



联袂推荐

微软技术丛书

Microsoft

- 沿袭深受读者欢迎的Step by Step经典写作风格
- 资深专家John Sharp再度演绎更SHARP的C#
- 覆盖面广，知识点新，充分展示C#新增特性

Visual C# 2012 从入门到精通

John Sharp 著
周靖 译

Step by Step



清华大学出版社

微软技术丛书

Microsoft

Visual C# 2012 从入门到精通

John Sharp 著
周靖 译



Step by Step

清华大学出版社

内 容 简 介

C#作为微软的旗舰编程语言,深受程序员喜爱,是编写高效应用程序的首选语言。Visual C# 2012 提供了大量新功能,本书围绕语言的基础知识和这些新功能全面介绍了如何利用 Visual Studio 2012 和.NET Framework 4.5 编写 C#应用程序。书中沿袭深受读者欢迎的 Step by Step 风格,通过合理的练习引导读者逐步构建在 Windows 7 和 Windows 8 上运行的应用程序,访问 SQL Server 数据库,开发多线程应用等。

全书共 27 章,结构清晰,叙述清楚。所有练习均在 Visual Studio 2012 简体中文版上进行过全面演练。无论是刚开始接触面向对象编程的新手,还是打算转移到 C#的具有 C、C++或者 Java 基础的程序员,都可以从本书汲取到新的知识,迅速掌握 C#编程技术。

© 2014 Tsinghua University Press Limited

Authorized translation of the English edition of **Microsoft Visual C# 2012 Step by Step**

Copyright © 2012 by CM Group, Ltd. This translation is published and sold by permission of O'Reilly Media, Inc., which owns or controls of all rights to publish and sell the same.

本书中文版由 O'Reilly Media, Inc. 授权给清华大学出版社出版发行。

北京市版权局著作权合同登记号 图字: 01-2013-4367

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话: 010-62782989 13701121933

图书在版编目(CIP)数据

Visual C# 2012 从入门到精通/(英)夏普(Sharp, J.)著.周靖译—北京:清华大学出版社,2014
(微软技术丛书)

书名原文: Microsoft Visual C# 2012 Step by Step

ISBN 978-7-302-34509-1

I. ① V… II. ①夏… ②周… III. ① C 语言—程序设计 IV. ① TP312

中国版本图书馆 CIP 数据核字(2013)第 274717 号

责任编辑: 文开棋

装帧设计: 杨玉兰

责任校对: 李玉萍

责任印制: 宋 林

出版发行: 清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址: 北京清华大学学研大厦 A 座 邮 编: 100084

社 总 机: 010-62770175 邮 购: 010-62786544

投稿与读者服务: 010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

课 件 下 载: <http://www.tup.com.cn>, 010-62791865

印 刷 者: 清华大学印刷厂

装 订 者: 北京市密云县京文制本装订厂

经 销: 全国新华书店

开 本: 185mm×260mm

印 张: 42

字 数: 873 千字

版 次: 2014 年 1 月第 1 版

印 次: 2014 年 1 月第 1 次印刷

印 数: 1~3500

定 价: 98.00 元



译者序

C#(读作“C sharp”)作为一种编程语言,宗旨是创建在.NET Framework上运行的各种应用程序。C#简单、功能强大、类型安全,而且完全面向对象。C# 凭借在许多方面的创新,在保持C语言风格的表现力和优雅特征的同时,实现了应用程序的快速开发。

Visual C#是Microsoft对C#语言的实现。而Visual Studio作为Microsoft的一款“交互开发环境”(IDE)产品,通过功能齐全的代码编辑器、编译器、项目模板、设计器、代码向导、强大且易用的调试器以及其他工具,实现了对Visual C#的支持。通过.NET Framework类库(FCL),可访问许多操作系统服务以及其他许多有用的、精心设计的类,这些类可显著加快开发过程。

本书是为Visual C#开发人员量身定制的一本“快速上手”指南。和市面上简单地罗列各种语法元素的书籍不同,本书使用了大量生动、实际的例子,逐步骤地指引你在Visual Studio中进行C#编程。

随着学习的深入,你将牢牢地掌握C#语言的各种概念,并很快就能掌握编写各种C#程序的技巧。这些程序涉及的领域广泛:从简单的控制台应用程序,到更高级的Windows Store应用程序;从简单的“Hello World”程序,到更实用的数据库应用程序,再到用C# 5.0的async/await关键字来实现的异步操作。

