TURING 图灵程序设计丛书

Apress®

# Pro C# 5.0 and the .NET 4.5 Framework
## Sixth Edition

# 精通C#（第6版）

Jolt大奖提名图书，世界级C#专家之作，难以超越的畅销经典


[美] Andrew Troelsen 著
姚琪琳 朱晔 肖逵 张大磊 王少葵 范睿 等 译

人民邮电出版社
POSTS & TELECOM PRESS

TURING 图灵程序设计丛书

# Pro C# 5.0 and the .NET 4.5 Framework
## Sixth Edition

# 精通C# (第6版)



[美] Andrew Troelsen 著
姚琪琳 朱晔 肖逵 张大磊 王少葵 范睿 等 译



人民邮电出版社
北京

**图书在版编目（CIP）数据**

精通C# : 第6版 / (美) 特罗尔森 (Troelsen, A.) 著 ; 姚琪琳等译. -- 北京 : 人民邮电出版社, 2013.7
(图灵程序设计丛书)
书名原文: Pro C# 5.0 and the .NET 4.5 framework, sixth edition
ISBN 978-7-115-32181-7

Ⅰ. ①精… Ⅱ. ①特… ②姚… Ⅲ. ①C语言－程序设计 Ⅳ. ①TP312

中国版本图书馆CIP数据核字(2013)第133321号

## 内 容 提 要

本书是C# 领域久负盛名的经典著作，深入全面地讲解了C# 编程语言和 .NET 平台的核心内容，并结合大量示例剖析相关概念。全书分为八部分：C# 和 .NET 平台、C# 核心编程结构、C# 面向对象编程、高级 C# 编程结构、用 .NET 程序集编程、.NET 基础类库、WPF 和 ASP.NET Web Forms。第 6 版是对第 5 版的进一步更新和完善，内容涵盖了最先进的 .NET 编程技术和技巧，并准确呈现出 C# 编程语言的最新变化和 .NET 4.5 Framework 的新特性。

本书由微软 C# MVP Andrew Troelsen 编写，第 6 版专门针对 C# 5.0 和 .NET 4.5 进行了细致入微的修订，是各层次 .NET 开发人员的必读之作。

◆ 著　[美] Andrew Troelsen
译　姚琪琳　朱　晔　肖　逵　张大磊
王少葵　范　睿 等
责任编辑　刘美英
执行编辑　刘美英　李　洁
责任印制　焦志炜

◆ 人民邮电出版社出版发行　北京市崇文区夕照寺街14号
邮编　100061　电子邮件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
北京鑫正大印刷有限公司印刷

◆ 开本：800×1000　1/16
印张：76.75
字数：1813千字　2013年7月第1版
印数：1-4 000册　2013年7月北京第1次印刷
著作权合同登记号　图字：01-2012-7085号

定价：159.00元

读者服务热线：(010)51095186转604　印装质量热线：(010)67129223
反盗版热线：(010)67171154
广告经营许可证：京崇工商广字第 0021 号

# 前　言

大约 2001 年，我有幸获得一个机会，为即将兴起的一种微软技术撰写一本书，那时这种技术被称为 NGWS（Next Generation Windows Software，下一代视窗服务）。在检查微软提供的源代码的过程中，我发现大量代码注释与一种 COOL（Common Object Oriented Language）编程语言相关。

在我使用预览版创作 *C# and the .NET Platform*（第 1 版）初稿的过程中，NGWS 最终更名为微软.NET 平台，而 COOL，你应该能猜到，就是今天众所周知的 C#。

本书第 1 版与.NET 1.0 Beta 2 同时推出。从那以后，随着 C#编程语言的更新以及新发布的.NET 平台引入大量 API，本书内容也不断修订。

这些年来，本书得到了出版社（曾入围 Jolt 大奖提名，获得 2003 年 Referenceware 编程类图书卓越大奖）、读者、计算机科学和软件工程类大学课程的认可。对此我甚是感激。

