

O'REILLY®

TURING

图灵程序设计丛书

第4版



# C#经典实例

C# 6.0 Cookbook

针对C# 6.0和.NET Framework 4.6全面更新；涵盖C#开发的各类陷阱和问题

[美] Jay Hilyard Stephen Teilhet 著  
徐敬德 译



中国工信出版集团



人民邮电出版社  
POSTS & TELECOM PRESS



图灵程序设计丛书

# C#经典实例（第4版）

## C# 6.0 Cookbook

[美] Jay Hilyard Stephen Teilhet 著  
徐敬德 译

O'REILLY®

*Beijing • Cambridge • Farnham • Köln • Sebastopol • Tokyo*

O'Reilly Media, Inc. 授权人民邮电出版社出版

人民邮电出版社

北 京

## 图书在版编目 (C I P) 数据

C#经典实例：第4版 / (美) 杰伊·希尔亚德  
(Jay Hilyard), (美) 斯蒂芬·泰耶  
(Stephen Teilhet) 著; 徐敬德译. -- 北京: 人民邮  
电出版社, 2016.10

(图灵程序设计丛书)

ISBN 978-7-115-43509-5

I. ①C… II. ①杰… ②斯… ③徐… III. ①C语言—  
程序设计 IV. ①TP312.8

中国版本图书馆CIP数据核字(2016)第216681号

## 内 容 提 要

本书共分为13章,每一章侧重于特定主题的C#解决方案。具体内容包括:类和泛型,集合、枚举器和迭代器,数据类型,语言集成查询和lambda表达式,调试和异常处理,反射和动态编程,正则表达式,文件系统I/O,网络和Web,XML,安全,线程、同步和并发,工具箱。本书使用大量范例,帮助开发人员快速理解并解决现实中的问题。

本书面向所有级别的C#和.NET开发人员。

---

◆ 著 [美] Jay Hilyard Stephen Teilhet

译 徐敬德

责任编辑 朱 巍

执行编辑 杨 琳 赵瑞琳

责任印制 彭志环

◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路11号

邮编 100164 电子邮件 315@ptpress.com.cn

网址 <http://www.ptpress.com.cn>

北京昌平百善印刷厂印刷

◆ 开本: 800×1000 1/16

印张: 37

字数: 874千字

2016年10月第1版

印数: 1-3 500册

2016年10月北京第1次印刷

著作权合同登记号 图字: 01-2016-4798号

---

定价: 129.00元

读者服务热线: (010)51095186转600 印装质量热线: (010)81055316

反盗版热线: (010)81055315

广告经营许可证: 京东工商广字第8052号

---

# 版权声明

© 2015 by Jay Hilyard, Stephen Teilhet.

Simplified Chinese Edition, jointly published by O'Reilly Media, Inc. and Posts & Telecom Press, 2016. Authorized translation of the English edition, 2016 O'Reilly Media, Inc., the owner of all rights to publish and sell the same.

All rights reserved including the rights of reproduction in whole or in part in any form.

英文原版由 O'Reilly Media, Inc. 出版，2015。

简体中文版由人民邮电出版社出版，2016。英文原版的翻译得到 O'Reilly Media, Inc. 的授权。此简体中文版的出版和销售得到出版权和销售权的所有者——O'Reilly Media, Inc. 的许可。

版权所有，未得书面许可，本书的任何部分和全部不得以任何形式重制。

---

# O'Reilly Media, Inc.介绍

O'Reilly Media 通过图书、杂志、在线服务、调查研究和会议等方式传播创新知识。自 1978 年开始，O'Reilly 一直都是前沿发展的见证者和推动者。超级极客们正在开创着未来，而我们关注真正重要的技术趋势——通过放大那些“细微的信号”来刺激社会对新科技的应用。作为技术社区中活跃的参与者，O'Reilly 的发展充满了对创新的倡导、创造和发扬光大。

O'Reilly 为软件开发人员带来革命性的“动物书”；创建第一个商业网站（GNN）；组织了影响深远的开放源代码峰会，以至于开源软件运动以此命名；创立了 Make 杂志，从而成为 DIY 革命的主要先锋；公司一如既往地通过多种形式缔结信息与人的纽带。O'Reilly 的会议和峰会集聚了众多超级极客和高瞻远瞩的商业领袖，共同描绘出开创新产业的革命性思想。作为技术人士获取信息的选择，O'Reilly 现在还将先锋专家的知识传递给普通的计算机用户。无论是通过书籍出版、在线服务或者面授课程，每一项 O'Reilly 的产品都反映了公司不可动摇的理念——信息是激发创新的力量。

