Stručný obsah

Část II: Visual Studio

Část III: Knihovny bázových tříd

Část IV: Data

Obsah

Část I: Jazyk C#

Část III: Knihovny bázových tříd

17. Sestavení 543

Část IV: Data

25. Práce se soubory a systémovým registrem 879

26. Přístup k datům 935

Úvod

Kdybychom označili jazyk C# a související prostředí .NET Framework za nejdůležitější novou technologii pro vývojáře za posledních mnoho let, rozhodně bychom nepřeháněli. Návrh technologie .NET poskytuje nové prostředí, ve kterém lze vyvíjet téměř libovolné aplikace určené pro systém Windows, a C# je nový programovací jazyk, který byl navržen zejména pro spolupráci s technologií .NET. Pomocí jazyka C# můžete například napsat dynamickou webovou stránku, webovou službu XML, komponentu distribuované aplikace, komponentu pro přístup k databázi, klasickou aplikaci pro pracovní plochu systému Windows, nebo dokonce novou inteligentní klientskou aplikaci s podporou režimů online i offline. Tato kniha se zabývá platformou .NET Framework 3.5. Programujete-li s verzí 1.0, 1.1, 2.0 nebo i 3.0, je možné, že některé části knihy nebudete moci využít. Pokusíme se upozorňovat na funkce, které se objevily až v platformě .NET Framework 3.5.

Nenechte se zmást označením .NET. NET jako součást názvu má zdůraznit přesvědčení společnosti Microsoft, že cestu vpřed představují distribuované aplikace, kde je výpočetní zpracování rozděleno mezi klienta a server. Jazyk C# však neslouží pouze k psaní internetových nebo síťových aplikací. Nabízí možnosti pro tvorbu téměř libovolných typů programů nebo komponent, které můžete v systému Windows požadovat. Jazyk C# spolu s technologií .NET mají zásadně změnit způsob psaní programů a usnadnit programování v systému Windows jako nikdy předtím.

To je poměrně odvážné tvrzení, které je nutné obhájit. Všichni nakonec víme, jak rychle se počítačové technologie mění. Každý rok přichází společnost Microsoft s novými programy, programovacími nástroji nebo verzemi systému Windows a prohlašuje přitom, že budou mimořádným přínosem pro vývojáře. V čem jsou tedy technologie .NET a jazyk C# jiné?

Důležitost technologie .NET a jazyka C#

Abychom porozuměli významu technologie .NET, je vhodné si připomenout podstatu několika technologií pro systém Windows, které se objevily v posledních asi deseti letech. Na první pohled se sice mohou značně lišit, ale všechny operační systémy Windows od verze Windows 3.1 (uvedena na trh roku 1992) až po Windows Server 2008 mají ve svém jádru stejné, důvěrně známé rozhraní Windows API. S novými verzemi systému Windows přibývalo v rozhraní API mnoho nových funkcí, ale jednalo se o vývoj a rozšiřování rozhraní, nikoli o jeho nahrazení.

Totéž lze říci o mnoha technologiích a platformách, pomocí nichž jsme vyvíjeli software pro systém Windows. Například model **COM** (**Component Object Model**) vycházel ze služeb **OLE** (**Object Linking and Embedding – propojování a vkládání objektů**). V té době se do značné míry jednalo pouze o metodu, jak propojit různé typy dokumentů sady Office, abyste mohli například vložit malou tabulku v Excelu do dokumentu ve Wordu. Dalším vývojem vznikl model COM, **DCOM** (**Distributed COM**) a nakonec COM+: pokročilá technologie, která se stala základem komunikace téměř všech komponent, stejně jako implementace transakcí, služeb zasílání zpráv a sdružování objektů.

Společnost Microsoft zvolila tento evoluční přístup k softwaru samozřejmě proto, že jí záleží na zpětné kompatibilitě. V průběhu let vyvinuli jiní dodavatelé rozsáhlou programovou základnu pro systém Windows. Kdyby v každé nové verzi společnost Microsoft zavedla novou technologii, která by neumožnila používat existující programy, nedosáhl by její systém takového úspěchu.

Zpětná kompatibilita sice byla klíčovou vlastností technologií Windows a jednou ze silných stránek této platformy, avšak má i velkou nevýhodu. Pokaždé, když se nějaká technologie vyvíjí a získává nové funkce, postupně se přitom komplikuje.