更重要的是，很高兴收到全世界读者和教育工作者发来的邮件，与他们交谈感觉很棒。感谢所有人的意见、评论，当然还有批评。也许有些邮件我来不及回复，但我充分考虑了每一封邮件提出的问题，这一点可以保证。

## 你和我，我们是一个团队

技术作家所面对的是一群苛刻的读者（我知道，因为我就是他们中的一员）。无论使用什么平台，对部门、公司、客户和任何课题来说，构建软件解决方案都是非常复杂而且有针对性的事情。可能你在电子出版行业工作，或者为政府开发系统，或者是在科研机构或军队的某个部门工作。就我自己而言，我开发过儿童教育软件（Oregon Trail、Amazon Trail 等游戏软件）、各种 *n* 层系统以及许多医疗和金融行业的项目。你工作时编写的代码和我编写的代码百分之百是不同的（除非我们恰巧以前在一起工作）。

因此，在这本书中，我特意避免选择那些和具体行业紧密相关的例子，而是用与行业无关的例子来解释 C#、OOP、CLR 和.NET 基础类库。我不使用诸如数据填充表格、薪水计算或者其他的一些例子，而是坚持用与我们都有联系的主题：汽车，另外再加上几何结构和雇员薪水系统作为补充示例。你不用担心会有什么陌生的背景知识。

我要做的是尽最大可能解释 C#编程语言和.NET 平台的核心内容。同时，我会尽可能把进一步学习本书的工具和策略提供给你。

你要做的是理解这些内容并将其付诸具体编程工作中。我很清楚，你的项目可能与具有友好昵称的汽车（比如，BMW 的 Zippy 和 Yugo 的 Clunker）根本无关，但是所用到的知识是相通的。

放心，只要理解了这本书中的概念，你便能够很好地构建一个和实际编程环境紧密相关的.NET 解决方案。

## 本书内容

本书从逻辑上分为8个部分，每个部分包含一些相关联的章节。下面先按部分，再按章来分解本书。

### 第一部分：C#与.NET平台

第一部分的目的在于让你初步适应并了解.NET平台以及在构造.NET应用中用到的各种开发工具（其中很多是开源的）。

**第1章：.NET之道**

这一章讲述本书其余部分的脉络。该章的主要目的是介绍许多.NET相关的构建块，如CLR（公共语言运行库）、CTS（公共类型系统）、CLS（公共语言规范）以及基础类库。该章让你对C#编程语言和.NET程序集格式有一个初步了解，此外，本章还会介绍Windows 8操作系统中.NET平台的作用，并讲述Windows 8应用程序和.NET应用程序之间的区别。

**第2章：构建C#应用程序**

这一章介绍如何使用各种工具和技术来编译C#源代码文件。先介绍了如何使用命令行编译器（csc.exe）和C#响应文件，接着介绍了许多代码编辑器和IDE（集成开发环境），包括Notepad++、SharpDevelop、Visual C# Express以及Visual Studio；同时也会介绍如何通过在本地安装.NET Framework 4.5 SDK文档来配置自己开发用的电脑。

### 第二部分：C#核心编程结构

这部分很重要，因为所有类型的.NET软件开发都必须用到它，如Web应用、GUI桌面应用、代码库和Windows服务。你将会在这里了解.NET基本数据类型，学习文本处理以及各种C#参数修饰符的作用（包括可选参数和命名参数）。

**第3章：C#核心编程结构Ⅰ**

这一章正式开始研究C#编程语言，其中介绍了`Main()`方法的作用和.NET平台的内部数据类型，以及使用了`System.String`和`System.Text.StringBuilder`的文本数据操作。本章还讨论了迭代和选择构造、宽化和窄化操作以及`unchecked`关键字的用法。

**第4章：C#核心编程结构Ⅱ**

这一章完成了C#核心方面的研究，首先介绍重载类型方法的结构以及如何通过`out`、`ref`和`params`关键字定义参数，接着介绍了C#的两个特性——参数（argument）和可选参数（optional parameter），然后介绍了如何创建和操作数据数组，定义可空数据类型（使用?和??操作符）以及值类型（包括枚举和自定义结构）和引用类型之间的区别。