## 业界评论

“O'Reilly Radar 博客有口皆碑。”

——Wired

“O'Reilly 凭借一系列（真希望当初我也想到了）非凡想法建立了数百万美元的业务。”

——Business 2.0

“O'Reilly Conference 是聚集关键思想领袖的绝对典范。”

——CRN

“一本 O'Reilly 的书就代表一个有用、有前途、需要学习的主题。”

——Irish Times

“Tim 是位特立独行的商人，他不光放眼于最长远、最广阔的视野，并且切实地按照 Yogi Berra 的建议去做了：‘如果你在路上遇到岔路口，走小路（岔路）。’回顾过去，Tim 似乎每一次都选择了小路，而且有几次都是一闪即逝的机会，尽管大路也不错。”

——Linux Journal

---

# 前言

C# 是一门面向 Microsoft .NET 平台开发者的语言。Microsoft 将 C# 描述为一种用于 .NET 平台开发的富于创新的现代化语言，并且在 C# 6.0 中增添了用于支持动态编程、并行编程以及编写更少代码的新功能，进一步实现了这个目标。C# 不仅同时支持声明式编程和函数式编程，还包含了强大的面向对象特性。简而言之，用 C# 可以针对特定问题采取不同的编程风格。

我们基于自己最初学习 C# 时遇到的编程问题开始了本书的编写，并且根据 C# 语言中的新问题和新功能不断地扩展内容。在这一版中，我们重新编写了许多解决方案，以充分利用 C# 最近的创新，例如新的表达式级别功能（nameof、字符串插值、null 条件运算符、索引初始值设定项）、成员声明功能（自动属性初始值设定项、getter-only 自动属性、表达式-函数体成员）和语句级别功能（异常过滤器）。同时，我们在原有及新的范例中纳入了动态编程（C# 4.0）和异步编程（C# 5.0）的新应用，帮助读者了解如何应用这些语言特性。

无论是首次学习 C#，还是探索其新的能力，抑或是处理开发周期中较为罕见的问题，每个人都会遇到一些常见（和不那么常见的）陷阱和问题；希望我们的以上增补能帮助读者解决难题。此外，尽管 Microsoft 已经提供了大量的功能以避免人们“重复创造轮子”，但是我们仍然发现 .NET Framework 类库（Framework class library, FCL）中有一些缺少的内容，并将其纳入了本书的范例。一些解决方案你可能马上会用到，也有一些你可能永远都用不到，但是无论如何，我们都希望这本书能够帮助你尽可能充分地利用 C# 和 .NET Framework。

本书的内容按照一个 C# 程序员在学习过程中要解决的问题类型来进行组织。这些解决方案称为范例（recipe）；每个范例都包含一个问题、其解决方案、对解决方案的讨论和其他相关信息，最后是一个资源列表，包括在 FCL 的哪里可以找到相关类的更多信息、相关文章和其他范例。这种问答模式提供了问题的完整解决办法，使得本书易于阅读和使用。几乎每个范例都包含完整的、有文档的代码示例，向读者展示如何解决特定的问题，同时也讨论了底层技术如何运作，并且列出了替代技术、限制条件和应用时要考虑的其他事项。

## 读者对象

即使你不是经验丰富的 C# 或 .NET 开发人员，也可以使用本书——本书面向所有级别的读

者。本书既提供了开发人员日常问题的解决方案，也包含了一些出现频率较低的问题。书中范例针对的是现实开发人员需要立刻解决的问题，不需要首先学习大量理论知识。虽然参考书和教程类书籍可以讲述通用概念，但通常不提供读者解决实际问题所需的帮助。我们选择通过示例来教学，因为这是大多数人自然的学习方式。

本书中解决的大多数问题是 C# 开发人员会频繁面对的，但有一些更高级的问题需要结合多项技术的复杂方案。每个范例旨在帮助开发人员快速理解问题，学习如何解决，并找出任何潜在的权衡选择来帮助你快速、高效、轻松地解决问题。