Bylo jasné, že něco se musí změnit. Společnost Microsoft nemohla věčně rozšiřovat stejné vývojové nástroje a jazyky a neustále zvyšovat jejich složitost, aby uspokojila konfliktní požadavky podpory nejnovějšího hardwaru a udržování zpětné kompatibility s technologiemi, které se používaly v době, kdy se systém Windows začátkem 90. let poprvé masově rozšířil. Chcete-li přijít s jednoduchou, ale přesto pokročilou sadou jazyků, prostředí a vývojových nástrojů, které vývojářům umožní snadnou tvorbu špičkových programů, musíte dříve nebo později začít s čistým stolem.

Přesně takový nový začátek představuje jazyk C# a technologie .NET. Zjednodušeně řečeno je .NET platforma (aplikační rozhraní, API) k programování pro Windows. Spolu s platformou .NET Framework se objevuje jazyk C#, který byl od základů navržen tak, aby spolupracoval s technologií .NET a zároveň využil veškerý pokrok ve vývojových prostředích a koncepcích objektově orientovaného programování, ke kterému došlo během posledních dvaceti let.

Na tomto místě bychom měli zdůraznit, že zpětná kompatibilita zůstala zachována. Existující programy budou fungovat i nadále a technologie .NET byla navržena tak, aby se stávajícím softwarem spolupracovala. Komunikace mezi softwarovými komponentami v systému Windows je v současnosti téměř výhradně založena na modelu COM. S ohledem na tento fakt může technologie .NET poskytnout obálky (wrapper) kolem existujících komponent COM, aby s nimi mohly komunikovat komponenty .NET.

Je pravda, že chcete-li psát kód pro technologii .NET, nemusíte se naučit jazyk C#. Společnost Microsoft rozšířila jazyk C++, nabídla jiný nový jazyk s názvem J# a zásadně přepracovala jazyk Visual Basic, který se změnil na výkonnější jazyk Visual Basic .NET. Tím umožnila uplatnit kód napsaný v libovolném z těchto jazyků v prostředí .NET. Tyto jazyky jsou však zatíženy dědictvím mnohaletého vývoje, neboť při jejich vývoji ještě neexistovaly dnešní technologie.

Tato kniha vás vybaví znalostmi programování v jazyku C# a současně poskytne potřebné základní informace o fungování architektury .NET. Nebudeme se zabývat pouze principy jazyka C#, ale uvedeme si také příklady aplikací, které používají různé související technologie, včetně přístupu k databázím, dynamických webových stránek, pokročilé grafiky a přístupu k adresářům. Předpokládáme pouze znalost minimálně jednoho jazyka vyšší úrovně, který se používá v systému Windows: může to být C++, Visual Basic nebo J++.

Výhody technologie .NET

Obecně jsme se zmínili o tom, jak je technologie .NET skvělá, ale zatím jsme příliš nevysvětlili, jak může vývojářům usnadnit život. V této části si stručně rozebereme některé vylepšené funkce této technologie.

- **Objektově orientované programování:** Platforma .NET Framework a jazyk C# jsou již od začátku kompletně založeny na objektově orientovaných principech.