### 第三部分：C# 面向对象编程

从这一部分开始我们深入学习C#语言的核心构造，其中包括面向对象编程语言（OOP）的细节。此外，我们还要研究一下如何处理运行时异常，并详细阐述强类型接口的使用方法。

**第 5 章：封装**

从这一章开始，我们会研究如何使用 C#编程语言进行面向对象编程。在探讨了 OOP 的支柱概念（封装、继承和多态）之后，会介绍如何使用构造函数、属性、静态成员、常量以及只读字段来构建健壮的类类型。我们最后会研究分部类型定义、初始化对象的语法和自动属性。

**第 6 章：继承和多态**

这一章研究 OOP 其余的支柱概念（继承和多态），通过它，我们就能构建相关类类型的家族。我们还会研究虚方法、抽象方法（和抽象基类）以及多态接口的本质。这一章最后介绍.NET 平台最高层基类 `System.Object` 的作用。

**第 7 章：结构化异常处理**

这一章的关键在于讨论如何使用结构化异常处理来处理运行时的异常情况。你不但可以学到 C#控制这些异常的关键字（`try`、`catch`、`throw` 和 `finally`），还将了解应用程序级异常和系统级异常的区别。另外，该章还讲述了 Visual Studio 中不同的调试工具，这些工具能让你调试那些被忽略的异常。

**第 8 章：接口**

这一章的内容建立在对面向对象开发的理解之上，涵盖了基于接口的编程主题。你将学到如何定义类和支持多行为的结构，如何在运行时发现这些行为，以及如何使用显式接口实现来选择性地隐藏特定的行为。除了创建大量自定义接口外，还介绍了如何实现.NET 平台中的标准接口，以及使用这些接口构建可以排序、复制、枚举和比较的对象。

## 第四部分：高级C#编程结构

这部分介绍了许多重要的高级技术，能让你对 C#语言有更深的理解。通过对接口和委托的学习，你将会了解.NET 类型系统；同时你会学到泛型的作用，初步了解 LINQ，以及更多 C#高级特性（例如扩展方法、分部方法和指针操作）。

**第 9 章：集合与泛型**

这一章首先研究泛型（generic）的概念。你将看到，泛型编程提供了一种创建类型和类型成员的方式，它包含由调用者指定的变量占位符。简而言之，泛型编程大大加强了应用程序的性能和类型安全。你不仅可以在 `System.Collections.Generic` 命名空间中看到各种泛型，而且可以学习如何创建自己的泛型方法和类型（有限制或没有限制）。

**第 10 章：委托、事件和 Lambda 表达式**

这一章的目的在于阐明委托（delegate）类型。可以简单地认为，一个.NET 委托就是指向应用程序中其他方法的一个对象。使用这个类型，可以构建允许多个对象进行双向会话的系统。在分析了.NET 委托的使用后，将介绍 C#的 `event` 关键字，使用这个关键字可以简化原始委托编程的操作。随后研究 C# Lambda 操作符`=>`，并探讨了委托、匿名方法和 Lambda 表达式之间的联系。

**第 11 章：高级C#语言特性**

这一章介绍了许多高级的.NET 编程技术，这些技术能够让你对 C#编程语言有更深的理解。例如，你将学到如何重载操作符、创建自定义类型转换例程（包括隐式的和显式的）、构建类型索引器并与之交互，使用扩展方法、匿名类型、分部方法以及使用不安全的代码上下文来操作 C#指针。

**第 12 章：LINQ to Object**

这一章开始研究 LINQ（语言集成查询），它可以用来构建强类型的查询表达式，并且可以把它应用到很多 LINQ 目标来操作最广义的数据。我们还会学习把 LINQ 表达式应用到数据容器（例如，数组、集合和自定义类型）的 LINQ to Object。这对于本书其余部分所涉及的其他 LINQ API 同样适用（如 LINQ to XML、LINQ to DataSet、PLINQ 和 LINQ to Entity）。

