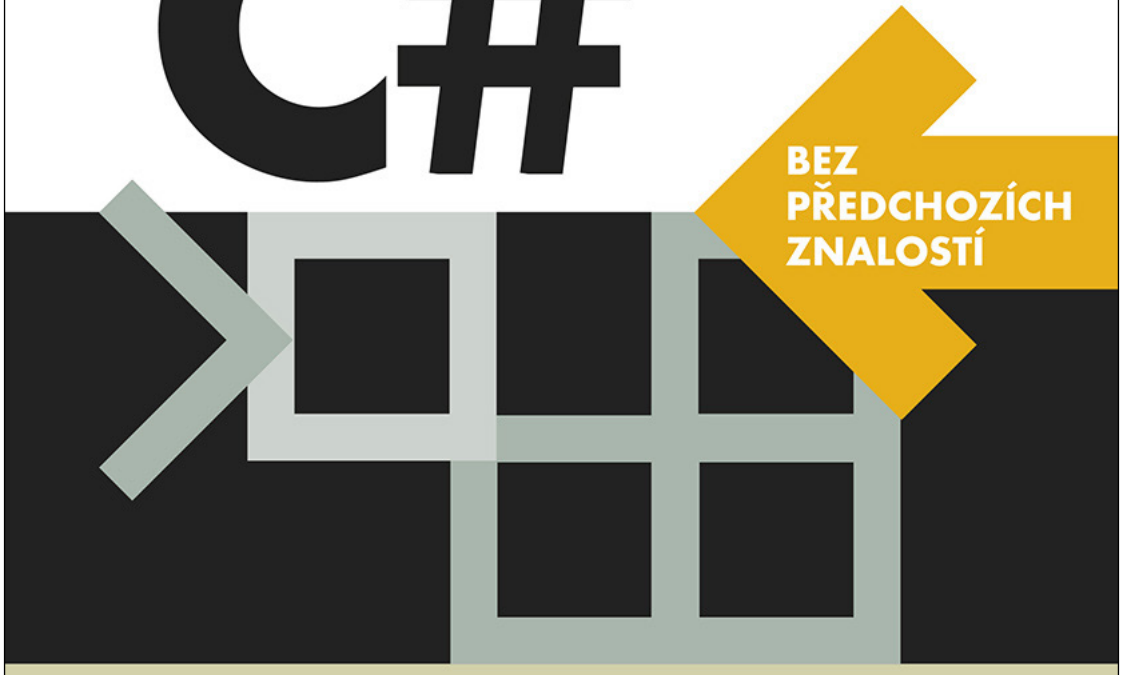
Pavel Bory

C#



BEZ PŘEDCHOZÍCH ZNALOSTÍ

Datové typy, operátory, řídicí struktury

Základy objektově orientovaného programování

Pole, řetězce, práce se soubory

Ladění programu, zpracování výjimek

computer press

C# bez předchozích znalostí

Vyšlo také v tištěné verzi

Objednat můžete na
www.cpress.cz
www.albatrosmedia.cz

Pavel Bory
C# bez předchozích znalostí – e-kniha

Obsah

O autorovi

Pavel Bory je seniorním vývojářem a řadu let pracuje hlavně na projektech společnosti Unicorn, a.s., především v oblasti financí. Kromě samotného vývoje softwaru se věnuje školení a výuce programování, zejména nad platformou .NET na vysoké škole Unicorn College, s.r.o. Tato kniha je jeho první monografií.

Poděkování

Rád bych poděkoval všem blízkým, bez jejichž podpory by tato kniha nevznikla. Dále bych rád poděkoval všem spolupracovníkům, od kterých jsem měl možnost se učit, a v neposlední řadě chci poděkovat svým učitelům, z nichž jsou mi někteří dodnes vzorem. Musím poděkovat i redaktorovi nakladatelství Computer Press panu Martinovi Herodkovi za jeho profesionální přístup a podporu během psaní této knihy.

Úvod

S pomocí této knihy se můžete sami naučit základům programovacího jazyka C# a objektově orientovaného programování. Kniha je koncipována tak, že je probíraná látka hojně ilustrována na příkladech, ke kterým jsou uvedeny jejich zdrojové kódy a výstupy programů. Zdrojové kódy jsou pro lepší srozumitelnost doprovázeny komentáři a u složitějších příkladů je zdrojový kód tvořen postupně a průběžně je jeho tvorba rozebírána. Na závěr každé kapitoly je uvedena sada kontrolních otázek pro ověření získaných znalostí a zároveň jsou zde uvedeny řešené příklady. Kromě kontrolních otázek a řešených příkladů jsou na závěr každé kapitoly uvedeny příklady k samostatnému procvičení.

Komu je tato kniha určena

Tato kniha nepředpokládá žádné znalosti programování. Je tedy určena každému, kdo má zájem se naučit základy programování v moderním objektově orientovaném jazyce C#. Kniha může posloužit jak čtenáři, který chce získat pouze základní přehled o programování, tak i čtenáři, který se chce programování věnovat dlouhodobě a hledá vhodný výchozí bod pro začátek studia.

Zpětná vazba od čtenářů

Nakladatelství a vydavatelství Computer Press, které pro vás tuto knihu připravilo, stojí o zpětnou vazbu a bude na vaše podněty a dotazy reagovat. Můžete se obrátit na následující adresy:

Computer Press
Albatros Media a.s., pobočka Brno
IBC
Příkop 4
602 00 Brno

nebo

sefredaktor.pc@albatrosmedia.cz

Computer Press neposkytuje rady ani jakýkoli servis pro aplikace třetích stran. Pokud budete mít dotaz k programu, obraťte se prosím na jeho tvůrce.

Zdrojové kódy ke knize

Z adresy *http://knihy.cpress.cz/K2202* si po klepnutí na odkaz Soubory ke stažení můžete přímo stáhnout archiv s ukázkovými kódy.

Errata

Přestože jsme udělali maximum pro to, abychom zajistili přesnost a správnost obsahu, chybám se úplně vyhnout nelze. Pokud v některé z našich knih nějakou najdete, ať už v textu nebo v kódu, budeme rádi, pokud nám ji oznámíte.

Veškerá existující errata zobrazíte na adrese *http://knihy.cpress.cz/K2202* po klepnutí na odkaz Soubory ke stažení. (Nejsou-li žádná errata zatím k dispozici, není odkaz Soubory ke stažení dostupný.)