- **Dobrý návrh:** Knihovna základních tříd je od základu navržena vysoce intuitivním způsobem.
- **Jazyková nezávislost:** V technologii .NET se všechny jazyky, tj. Visual Basic .NET, C#, J# a spravované (managed) C++, překládají do společného *zprostředkujícího jazyka* (Intermediate Language). To znamená, že jazyky mohou spolupracovat způsobem, který dříve nebyl k dispozici.
- **Lepší podpora dynamických webových stránek:** Prostředí ASP sice poskytovalo značnou pružnost, ale zároveň nebylo příliš efektivní, protože používalo interpretované skriptovací jazyky. Nedostatečný objektově orientovaný návrh také často vedl ke vzniku nepřehledného kódu ASP. Technologie .NET nabízí integrovanou podporu webových stránek pomocí nové technologie ASP.NET. V rámci ASP.NET je kód stránek překládán a lze jej psát v jazyce vysoké úrovně kompatibilním s technologií .NET, jako je např. C#, J# nebo Visual Basic 2008.
- **Účinný přístup k datům:** Sada komponent .NET souhrnně označovaná jako ADO.NET zajišťuje efektivní přístup k relačním databázím a různým zdrojům dat. Jsou také k dispozici komponenty, které poskytují přístup k systému souborů a k adresářům. Zejména je do technologie .NET integrována podpora jazyka XML, což umožňuje manipulaci s daty, která lze importovat nebo exportovat na jiné platformy než Windows.
- **Sdílení kódu:** Technologie .NET zcela mění způsob sdílení kódu mezi aplikacemi. Zavádí koncepci **sestavení** (assembly), která nahrazují tradiční knihovny DLL. Sestavení mají formální funkce pro správu verzí a je možné, aby vedle sebe existovaly různé verze sestavení.
- **Zlepšené zabezpečení:** Každé sestavení může také obsahovat integrované informace o zabezpečení, jež mohou přesně určovat, který uživatel nebo která kategorie uživatelů či procesů smí volat určité metody definovaných tříd. Tak získáváte velmi podrobnou kontrolu nad tím, jak lze zaváděná sestavení používat.
- **Instalace s nulovým dopadem:** Existují dva typy sestavení: sdílená (shared) a soukromá (private). Sdílená sestavení jsou společné knihovny dostupné všem programům, zatímco soukromá sestavení jsou určena pouze k použití s konkrétním softwarem. Soukromé sestavení je zcela soběstačné, takže se zjednodušuje proces instalace. Nevznikají žádné položky registru. Stačí umístit příslušné soubory do správné složky v systému souborů.
- **Podpora webových služeb:** Technologie .NET zahrnuje plně integrovanou podporu vývoje webových služeb, které lze vytvářet stejně snadno jako libovolný jiný typ aplikací.
- **Visual Studio 2008:** Pro technologii .NET se dodává vývojové prostředí Visual Studio 2005, které umožňuje stejně snadno pracovat s jazyky C++, C#, J# a Visual Basic 2005 i s kódem ASP.NET. Visual Studio 2008 integruje nejlepší funkce příslušných jazykově specifických prostředí Visual Studia .NET 2002/2003/2005 a Visual Studia 6.
- **C#:** C# je nový objektově orientovaný jazyk, který je určen pro spolupráci s technologií .NET.

V kapitole 1 se podrobněji zaměříme na výhody architektury .NET.

Co je nového v platformě .NET Framework 3.5

Vydání první verze platformy .NET Framework (1.0) roku 2002 se setkalo s velkým nadšením..NET Framework 2.0 byla představena roku 2005 a byla považována za klíčové vydání této platformy. Ačkoli .NET Framework 3.5 nepřináší tak rozsáhlé změny jako verze 2.0, je stále považována za významnou verzi produktu, která obsahuje mnoho podstatných nových vlastností.

Při každém vydání této platformy se společnost Microsoft snažila zajistit, aby byly dopady na vyvinutý kód co nejmenší. Dosud se jí to velmi dobře dařilo.

> Nezapomeňte instalovat testovací server, abyste mohli upgrade svých aplikací na platformu .NET Framework 3.5 podrobně vyzkoušet. Není vhodné přímo aktualizovat aktivně používanou aplikaci.

Následující podkapitoly popisují některé novinky v platformě .NET Framework 3.5, C# 2008 a také nové funkce vývojového prostředí Visual Studio 2008, které je pro tuto platformu určeno.

Implicitně typované proměnné

C# 2008 nyní umožňuje deklarovat proměnnou s tím, že její typ určí implicitně překladač. Zjistíte, že LINQ tuto možnost využívá při práci s dotazy. Využít této funkcionality lze s pomocí klíčového slova `var`:

```
var x = 5;
```

Překladač se u takového příkazu pokusí určit typ proměnné s pomocí hodnoty 5. V tomto případě bude příkaz interpretován takto:

```
int x = 5;
```

Automaticky implementované vlastnosti (properties)

V C# 2008 je často opakovaný proces deklarování vlastností zase o něco snazší. Před touto verzí byly vlastnosti deklarovány například takto:

```
private int _myItem;
public int MyItem
{
  get {
    return myItem;
  }

  set {
    myItem = value;
  }
}
```

Nyní již můžete tuto práci přenechat překladači. Místo neustále opakovaného vkládání výše uvedeného kódu je teď možné využít následující konstrukci:

```
public int MyItem { get; set; }
```

Tento zápis vygeneruje výsledek zcela shodný s výše uvedeným (a značně delším) kódem. Překladač provede konverzi zkráceného zápisu na původní zápis vlastností za vás a umožní vám tak psát kód přehledněji a rychleji než kdy dřív.