**第 13 章：对象的生命周期**

这一章分析了 CLR 如何使用.NET 垃圾收集器来管理内存。你将会了解应用程序根、对象产生和 System.GC 类型的作用。理解了这些之后，接下来学习可处置的对象（通过 IDisposable 接口）和终结过程（通过 System.Object.Finalize()方法）。这一章还将研究 Lazy<T>类，它用来定义一个直到调用者发出请求时才会分配的数据。你将看到，当你的程序不希望托管堆散乱不堪时，这个特性将有助于确保堆的整洁。

## 第五部分：用.NET程序集编程

这一部分深入分析了.NET 程序集格式的细节。你不仅会学到如何部署和配置.NET 代码库，而且会理解.NET 二进制映像的内部结构。这部分也阐述了.NET 特性的作用和运行时解析类型信息的作用，还介绍了动态语言运行时（DLR, Dynamic Language Runtime）和 C# dynamic 关键字的作用。后面的章节分析了一些程序集相关的高级主题（如应用程序域、CIL 语法和在内存中构建程序集）。

**第 14 章：.NET 程序集入门**

从一个比较高的层次来看，程序集（assembly）是用于描述托管的*.dll或*.exe二进制文件的一个术语，但是其真正内涵实际上远远不仅于此。通过这一章，你将学到单文件程序集和多文件程序集的区别，以及如何构建和部署每一个实体；你还会学到如何利用基于XML的*.config文件和发布策略程序集来配置私有的和共享的程序集。通过这些内容，你将看到GAC（全局程序集缓存）的内部结构。

**第 15 章：类型反射、晚期绑定和基于特性的编程**

这一章通过 System.Reflection 命名空间，分析了运行时类型发现的过程，来继续探讨.NET 程序集。使用这些类型，你能够创建一个可以实时读取程序集元数据的应用程序。该章还将介绍如何在运行时使用晚期绑定来动态加载和创建类型。最后的主题是.NET 特性（包括标准的和自定义的）的作用。为了说明每个主题的用法，该章最后将构造一个可扩展的 Windows Forms 应用程序。

**第 16 章：动态类型和动态语言运行时**

.NET 4.0 引入了一个新的.NET 运行时环境，叫做动态语言运行时。使用 DLR 和 C# 2010 dynamic 关键字，可以定义直到运行时才真正处理的数据。这些特性可以显著地简化一些极其复杂的.NET 编程任务。在这一章中，你将学习一些动态数据的实际用法，包括如何用简单的方式使用.NET 反射 API，以及如何用最小的代价与遗留的 COM 库通信。

**第 17 章：进程、应用程序域和对象上下文**

既然你已经对程序集有了一定的了解，这一章将深入探讨加载的.NET 执行单元的组成。这一章的目标是阐明进程、应用程序域和上下文边界的关系。该章所叙述的内容为第 19 章做了很好的铺垫，那一章讨论的是多线程应用程序的构造。

**第 18 章：CIL 和动态程序集的作用**

这一章有两个目的。在前半部分（大致）中，将会比之前章节更具体地介绍 CIL 的语法和语义，余下的部分主要讲述 System.Reflection.Emit 命名空间的作用。使用这些类型，可以构建一个能在运行时在内存中产生.NET 程序集的软件。正式地说，一个能在内存中定义并执行的程序集称为动态程序集。

## 第六部分：.NET基础类库

到本书的这一部分，你应该已经很好地掌握了 C#语言以及.NET 程序集格式的细节。这一部分将通过探索基础类库中的一些常用服务程序来讲授一些新的知识，包括多线程应用程序的创建、文件的输入/输出和利用 ADO.NET 的数据库访问，通过 WCF 构造分布式应用程序以及构建使用了 WF API 和 LINQ to XML API 的支持工作流的应用程序。

**第 19 章：多线程、并行和异步编程**

这一章介绍了如何构建多线程应用程序，演示了大量用于编写线程安全代码的技术。首先复习了.NET 委托类型，解释了委托对于异步方法调用的内在支持。接下来，研究了 System.Threading 命名空间中的类型。最后，介绍了 TPL（Task Parallel Library，任务并行库）。使用 TPL，.NET 开发者可以用一种极其简单的方式，将应用程序的工作分配给所有可用的 CPU。与此同时，你还将学习 PLINQ 的作用，它提供了一种在多个机器内核中执行 LINQ 查询的方法。本章结尾处总结了.NET 4.5 引入的几个新的 C#关键字，它们能够将异步方法调用直接集成到语言中。