KAPITOLA 1

První program

V této kapitole:

- Instalace potřebného softwaru
- Náš první program v jazyce C#
- Co je to vlastně program a programovací jazyk?
- Základy práce s Microsoft Visual Studiem
- Kontrolní otázky
- Cvičení

V této kapitole se dozvíte, jaký software budeme v rámci studia programování pomocí této knihy používat, jak jej získat, nainstalovat a jak s ním pracovat. Dozvíte se, co je to program, programovací jazyk a vytvoříte si svůj první program v jazyce C#.

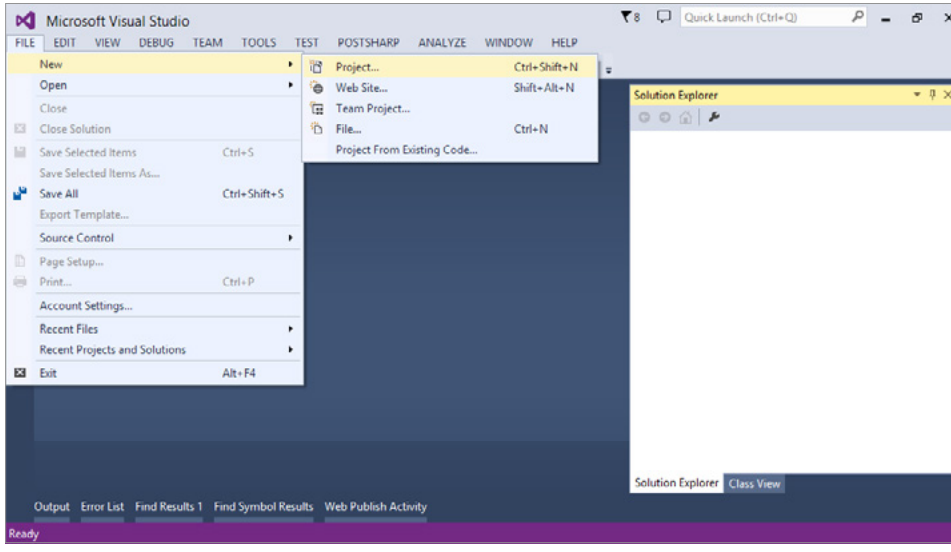
Instalace potřebného softwaru

Než začnete se studiem programování pomocí této knihy, je nezbytné si nainstalovat vývojové prostředí Microsoft Visual Studio, pomocí kterého budeme naše programy vytvářet. V této knize budeme používat verzi Microsoft Visual Studio 2015 Community Edition, která je k dispozici zdarma ke stažení zde: *https://www.visualstudio.com/products/visual-studio-community-vs*.

Vývojové prostředí nainstalujte podle pokynů uvedených na webových stránkách a v instalačním průvodci vývojového prostředí Microsoft Visual Studio.

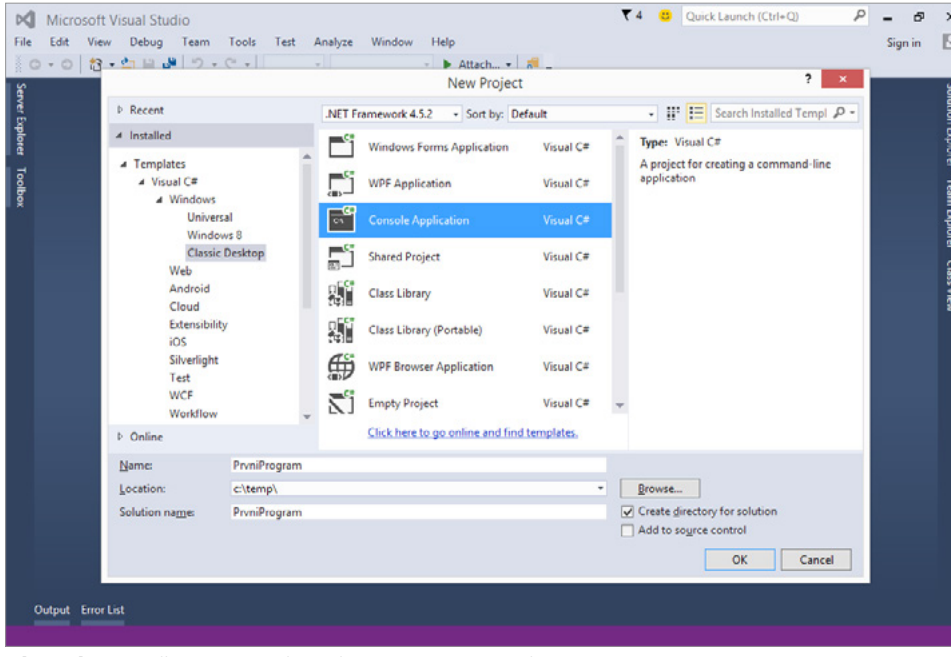
Náš první program v jazyce C#

Pokud máte nainstalované vývojové prostředí Microsoft Visual Studio, se můžete pustit do vytvoření svého prvního programu v jazyce C#. Spusťte vývojové prostředí Microsoft Visual Studio. Zobrazí se vám úvodní obrazovka, na které vyberte z hlavního menu možnost **File → New → Project**.



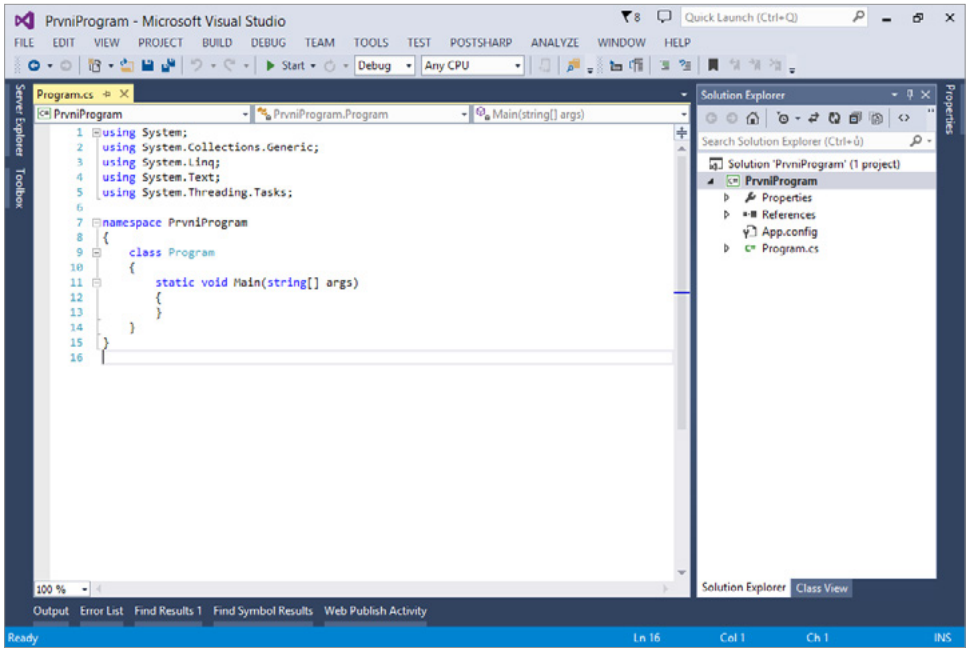
Obrázek 1.1 Vytvoření nového projektu

Zobrazí se poměrně pestrá nabídka, protože Microsoft Visual Studio umožňuje vytvářet celou řadu různých typů programů. Vyberte projekt typu **Visual C# → Windows → Classic Desktop → Console Application**. Do pole **Name** vyplňte název projektu `PrvniProgram`. V poli **Location** vyberte umístění, kde na disku bude uložen vytvořený projekt.