为了免去读者手动输入解决方案的麻烦，我们在 O'Reilly 的网站上提供了本书的示例代码，以方便“编辑继承”模式的开发（复制和粘贴），同时有助于经验较少的开发人员看到优秀的编程实践。示例代码提供了利用到每个解决方案的可运行测试，不过本书也在每个解决方案中包含了足够的代码，读者不使用示例代码也能够实现解决方案。示例代码可以从本书的产品页面（[https://github.com/oreillymedia/c\\_sharp\\_6\\_cookbook](https://github.com/oreillymedia/c_sharp_6_cookbook)）上获取<sup>1</sup>。

## 硬件和软件要求

要运行本书中的示例，你需要一台运行 Windows 7 或更高版本的计算机。一部分网络和 XML 解决方案需要 Microsoft IIS 7.5 或更高版本，第 9 章中 FTP 的范例需要一个本地配置好的 FTP 服务器。

打开和编译本书中的示例需要 Visual Studio 2015。如果你精通可下载的 Framework SDK 及其命令行编译器，也可以顺利地使用本书和示例代码。

## 平台说明

本书中的解决方案都是使用 Visual Studio 2015 开发的。C# 6.0 和 C# 3.0 之间的差异是非常显著的，本书的示例代码与第 3 版中有很大不同，反映了其中的差异。

值得一提的是，尽管 C# 现在已经是 6.0 版，.NET Framework 则仍是 4.6 版。C# 随着 .NET Framework 的每次发布持续创新，现在的 C# 6.0 中包含许多功能，使得开发人员可以用最适合手头任务的风格进行编程。

## 内容结构

本书共分为 13 章，每一章侧重于特定主题的 C# 解决方案。下面总结了每一章的要点，概述了本书的内容。

- 第 1 章 类和泛型

这一章篇幅很长，包含处理类和结构数据类型的范例，以及泛型的使用。泛型能够让你的代码在不同类型的值上运行一致。这一章涵盖的范例范围很广，包括闭包、类转换、完善的命令行参数处理系统以及类设计的主题。有的范例增强了读者对泛型的整

---

注 1：示例代码也可以在图灵社区本书主页（<http://www.ituring.com.cn/book/1746>）下载。——编者注

体理解；有的范例涵盖了泛型何时适用，框架中提供了哪些支持，以及如何实现自定义集合。

- 第2章 集合、枚举器和迭代器

这一章范例考察了集合、枚举器和迭代器的用法。集合的范例使用了数组（单维、多维和锯齿状）、`List<T>` 以及其他集合类，并且扩展了它们的功能。这一章讨论了泛型集合以及创建自定义强类型集合的多种方式。我们探讨了自定义枚举器的创建，向读者展示了如何为泛型和非泛型类型实现迭代器以及如何使用迭代器实现 `foreach` 功能，并涵盖了自定义迭代器实现。

- 第3章 数据类型

这一章包括字符串、数字和枚举。这些范例展示了如何完成诸如编码 / 解码字符串、执行数值转换，以及测试字符串以确定它们是否包含数字值的任务。我们还介绍了如何显示、转换和测试枚举类型，以及如何使用包含位标志的枚举。

- 第4章 语言集成查询和 `lambda` 表达式

这一章涵盖语言集成查询（language integrated query, LINQ）及其用法，包括并行 LINQ（parallel LINQ, PLINQ）的一个示例。有些范例使用了诸多标准查询运算符，也有些展示了如何使用功能强大但并不是语言关键字的查询运算符。这一章也对 `lambda` 表达式进行了探讨，通过范例展示如何用其代替旧风格的委托。

- 第5章 调试和异常处理

这一章介绍调试和异常处理。我们提供了使用 `System.Diagnostics` 命名空间下数据类型的范例，例如事件日志、进程、性能计数器和类型的自定义调试器显示。我们同时关注了在应用程序中实现异常处理的最佳方式。这一章还包括了避免未处理异常，读取和显示栈跟踪，引发和重新引发异常的范例。最后，我们提供的范例展示了如何克服某些棘手的情形，如后期绑定调用的方法中出现的异常和异步异常处理。

- 第6章 反射和动态编程

这一章展示了如何使用 .NET Framework 内置的程序集检视系统，确定一个程序集中实现了哪些类型、接口和方法，以及如何以后期绑定的方式访问它们。这一章也说明了如何使用 `dynamic`、`ExpandoObject` 和 `DynamicObject` 在应用程序中实现动态编程。