**第 20 章：文件输入输出和对象序列化**

System.IO 命名空间允许与机器的文件和目录结构交互。通过这一章的学习，你将学会如何以编程方式创建（和删除）一个目录系统，以及如何将数据从不同的（例如，基于文件的、基于字符串的、基于内存的等）数据流中移进移出。本章的后半部分探讨了.NET 平台的对象序列化服务。简单地说，序列化（serialization）就是将一个对象（或一组相关对象）的状态持久化为流，以便今后使用。反序列化（Deserialization）是一个从流中取出对象并放入供应用程序使用的内存中的过程。只要理解了基本原理，你就会学到如何通过 ISerializable 接口和一组.NET 特性来定制序列化过程。

**第 21 章：ADO.NET 之一：连接层**

本书中有 3 章介绍数据库相关操作，这是第一章，介绍.NET 平台 ADO.NET 的数据库访问 API。具体而言，涉及.NET 数据提供程序的作用以及如何使用由连接对象、命令对象、事务对象以及数据读取器对象构成的 ADO.NET 连接层与关系数据库进行通信。该章引导你创建一个在本书剩余部分会用到的自定义数据库和自定义数据访问类库（AutoLotDAL.dll）。

**第 22 章：ADO.NET 之二：断开连接层**

这一章将通过研究 ADO.NET 的*断开连接层*继续探讨数据库操作。在这里，我们会学习 DataSet 类型、数据适配器对象以及各种可以大幅简化创建数据驱动应用程序的 Visual Studio 工具。此后，我们会学习如何把 DataTable 对象绑定到用户界面元素以及如何使用 LINQ to DataSet 将 LINQ 查询应用于内存对象 DataSet。

**第 23 章：ADO.NET 之三：Entity Framework**

这一章研究了 ADO.NET 的最后一部分内容——Entity Framework（EF）的作用。实质上，EF 是

一种用直接映射到业务模型上的强类型类编写数据访问代码的方式。在这里，你将了解 EF 对象服务的作用、实体客户端和对象上下文，以及*.edmx 文件的构成。同时，还将学习如何使用 LINQ to Entity 与关系型数据库交互。你还将构建最终版本的自定义数据访问库（AutoLotDAL.dll），本书其他章节中将会使用这个库。

**第 24 章：LINQ to XML 简介**

第 14 章介绍核心的 LINQ 编程模型，特别是 LINQ to Object。在这里，我们会通过研究如何把 LINQ 查询应用到 XML 文档来深入理解 LINQ。我们首先学习在使用 System.XML.dll 程序集的类型进行.NET XML 操作时所暴露出的一些“缺点”，然后探索如何使用 LINQ 编程模型（LINQ to XML）在内存中创建 XML 文档、将文档持久化到硬盘驱动以及对其内容进行导航。

**第 25 章：WCF**

到现在为止，所有的示例程序都是在一台计算机上执行的。在这一章中，我们将学习WCF API以便无须考虑底层管道，以系统的方式构建分布式应用程序。该章会介绍WCF服务、服务器端以及客户端的构建。我们还会看到，WCF服务非常灵活，在客户端和服务器端中都可以使用基于XML的配置文件以声明方式指定地址、绑定和契约。

**第 26 章：Windows Workflow Foundation 简介**

本章你首先会学习启用工作流的应用程序的作用，随后了解使用.NET 4.0 WF API 对业务流程进行建模的不同方式，接下来学习 WF 活动库的范围；以及如何构建自定义活动。在这个活动中，将使用本书前面所创建的自定义数据库访问库。

## 第七部分：WPF

.NET 3.0 向程序员介绍了一个绝妙的 API——WPF（Windows Presentation Foundation），该 API 很快就成为 Windows Forms 桌面编程模型的继承者。实质上，WPF 所构建的桌面应用程序可以包含向量图形、交互式动画，以及使用声明式标记语法 XAML 所进行的数据绑定操作。

