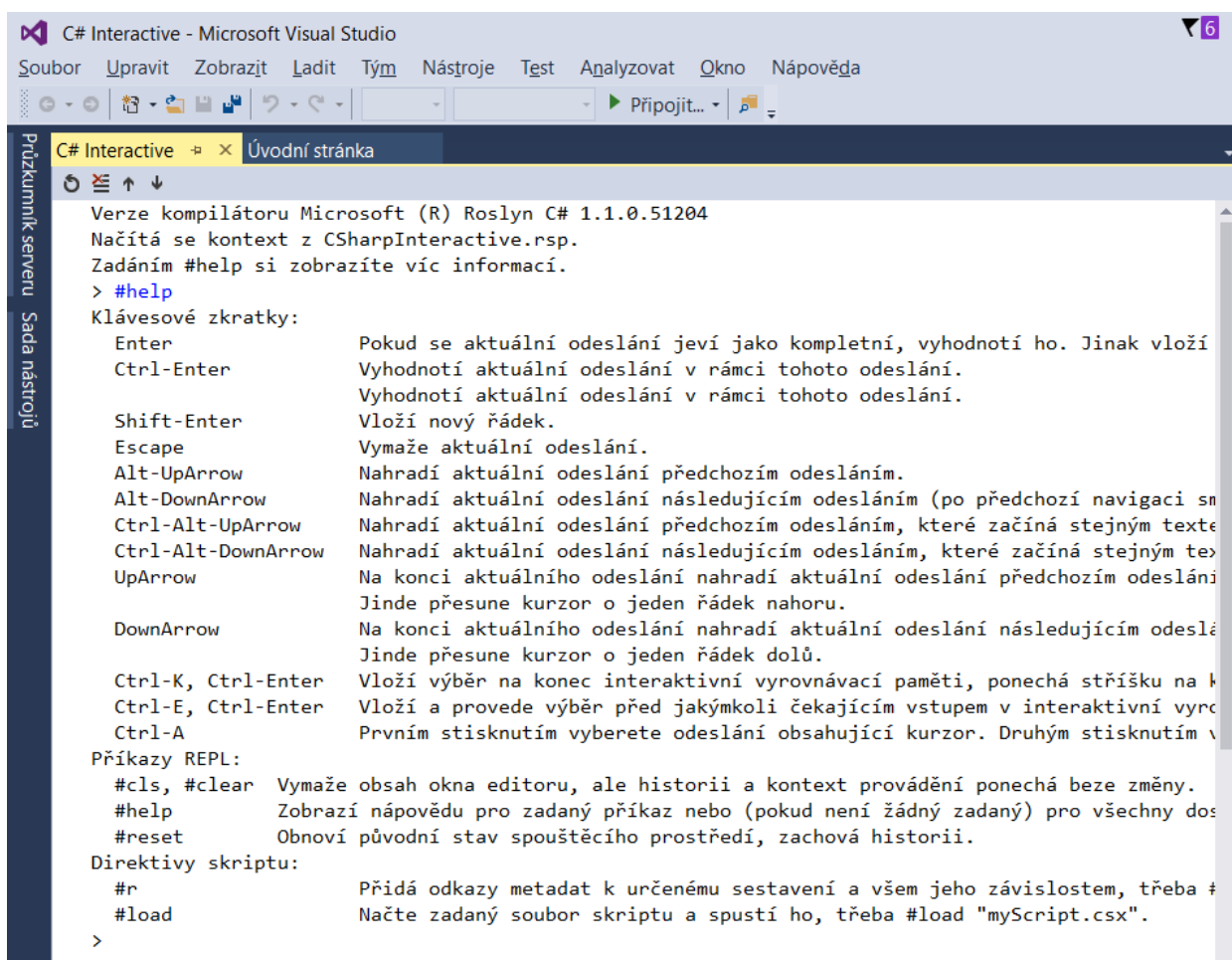
## Spuštění interaktivního režimu

Spustíte Visual Studio. Nemusíte ani zakládat nový projekt. Pokud ho ale již máte otevřený, nic se neděje. Interaktivní režim je takové pískoviště, které funguje nezávisle na případně otevřeném projektu.

Z nabídky Visual Studia zvolte *Zobrazit > Ostatní okna > C# Interactive*.