- 第7章 正则表达式

这一章介绍了一组有用的类，用于对字符串运行正则表达式。范例包括列举正则表达式匹配，将字符串解析为一组标记，查找 / 替换字符，以及验证正则表达式的语法。此外我们还加入了一个范例，其中包含许多常见的正则表达式模式。

- 第8章 文件系统 I/O

这一章涉及与文件系统交互的三种不同方式：典型的文件交互；基于目录或文件夹的交互；文件系统 I/O 的高级主题。

- 第9章 网络和 Web

这一章探讨了由 .NET Framework 提供的连接选项，以及如何以编程方式访问网络资源和 Web 上的内容。这一章中的范例包含了直接使用 TCP/IP，使用命名管道通信，构建自己的端口扫描程序，以编程方式确定网站配置，等等。

- 第10章 XML

如果你在使用 .NET，那么很可能需要在一定程度上处理 XML。在这一章中，我们将探讨 XML 的一些作用，以及如何使用 LINQ to XML、XmlReader/XmlWriter 类和 XmlDocument 类来进行 XML 编程。这一章包含了使用 XPath 和 XSLT 的示例，以及验证 XML、将 XML 转换到 HTML 等主题。

- 第11章 安全

编写不安全代码的方式有很多种，但仅有少数几个途径可以编写安全的代码。在这一章中，我们探讨了类型的访问控制、加密和解密、安全地存储数据、程式安全声明式安全等领域。

- 第12章 线程、同步和并发

这一章的主题是在 .NET 程序中使用多个执行线程，讨论的问题有：在应用中实现多线程，避免资源的并行访问，允许安全的并行访问，存储每个线程的数据，顺序执行任务，在 .NET 中使用同步原语以编写线程安全的代码，等等。

- 第13章 工具箱

这一章包含的范例是开发人员会反复遇到的随机操作类型，例如确定系统资源的位置，发送电子邮件，以及使用服务。这一章还包括一些较少用到但非常有用的应用程序块，如消息队列，在单独的应用程序域中运行代码，以及在全局程序集缓存（global assembly cache，GAC）中查找应用程序集的版本。

有一些范例是相关联的；在这类范例中，参阅部分和讨论部分的文本中会注明这些关联关系。

## 未涉及的内容

这本书并不是 C# 的参考手册或入门书。O'Reilly 出版了一些优秀的入门书和参考手册，如 Joseph Albahari 和 Ben Albahari 的 *C# 6.0 in a Nutshell* (<http://shop.oreilly.com/product/0636920040323.do>) 和 *C# 6.0 Pocket Reference* (<http://shop.oreilly.com/product/0636920040675.do>)，以及 Stephen Cleary 的《C# 并发编程经典实例》(*Concurrency in C# Cookbook*, <http://shop.oreilly.com/product/0636920030171.do>)。MSDN 库也是极为有用的。它包含在 Visual Studio 2015 中，也可以在网站 <http://msdn.microsoft.com> 上在线查看。

## 排版约定

在本书中将会使用以下排版约定。

- 等宽字体 (*Constant width*)

用于程序清单和代码元素，如命令、选项、开关、变量、特性、键、函数、类型、类、命名空间、方法、模块、属性、参数、值、对象、事件、事件处理器、XML 标记、HTML 标记、宏、文件的内容和命令输出。

- 加粗等宽字体 (*Constant width bold*)

用于在程序清单中突出显示代码中的重要部分。

- 斜体等宽字体 (*Constant width italic*)

用于指示代码中可替换的部分。

- *//...*

C# 代码中的省略号表示为了段落清晰而省略掉的文本。

- *<!--...-->*

在 XML 模式和文档的代码中的省略号表示为了段落清晰而省略掉的文本。



此标志指示提示、建议或一般注意事项。



此标志表示警告或警示。

## 关于代码

本书中几乎每个范例都包含一个或多个代码示例。这些示例包含在解决方案中，代码片段和完整项目都可以直接用于你的应用程序。大多数代码示例都写在一个类或结构中，使得它们更易于在应用程序中使用。除此之外，所有的 `using` 指令都包含在各个范例中，因此你不需要查明在你的代码中要包含哪个命名空间。

只有关键的部分才包括完整的错误处理，例如输入参数。这允许你轻松地查看什么是正确的输入，什么是错误的输入。许多范例省略了错误处理，关注重点概念会使得解决方案更易于理解。