整个学习过程非常清晰和直接。在本书上一版《Visual C# 2010从入门到精通》的基础上,新的一版进行了大量修订和增补。在内容的衔接和对新的C# 5.0的侧重上,更是下足功夫。如果是C#的新手,可选择从头读到尾的方式,整个阅读过程是流畅、没有阻碍的。如果有经验的C#开发者,可以有针对性地阅读自己感兴趣的主体,例如自己感觉比较薄弱的环节以及和C# 5.0新特性有关的章节。具体可以参见本书前言的“导读”。

其实任何书都是有瑕疵的。翻译一本书的过程其实和写一个程序的过程差不多。无论在这个过程中的感觉有多么“完美”,最后总能找出这样或那样的错误或者并不完美的地方。因此,一本没有勘误、没有后期维护的书不能算是真正的好书。根据传统,本书在付印之后,我的博客会开辟它的专栏,提供相关资源(如源代码、练习文件)以及勘误的下载,详情请访问<http://transbot.blog.163.com>。本书需要重印的时候,我也会敦促出版商将已确定的勘误反映到新的印次中。

在阅读本书的同时,推荐关注同样由我翻译的《CLR via C#》。这本书从更底层的角度讲解了C#以及它面向的“公共语言运行时”(CLR),可以帮助你加深对语言精妙之处的

体验，同时更扎实地掌握 C# 语言，加深和巩固你在本书中学到的知识。

简单地说，像《Visual Studio 2012 从入门到精通》这样的书旨在将重点集中于特定的应用程序类型，帮助你“自上而下”地学习；而《CLR via C#》这样的书是将重点放在开发平台上，帮助你“自下而上”地学习。两种方式的结合，能让你成为一名卓尔不群的开发人员。

祝学习愉快！

周 靖

2013.12@北京

前 言

Microsoft Visual C# 是一种功能强大、使用简单的语言，主要面向需要使用 Microsoft .NET Framework 来创建应用程序的开发者。它在 C++ 和 Microsoft Visual Basic 的基础上去芜存菁，最终成了一种更加清晰、更富有逻辑的语言。C# 1.0 于 2001 年亮相。几年后随着 C# 2.0 和 Visual Studio 2005 的问世，语言新增了几个重要功能，包括泛型、迭代器和匿名方法等。随 Microsoft Visual Studio 2008 发布的 C# 3.0 添加了更多功能，包括扩展方法、Lambda 表达式以及语言集成查询(Language Integrated Query, LINQ)。2010 年发布的 C# 4.0 提供了进一步的增强，它改善了与其他语言和技术的互操作性。新增功能包括具名参数和可选参数，另外还有 dynamic 类型(告诉语言的“运行时”实现对象的晚期绑定)。在随 C# 4.0 发布的 .NET Framework 中，最重要的新功能就是“任务并行库”(Task Parallel Library, TPL)。可用 TPL 构建具有良好伸缩性的应用程序，从而快速和简单地发挥出多核处理器的潜力。C# 5.0 则通过 async 修饰符和 await 操作符提供了对异步任务的原生支持。

Windows 8 是 Microsoft 公司近年来最具革命性的一款操作系统。新操作系统支持高度交互式的应用程序，它们能相互分享和协作，还能轻松连接云端服务。Visual Studio 2012 开发环境使这些强大功能变得很容易使用，大量新向导和增强功能显著提高了开发效率。Visual Studio 2012、Windows 8 和 C# 5.0 三剑客提供了完善的平台和工具集来帮助你开发下一代功能强大的、直观的而且容易移植的应用程序。但是，即便不用 Windows 8 进行开发，Visual Studio 2012 和 C# 5.0 这两者的组合也能提供强大的助力。

本书面向的读者

本书假定你要使用 Visual Studio 2012 和 .NET Framework 4.5 学习基础的 C# 编程知识。学完本书后，会对 C# 有一个全面、透彻的理解，会用它开发出响应速度快的、易于伸缩的、能在 Windows 7 和 Windows 8 上运行的应用程序。

可构建并运行在 Windows 7 和 Windows 8 上运行的 C# 5.0 应用程序。但两种操作系统的用户界面显著不同，所以第 I 部分到第 III 部分的练习和示例在两种环境下都能运行，而第 IV 部分专注于 Windows 8 应用开发。