## Ovládání

V okně zadáváte C# příkazy a odesíláte obvykle *Enterem*. Seznam klávesových zkratk najdete po zadání a odeslání příkazu `#help`.



```

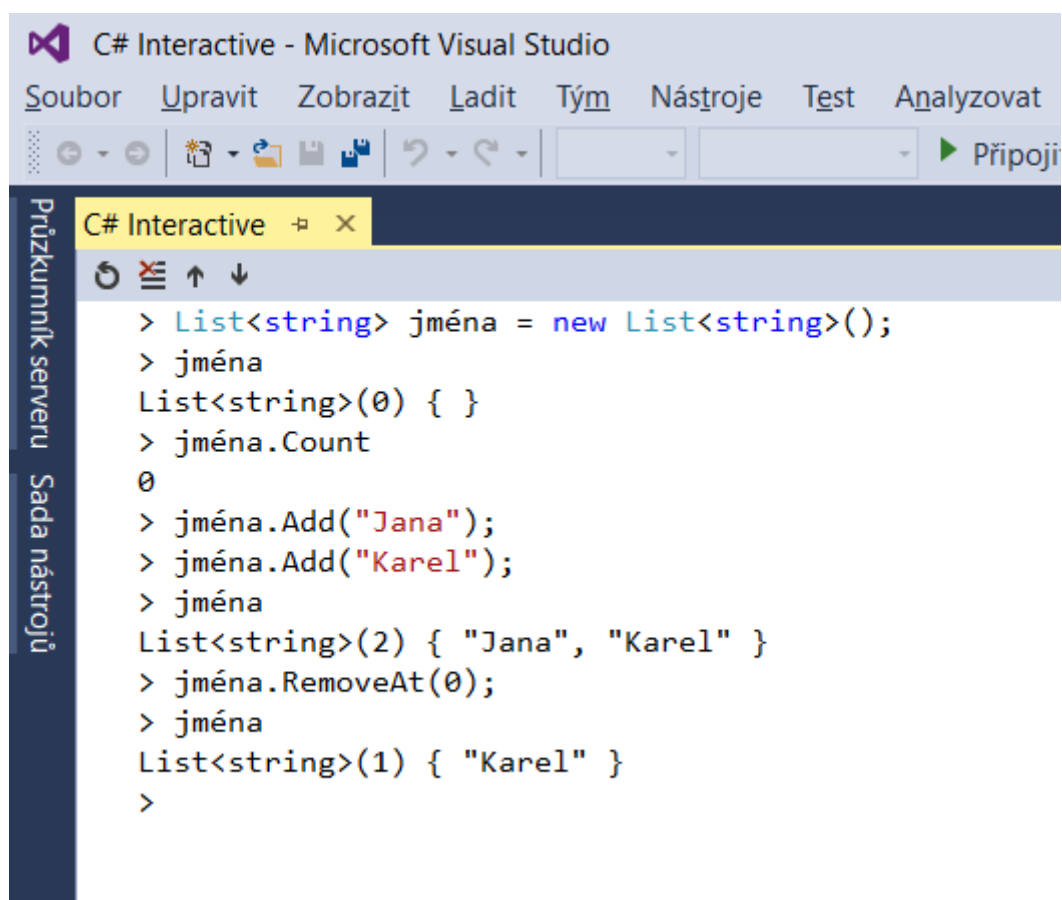
C# Interactive - Microsoft Visual Studio
Soubor  Upravit  Zobrazit  Ledit  Tým  Nástroje  Test  Analyzovat  Okno  Nápověda
C# Interactive  Úvodní stránka
Verze kompilátoru Microsoft (R) Roslyn C# 1.1.0.51204
Načítá se kontext z CSharpInteractive.rsp.
Zadáním #help si zobrazíte víc informací.
> #help
Klávesové zkratky:
  Enter           Pokud se aktuální odeslání jeví jako kompletní, vyhodnotí ho. Jinak vloží
  Ctrl-Enter      Vyhodnotí aktuální odeslání v rámci tohoto odeslání.
                  Vyhodnotí aktuální odeslání v rámci tohoto odeslání.
  Shift-Enter     Vloží nový řádek.
  Escape          Vymaže aktuální odeslání.
  Alt-UpArrow     Nahradí aktuální odeslání předchozím odesláním.
  Alt-DownArrow   Nahradí aktuální odeslání následujícím odesláním (po předchozí navigaci sn
  Ctrl-Alt-UpArrow Nahradí aktuální odeslání předchozím odesláním, které začíná stejným text
  Ctrl-Alt-DownArrow Nahradí aktuální odeslání následujícím odesláním, které začíná stejným text
  UpArrow         Na konci aktuálního odeslání nahradí aktuální odeslání předchozím odesláním
  Jinde přesune kurzor o jeden řádek nahoru.
  DownArrow       Na konci aktuálního odeslání nahradí aktuální odeslání následujícím odesláním
  Jinde přesune kurzor o jeden řádek dolů.
  Ctrl-K, Ctrl-Enter Vloží výběr na konec interaktivní vyrovnávací paměti, ponechá stříšku na k
  Ctrl-E, Ctrl-Enter Vloží a provede výběr před jakýmkoli čekajícím vstupem v interaktivní vyrovnávací paměti
  Ctrl-A          Prvním stisknutím vyberete odeslání obsahující kurzor. Druhým stisknutím vyberete
Příkazy REPL:
  #cls, #clear     Vymaže obsah okna editoru, ale historii a kontext provádění ponechá beze změny.
  #help           Zobrazí nápovědu pro zadaný příkaz nebo (pokud není žádný zadaný) pro všechny dostupné příkazy.
  #reset          Obnoví původní stav spouštěcího prostředí, zachová historii.
Direktivy skriptu:
  #r              Přidá odkazy metadat k určenému sestavení a všem jeho závislostem, třeba #load "myScript.csx".
  #load          Načte zadaný soubor skriptu a spustí ho, třeba #load "myScript.csx".
>

