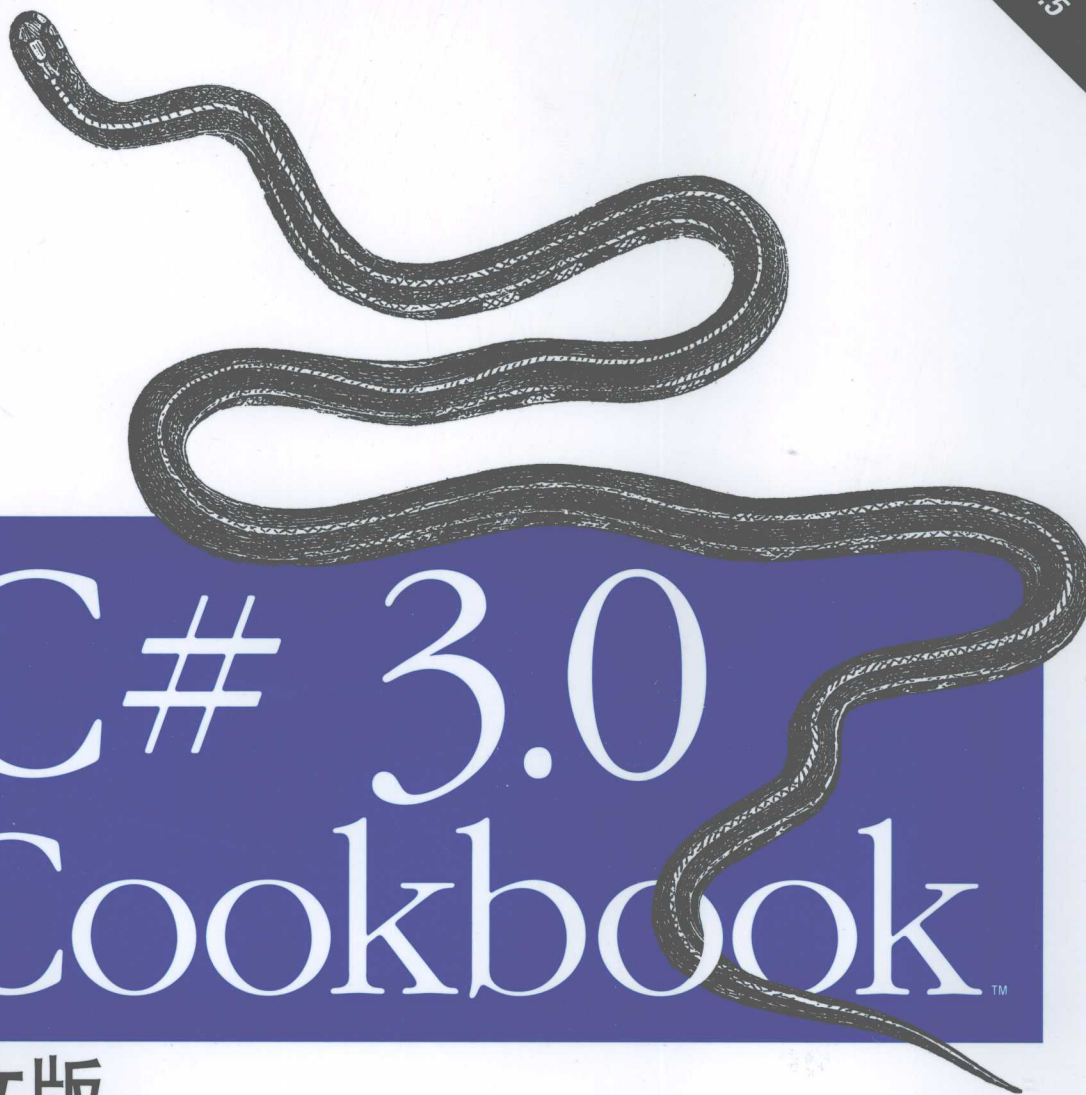


C# 3.0 Cookbook

第3版
Thoroughly revised
for C# 3.0 and .NET 3.5



C# 3.0 Cookbook™

中文版

Jay Hilyard & Stephen Teilhet 著

陈宗斌 等译

O'REILLY®



清华大学出版社

第3版

C# 3.0 Cookbook 中文版

Jay Hilyard & Stephen Teilhet 著

陈宗斌 等译

O'REILLY®

Beijing • Cambridge • Farnham • Köln • Sebastopol • Taipei • Tokyo

O'Reilly Media, Inc. 授权清华大学出版社出版

清华大学出版社

Copyright ©2007 by O'Reilly Media, Inc.