Inicializátory kolekcí a objektů

C# 2008 nyní umožňuje přiřadit hodnotu vlastnostem (property) objektu v okamžiku, kdy je tato vlastnost inicializována. Mějme například následující kód:

```
public class MyStructure
{
  public int MyProperty1 { get; set; }
  public int MyProperty2 { get; set; }
}
```

V C# 2008 pak můžeme objekt MyStructure inicializovat následovně:

```
MyStructure myStructure = new MyStructure() { MyProperty1 = 5, MyProperty2 = 10 };
```

Tato funkcionalita nám také umožňuje deklarovat několik prvků kolekce zároveň:

```
List <int>  myInts = new List <int> () { 5, 10, 15, 20, 25 };
```

Tímto způsobem jsou všechna zadaná čísla přidána do kolekce stejně, jako bychom je přidali metodou `Add()`.

Vestavěná podpora ASP.NET AJAX

Ačkoli bylo možné vytvářet stránky využívající technologii ASP.NET Ajax již s pomocí .NET Framework 2.0, vyžadovalo to instalaci dalších komponent. Nyní je podpora ASP.NET AJAX zabudována přímo do ASP.NET 3.5 a Visual Studia 2008.

Každá stránka, kterou s pomocí ASP.NET s .NET Frameworkem 3.5 vytvoříte, nyní může využívat Ajax (veškerá nastavení Ajaxu jsou v souboru `Web.config`). V sadě nástrojů pro ASP.NET teď také najdete nové serverové ovládací prvky, které vám umožní doplnit do vašich webových aplikací funkcionalitu založenou na technologii AJAX. Více informací o ASP.NET Ajaxu je v kapitole 39.

.NET Language Integrated Query Framework (LINQ)

LINQ je jednou z nejpříjemnějších – a nejočekávanějších – vlastností nového .NET. Microsoft poskytuje LINQ jako silně typované rozhraní pro přístup k externím zdrojům dat. LINQ dává vývojářům možnost pracovat s externími daty v rámci prostředí, na které jsou zvyklí, s pomocí objektů, které spolupracují s IDE, IntelliSense, a dokonce i s ladicími nástroji.

S použitím LINQ lze provádět dotazy vůči objektům, kolekcím, SQL databázím, XML souborům a mnoha dalším datovým zdrojům. Příjemné je, že bez ohledu na to, z jakého zdroje jsou data načítána, se k nim přistupuje vždy stejným způsobem, protože LINQ k nim poskytuje univerzální strukturovaný přístup.

Následující příklad ukazuje jak lze z XML dokumentu získat jména zákazníků:

```
XDocument xdoc = XDocument.Load(@"C:\Customers.xml");
var query = from people in xdoc.Descendants("CustomerName")
            select people.Value;

Console.WriteLine("{0} Customers Found", query.Count());
```

```
Console.WriteLine();
foreach (var item in query)
{
 Console.WriteLine(item);
}
```

Další aspekty využití LINQu jsou popsány v kapitolách 11, 27 a 29.

Podpora různých verzí .NET Framework

Vývojáři .NET dnes v mnoha případech pracují s několika aplikacemi, které jsou určeny pro různé verze .NET Frameworku (2.0, 3.0, nyní 3.5). Bylo by těžkopádné udržovat na počítači souběžně několik verzí Visual Studia jen proto, aby bylo možné s těmito aplikacemi pracovat. Nejnovější verze Visual Studia 2008 proto umožňuje zvolit verzi platformy, se kterou chcete pracovat. Při vytváření nové aplikace tak máte možnost rozhodnout se, zda má být určena pro .NET Framework verze 2.0, 3.0 nebo 3.5.

Podpora nejnovějších typů aplikací

Nedávno uvedený .NET Framework 3.0 přinesl některé zcela zásadní možnosti – mezi nimi i přístup k aplikacím využívajícím Windows Presentation Foundation (WPF) či knihovny Windows Communication Foundation (WCF).

Ve Visual Studiu 2008 naleznete prostředky pro sestavování těchto aplikací – všechny jsou k dispozici jako samostatný projekt, včetně nových nastavení, průvodců (wizard) a specifické podpory ze strany vývojového prostředí.

Porovnání jazyka C# s jinými jazyky

Z určitého hlediska lze jazyk C# považovat za stejnou revoluci mezi programovacími jazyky, jakou v prostředí Windows představuje technologie .NET. Současně s tím, jak společnost Microsoft přidávala v posledním desetiletí další a další funkce do systému Windows a do rozhraní Windows API, rozrostly se i jazyky Visual Basic 2008 a C++. Jazyky Visual Basic a C++ sice díky tomu získaly mimořádné možnosti, ale oba jazyky také kvůli způsobu svého vývoje trpí různými problémy.