此外，WPF 控件架构提供了一种非常简单的方式，仅仅使用一些格式良好的 XAML，就可以彻底改变常用控件的外观。

**第 27 章：WPF 和 XAML**

本质上，WPF 可以为桌面应用程序（以及非直接的 Web 应用程序）构建交互性极好的富媒体前端。和 Windows Forms 不同，这个超强的 UI 框架把很多关键服务（例如 2D 和 3D 图形、动画、富文档等）整合到了一个统一的对象模型中。在这一章中，我们会从 WPF 以及 XAML（可扩展应用程序标记语言）开始研究，还会学习如何创建不使用 XAML、只使用 XAML 以及两者结合的 WPF 程序。最后，我们创建了在其余 WPF 相关章节中都会用到的自定义 XAML 编辑器。

**第 28 章：使用 WPF 控件编程**

这一章将介绍使用内置的 WPF 控件和布局管理器的过程，例如，构建菜单系统、拆分窗口、工具条和状态栏，还将介绍大量 WPF API（及其相关控件），包括 WPF Documents API、WPF Ink API 和数据绑定模型。同样重要的是，这一章将开始研究 Expression Blend IDE，它将简化为 WPF 应用程序创建富 UI 的任务。

**第 29 章：WPF 图形呈现服务**

WPF 是一个图形密集型 API，它提供了 3 种呈现图形的方式：形状、绘图和几何图形、可视化。在这一章中，我们将介绍这几种方法，并讨论大量重要的图形基元（如画刷、画笔和图形变换），还将学习 Expression Blend 用于简化创建 WPF 图形过程的多种方式，以及如何对图形数据执行命中测试操作。

**第 30 章：WPF 资源、动画和样式**

这一章介绍了 3 个重要的（也是相互关联的）主题，它们将加深你对 WPF API 的理解。第一个主题是逻辑资源的作用。你将看到逻辑资源（也叫对象资源）系统提供了一种方式，可以在 WPF 应用程序中命名和引用常用的对象。接下来，你将学习如何定义、执行和控制动画序列。你可能会认为 WPF 动画仅局限于视频游戏和多媒体应用，但实际上绝不是这样。最后你将学习 WPF 样式的作用。与使用 CSS 或 ASP.NET 主题引擎的 Web 页面类似，WPF 应用程序也可以定义常用控件的外观。

**第 31 章：依赖属性、路由事件和模板**

本章先介绍了创建自定义控件时涉及的两个重要话题：依赖属性和路由事件。理解了这些内容之后，我们将学习默认模板的作用，以及如何在运行时以编程的方式查看它们。打好这些基础之后，最后将学习如何创建自定义模板。

## 第八部分：ASP.NET Web Form

这一部分主要研究使用 ASP.NET 编程 API 来构建 Web 应用程序。我们会看到，ASP.NET 基于标准的 HTTP 请求/响应对事件驱动的面向对象框架分层，以此来对桌面用户界面的创建进行建模。

**第 32 章：ASP.NET Web Form**

本章开始介绍使用 ASP.NET 进行 Web 应用开发。如你所见，服务器端的脚本代码现在由“真正的”面向对象的语言（如 C#和 VB .NET 等）所替代。这一章将介绍 ASP.NET 网页的构造、基础的编程模型以及 ASP.NET 的其他关键主题，如 Web 服务器的选择和 web.config 文件的使用。

**第 33 章：ASP.NET Web 控件、母版页和主题**

由于前几章介绍了 ASP.NET 页面对象的构建，这一章将会关注组成内部控件树的控件。在这里，我们会研究包括验证控件、内置站点导航控件以及各种数据绑定操作在内的核心 ASP.NET Web 控件。同样，还会演示母版页的作用以及与传统样式表对应的服务器端 ASP.NET 主题引擎。