Authorized Simplified Chinese translation edition, by O'Reilly Media, Inc., is published by Tsinghua University Press, 2008. Authorized translation of the original English edition, 2007 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.

本书之英文原版由 O'Reilly Media, Inc. 于 2007 年出版。

本中文简体翻译版由 O'Reilly Media, Inc. 授权清华大学出版社于 2008 年出版。此翻译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, 未经书面许可, 本书的任何部分和全部不得以任何形式复制。

北京市版权局著作权合同登记

图字: 01-2008-1925 号

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

C# 3.0 Cookbook 中文版 (第 3 版) / (美) 希尔雅德 (Hilyard, J.), (美) 特尔赫特 (Teilhet, S.) 著; 陈宗斌等译. —北京: 清华大学出版社, 2008.12

书名原文: C# 3.0 Cookbook, Third Edition

ISBN 978-7-302-18109-5

I. C… II. ①希… ②特… ③陈… III. C 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2008) 第 101910 号

责任编辑: 龙啟铭

封面设计: Karen Montgomery, 张 健

责任校对: 徐俊伟

责任印制: 王秀菊

出版发行: 清华大学出版社

地 址: 北京清华大学学研大厦 A 座

<http://www.tup.com.cn>

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

印 刷 者: 清华大学印刷厂

装 订 者: 三河市金元印装有限公司

经 销: 全国新华书店

开 本: 178 毫米×233 毫米 印 张: 45.25 字 数: 1063 千字

版 次: 2008 年 12 月第 1 版 印 次: 2008 年 12 月第 1 次印刷

印 数: 1~3000 册

定 价: 99.00 元 (册)

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题, 请与清华大学出版社出版部联系调换。联系电话: (010)62770177 转 3103 产品编号: 028161-01

O'Reilly Media, Inc. 介绍

为了满足读者对网络 and 软件技术知识的迫切需求，世界著名计算机图书出版机构 O'Reilly Media, Inc. 授权清华大学出版社，翻译出版一批该公司久负盛名的英文经典技术专著。

O'Reilly Media, Inc. 是世界上在 UNIX、X、Internet 和其他开放系统图书领域具有领导地位的出版公司，同时也是联机出版的先锋。

从最畅销的 *The Whole Internet User's Guide & Catalog* (被纽约公共图书馆评为 20 世纪最重要的 50 本书之一) 到 GNN (最早的 Internet 门户和商业网站)，再到 WebSite (第一个桌面 PC 的 Web 服务器软件)，O'Reilly Media, Inc. 一直处于 Internet 发展的最前沿。

许多书店的反馈表明，O'Reilly Media, Inc. 是最稳定的计算机图书出版商——每一本书都一版再版。与大多数计算机图书出版商相比，O'Reilly Media, Inc. 具有深厚的计算机专业背景，这使得 O'Reilly Media, Inc. 形成了一个非常不同于其他出版商的出版阵容。O'Reilly Media, Inc. 所有的编辑人员以前都是程序员，或者是顶尖级的技术专家。O'Reilly Media, Inc. 还有许多固定的作者群体——他们本身是相关领域的技术专家、咨询专家，而现在着手编写著作，O'Reilly Media, Inc. 依靠他们及时地推出图书。因为 O'Reilly Media, Inc. 紧密地与计算机业界相联系，所以 O'Reilly Media, Inc. 知道市场上真正需要什么样的图书。

译者序

C#语言是Microsoft公司为推行.NET战略而发布的一种全新的编程语言。它是一种全新的、纯面向对象的编程语言,具有清晰明了的语法结构、优秀的编程开发环境和高效率的编译工具。C#语言从C++语言发展而来,继承了C++语言的所有优点,同时根据.NET战略的需要增强了自身的功能,抛弃了C++中各种不安全因素,从而使程序员能够更有效率地使用C#开发应用软件。