V případě jazyka Visual Basic 6 a předchozích verzí spočívala hlavní síla jazyka v tom, že jej bylo možné snadno pochopit a že usnadňoval mnoho programátorských úkolů, protože před vývojářem z větší části skrýval podrobnosti rozhraní Windows API a infrastruktury komponent COM. Kvůli tomu však jazyk Visual Basic nikdy nebyl plně objektově orientovaný a kód velkých aplikací proto brzy ztrácel přehlednost a obtížně se udržoval. Syntaxe jazyka Visual Basic byla navíc převzata ze starších verzí jazyka BASIC (který byl opět navržen tak, aby se jej mohli intuitivně naučit začínající programátoři, nikoli aby se v něm psaly velké komerční aplikace). Proto se naprosto nehodil k tvorbě kvalitně strukturovaných či objektově orientovaných programů.

Jazyk C++ má na druhé straně své kořeny v definici standardu ISO jazyka C++. Nesplňuje normu ISO úplně z toho důvodu, že společnost Microsoft napsala svůj první kompilátor C++ ještě před oficiálním vydáním standardu ISO. Velmi se však této normě blíží. Naneštěstí to vedlo ke dvěma problémům. Za prvé má norma ISO C++ své kořeny v technologii staré desítky let, což se projevuje v nedostatečné podpoře moderních koncepcí (např. řetězců v kódování Unicode a generování

dokumentace v XML) a v určitých zastaralých strukturách určených pro dřívější kompilátory (jako je oddělení deklarace od definice členských funkcí). Za druhé se společnost Microsoft současně snažila vyvinout C++ jako jazyk, který je určen k náročným úkolům v systému Windows. Aby toho dosáhla, musela jazyk rozšířit o velké množství specifických klíčových slov a také různých knihoven. Výsledek je, že v systému Windows se z tohoto jazyka stal naprostý „guláš". Stačí se zeptat vývojářů v C++, na kolik definic řetězce si dokážou vzpomenout: `char*`, `LPTSTR`, `string`, `CString` (verze MFC), `CString` (verze WTL), `wchar_t*`, `OLECHAR*` atd.

V této fázi na scénu vstupuje zcela nové prostředí .NET, které bude zahrnovat nová rozšíření obou jazyků. Společnost Microsoft tento problém vyřešila přidáním dalších vlastních klíčových slov do jazyka C++ a kompletním přepracováním jazyka Visual Basic v prostředí Visual Basic .NET na jazyk Visual Basic 2008. Tento jazyk si zachovává některé základní syntaktické prvky jazyka VB, ale jeho návrh se natolik liší, že jej z praktického hlediska můžeme považovat za nový jazyk.

Za této situace se společnost Microsoft rozhodla poskytnout vývojářům alternativu – jazyk navržený speciálně pro technologii .NET a vytvořený zcela nově. Výsledkem je C#. Společnost Microsoft oficiálně popisuje jazyk C# jako „jednoduchý, moderní, objektově orientovaný a typově bezpečný programovací jazyk, který je odvozen od jazyků C a C++". Většina nezávislých pozorovatelů by tento popis pravděpodobně opravila na „odvozený od jazyků C, C++ a Java". Tyto definice jsou technicky přesné, ale příliš nezachycují krásu či eleganci jazyka. Syntakticky se jazyk C# velmi podobá jazykům C++ i Java do takové míry, že mnohá klíčová slova jsou shodná. C# také sdílí stejnou strukturu bloků se závorkami ({}) pro označení bloků kódu a odděluje příkazy pomocí středníků. Výpis kódu C# na první pohled velmi připomíná kód v jazycích C++ nebo Java. Navzdory této prvotní podobnosti se však jazyk C# lze naučit mnohem snáze než C++ a jeho obtížnost je srovnatelná s jazykem Java. Jeho struktura lépe vyhovuje moderním vývojovým prostředím než oba starší jazyky a byl navržen tak, aby programátorům současně poskytl jednoduchost používání jazyka Visual Basic a v případě potřeby vysoký výkon a nízkoúrovňový přístup k paměti jako jazyk C++. Uveďme si některé vlastnosti jazyka C#:

- plná podpora tříd a objektově orientovaného programování, včetně dědičnosti rozhraní i implementace, virtuálních funkcí a přetížení operátorů,
- konzistentní a vhodně definovaná sada základních typů,
- integrovaná podpora automatického generování dokumentace ve formátu XML,
- automatické čištění dynamicky přidělované paměti,
- možnost označit třídy nebo metody uživatelsky definovanými atributy. Tato funkce může být užitečná pro dokumentaci a někdy má určitý vliv na překlad (např. při označení metod, aby se překládaly pouze v ladicích sestaveních),
- plný přístup ke knihovně základních tříd .NET a také snadná dostupnost rozhraní Windows API (pokud ho skutečně potřebujete, k čemuž nedochází příliš často),
- v případě potřeby jsou dostupné ukazatele a přímý přístup do paměti, ale jazyk je navržen takovým způsobem, že lze bez nich pracovat téměř ve všech situacích,
- podpora vlastností a událostí ve stylu jazyka Visual Basic,
- pouhou změnou možností překladače můžete překládat buď spustitelný soubor, nebo knihovnu komponent .NET, kterou může volat jiný kód stejným způsobem jako ovládací prvky ActiveX (komponenty COM),
- pomocí jazyka C# lze psát dynamické stránky ASP.NET a webové služby založené na XML.

Je vhodné uvést, že většina výše zmíněných vlastností se vztahuje i na jazyky Visual Basic 2008 a spravované C++. Nicméně díky tomu, že jazyk C# je od začátku navržen pro spolupráci s technologií .NET, je jeho podpora funkcí této platformy kompletnější a dostupná na základě vhodnější syntaxe, než je tomu u výše uvedených jazyků. Jazyk C# je sice velmi podobný jazyku Java, ale poskytuje některá vylepšení. Jazyk Java zejména není určen ke spolupráci s prostředím .NET.

Na závěr tohoto tématu bychom měli uvést několik omezení jazyka C#. Jednou z oblastí, pro které tento jazyk není určen, je vývoj časově kritického nebo extrémně zatěžovaného kódu, tj. kódu, kde skutečně záleží na tom, zda průběh smyčky vyžaduje 1 000 nebo 1 050 cyklů procesoru, a potřebujete uvolnit prostředky v té milisekundě, kdy již nejsou potřeba. Mezi nízkoúrovňovými jazyky si v této oblasti nejspíš i nadále udrží výsadní postavení C++. Jazyk C# postrádá některé klíčové funkce, které jsou nutné pro aplikace s mimořádně vysokým výkonem, včetně možnosti specifikovat funkce s modifikátorem inline a destruktory, u kterých je zajištěno spuštění v definovaných bodech kódu. Do této kategorie však patří pouze malá skupina aplikací.

Požadavky na psaní a spouštění kódu v jazyce C#

Platformu .NET Framework 3.5 lze spustit v systémech Windows XP, Windows Server 2003, Windows Vista a nejnovějším Windows Server 2008. Chcete-li psát kód pomocí technologie .NET, musíte si nejdříve nainstalovat sadu .NET 3.5. Jestliže navíc neuvažujete o tom, že byste psali kód C# v textovém editoru nebo vývojovém prostředí jiného dodavatele, budete téměř jistě potřebovat i prostředí Visual Studio 2008. Spravovaný kód lze spustit i bez plné sady SDK, je ale nutné běhové prostředí .NET. Někdy je vhodné běhové prostředí .NET distribuovat spolu s kódem kvůli těm klientům, kteří si ho zatím nenainstalovali.

Témata popsaná v této knize

Tuto knihu zahájíme v následující kapitole celkovým přehledem architektury .NET, abyste získali základní znalosti potřebné k psaní spravovaného kódu. Zbytek knihy je rozdělen na části, které se týkají jazyka C# a jeho aplikací v mnoha oblastech.

Část I: Jazyk C#

Tato část představuje dostatečný úvod do vlastního jazyka C#. Nevyžaduje znalosti žádného konkrétního jazyka, ačkoli předpokládá, že jste zkušenými programátory. Začneme přehledem základní syntaxe a datových typů jazyka C# a potom si rozebereme objektově orientované funkce tohoto jazyka, abychom mohli přejít k pokročilejším tématům programování v C#.

Část II: Visual Studio

Tato část se zabývá celosvětově nejvyužívanějším IDE pro vývojáře C# – Visual Studiem 2008. Dvě kapitoly této části se zabývají nejlepšími způsoby využití tohoto nástroje k vytváření aplikací založených na .NET Framework 2.0 nebo 3.0. Dále pak osvětlují možnosti nasazování (deployment) projektů.