**第 34 章：ASP.NET 状态管理技术**

这一章将通过研究.NET 下处理状态管理的多种方式来扩展你对 ASP.NET 的理解。和传统的 ASP 一样，使用 ASP.NET 可以轻松创建 cookie 以及应用程序级变量和会话级变量。然而，ASP.NET 还引入了一项新的状态管理技术：应用程序高速缓存。只要知道了使用 ASP.NET 处理状态的多种方式，你就会明白 `HttpApplication` 基类的作用，以及如何用 web.config 文件动态改变 Web 应用程序的运行时行为。

## 附录

除了 34 章正文，本书英文版还包括附录 A 和附录 B，其内容需要从 Apress 网站（www.apress.com）的本书主页下载。附录 A 涵盖基本 Windows Forms API，正文中有几个 UI 示例用到了。附录 B 通过

Mono 平台研究了.NET 不依赖于平台的特性。

**附录 A：Windows Forms 编程**

.NET 平台最初发布的桌面 GUI 工具包是 Windows Forms。该附录将介绍这个 UI 框架的作用，并演示如何构建主窗口、对话框和菜单系统，还将说明窗体继承的作用，以及如何使用 `System.Drawing` 命名空间呈现二维图形数据。最后，该附录将构建一个绘图程序（半成品），演示附录中讨论的各个主题。

**附录 B：使用 Mono 进行平台无关的.NET 开发**

最后，附录 B 介绍了一个叫做 Mono 的.NET 平台的开源实现。使用 Mono 可以在 Mac OS X、Solaris 以及各种 Linux 版本的操作系统中创建、部署和执行富特性的.NET 应用程序。由于 Mono 和微软的.NET 平台非常相似，因此你应该知道 Mono 提供的大部分内容。因此，附录 B 会重点介绍 Mono 的安装过程、Mono 的开发工具以及 Mono 运行时引擎。

## 本书源代码

本书所包含的所有代码示例都可以从 Apress 网站上的 Source Code/Download 中免费下载（也可以从图灵社区本书主页 http://www.ituring.com.cn/book/1046 免费下载）。访问网址 http://www.apress.com，选择 Source Code/Download 链接，然后按书名查找。找到本书的主页后，就可以下载一个压缩的*.zip 文件。解开压缩文件就会看到，所有代码都是按章编排的。

需要提醒你注意的是，本书的很多章节都包含有如下所示的源代码说明，书中讨论的例子都可以依此线索下载，并加载到 Visual Studio 中，以便进一步讨论和修改。

---

**源代码** 这里给出了源代码在压缩文件中的具体目录。

---

要打开一个 Visual Studio 解决方案，可以使用 File→Open→Project/Solution 菜单选项，然后导航到解压缩文档所在的子目录，找到正确的*.sln 文件。

## 勘误信息

阅读本书过程，你或许偶尔会发现一些语法错误或代码错误（很显然我不希望看到这些）。如果真发现了，我在此道歉。作为一个凡人，尽管我已经很尽力了，但是一两个小错误总是难免的。你可以从 Apress 的网页上获得勘误表（还是在这本书的“主页”上）。如果你发现错误的话，请在那里找到我的联系方式，与我联系。

## 联系作者

如果你有任何关于本书源代码的问题，或者需要进一步阐明这里所举的例子，亦或者只是想简单地向我传达你关于.NET 平台的想法，请通过以下电子邮件地址与我联系：atroelsen@intertech.com（为了确保你的邮件不会被我的信箱划为垃圾邮件，请在主题栏中包含“C# SixthEd”）。

请你们相信，我会尽我所能在较短的时间里回复你们。但是，就如同各位一样，我有时也会比较忙。如果我没能在一周或两周的时间里回复你们，请不要认为我是一个怪异的人或者不屑与你们交流，我可能只是比较忙而已（或者，如果足够幸运的话，在某处度假也不一定）。最后，感谢购买本书（或者至少在你决定是否购买的时候，曾经在书店里翻看过）。我希望你喜欢阅读本书，并且可以灵活运用书中所学的知识。

Andrew Troelsen

# 目　录

## 第一部分　C#与.NET平台

## 第二部分 C#核心编程结构

## 第三部分　C#面向对象编程