本书不面向的读者

本书面向刚开始涉足 C# 应用开发的人士，重点放在 C# 语言上面。本书不涉及企业级 Windows 应用程序开发，比如 ADO.NET，ASP.NET，Windows Communication Foundation 或者 Workflow Foundation。要了解这些方面的知识，可阅读“从入门到精通”系列的其他书籍，包括《ASP.NET 4 从入门到精通》、《ADO.NET 4 从入门到精通》以及《Windows

Communication Foundation 4 从入门到精通》。

本书的组织

本书分为以下四大部分。

- 第 I 部分“Visual C#和 Visual Studio 2012 概述”介绍 C#语言的核心语法，演示了 Visual Studio 编程环境。
- 第 II 部分“理解 C#对象模型”深入探讨如何用 C#创建和管理新类型，如何管理这些类型引用的资源。
- 第 III 部分“用 C#定义可扩展类型”全面讨论如何利用 C#语言元素来构建能在多个应用程序中重用的类型。
- 第 IV 部分“使用 C#构建专业 Windows 8 应用程序”描述 Windows 8 编程模型，如何用 C#为新模型构建交互式应用程序。

 **注意** 虽然第 IV 部分面向 Windows 8，但第 23 章和第 24 章的一些概念仍然适用于 Windows 7 应用程序。

导读

本书帮助你掌握多个基本领域的开发技能。无论刚开始学习编程，还是从另一种语言(C、C++、Java 或 Visual Basic)转向 C#，本书都能提供帮助。可以参考下表找到自己的最佳起点。

读者类型	步骤
面向对象编程的新手	1. 按照“范例代码”一节的步骤安装练习文件。 2. 顺序阅读第 I 部分、第 II 部分和第 III 部分。 3. 有了一定经验后，如果有兴趣，继续完成第 IV 部分的学习。
熟悉 C 语言等过程编程语言，但新涉足 C#	1. 按照“范例代码”一节的步骤安装练习文件。略读前 5 章来获得对 C#和 Visual Studio 2012 的大致印象，重点阅读第 6~22 章。 2. 有了一定经验后，如果有兴趣就继续完成第 IV 部分的学习。
从面向对象语言 C++或 Java 等迁移到 C#	1. 按照“范例代码”一节的步骤安装练习文件。 2. 略读前 7 章，获得对 C#语言和 Visual Studio 2012 的大致印象，重点阅读第 8~22 章。 3. 要想了解 Windows 8 应用程序开发，请阅读第 IV 部分。

续表

读者类型	步骤
从 Visual Basic 6 迁移到 C#	1. 按照“范例代码”一节的步骤安装练习文件。 2. 顺序阅读第 I 部分、第 II 部分和第 III 部分。 3. 如要了解 Windows 8 应用程序开发, 请阅读第 IV 部分。 4. 阅读每章末尾的“快速参考”小节, 了解 C#特有的构造。
做完所有练习后再将本书作为参考书使用的读者	1. 根据目录寻找特定主题的信息。 2. 阅读每章最后的“快速参考”, 查看对当前章所介绍的语法和技术的简单回顾。

本书大多数章节都通过实际的例子来方便读者试验刚学到的概念。无论感兴趣的是哪个主题, 都需要下载并安装范例代码。

本书的约定和特色

本书通过一些约定来增强内容的可读性, 以便于读者理解。

- 每个练习都用编号的操作步骤来完成。
- “注意”等特色段落提供了成功完成一个步骤需要了解的额外信息或替代方案。
- 要求读者输入的文本加粗显示。
- 两个键名之间的加号(+)意味着必须同时按下这两个键。例如, “按 Alt+Tab 键”意味着按住 Alt 键不放, 然后按 Tab 键。
- 描述菜单操作时, 采取“文件”|“打开”的形式, 意思是从“文件”菜单中选择“打开”命令。

系统需求

为了完成本书的练习, 需准备以下硬件和软件:

- Windows 7(x86 或 x64), Windows 8(x86 或 x64), Windows Server 2008 R2(x64)或者 Windows Server 2012(x64)

 **注意** Visual Studio 2012 还支持 Windows Vista、Windows XP 和 Windows Server 2003。但本书的练习和代码未在这些平台上测试。

- Visual Studio 2012(除 Visual Studio Express for Windows 8 之外的任意版本)