Část III: Knihovny bázových tříd (Base class libraries)

V této části se podíváme na principy programování v prostředí .NET. Zejména se zaměříme na Visual Studio .NET, zabezpečení, zavádění podprocesů do aplikací pro .NET a na metody vytváření vlastních knihoven a sestavení.

Část IV: Data

Budeme se zabývat přístupem k databázím pomocí ADO.NET a LINQ a interakcí se soubory a adresáři. Podrobně také analyzujeme podporu jazyka XML v technologii .NET a na straně operačního systému Windows a funkce platformy .NET v SQL Serveru 2008. V rámci rozsáhlého prostoru věnového LINQ se budeme především soustředit na LINQ v SQL a LINQ v XML.

Část V: Prezentace

V této části se soustředíme na tvorbu klasických okenních aplikací pro Windows, které se v technologii .NET označují jako formuláře (Windows Forms). Formuláře jsou aplikace typu silných klientů a v prostředí .NET lze tyto typy aplikací vytvářet snadno a rychle. Kromě informací o formulářích pro Windows se také zaměříme na GDI+. Pomocí této technologie budeme vytvářet aplikace, které zahrnují pokročilou grafiku.

V této části si také vysvětlíme psaní komponent, které jsou spouštěny na webových serverech a fungují jako webové stránky. Patří sem mimořádné množství nových funkcí, které poskytuje prostředí ASP.NET 3.5. Konečně zde naleznete také postupy pro vytváření aplikací založených na Windows Presentation Foundation a VSTO.

Část VI: Komunikace

Tato část je kompletně věnována komunikaci. Zabýváme se zde webovými službami pro komunikaci nezávislou na platformě, vzdálenou komunikací v .NET (remoting) pro komunikaci mezi klienty a servery v .NET, službami Enterprise Services pro služby na pozadí a komunikací s DCOM. Ukážeme si asynchronní odpojenou komunikaci pomocí služby Message Queuing. Taktéž se zde zmíníme o využití Windows Communication Foundation a Windows Workflow Foundation.

Část VII: Přílohy

Tato část zahrnuje tři přílohy, které podrobně analyzují principy objektově orientovaného programování a obsahují také specifické programátorské informace o jazyku C#.

Konvence

V knize používáme několik různých stylů textu a grafické úpravy, které pomáhají rozlišit různé typy informací. Následují příklady používaných stylů spolu s vysvětlením, co znamenají:

Odrážky jsou odsazené a každá nová odrážka je označena takto:

- Důležitá slova jsou formátována *kurzivou*.
- Klávesy, které je nutné stisknout na klávesnici, jsou zobrazeny takto: Ctrl, Enter, Ctrl+Enter.

Kód je uveden několika různými způsoby. Pokud je diskutované slovo součástí textu (například při vysvětlení cyklu `if...else`), je označeno `tímto písmem`. Jestliže se jedná o blok kódu, který lze zadat jako program a spustit, naleznete ho také v šedém rámečku:

```
public static void Main()
{
   AFunc(1,2,"abc");
}
// Pokud jsme se nedostali na konec, vrať hodnotu true, jinak
// nastav neplatnou pozici a vrať hodnotu false.
pos++;
if (pos < 4)
   return true;
else {
   pos = -1;
   return false;
}
```

Takovéto panely s tučným písmem obsahují kriticky důležité informace, které přímo souvisí s okolním textem a které byste neměli zapomenout.

Poznámky psané kurzivou vyjadřují různé tipy, triky a doplňující informace k aktuálnímu výkladu.

Syntaxe zápisu metod a vlastností má následující formát.

```
Regsvcs BookDistributor.dll [NázevAplikaceCOM+] [KnihovnaTypů.tbl]
```

Kurziva zde označuje odkazy na objekty, proměnné nebo hodnoty parametrů, které je nutné zadat. Hranaté závorky představují volitelné parametry.

Zdrojový kód a přílohy

Při procházení příkladů v knize můžete podle svého uvážení veškerý kód zadávat ručně, nebo můžete použít soubory se zdrojovými kódy, které jsou doplňkem knihy. Všechny zdrojové kódy uvedené v této knize jsou k dispozici ke stažení na webu `www.wrox.com`. Po zobrazení úvodní stránky webu jednoduše vyhledejte název knihy (pomocí pole Search či jednoho ze seznamů titulů) a klepněte na odkaz Download Code na stránce podrobností o knize, abyste si stáhli kompletní zdrojový kód ke knize.

Hodně knih mívá podobné tituly, a proto může být jednodušší vyhledávat podle jejího čísla ISBN – tato kniha má v originálním anglickém vydání ISBN 978-0-470-19137-8.