Obrázek 1.2 Volba typu projektu, jeho pojmenování a výběr umístění

Jakmile budete mít vyplněno, stiskněte tlačítko **OK** a Microsoft Visual Studio vytvoří nový projekt.



Obrázek 1.3 Výchozí stav po vytvoření nového projektu

Možná jste očekávali, že bude projekt zcela prázdný, a on přitom obsahuje pro nás zatím neznámý zdrojový kód a několik souborů. To je dáno tím, že na základě vybraného typu projektu dojde k zjednodušeně řečeno vygenerování určitého výchozího stavu, který je pro daný typ programu typický. Zatím do onoho výchozího stavu projektu nezasahujme, ale nebojte se, postupně se při čtení této knihy s významem jednotlivých příkazů seznámíte.

Než se pustíme do vytvoření prvního programu, pojďme si vysvětlit dva pojmy: `Project` a `Solution`.

`Project` obsahuje zdrojové kódy programu, které jsou umístěny v jednom či více souborech. `Solution` může obsahovat jeden či více projektů.

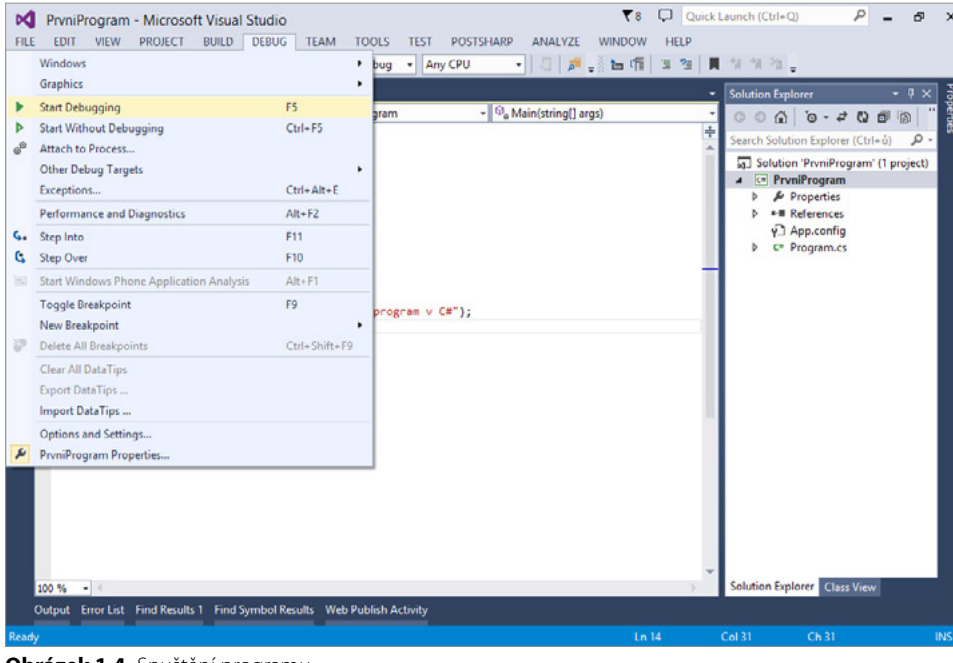
My jsme tedy vytvořili jednu `Solution`, která obsahuje právě jeden `Project`, ve kterém napíšete svůj první program v jazyce C#. V rámci ukázkových příkladů k této knize se můžete setkat s tím, že pro jednu kapitolu existuje jedna `Solution`, která obsahuje více projektů, přičemž jednotlivé projekty obsahují jednotlivé ukázkové programy. Pojďte vytvořit svůj první program v jazyce C#! Doplňte následující dva řádky do souboru `Program.cs`,

```
Console.WriteLine("Muj prvni program v C#");
Console.ReadKey();
```

tak aby přesně odpovídal následujícímu zápisu:

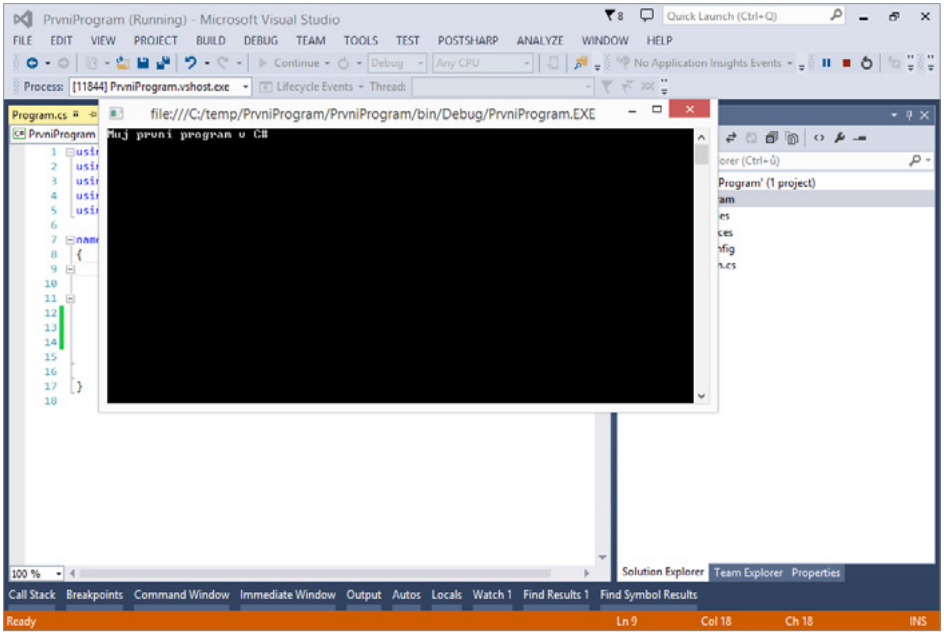
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace PrvniProgram
{
  class Program
  {
    static void Main(string[] args)
    {
      Console.WriteLine("Muj prvni program v C#");
      Console.ReadKey();
    }
  }
}
```

Nyní program spusťte pomocí možnosti **Start Debugging**, která se nachází v menu **Debug**.



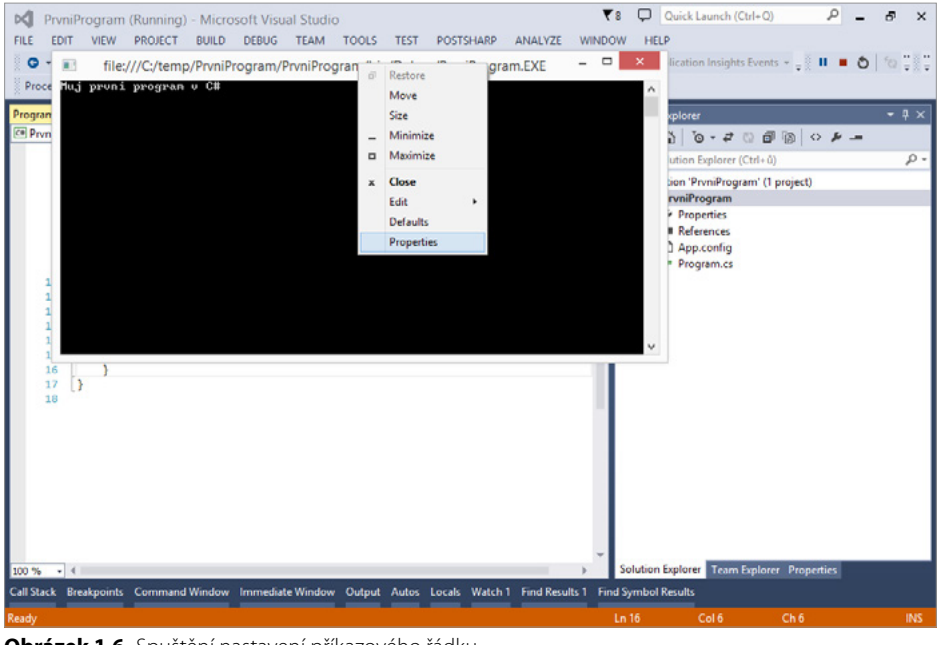
Obrázek 1.4 Spuštění programu

Zobrazí se vám příkazový řádek, ve kterém program vypíše zadaný text, který jste vložili do souboru `Program.cs`.



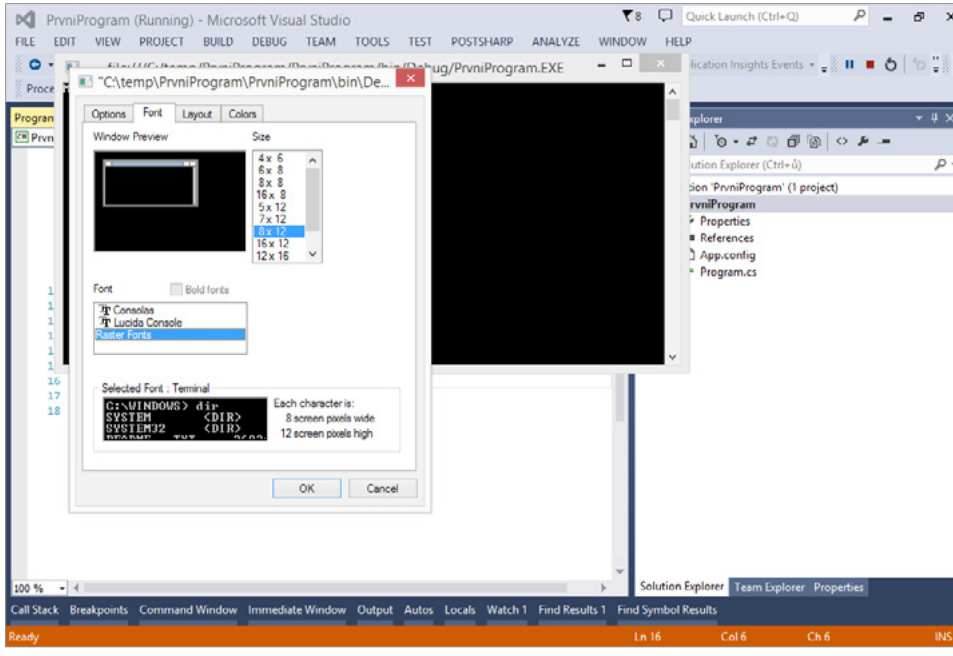
Obrázek 1.5 Příkazový řádek s výstupem vašeho prvního programu

Je možné, že se vám bude velikost písmen v příkazovém řádku zdát příliš malá. Proto si nyní ukažme jak ji zvětšit. Klikněte pravým tlačítkem do vrchní části rámu příkazového řádku.



Obrázek 1.6 Spuštění nastavení příkazového řádku

Z kontextové nabídky vyberte možnost **Properties**. Zobrazí se vám následující obrazovka s nastavením příkazového řádku.



Obrázek 1.7 Nastavení fontu příkazového řádku

Zde můžete změnit velikost písma v příkazovém řádku. Pokud chcete, prozkoumejte i jiné možnosti nastavení příkazového řádku a přizpůsobte si jeho zobrazení tak, aby se vám s ním dobře pracovalo.

Poznámka: Program můžeme spouštět jak pomocí **Debug → Start Debugging**, tak i pomocí stisknutí klávesy **F5**.

Co je to vlastně program a programovací jazyk?

Abychom si mohli vysvětlit, co je to vlastně program, je potřeba si nejprve definovat pojem algoritmus. Pro tento pojem neexistuje jednoznačná, dokonalá a všemi uznávaná definice. My si algoritmus definujme jako určitý návod, postup jak řešit určitý problém. Tento postup musí být zejména jednoznačný a nesmí umožňovat více výkladů. Příkladem neprogramátorského algoritmu může být recept v kuchařce. Ten přesně definuje postup přípravy pokrmu a měl by být jednoznačný a neumožňovat více způsobů výkladu.

Když jsme si přiblížili pojem algoritmus, můžeme si počítačový program popsat jako zápis algoritmu ve zvoleném jazyce, který je srozumitelný počítači. Tyto jazyky se nazývají programovací jazyky a slouží právě k dorozumění mezi člověkem (programátorem) a počítačem. A vy se pro dorozumění s počítačem naučíte právě programovací jazyk C#.

Ve spojitosti s programovacím jazykem C# se často setkáte s pojmem *.NET Framework*. .NET Framework je softwarová platforma poskytující širokou škálu prostředků pro programy. Její popis by zcela jistě vydal na celou řadu knih a není cílem této knihy ji detailně popisovat. Prostředky z této platformy budeme při programování v jazyce C# používat, a dokonce jste je již použili, pokud jste si vytvořili předchozí program. Použili jsme její prostředky např. pro výpis na konzoli. Kromě programovacího jazyka C# se tedy seznámíte i s řadou základních prostředků platformy .NET.

Řekli jsme si, že programovací jazyk je jazyk srozumitelný počítači. K tomuto je potřeba ještě vysvětlit, že než program spustíte, je potřeba spustit tzv. **Build**. Ten zajistí ověření, že program neobsahuje formální chyby, a vytvoří spustitelný soubor. Možná si teď říkáte, že když jsme spouštěli náš první program, **Build** jsme neprováděli. Tím, že jsme zvolili možnost **Start Debugging**, se provedly v podstatě dva kroky zároveň, a to **Build** a následné spuštění programu.

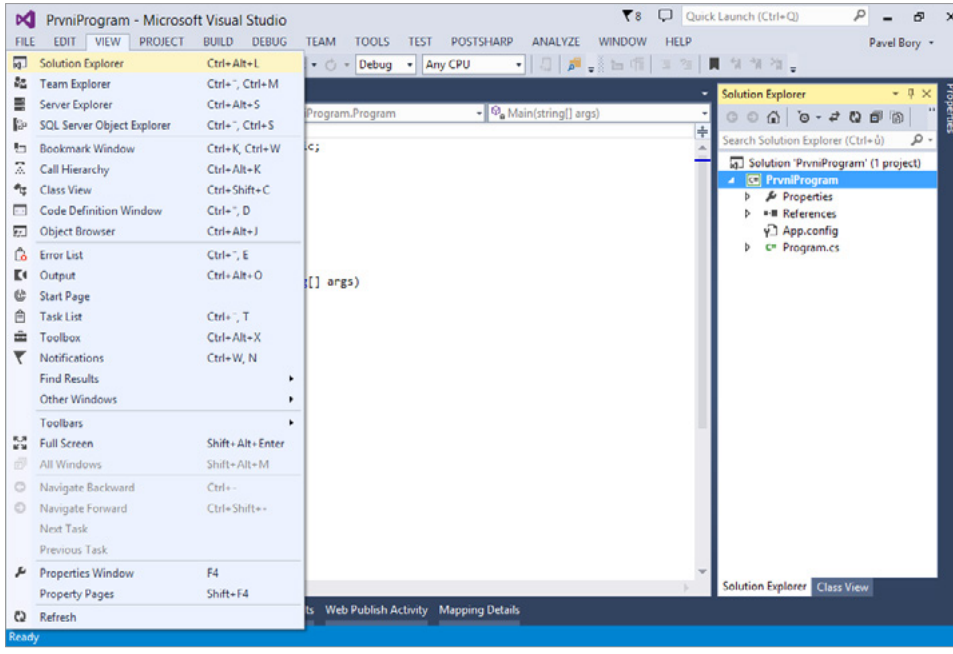
Ve skutečnosti je tato problematika podstatně komplikovanější, ale cílem této knihy není zabíhat do přílišných detailů. Naším cílem je, abyste se naučili základy programování a byli schopni na těchto zejména praktických základech stavět.

Základy práce s Microsoft Visual Studiem