 **注意** 可以使用 Visual Studio Express 2012 for Windows Desktop, 但就只能执行本书的 Windows 7 版本的练习。不可用它执行第 IV 部分的练习。

- 1.6 GHz 或更快的处理器(推荐 2 GHz 以上)
- 32 位操作系统至少 1 GB RAM, 64 位至少 2 GB RAM。在虚拟机中运行再加 512 MB
- 10 GB 剩余硬盘空间
- 支持 DirectX 9 的显卡, 1024 × 768 或更高分辨率。Windows 8 推荐 1366 × 768 或更高分辨率
- DVD-ROM 驱动器(如果从 DVD 安装 Visual Studio)
- 下载软件和范例代码需要 Internet 连接

此外, 还需要以管理员身份安装和配置 Visual Studio 2012。

范例代码

本书的配套网络资源包含练习时会用到的范例代码。使用这些范例代码, 读者不再需要浪费时间创建和练习无关的文件。借助于这些练习文件和课程中描述的步骤, 读者可以在实践中学习, 这是迅速掌握并记住新的编程技能的一种简单而高效的方式。

按以下步骤在计算机上安装本书配套代码。

1. 在 Internet Explorer 或其他浏览器的地址栏中输入 ***transbot.blog.163.com***。
2. 找到《Visual C# 2012 从入门到精通》博客文章, 按提示下载源代码文件压缩包。
3. 解压到以下位置:

文档^①\Microsoft Press\

也可从译者的网盘 <http://transbot.ys168.com> 下载练习文件和其他资源。

使用练习文件

本书每一章都解释了在什么时候以及如何使用必要的练习文件。需要使用练习文件的时候, 书中会给出相应的指示, 帮助你打开正确的文件。

如果想知道所有细节, 可以参见下表, 其中列出了本书要用到的所有 Visual Studio 2012

^① 本书将路径 “C:\Users\YourName\Documents” 简单称为 “文档” 文件夹。——译注

项目和解决方案，它们以文件夹的形式进行分组以便查找。练习通常会为同一个项目提供初始文件和完成之后的版本。有的练习提供 Windows 7 和 Windows 8 两个版本，操作步骤会针对不同操作系统给出相应的指示。已完成的项目存储在带有“- Complete”后缀的文件夹中。

 **注意** 如果使用 Windows Server 2008 R2，则按照 Windows 7 的步骤操作。在使用 Windows Server 2012 时，则按 Windows 8 的步骤操作。

项目名称	说明
第 1 章	
TextHello	作为第一个项目，它将指导你创建一个简单程序来显示欢迎文本
WPFHello	使用 WPF 技术在窗口中显示欢迎文本
第 2 章	
PrimitiveDataTypes	演示如何使用基元类型来声明变量，如何向变量赋值，如何在窗口中显示值
MathsOperators	演示了算术操作符(+、-、*、/、%)
第 3 章	
Methods	改进上个项目的代码，体会如何使用方法来建立代码的结构
DailyRate	指导你写自己的方法，运行方法，使用 Visual Studio 2012 调试器来单步执行方法
使用可选参数的 DailyRate	演示如何让方法获取可选参数，如何使用具名参数来调用方法
第 4 章	
Selection	演示如何用嵌套 if 语句实现复杂逻辑，例如比较两个日期的相等性
SwitchStatement	这个简单的程序使用一个 switch 语句将字符转换成相应的 XML 形式
第 5 章	
WhileStatement	使用 while 语句逐行读取源文件的内容，并在 Windows 窗体的文本框中显示每一行
DoStatement	使用 do 语句将十进制数转换成八进制
第 6 章	
MathsOperators	对第 2 章的 MathsOperators 项目进行改进，试验会造成程序执行失败的各种未处理的异常。然后，用 try 和 catch 关键字使应用程序更健壮，不会因为错误输入或操作而失败
第 7 章	
Classes	演示如何定义自己的类，为它添加了公共构造器、方法和私有字段；还演示如何用 new 关键字创建类的实例，如何定义静态方法和字段
第 8 章	
Parameters	演示值类型和引用类型的参数的区别，还演示如何使用 ref 和 out 关键字