另外,C#在许多方面与Java相似;Java程序员一旦学习了Framework SDK,他们在C#中就会有一种宾至如归的感觉。C#还非常适合于Visual Basic .NET程序员,当他们需要能够更多地控制代码执行的任务并且不想为此编写C++代码时就可以使用C#。

本书侧重于解决C#程序员在开发应用程序时遇到的各类问题,并据此组织全书内容。这些解决方案称为秘诀;每个秘诀都包含一个问题,它的解决方案及相关信息的讨论,最后还提供了关于使用FCL中的类的详细信息的出处、针对这个主题的其他书籍、相关的文章以及其他秘诀。这种问题—答案格式提供了问题的完整解决方案,使得本书易于阅读和使用。几乎每个秘诀都包含一个完整的书面代码示例,展示了如何解决特定的问题,还讨论了底层技术的工作原理,并且根据需要列出了一些替代方法、限制条件及其他考虑事项。

本书适合于C#或.NET开发人员阅读,本书提供了开发人员每天所面临的各种问题以及一些可能不太常见的问题的解决方案;另外,本书在介绍每个主题时,考虑到初学者的需要,都会给出相应的程序示例,非常适合程序设计初学者阅读。

参加本书翻译的人员有:陈宗斌、陈红霞、张景友、易小丽、陈婷、管学岗、王新彦、金惠敏、张海峰、徐晔、戴锋、张德福、张士华、张锁玲、杜明宗、高玉琢、王涛、申川、孙玲、李振国、高德杰、宫飞、侯经国、刘淑妮、张春林、李大成、程明、张路红、张淑芝、孙先国、刘冀得、梁永翔、张广东、郁琪琳、邵长凯、蒲书箴、潘曙光、刘瑞东、李军、焦敬俭、赵中元、金鑫、赵宏伟、赵宏顺、尹周、王开年、贾震、陆晓萍、金国良、俞群等。

由于时间紧迫,加之译者水平有限,错误在所难免,恳请广大读者批评指正。

作者简介

Jay Hilyard 开发 Windows 应用程序的时间超过了 15 年，其中有 7 年多的时间在开发 .NET 应用程序。他在 MSDN Magazine 中发表了众多文章，他目前在位于美国新罕布什尔州朴次茅斯市的 Newmarket International 的 New Product 团队工作。

Stephen Teilhet 自从 Microsoft 开发了 .NET 1.0 Framework 预发行版本以来，就一直在使用 .NET 平台。目前，他为 Ounce Labs 工作，增强它们的静态安全代码分析工具，用以找出多种语言（包括 C# 和 Visual Basic）中的漏洞。

封面介绍

本书封面上的动物是一只束带蛇。之所以这样命名，是因为它们的纵向条纹类似于曾用于支撑人类袜子的吊袜带，可以通过它们与众不同的条纹轻松地识别束带蛇：后背中间有一条窄条纹，它两边各有一条宽条纹。颜色和图案变化使它们能够混入天然环境中，帮助它们躲避食肉动物。它们是北美洲最常见的蛇，并且是阿拉斯加州唯一的蛇类。

束带蛇具有棱鳞——沿着鳞中轴的一条或多条脊，这使得它们具有粗糙的组织 and 暗淡的外表。成年束带蛇的长度一般在 46~130 厘米（1.5 英尺到 4 英尺以上）之间。雌性束带蛇通常比雄性束带蛇大一些，尾巴要短一些，并且在躯干和尾巴的结合部有一个凸出部分。

雌性束带蛇是卵胎生的，这意味着它们在软蛋中生育“有生命的”幼崽。在分娩时，大多数软蛋和粘膜都会破裂，这使得它们生育的孩子看上去是有生命的。偶尔，新生的幼崽仍会在其软壳内。雌性束带蛇通常会生育 10~40 只幼崽：束带蛇的最高记录是生育了 98 只活的幼崽。一旦从母体生育出来，幼蛇就完全独立并且必须开始自谋生路。在这段时间，它们极易学会以捕食其他动物为生的方法，并且超过半数的幼蛇会在一岁前死亡。