Po stažení kódu jej stačí dekomprimovat vhodným komprimačním nástrojem. Můžete také přejít na hlavní stránku společnosti Wrox a stáhnout si ho na adrese `www.wrox.com/dynamic/books/download.aspx`, kde naleznete dostupné kódy této knihy a všech dalších knih nakladatelství Wrox.

Lokalizované zdrojové kódy všech příkladů jsou ke stažení na adrese http://knihy.cpress.cz/1472.

Errata

Snažíme se v maximální možné míře zajistit, aby text ani kód neobsahovaly žádné chyby. Nikdo však není dokonalý a k chybám dochází. Naleznete-li v některé z našich knih chybu, jako je překlep nebo chybná část kódu, budeme vám velmi vděčni, když nám to oznámíte. Odesláním opravy možná jiným čtenářům ušetříte hodiny frustrací a současně pomůžete vylepšit kvalitu našich budoucích knih.

Pokud chcete otevřít stránku s opravami k této knize, přejděte na adresu www.wrox.com a vyhledejte název knihy pomocí pole Search nebo jednoho ze seznamů titulů. Na stránce podrobností o knize pak klepněte na odkaz Book Errata. Tato stránka obsahuje seznam všech oprav, které redaktoři nakladatelství Wrox zadali a zveřejnili. Úplný seznam knih včetně odkazů na opravy každé z nich je také k dispozici na adrese http://www.wrox.com/misc-pages/booklist.shtml.

Pokud na stránce Book Errata „svou" chybu nenajdete, přejděte na stránku http://www.wrox.com/contact/techsupport.shtml a vyplňte formulář pro odeslání nalezené chyby. Informaci zkontrolujeme, a jestliže bude správná, vystavíme zprávu na stránce oprav ke knize a vyřešíme problém v následujících vydáních knihy.

p2p.wrox.com

Chcete-li se zúčastnit diskuse s autory a uživateli knihy, přihlaste se do fór P2P na adrese p2p.wrox.com. Webový systém fór dovoluje odesílat zprávy týkající se knih nakladatelství Wrox a souvisejících technologií a komunikovat s jinými čtenáři a odborníky. Fóra umožňují, abyste přihlásili svou e-mailovou adresu k odběru zvolených témat, když se na fórech objeví nové příspěvky. Těchto fór se účastní autoři a redaktoři nakladatelství Wrox, jiní experti z oboru a další čtenáři.

Na adrese http://p2p.wrox.com naleznete mnoho různých fór, která můžete využít nejen při čtení této knihy, ale také při vývoji svých vlastních aplikací. Chcete-li se k fórům připojit, postupujte podle následujících kroků:

1. Přejděte na adresu p2p.wrox.com a klepněte na odkaz Register.
2. Přečtěte si podmínky používání a klepněte na tlačítko Agree.
3. Zadejte informace požadované pro přihlášení a případně také nepovinné informace, které o sobě chcete uvést, a klepněte na tlačítko Submit.

Obdržíte e-mail s popisem, jak potvrdit svůj účet a dokončit proces registrace.

> Číst zprávy na fórech můžete i bez přihlášení k P2P, ale pro posílání vašich vlastních zpráv již musíte být jejich členem.

Po přihlášení můžete číst nové zprávy a odpovídat na příspěvky jiných uživatelů. Zprávy si můžete číst kdykoli, když máte přístup k webu. Pokud byste chtěli dostávat nové zprávy z určitého fóra e-mailem, klepněte na ikonu Subscribe to this Forum vedle názvu fóra v jejich seznamu.

Chcete-li získat další informace o práci se systémem Wrox P2P, nezapomeňte si na stránce často kladených otázek P2P FAQs přečíst odpovědi na dotazy o fungování softwaru fóra a také na mnoho běžných otázek, které se týkají systému P2P a knih nakladatelství Wrox. Tyto často kladené otázky naleznete po klepnutí na odkaz FAQ na libovolné stránce systému P2P.

Poznámka redakce českého vydání

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

Computer Press
redakce počítačové literatury
náměstí 28. dubna 48
635 00 Brno-Bystrc

nebo

knihy@cpress.cz.

Další informace a případné opravy českého vydání knihy najdete na internetové adrese http://knihy.cpress.cz/k1472. Prostřednictvím uvedené adresy můžete též naší redakci zaslat komentář nebo dotaz týkající se knihy. Na vaše reakce se srdečně těšíme.