V této kapitole se seznámíte se základy ovládání Microsoft Visual Studia, abyste mohli pohodlně pracovat s touto knihou, zkoušet si ukázkové programy, a zejména tvořit programy vlastní. Většina prvků Microsoft Visual Studia, které budeme používat, je reprezentována okny, která lze zavírat, přesouvat a minimalizovat. Zkrátka s nimi lze pracovat, jako jste zvyklí pracovat s okny v prostředí Microsoft Windows.

Tip: Každý ovládací prvek (okno), který zde budeme popisovat, lze otevřít pomocí menu **View**.

Na tuto kapitolu budeme v průběhu studia v dalších kapitolách volně navazovat a znalosti práce s Visual Studiem postupně dále rozšiřovat. První, co si ukážeme, je postup, jak se orientovat v `Solution`, projektech a souborech těchto projektů. K tomuto slouží tzv. **Solution Explorer**.



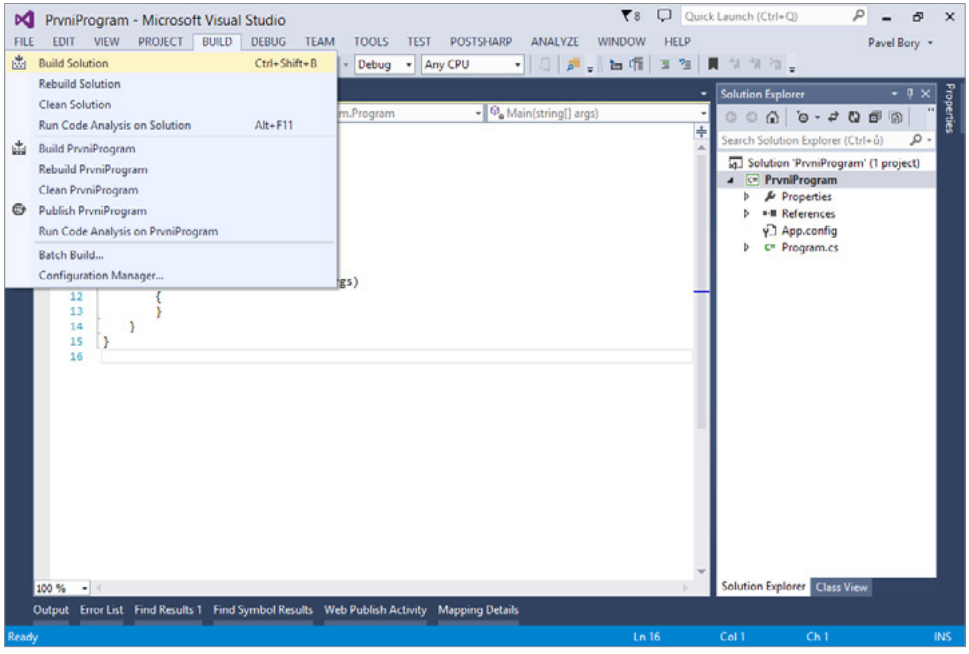
Obrázek 1.8 Solution Explorer

Toto okno vám ukáže, které projekty se v `Solution` nachází, a můžete zde otevírat jednotlivé soubory obsahující zdrojové kódy a ty editovat. Zkuste si v **Solution Explorer** otevřít soubor `Program.cs`, do kterého jsme napsali náš první program.

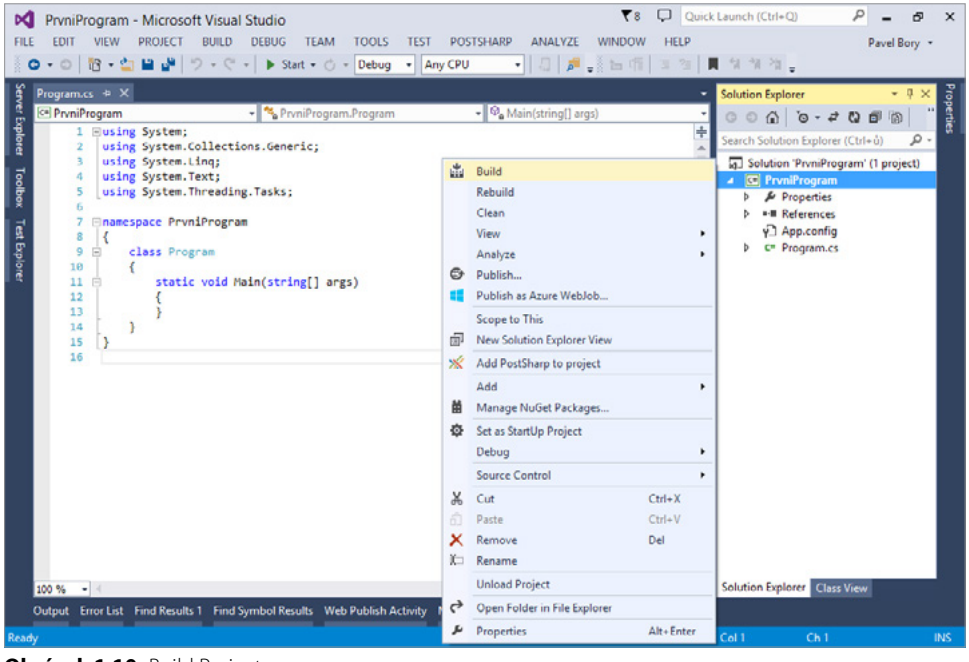
V předchozí kapitole jsme hovořili o tzv. **Build**, který vytvoří spustitelný soubor programu. Máme v základu dvě možnosti jak **Build** spustit. První možnost je použít **Build Solution** z menu **Build**.

Víme, že `Solution` může obsahovat více projektů. Z každého projektu obsaženého v `Solution` se vytvoří jeden spustitelný soubor s příponou `.exe` a několik dalších souborů, které nejsou nutně zapotřebí pro samotné spuštění programu, a nebudeme se jimi proto v této knize zabývat. Pokud máte vytvořenu `Solution` obsahující jeden projekt z kapitoly Náš první program v jazyce C# v umístění `C:\temp\PrvniProgram`, naleznete spustitelný `.exe` soubor programu zde: `C:\temp\PrvniProgram\bin\Debug\PrvniProgram.exe`. Zkuste si tento soubor najít a program spustit. Měla by se vám zobrazit konzole, na kterou se vypíše text stejně, jako když jsme program spouštěli pomocí **Debug → Start Debugging** přímo z Microsoft Visual Studia.

Další možností je provést **Build** nad konkrétním projektem. Můžeme jej spustit buď z menu **Build** pomocí příkazu **Build PrvniProgram** (nebo jiný název projektu), nebo z kontextového menu nad projektem v **Solution Explorer**. Klikněte pravým tlačítkem na název projektu v **Solution Explorer** a vyberte možnost **Build**.



Obrázek 1.9 Build Solution

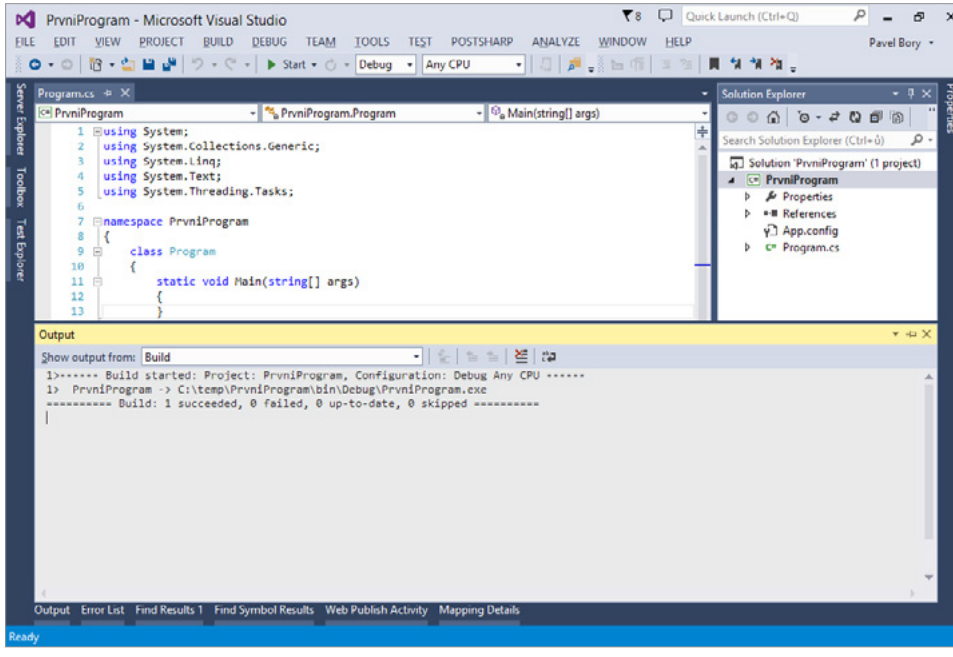


Obrázek 1.10 Build Project

Příkazem **Build** jsme vždy pouze vytvořili spustitelný soubor programu.

Při studiu a samostatném vytváření programů budete nejčastěji potřebovat programy rovnou spouštět z vývojového prostředí. Za tímto účelem můžete použít volbu **Debug → Start Debugging**, stejně jako když jsme spouštěli náš první program.

Někdy se může stát, že v programu uděláte chybu, která zamezí jeho spuštění. Např. se pokusíte použít neexistující příkaz, zapomenete někde napsat závorku apod. V tomto případě se spustitelný soubor programu nevytvoří, program se nespustí a Microsoft Visual Studio bude hlásit chybu. Otevřeme si další okno v nabídky **View → Output**, ve kterém budeme zpravidla sledovat, jak dopadl příkaz **Build**.



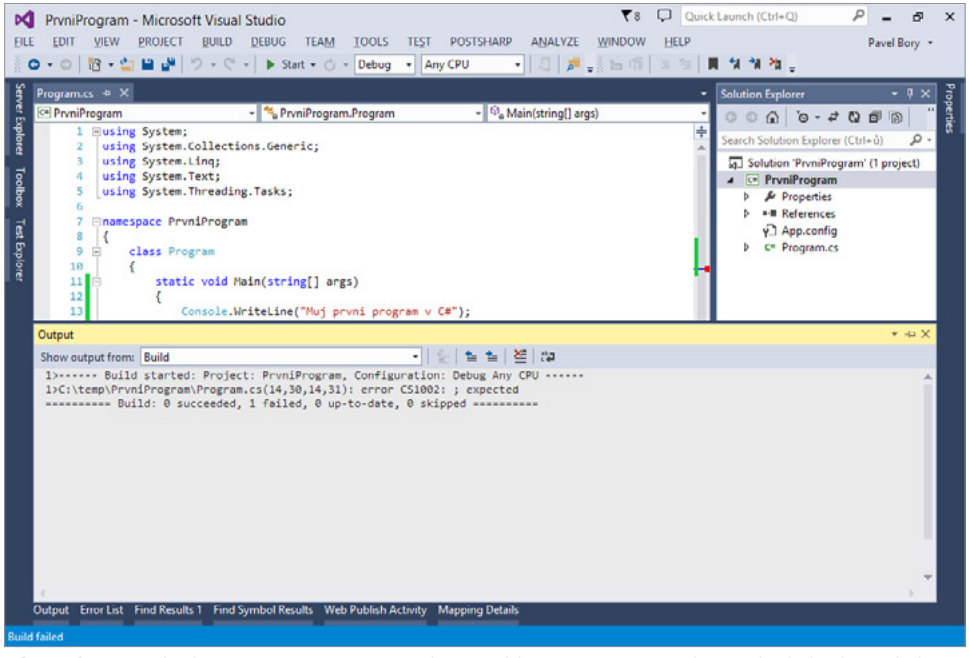
Obrázek 1.11 Ukázka výstupu po spuštění příkazu Build Project

V okně **Output** se po spuštění příkazu **Build** dozvíme, pro kolik projektů dopadl **Build** úspěšně (succeeded), pro kolik neúspěšně (failed) a pro kolik se nespustil, protože již byl pro ně spuštěn a nebyla v nich provedena žádná úprava (up-to-date). Zkusme se záměrně dopustit chyby v programu. Odstraňte např. středník za příkazem `Console.ReadKey()`. Záměrně chybně zapsaný program by mohl vypadat takto.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
```