## 使用代码示例

这本书的示例代码可从网页 [https://github.com/oreillymedia/c\\_sharp\\_6\\_cookbook](https://github.com/oreillymedia/c_sharp_6_cookbook) 上获取。

这本书是用来帮助你完成工作的。一般来说，你可以在自己的程序和文档中使用本书的代码。你没有必要联系我们来获得授权，除非想对代码做出大规模的重构。比如，使用本书中的若干程序来编写自己的代码是无需授权的，但是销售或分发 O'Reilly 出版书籍配套光盘中的代码是需要授权的；引用本书中的内容或示例代码来回答问题是无需授权的；而把本书中的大量代码合并到你的产品文档中就需要授权。

我们并不要求你注明引用内容的出处，但是非常感激你这么做。一条引用说明通常包括书名、作者、出版商和 ISBN。例如，“*C# 6.0 Cookbook*, Fourth Edition, by Jay Hilyard and Stephen Teilhet. Copyright ©2015 Jay Hilyard and Stephen Teilhet, 978-1-4919-2146-3”。

如果你感觉你使用代码示例的方式不属于以上所述的任何方式，可以随时与我们联系，我们的电子邮箱地址为 [permissions@oreilly.com](mailto:permissions@oreilly.com)。

## Safari®在线图书



Safari 在线图书 (<http://safaribooksonline.com/>) 是一个基于用户需求，发行全球技术和商业领域顶级作者的优质图书和视频 (<https://www.safaribooksonline.com/explore/>) 的数字图书馆。

技术专家、软件开发人员、网页设计师以及商业和创意专家都选择将 Safari 在线图书作为研究、解决问题、学习和证书培训的首要资源。

Safari 在线图书为企业 (<https://www.safaribooksonline.com/enterprise/>)、政府 (<https://www.safaribooksonline.com/government/>)、教育机构 (<https://www.safaribooksonline.com/academic-public-library/>) 和个人提供了不同的产品组合和价格方案 (<https://www.safaribooksonline.com/pricing/>)。

用户可通过 O'Reilly Media、Prentice Hall Professional、Addison-Wesley Professional、Microsoft Press、Sams、Que、Peachpit Press、Focal Press、Cisco Press、John Wiley & Sons、Syngress、Morgan Kaufmann、IBM Redbooks、Packt、Adobe Press、FT Press、Apress、Manning、New Riders、McGraw-Hill、Jones & Bartlett 和 Course Technology 等数百个出版商的数据库 (<https://www.safaribooksonline.com/our-library/>) 搜寻上千种图书、培训视频和预出版的手稿。要了解有关 Safari 在线图书的更多信息，请访问我们的在线网站 (<http://safaribooksonline.com/>)。

## 联系我们

请把对本书的评价和问题发给出版社。

美国：

O'Reilly Media, Inc.  
1005 Gravenstein Highway North  
Sebastopol, CA 95472

中国：

北京市西城区西直门南大街 2 号成铭大厦 C 座 807 室 (100035)

奥莱利技术咨询(北京)有限公司

O'Reilly 的每一本书都有专属网页，你可以在那儿找到本书的相关信息，包括勘误表、示例代码以及其他信息。本书的网站地址是：

<http://shop.oreilly.com/product/0636920037347.do>

对于本书的评论和技术性问题，请发送电子邮件到：

[bookquestions@oreilly.com](mailto:bookquestions@oreilly.com)

要了解更多 O'Reilly 图书、培训课程、会议和新闻的信息，请访问以下网站：

<http://www.oreilly.com>

我们在 Facebook 的地址如下：<http://facebook.com/oreilly>

请关注我们的 Twitter 动态：<http://twitter.com/oreillymedia>

我们的 YouTube 视频地址如下：<http://www.youtube.com/oreillymedia>

## 致谢

这本书始于我们最初接触 C#，并在我们多年来对该语言的探索和创新应用中演变成长。自本书的上一版出版之后，随着 C# 6.0 的发布以及 C# 4.0 和 C# 5.0 的新功能出现，我们决定复查前三个版本以确认如何改进原有的范例，并探索使用 C# 更好地完成编程任务的方法。通过对 C# 和 Framework 大量知识的不断学习，我们致力于在本书中为读者呈现 C# 是如何演变的，以及如何使用它更好地完成工作。