续表

项目名称	说明
第 9 章	
StructsAndEnums	定义结构来表示日期
第 10 章	
Cards	使用数组来建模纸牌游戏中的一手牌
第 11 章	
ParamsArrays	演示如何使用 <code>params</code> 关键字使方法能接受任意数量的参数
第 12 章	
Vehicles	用继承创建简单的交通工具类，还演示如何定义虚方法
ExtensionMethod	演示如何为 <code>int</code> 类型创建扩展方法，提供方法将整数从十进制转换成其他进制
第 13 章	
Drawing Using Interfaces	实现图形绘图包的一部分。用接口定义要由几何图形对象公开并实现的方法
Drawing Using Abstract Classes	扩展了 Drawing Using Interfaces 项目，将几何图形对象的常用功能集成到抽象类中
第 14 章	
GarbageCollectionDemo	演示如何使用 <code>Dispose</code> 模式来实现异常安全的资源清理
第 15 章	
Drawing Using Properties	扩展第 13 章的 Drawing Using Abstract Classes 项目，用属性将数据封装到类中
AutomaticProperties	演示如何为类创建自动属性，如何用它们初始化类的实例
第 16 章	
Indexers	该项目使用了两个索引器，一个根据姓名查找某人的电话号码，另一个根据电话号码查找某人的姓名
第 17 章	
BinaryTree	演示如何使用泛型生成类型安全的结构，可包含任何类型的元素
BuildTree	演示如何使用泛型实现类型安全的方法，可获取任何类型的参数
第 18 章	
Cards	更新第 10 章的代码，演示如何用集合建模一手牌
第 19 章	
BinaryTree	演示如何实现泛型 <code>IEnumerator<T></code> 接口，为泛型 <code>Tree</code> 类创建枚举器
IteratorBinaryTree	用迭代器为泛型 <code>Tree</code> 类生成枚举器
第 20 章	
Delegates	演示如何通过委托调用方法，将方法的逻辑和调用方法的应用程序分开
Delegates With Event	演示如何用事件提醒对象发生了某事，如何捕捉事件并执行需要的处理
第 21 章	
QueryBinaryTree	演示如何通过 LINQ 查询从二叉树对象获取数据

续表

项目名称	说明
第 22 章	
ComplexNumbers	定义新类型来建模复数，并为这种类型实现常用的操作符
第 23 章	
GraphDemo	生成并在 WPF 窗体上显示复杂图表。使用单线程执行计算
GraphDemo With Tasks	创建多个任务，并行执行图表计算
Parallel GraphDemo	使用 Parallel 类对创建和管理任务的过程进行抽象
GraphDemo With Cancellation	中途得体地取消任务
ParallelLoop	演示何时不该使用 Parallel 类创建和运行任务
第 24 章	
GraphDemo	对第 23 章的同名项目进行修改，使用 async 关键字和 await 操作符来异步计算图表数据
PLINQ	使用并行任务，用 PLINQ 查询数据
CalculatePI	使用统计学采样计算 PI 的近似值。使用了并行任务
第 25 章	
Customers Without Scalable UI	使用默认 Grid 控件进行 Adventure Works 公司的 Customers 应用程序的 UI 布局。控件使用绝对定位，屏幕分辨率和设备大小改变时，不会自动地伸缩
Customers With Scalable UI	使用嵌套 Grid 控件并定义行和列来实现控件的相对定位。这个版本的 UI 能自动适应不同的屏幕分辨率和设备大小。但在贴靠视图中表现不佳
Customers With Adaptive UI	对上一个可伸缩 UI 进行扩展。使用 Visual State Manager 检测切换到贴靠视图的事件，并相应修改控件布局
Customers With Styles	使用 XAML 样式更改字体和背景图片
第 26 章	
DataBinding	使用数据绑定在 UI 中显示从数据源获取的客户资料；还演示了如何实现 INotifyPropertyChanged 接口，从而允许 UI 更新客户资料，并将改动发送回数据源
ViewModel	通过实现 Model-View-ViewModel 模式，将 UI 同数据源访问逻辑分开
Search	实现 Windows 8 搜索合约，按名字或姓氏搜索客户
第 27 章	
Data Service	这个解决方案包含一个 Web 应用程序来提供 WCF Data Service，使 Customers 应用程序能从 SQL Server 数据库获取客户数据。WCF Data Service 通过由实体框架创建的实体模型来访问数据库
Updatable ViewModel	这个解决方案中的 Customers 项目包含一个扩展的 ViewModel，它提供许多命令，允许 UI 通过 WCF Data Service 插入和更新客户资料