```

Co zde není, je použití středníku na konci. Když jej dáte, bere se to jako příkaz, když ne, bere se to jako výraz, jehož hodnota se vypíše.

## Příklad použití - studium seznamů

Chcete studovat třeba seznamy?



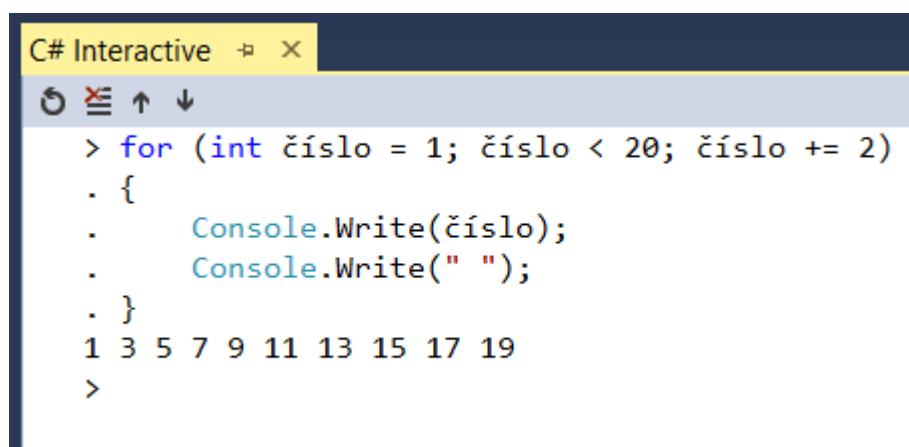
The screenshot shows the C# Interactive window in Microsoft Visual Studio. The window title is "C# Interactive - Microsoft Visual Studio". The menu bar includes "Soubor", "Upravit", "Zobrazit", "Ladit", "Tým", "Nástroje", "Test", and "Analyzovat". The toolbar contains various icons for navigation and execution. The main area shows the following code and its output:

```
> List<string> jména = new List<string>();
> jména
List<string>(0) { }
> jména.Count
0
> jména.Add("Jana");
> jména.Add("Karel");
> jména
List<string>(2) { "Jana", "Karel" }
> jména.RemoveAt(0);
> jména
List<string>(1) { "Karel" }
>
```

Není to krásné? Vždy hned vidíte výsledek...

## Další příklad - trénink cyklů

Chcete se studenty trénovat cykly? Ať vypíší všechna lichá čísla do dvacítky:



The screenshot shows the C# Interactive window in Microsoft Visual Studio. The window title is "C# Interactive". The menu bar includes "Soubor", "Upravit", "Zobrazit", "Ladit", "Tým", "Nástroje", "Test", and "Analyzovat". The toolbar contains various icons for navigation and execution. The main area shows the following code and its output:

```
> for (int číslo = 1; číslo < 20; číslo += 2)
. {
.     Console.Write(čísló);
.     Console.Write(" ");
. }
1 3 5 7 9 11 13 15 17 19
>
```

## Pozor! Podmínky...

Když zapisujete `if-else` s větvemi neuzavřenými mezi složené závorky, **je třeba po prvním středníku** (konec větve `if`) **stisknout *Shift+Enter***. Obyčejný *Enter* by celý příkaz hned odeslal, nečekal by na to, že chcete pokračovat.

```
C# Interactive
> int číslo = 7;
> if (číslo % 2 == 0)
.     Console.WriteLine("Číslo je sudé");
. else
.     Console.WriteLine("Číslo je liché");
Číslo je liché
>
```