以下这些人对本书的完成具有不可或缺的作用，我们在此对他们给予的帮助表示感谢。

感谢 Brian MacDonald (我们的编辑)、Heather Scherer、Rachel Monaghan、Nick Adams 和 Sara Peyton。在你们的督促下本书才得以快速完成并出版。感谢你们的付出。

同时我们还要感谢我们的技术复审小组成员：Steve Munyan、Lee Coward 和 Nick Pinkham。感谢你们对完善本书的全身心投入和富有洞察力的见解。没有你们有价值的反馈，这本书就不可能存在，感谢你们。

## Jay Hilyard 的致谢

感谢 Steve Teihet 的创意、幽默感以及愿意再次与我踏上写书的征途。我一直很享受与你共事，尽管都是在晚上和周末，并且几乎都是在网络上。

感谢我的妻子 Brooke。尽管明白写书意味着我们相处时间的减少，但你依然支持我、鼓励我、帮助我。这本书没有你就不可能诞生。谢谢你，我爱你！

感谢我的两个儿子，Owen 和 Drew。你们从新角度看待事物的能力每每让我惊讶，我为你们所完成的一切感到骄傲。我很高兴你们都对我为之奋斗一生的领域感兴趣，你们是我无可替代的珍宝。

感谢 Phil 和 Gail 对我不得不牺牲假期而工作的支持和理解，并肩负起祖父母的责任帮助我照顾孩子，同时还要感谢妈妈每个月的心灵鸡汤。

感谢我的一群好友：Seth、Katie Fiermonti、Tom Bebbington 和 Jenna Roberts。世上再无难事，只要有朋友和一杯好酒。

感谢 Scott Cronshaw、Bill Bolevic、Melissa Jurkoic、Mike Kennie、Alex Shore、Dave Flanders、Aaron Reddish、Rakshit Jain、Jason Phelps、Josh Clairmont、Bob Blais、Kim Serpa、Stu Savage、Gaurang Patel、Jesse Peters、Ken Jones、Mahesh Unnikrishnan、T Antonio、Mary Ellen Sawyer、Jon Godbout、Atul Kaul、Mark Miller、Rich Labenski、Lance Simpson、Tim Beaulieu 和 Lee Horgan。你们是最棒的团队，你们工作都非常努力，我感谢你们所做的一切。

最后，再次感谢我的家人和朋友。你们对一本自己并不了解的书表示了极大的关心，并为我感到骄傲。

## Steve Teilhet的致谢

我很骄傲能与 Jay Hilyard 成为好友，他是一个出色的合作者，一个勤奋的合著者。不是每天都能找到一个既值得信任又能合作无间的好朋友。再次与你合著是我的荣幸。

感谢我的妻子 Kandis Teilhet 给了我坚持前行每一步的力量。我对你的爱已尽在不言中。感谢我的两个儿子 Patrick 和 Nicholas，是你们让艰难的时刻变得顺利。你们对我来说无可替代。现在你们都步入了人生的下一个阶段，我为能看到你们即将获取的成就感到激动，没准你们也会写一本书。

感谢我的妈妈、爸爸和兄弟，感谢他们一直以来的倾听和支持。

最后同样要感谢的是 IBM 团队、Larry Rose、Babita Sharma、Jessica Berliner、Jeff Turnham、John Peyton、Kris Duer、Robert Stanzel、Shu Wang、Bingzhou Zheng、Dave Steinberg、Dave Stewart、Jason Todd、Alexei Pivkine、Joshua Clark、William Frontiero、Matthew Murphy、Omer Trip、Marco Pistoia、Enrique Varillas、Guillermo Hurtado、Bao Lu、Mary Santo、Diane Redfearn、Urmi Chatterjee、Joshua Ho、Kenneth Cheung、Andrew Mak、Daniel Nguyen、Jennifer Calder、Tahseen Shabab、Srinivas Sripada、David Marshak、Larry Gerard、Douglas Wilson、Steve Hikida 以及其他许多人，你们的辛勤工作和才华一直激励着我。