目 录

第 I 部分 Visual C#和 Visual Studio 2012 概述

第 1 章 欢迎进入 C#编程世界..... 3

1.1 开始在 Visual Studio 2012 环境中 编程.....	3
1.2 编写第一个程序.....	7
1.3 使用命名空间.....	11
1.4 创建图形应用程序.....	14
1.4.1 探索 Windows Store 应用 程序.....	23
1.4.2 探索 WPF 应用程序.....	26
1.4.3 向图形应用程序添加代码.....	28
小结.....	30
第 1 章快速参考.....	30

第 2 章 使用变量、操作符和表达式..... 32

2.1 理解语句.....	32
2.2 使用标识符.....	33
2.3 使用变量.....	34
2.3.1 命名变量.....	34
2.3.2 声明变量.....	35
2.4 使用基元数据类型.....	35
2.4.1 未赋值的局部变量.....	36
2.4.2 显示基元数据类型的值.....	36
2.5 使用算术操作符.....	42
2.5.1 操作符和类型.....	42
2.5.2 深入了解算术操作符.....	44
2.5.3 控制优先级.....	49
2.5.4 使用结合性对表达式进行 求值.....	49
2.5.5 结合性和赋值操作符.....	50
2.6 变量递增和递减.....	51
前缀和后缀.....	51
2.7 声明隐式类型的局部变量.....	52

小结.....	53
---------	----

第 2 章快速参考.....	53
----------------	----

第 3 章 方法和作用域..... 54

3.1 创建方法.....	54
3.1.1 声明方法.....	54
3.1.2 从方法返回数据.....	55
3.1.3 调用方法.....	57
3.2 使用作用域.....	59
3.2.1 定义局部作用域.....	60
3.2.2 定义类作用域.....	60
3.2.3 重载方法.....	61
3.3 编写方法.....	62
3.4 使用可选参数和具名参数.....	68
3.4.1 定义可选参数.....	70
3.4.2 传递具名参数.....	70
3.4.3 消除可选参数和具名参数的 歧义.....	71
小结.....	75
第 3 章快速参考.....	75

第 4 章 使用判断语句..... 77

4.1 声明布尔变量.....	77
4.2 使用布尔操作符.....	77
4.2.1 理解相等和关系操作符.....	78
4.2.2 理解条件逻辑操作符.....	78
4.2.3 短路求值.....	79
4.2.4 操作符的优先级和结合性 总结.....	79
4.3 使用 if 语句做出判断.....	80
4.3.1 理解 if 语句的语法.....	80
4.3.2 使用代码块分组语句.....	81
4.3.3 嵌套 if 语句.....	82

4.4 使用 switch 语句	87	6.1 处理错误	111
4.4.1 理解 switch 语句的语法	88	6.2 尝试执行代码和捕捉异常	111
4.4.2 遵守 switch 语句的规则	89	6.2.1 未处理的异常	113
小结	92	6.2.2 使用多个 catch 处理程序	113
第 4 章快速参考	92	6.2.3 捕捉多个异常	114
第 5 章 使用复合赋值和循环语句	94	6.2.4 传播异常	119
5.1 使用复合赋值操作符	94	6.3 使用 checked 和 unchecked 整数	
5.2 使用 while 语句	95	运算	121
5.3 编写 for 语句	100	6.3.1 编写 checked 语句	122
5.4 编写 do 语句	102	6.3.2 编写 checked 表达式	122
小结	109	6.4 引发异常	125
第 5 章快速参考	109	6.5 使用 finally 块	129
第 6 章 管理错误和异常	111	小结	130
		第 6 章快速参考	131