束带蛇是能够吃蟾蜍、蝶螈以及其他具有强大防化能力的两栖动物的少数几种动物之一。尽管它们通常所吃的食物因环境而异，束带蛇通常会吃蚯蚓和两栖动物；不过，它们偶尔也会吃幼鸟、鱼和较小的啮齿动物。束带蛇的唾液有毒（对人类无害），束带蛇在完全吞食猎物之前，将会使用其唾液使猎物昏厥或杀死它们。

目录

前言	1
第 1 章 语言级集成查询 (LINQ)	11
1.0 简介	11
1.1 查询消息队列	12
1.2 使用带有数据的集合语义	14
1.3 利用 LINQ to SQL 重用参数化查询	17
1.4 以一种文化敏感的方式对结果排序	19
1.5 添加用于 LINQ 的功能扩展	21
1.6 跨数据库执行查询和连接	24
1.7 利用 LINQ 查询配置文件	27
1.8 直接从数据库创建 XML 文件	30
1.9 有选择地输出查询结果	36
1.10 结合使用 LINQ 与不支持 IEnumerable<T> 的集合	38
第 2 章 字符串和字符	41
2.0 简介	41
2.1 确定 char 包含的字符种类	41
2.2 在比较两个字符时控制大小写敏感性	44
2.3 在一个字符串内寻找另一个字符串出现的所有位置	46

2.4	在比较两个字符串时控制大小写敏感性	50
2.5	把一个字符串与另一个字符串的头部或尾部作比较	51
2.6	插入文本到字符串中	52
2.7	删除或替换字符串内的字符	54
2.8	把二进制数据编码为 base64 格式	56
2.9	解码 base64 编码的二进制数据	57
2.10	把作为 byte[] 返回的字符串转换为字符串	58
2.11	把字符串传递给只接受 byte[] 的方法	59
2.12	把字符串转换为其他类型	61
2.13	创建定界字符串	64
2.14	从定界字符串中提取数据项	65
2.15	遍历字符串中的每个字符	66
2.16	从字符串的头部和 / 或尾部删除字符	67
2.17	测试字符串是否为 null 或者长度为 0	68
2.18	追加行	68
第 3 章 类和结构		70
3.0	简介	70
3.1	创建联合类型的结构	72
3.2	使类型可排序	74
3.3	使类型可检索	78
3.4	间接重载 +=、-=、/= 和 *= 运算符	80
3.5	间接重载 &&、 和 ?: 运算符	82
3.6	创建无错的表达式	84
3.7	简化布尔逻辑	87
3.8	以程序设计语言无关的方式在简单类型之间转换	90
3.9	确定何时使用强制转换运算符、as 运算符或 is 运算符	96
3.10	利用 as 运算符进行强制转换	97
3.11	利用 is 运算符确定变量的类型	99
3.12	从一个方法返回多个数据项	102
3.13	解析命令行参数	104
3.14	在运行时初始化常量字段	113
3.15	构建可复制的类	115

3.16	确保处理对象	118
3.17	处理非托管资源	120
3.18	确定装箱和拆箱的位置	126
第 4 章 泛型		129
4.0	简介	129
4.1	确定使用泛型的时间和位置	129
4.2	理解泛型类型	130
4.3	利用对应的泛型替换 ArrayList	137
4.4	用对应的泛型替换 Stack 和 Queue	140
4.5	使用链表	144
4.6	创建可以初始化为 null 的值类型	147
4.7	反转有序链表中的内容	149
4.8	以泛型方式创建只读集合	151
4.9	利用对应的泛型替换 Hashtable	152
4.10	对泛型 Dictionary 类型使用 foreach	155
4.11	约束类型参数	156
4.12	将泛型变量初始化为它们的默认值	160
第 5 章 集合		162
5.0	简介	162
5.1	交换数组中的两个元素	164
5.2	快速反转数组	165
5.3	编写更灵活的 StackTrace 类	167
5.4	确定一个数据项在 List<T> 中出现的次数	169
5.5	检索 List<T> 中的特定数据项的所有实例	171
5.6	从数组中插入和删除数据项	173
5.7	保持 List<T> 有序	175
5.8	对 Dictionary 的键和 / 或值排序	177
5.9	创建具有最大值和最小值界限的 Dictionary	178
5.10	在数组中存储列表的快照	181
5.11	维持应用程序会话之间的集合	182
5.12	测试 Array 或 List<T> 中的每个元素	183