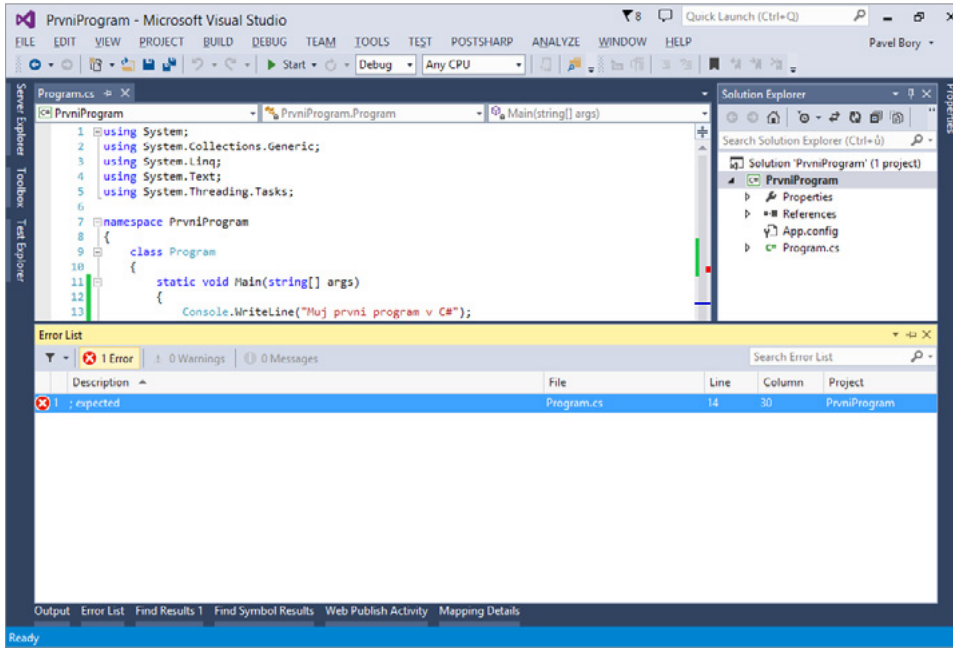
```
namespace PrvniProgram
{
  class Program
  {
    static void Main(string[] args)
    {
      Console.WriteLine("Muj prvni program v C#");
      Console.ReadKey()
    }
  }
}
```