第 II 部分 理解 C# 对象模型

第 7 章 创建并管理类 and 对象	135	8.3.1 创建 ref 参数	162
7.1 理解分类	135	8.3.2 创建 out 参数	163
7.2 封装的目的	135	8.4 计算机内存的组织方式	165
7.3 定义并使用类	136	8.5 System.Object 类	167
7.4 控制可访问性	137	8.6 装箱	168
7.4.1 使用构造器	138	8.7 拆箱	168
7.4.2 重载构造器	139	8.8 数据类型的安全转换	170
7.5 理解静态方法和数据	146	8.8.1 is 操作符	170
7.5.1 创建共享字段	147	8.8.2 as 操作符	170
7.5.2 使用 const 关键字创建静态		小结	172
字段	148	第 8 章快速参考	173
7.5.3 静态类	148	第 9 章 使用枚举和结构创建值类型	175
7.5.4 匿名类	150	9.1 使用枚举	175
小结	151	9.1.1 声明枚举	175
第 7 章快速参考	152	9.1.2 使用枚举	175
第 8 章 理解值和引用	154	9.1.3 选择枚举文字常量值	176
8.1 复制值类型的变量和类	154	9.1.4 选择枚举的基本类型	177
8.2 理解 null 值和可空类型	159	9.2 使用结构	179
8.2.1 使用可空类型	160	9.2.1 声明结构	181
8.2.2 理解可空类型的属性	161	9.2.2 理解结构和类的区别	182
8.3 使用 ref 和 out 参数	162	9.2.3 声明结构变量	183

9.2.4 理解结构的初始化	184	12.2.6 声明重写方法	230
9.2.5 复制结构变量	187	12.2.7 理解受保护的访问	233
小结	191	12.3 理解扩展方法	239
第 9 章快速参考	191	小结	242
第 10 章 使用数组	192	第 12 章快速参考	243
10.1 声明和创建数组	192	第 13 章 创建接口和定义抽象类	245
10.1.1 声明数组变量	192	13.1 理解接口	245
10.1.2 创建数组实例	193	13.1.1 定义接口	246
10.1.3 填充和使用数组	194	13.1.2 实现接口	246
10.1.4 创建隐式类型的数组	194	13.1.3 通过接口来引用类	248
10.1.5 访问单独的数组元素	195	13.1.4 使用多个接口	248
10.1.6 遍历数组	196	13.1.5 显式实现接口	249
10.1.7 数组作为方法参数 和返回值传递	197	13.1.6 接口的限制	251
10.1.8 复制数组	198	13.1.7 定义和使用接口	251
10.1.9 使用多维数组	200	13.2 抽象类	260
10.1.10 创建交错数组	200	13.3 密封类	262
小结	210	13.3.1 密封方法	262
第 10 章快速参考	210	13.3.2 实现并使用抽象类	263
第 11 章 理解参数数组	212	小结	268
11.1 回顾重载	212	第 13 章快速参考	269
11.2 使用数组参数	213	第 14 章 使用垃圾回收和资源管理	271
11.2.1 声明参数数组	214	14.1 对象的生存期	271
11.2.2 使用 params object[]	216	14.1.1 编写析构器	272
11.2.3 使用参数数组	217	14.1.2 为什么要使用垃圾回收器	273
11.3 比较参数数组和可选参数	220	14.1.3 垃圾回收器的工作原理	275
小结	222	14.1.4 慎用析构器	275
第 11 章快速参考	222	14.2 资源管理	276
第 12 章 使用继承	223	14.2.1 资源清理方法	276
12.1 什么是继承	223	14.2.2 异常安全的资源清理	276
12.2 使用继承	224	14.2.3 using 语句和 IDisposable 接口	277
12.2.1 复习 System.Object 类	225	14.2.4 从析构器中调用 Dispose 方法	279
12.2.2 调用基类构造器	226	14.3 实现异常安全的资源清理	281
12.2.3 类的赋值	227	小结	287
12.2.4 声明新方法	228	第 14 章快速参考	288
12.2.5 声明虚方法	229		