## 电子书

扫描如下二维码，即可购买本书电子版。



# 目录

|                       |    |
|-----------------------|----|
| 前言                    | xi |
| 第 1 章 类和泛型            | 1  |
| 1.0 简介                | 1  |
| 1.1 创建联合类型的结构         | 3  |
| 1.2 使类型可排序            | 5  |
| 1.3 使类型可查找            | 9  |
| 1.4 从一个方法返回多个数据项      | 12 |
| 1.5 解析命令行参数           | 15 |
| 1.6 在运行时初始化常量字段       | 25 |
| 1.7 构建可克隆的类           | 28 |
| 1.8 确保对象的处置           | 31 |
| 1.9 确定何时何处使用泛型        | 33 |
| 1.10 理解泛型类型           | 34 |
| 1.11 反转有序列表中的内容       | 41 |
| 1.12 约束类型参数           | 43 |
| 1.13 将泛型变量初始化为默认值     | 46 |
| 1.14 向生成的实体中添加钩子      | 48 |
| 1.15 控制如何触发多播委托中的一个委托 | 50 |
| 1.16 在 C# 中使用闭包       | 56 |
| 1.17 使用函数对象在列表中执行多种操作 | 61 |
| 1.18 控制结构类型字段初始化      | 64 |
| 1.19 以更简洁的方式检查 null 值 | 68 |

|                                      |     |
|--------------------------------------|-----|
| 第 2 章 集合、枚举器和迭代器                     | 72  |
| 2.0 简介                               | 72  |
| 2.1 寻找 List<T> 中的重复数据项               | 74  |
| 2.2 保持 List<T> 有序                    | 78  |
| 2.3 对 Dictionary 的键和 / 或值排序          | 80  |
| 2.4 创建具有最小值和最大值边界的 Dictionary        | 82  |
| 2.5 在应用程序会话间持久化一个集合                  | 84  |
| 2.6 测试 Array 或 List<T> 中的每个元素        | 86  |
| 2.7 创建自定义枚举器                         | 88  |
| 2.8 处理 finally 语句块和迭代器               | 91  |
| 2.9 在类中实现嵌套的 foreach 功能              | 95  |
| 2.10 使用线程安全的字典进行并发访问，不手动加锁           | 99  |
| 第 3 章 数据类型                           | 106 |
| 3.0 简介                               | 106 |
| 3.1 把二进制数据编码为 base64 格式              | 108 |
| 3.2 解码 base64 编码的二进制数据               | 109 |
| 3.3 把作为 byte[] 返回的字符串转换为字符串          | 110 |
| 3.4 把字符串传递给只接受 byte[] 的方法            | 112 |
| 3.5 确定一个字符串是否为有效的数字                  | 113 |
| 3.6 舍入浮点值                            | 114 |
| 3.7 选择一种舍入算法                         | 115 |
| 3.8 安全地执行窄化数据转换                      | 116 |
| 3.9 测试有效的枚举值                         | 118 |
| 3.10 在位掩码中使用枚举成员                     | 120 |
| 3.11 确定是否设置了一个或多个枚举标志                | 122 |
| 第 4 章 语言集成查询和 lambda 表达式             | 126 |
| 4.0 简介                               | 126 |
| 4.1 查询消息队列                           | 128 |
| 4.2 对数据使用集合语义                        | 132 |
| 4.3 利用 LINQ to SQL 重用参数化查询           | 136 |
| 4.4 以文化敏感的方式对结果排序                    | 138 |
| 4.5 添加用于 LINQ 的函数式扩展                 | 141 |
| 4.6 跨数据库执行查询和联接                      | 144 |
| 4.7 利用 LINQ 查询配置文件                   | 147 |
| 4.8 从数据库直接创建 XML 文件                  | 150 |
| 4.9 有选择地输出查询结果                       | 162 |
| 4.10 将 LINQ 用于不支持 IEnumerable<T> 的集合 | 165 |