Spusťte příkaz **Build** pro projekt a podívejte se do okna **Output**, ve kterém se dozvíme, že byl pro projekt neúspěšný.



Obrázek 1.12 Ukázka výstupu po spuštění příkazu Build Project, přičemž zdrojový kód obsahuje chybu

Pokud se v programu vyskytuje chyba bránící jeho spuštění, je pro nás podstatně zajímavější okno **Error List**, které je opět možné spustit z nabídky **View**. V tomto okně se dozvíte o tom, kde se v programu vyskytují chyby.



Obrázek 1.13 Error List

Je zde uveden tabulkový výpis, kde každý řádek popisuje jednu chybu. Pro každou chybu vás bude v tomto výpisu především zajímat její popis, soubor, ve kterém se vyskytuje, a řádek, na kterém se vyskytuje. Dvojklikem na řádek s chybou vás Visual Studio nasměruje na umístění příslušné chyby. Je zapotřebí říci, že v některých případech nemusí být na první pohled zřejmé, v čem chyba spočívá. Nicméně postupně v tomto ohledu získáte zkušenosti a odhalení příčin chyb pro vás bude jednodušší. V rámci ukázek programů z této knihy máte vždy možnost si otevřít přiložené ukázkové projekty, porovnat je s vlastním řešením, nalézt odlišnosti a odvodit, v čem chyba spočívá. Při vašem studiu může být rovněž přínosné nalézt řešení obdobného problému, který budete řešit, a inspirovat se z něj nebo poznat, v čem jste se dopustili chyby.