第 III 部分 用 C# 定义可扩展类型

第 15 章 实现属性以访问字段	293	17.5.1 协变接口	350
15.1 使用方法实现封装	293	17.5.2 逆变接口	351
15.2 什么是属性	295	小结	353
15.2.1 使用属性	297	第 17 章快速参考	354
15.2.2 只读属性	297	第 18 章 使用集合	355
15.2.3 只写属性	298	18.1 什么是集合类	355
15.2.4 属性的可访问性	298	18.1.1 List<T>集合类	356
15.3 理解属性的局限性	299	18.1.2 LinkedList<T>集合类	358
15.4 在接口中声明属性	300	18.1.3 Queue<T>集合类	360
15.5 生成自动属性	305	18.1.4 Stack<T>集合类	361
15.6 使用属性来初始化对象	307	18.1.5 Dictionary<TKey, TValue> 集合类	362
小结	311	18.1.6 SortedList<TKey, TValue> 集合类	363
第 15 章快速参考	311	18.1.7 HashSet<T>集合类	364
第 16 章 使用索引器	313	18.2 使用集合初始化器	366
16.1 什么是索引器	313	18.3 Find 方法、谓词和 Lambda 表达式	366
16.1.1 不用索引器的例子	313	18.4 比较数组和集合	368
16.1.2 使用索引器的同一个例子	315	小结	372
16.1.3 理解索引器的访问器	317	第 18 章快速参考	373
16.1.4 对比索引器和数组	317	第 19 章 枚举集合	375
16.2 接口中的索引器	320	19.1 枚举集合中的元素	375
16.3 在 Windows 应用程序中使用 索引器	321	19.1.1 手动实现枚举器	376
小结	326	19.1.2 实现 IEnumerable 接口	380
第 16 章快速参考	327	19.2 使用迭代器来实现枚举器	382
第 17 章 泛型概述	328	19.2.1 一个简单的迭代器	382
17.1 object 的问题	328	19.2.2 使用迭代器为 Tree<TItem> 类定义枚举器	384
17.2 泛型解决方案	331	小结	386
17.2.1 对比泛型类与常规类	333	第 19 章快速参考	386
17.2.2 泛型和约束	334	第 20 章 分离应用程序逻辑并处理 事件	388
17.3 创建泛型类	334	20.1 理解委托	388
17.3.1 二叉树理论	334		
17.3.2 使用泛型构造二叉树类	337		
17.4 创建泛型方法	345		
17.5 可变性和泛型接口	348		

20.1.1 .NET Framework 类库的委托例子	389	21.2.5 使用查询操作符	426
20.1.2 自动化工厂的例子	391	21.2.6 查询 Tree<TItem>对象中的数据	429
20.1.3 不使用委托来实现工厂	391	21.2.7 LINQ 和推迟求值	434
20.1.4 使用委托来实现工厂	392	小结	438
20.1.5 声明和使用委托	394	第 21 章快速参考	438
20.2 Lambda 表达式和委托	402	第 22 章 操作符重载	440
20.2.1 创建方法适配器	402	22.1 理解操作符	440
20.2.2 Lambda 表达式的形式	403	22.1.1 操作符的限制	440
20.3 启用事件通知	404	22.1.2 重载的操作符	441
20.3.1 声明事件	405	22.1.3 创建对称操作符	442
20.3.2 订阅事件	406	22.2 理解复合赋值	444
20.3.3 取消订阅事件	406	22.3 声明递增和递减操作符	445
20.3.4 引发事件	406	22.4 比较结构和类中的操作符	446
20.4 理解用户界面事件	407	22.5 定义成对的操作符	447
小结	414	22.6 实现操作符	448
第 20 章快速参考	415	22.7 理解转换操作符	453
第 21 章 使用查询表达式来查询内存中的数据	418	22.7.1 提供内建转换	454
21.1 什么是语言集成查询	418	22.7.2 实现用户自定义的转换操作符	454
21.2 在 C# 应用程序中使用 LINQ	419	22.7.3 再论创建对称操作符	455
21.2.1 选择数据	420	22.7.4 添加隐式转换操作符	456
21.2.2 筛选数据	423	小结	458
21.2.3 排序、分组和聚合数据	423	第 22 章快速参考	459
21.2.4 联接数据	425		
第 IV 部分 使用 C# 构建 Windows 8 专业应用			
第 23 章 使用任务提高吞吐量	463	23.2.5 什么时候不要使用 Parallel 类	482
23.1 使用并行处理来执行多任务处理	463	23.3 取消任务和异常	484
多核处理器的崛起	464	23.3.1 协作式取消的原理	484
23.2 用 .NET Framework 实现多任务处理	465	23.3.2 为 Canceled 和 Faulted 任务使用延续任务	496
23.2.1 任务、线程和线程池	466	小结	496
23.2.2 创建、运行和控制任务	467	第 23 章快速参考	497
23.2.3 使用 Task 类实现并行处理	469	第 24 章 通过异步操作提高响应速度	499
23.2.4 使用 Parallel 类对任务进行抽象	478	24.1 实现异步方法	500