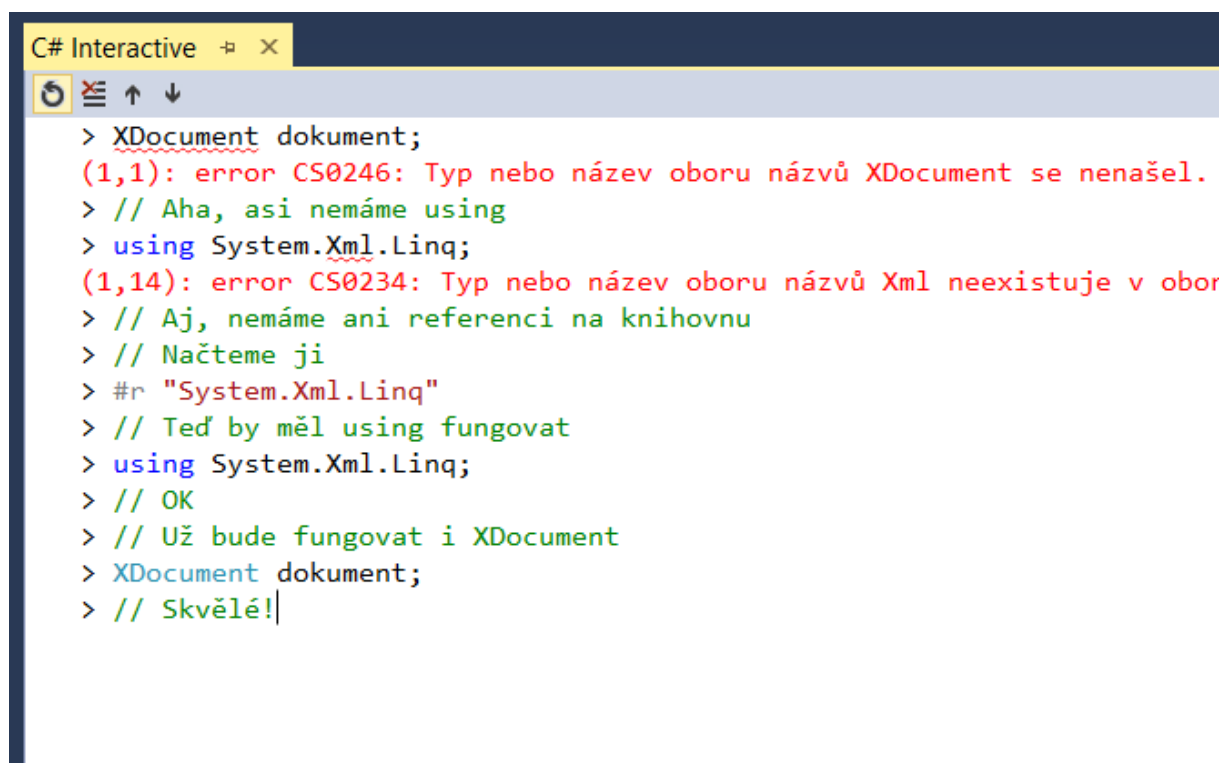
## Práce se soubory

Učíte se pracovat se soubory? Na ploše jsem si udělal textáček `pokus.txt` s obsahem v UTF-8. Čtení souboru i s mým překlepem:

```
C# Interactive
> string plocha = Environment.GetFolderPath(Environment.SpecialFolder.Desktop);
> plocha
"C:\\Users\\vystavel\\Desktop"
> string jménoSouboru = Path.Combine(plocha, "pokus.txt");
> jménoSouboru
"C:\\Users\\vystavel\\Desktop\\pokus.txt"
> string[] řádky = File.ReadLines(jménoSouboru);
(1,18): error CS0266: Typ System.Collections.Generic.IEnumerable<string> nejde implicitně
> string[] řádky = File.ReadAllLines(jménoSouboru);
> řádky
string[2] { "Příliš žluťoučký kůň", "úpěl ďábelské ódy." }
>
```

## Jmenné prostory a knihovny

Učíte se pracovat s XML soubory. Zjistíte, že potřebujete odkazy na příslušnou knihovnu a jmenný prostor:



```
C# Interactive
> XDocument dokument;
(1,1): error CS0246: Typ nebo název oboru názvů XDocument se nenašel.
> // Aha, asi nemáme using
> using System.Xml.Linq;
(1,14): error CS0234: Typ nebo název oboru názvů Xml neexistuje v obor
> // Aj, nemáme ani referenci na knihovnu
> // Načteme ji
> #r "System.Xml.Linq"
> // Teď by měl using fungovat
> using System.Xml.Linq;
> // OK
> // Už bude fungovat i XDocument
> XDocument dokument;
> // Skvělé!
```

## Závěr

Je to mocnější nástroj nežli Příkazové podokno (Immediate window).

Líbí se vám? Pohrajte si...

*Radek Vystavěl, 12. února 2016*

*Pokud Vám Zpravodaje moderníProgramování připadají užitečné, doporučte jejich odběr svým známým. Mohou se přihlásit na webu [www.moderniProgramovani.cz](http://www.moderniProgramovani.cz).*