Kontrolní otázky

1. K čemu slouží vývojové prostředí Microsoft Visual Studio?
2. Co je to `Project` a `Solution` a jaký je mezi nimi vztah?
3. Co je to program a programovací jazyk?
4. Jakým příkazem vytvoříme spustitelný soubor z programu, aniž bychom jej rovnou spustili? Kde tento soubor naleznete?
5. Jakým příkazem spustíme program přímo z Microsoft Visual Studia?

6. Pokud se program nepodaří kvůli chybám spustit, kde nalezneme informace o výskytu chyb?

Cvičení

1. Vyzkoušejte si založení nové `Solution` a `Project`. Jakmile budete mít projekt vytvořen, zkuste si doplnit kód v souboru `Program.cs`, aby pozdravil uživatele slovy „Ahoj clovece“ obdobně, jako jste učinili ve vašem prvním programu v jazyce C#.
2. Zkuste si záměrně zanést do programu chybu, např. tím že odstraníte středník nebo závorky apod. Následně zkuste program spustit a pomocí **Error List** nalézt informace o chybě a jejím umístění.

KAPITOLA 2

Datové typy a proměnné

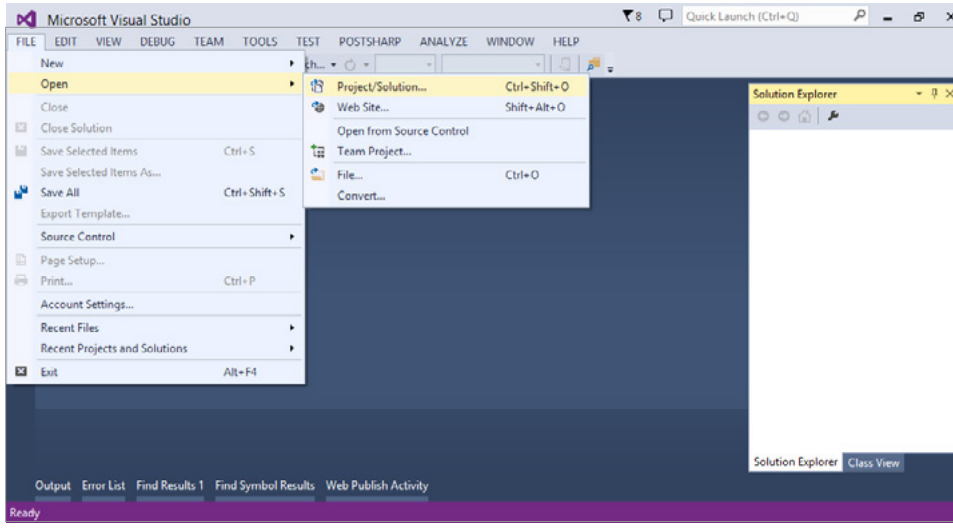
V této kapitole:

- Otevření existujícího projektu
- Základní struktura programu
- Komentáře zdrojového kódu
- Proměnné
- Datové typy
- Konstanty
- Základní operace s čísly
- Výpis na příkazový řádek
- Načtení vstupu od uživatele
- Konverze řetězce na číslo
- Konverze čísla na řetězec
- Intelisense ve Visual Studiu
- Kontrolní otázky
- Řešená cvičení
- Cvičení

V této kapitole si rozšíříte své znalosti práce s vývojovým prostředím Microsoft Visual Studio. Naučíte se, jak otevřít ukázkové příklady, které doprovázejí tuto knihu. Poté se naučíte, jak si v programu uchovávat data, se kterými program může pracovat. Nemalé množství programů potřebuje pro svou činnost komunikovat s jejich uživatelem, proto se dozvíte, jak může program uživateli zobrazovat textové zprávy v příkazovém řádku, a zároveň se naučíte, jak do programu předat data, která uživatel do příkazového řádku zapíše.

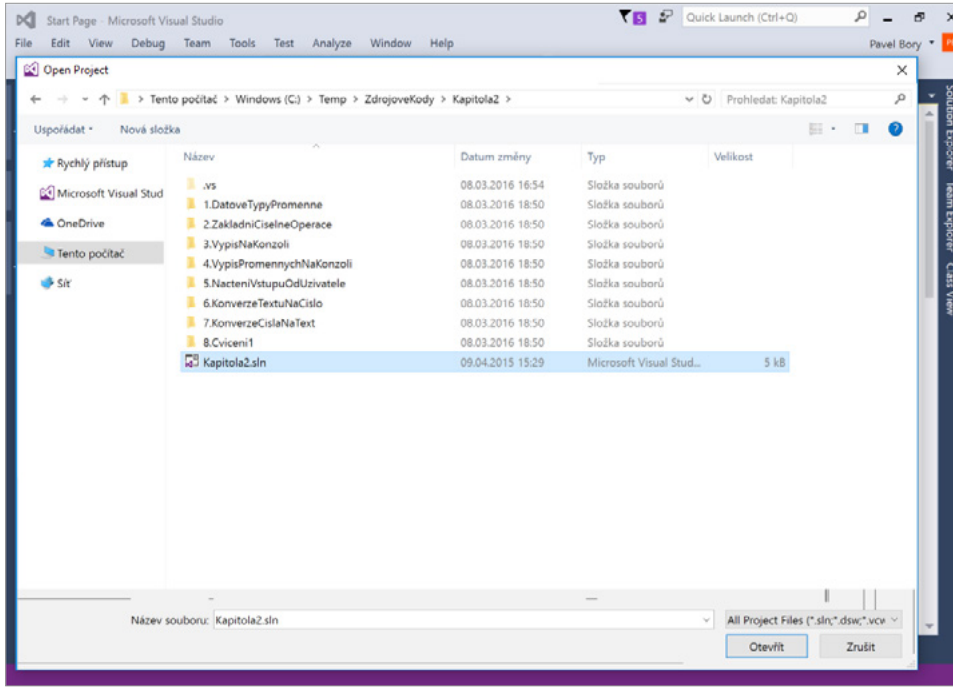
Otevření existujícího projektu

Nyní i v následujících kapitolách budeme při výkladu vycházet z ukázkových projektů. Proto se naučíte, jak nějaký již existující projekt otevřít. Spusťte Visual Studio. Zobrazí se vám úvodní obrazovka, na které vyberte z hlavního menu možnost **File → Open → Project/Solution**.



Obrázek 2.1 Otevření existujícího Project/Solution – krok 1

Zobrazí se vám standardní okno pro výběr souboru. Pomocí něj najděte složku, ve které máte zdrojové kódy k této knize, a otevřete složku `Kapitola2`. V této složce vyberte soubor `Kapitola2.sln` a stiskněte tlačítko **Open**.



Obrázek 2.2 Otevření existujícího Project/Solution – krok 2