|              |                                |            |
|--------------|--------------------------------|------------|
| 4.11         | 执行高级接口查找                       | 167        |
| 4.12         | 使用 lambda 表达式                  | 168        |
| 4.13         | 在 lambda 表达式中使用不同的参数修饰符        | 173        |
| 4.14         | 用并行来加速 LINQ 操作                 | 176        |
| <b>第 5 章</b> | <b>调试和异常处理</b>                 | <b>187</b> |
| 5.0          | 简介                             | 187        |
| 5.1          | 知道何时捕获并重新引发异常                  | 193        |
| 5.2          | 处理通过反射调用的方法引发的异常               | 194        |
| 5.3          | 创建新的异常类型                       | 197        |
| 5.4          | 在首次异常上中断                       | 204        |
| 5.5          | 处理从异步委托中引发的异常                  | 209        |
| 5.6          | 利用 Exception.Data 为异常提供所需的额外信息 | 211        |
| 5.7          | 在 WinForms 应用程序中处理未经处理的异常      | 213        |
| 5.8          | 在 WPF 应用程序中处理未经处理的异常           | 214        |
| 5.9          | 确定一个进程是否停止了响应                  | 217        |
| 5.10         | 在应用程序中使用事件日志                   | 219        |
| 5.11         | 监视事件日志中的特定条目                   | 229        |
| 5.12         | 实现一个简单的性能计数器                   | 230        |
| 5.13         | 为类创建自定义的调试显示                   | 233        |
| 5.14         | 跟踪异常从何而来                       | 235        |
| 5.15         | 在异步情境下处理异常                     | 237        |
| 5.16         | 有选择地处理异常                       | 243        |
| <b>第 6 章</b> | <b>反射和动态编程</b>                 | <b>247</b> |
| 6.0          | 简介                             | 247        |
| 6.1          | 列出引用的程序集                       | 248        |
| 6.2          | 确定程序集中的类型特征                    | 252        |
| 6.3          | 确定继承特征                         | 256        |
| 6.4          | 使用反射调用成员                       | 261        |
| 6.5          | 访问局部变量信息                       | 264        |
| 6.6          | 创建一个泛型类型                       | 267        |
| 6.7          | 使用 dynamic 与使用 object          | 268        |
| 6.8          | 动态构建对象                         | 271        |
| 6.9          | 使对象可扩展                         | 275        |
| <b>第 7 章</b> | <b>正则表达式</b>                   | <b>284</b> |
| 7.0          | 简介                             | 284        |
| 7.1          | 从 MatchCollection 中提取组         | 285        |

|              |                       |            |
|--------------|-----------------------|------------|
| 7.2          | 验证正则表达式的语法            | 288        |
| 7.3          | 增强基本的字符串替换函数          | 289        |
| 7.4          | 实现一个更好的分词器            | 292        |
| 7.5          | 返回匹配所在的整行内容           | 293        |
| 7.6          | 找到特定次数的匹配             | 297        |
| 7.7          | 使用常见模式                | 299        |
| <b>第 8 章</b> | <b>文件系统 I/O</b>       | <b>303</b> |
| 8.0          | 简介                    | 303        |
| 8.1          | 使用通配符查找目录和文件          | 304        |
| 8.2          | 获取目录树                 | 309        |
| 8.3          | 解析路径                  | 313        |
| 8.4          | 启动并与控制台工具交互           | 314        |
| 8.5          | 锁定文件的一部分              | 316        |
| 8.6          | 等待文件系统中的动作发生          | 320        |
| 8.7          | 比较两个可执行模块的版本信息        | 322        |
| 8.8          | 查询系统上所有驱动器的信息         | 325        |
| 8.9          | 压缩和解压缩文件              | 327        |
| <b>第 9 章</b> | <b>网络和 Web</b>        | <b>337</b> |
| 9.0          | 简介                    | 337        |
| 9.1          | 处理 Web 服务器错误          | 338        |
| 9.2          | 与 Web 服务器通信           | 339        |
| 9.3          | 通过代理服务器               | 341        |
| 9.4          | 从一个 URL 获取 HTML       | 343        |
| 9.5          | 使用 Web 浏览器控件          | 344        |
| 9.6          | 以编程方式预构建一个 ASP.NET 网站 | 346        |
| 9.7          | 为 Web 应用对数据进行转义和取消转义  | 349        |
| 9.8          | 检查 Web 服务器的自定义错误页     | 351        |
| 9.9          | 编写一个 TCP 服务器          | 355        |
| 9.10         | 编写一个 TCP 客户端          | 362        |
| 9.11         | 模拟表单执行                | 370        |
| 9.12         | 通过 HTTP 传输数据          | 373        |
| 9.13         | 使用命名管道进行通信            | 377        |
| 9.14         | 以编程方式发送 ping          | 384        |
| 9.15         | 使用 SMTP 服务发送 SMTP 邮件  | 386        |
| 9.16         | 使用套接字扫描机器的端口          | 388        |
| 9.17         | 使用当前的互联网连接设置          | 392        |
| 9.18         | 使用 FTP 传输文件           | 398        |