5.13	对 Array 或 List<T> 中的每个元素执行一种操作	185
5.14	创建只读的 Array 或 List<T>	186
第 6 章 迭代器、分部类型和分部方法		189
6.0	简介	189
6.1	在泛型类型上创建迭代器	190
6.2	在非泛型类型上创建迭代器	191
6.3	创建自定义的枚举器	193
6.4	实现迭代器逻辑	196
6.5	强制迭代器停止迭代	199
6.6	处理 finally 块和迭代器	201
6.7	在类中实现嵌套的 foreach 功能	204
6.8	组织接口实现	208
6.9	生成不再位于主代码路径中的代码	212
6.10	向生成的实体中添加挂钩	214
第 7 章 异常处理		217
7.0	简介	217
7.1	知道何时捕获和重新抛出异常	222
7.2	使用 finally 块时确保异常不会丢失	223
7.3	处理从通过反射调用的方法中抛出的异常	226
7.4	防止未处理的异常	228
7.5	获取异常信息	230
7.6	快速找到问题的根源	233
7.7	创建新的异常类型	234
7.8	获得栈记录	241
7.9	在第一次机会的异常上中断	243
7.10	处理从异步委托抛出的异常	246
7.11	利用 Exception.Data 给异常提供它们所需的额外信息	247
7.12	在 WinForms 应用程序中处理未处理的异常	249
7.13	处理 Windows Presentation Foundation (WPF) 应用程序中未处理的异常	251
7.14	为常见错误分析异常	253

第 8 章 诊断	255
8.0 简介	255
8.1 对调试 / 跟踪输出提供细粒度的控制	255
8.2 确定进程是否停止了响应	258
8.3 在应用程序中使用事件日志	260
8.4 查找事件日志条目	266
8.5 监视事件日志中的特定条目	269
8.6 实现一个简单的性能计数器	270
8.7 启用和禁用复杂的跟踪代码	273
8.8 捕获进程的标准输出	276
8.9 为类创建自定义的调试显示	278
 第 9 章 委托、事件和 lambda 表达式	 280
9.0 简介	280
9.1 控制何时以及是否在多播委托中触发一个委托	282
9.2 从多播委托中的每个委托获得返回值	285
9.3 为多播委托中的每个委托单独处理异常	287
9.4 把同步委托调用转换成异步委托调用	289
9.5 一种高级接口查找机制	291
9.6 观察字典的添加和修改	293
9.7 使用 lambda 表达式	302
9.8 清晰地建立事件处理程序	306
9.9 在 lambda 表达式中使用不同的参数修饰符	309
9.10 在 C# 中使用闭包	313
9.11 使用函子对列表执行多种操作	317
 第 10 章 正则表达式	 321
10.0 简介	321
10.1 枚举匹配	322
10.2 从 MatchCollection 中提取组	324
10.3 验证正则表达式的语法	327
10.4 只快速查找字符串中的最后一个匹配	328
10.5 增强基本的字符串替换功能	329

10.6	实现更好的标记器	331
10.7	统计文本行数	333
10.8	返回在其中找到匹配的一整行内容	335
10.9	找到出现的特定匹配	338
10.10	使用常见的模式	340
第 11 章 数据结构与算法		344
11.0	简介	344
11.1	为一种数据类型生成散列代码	344
11.2	生成优先级队列	351
11.3	生成一对多映射 (MultiMap)	357
11.4	生成一棵二叉查找树	364
11.5	生成一棵 n 叉树	375
11.6	使用 HashSet 对象	385
第 12 章 文件系统 I/O		390
12.0	简介	390
12.1	操作文件属性	391
12.2	重命名文件	393
12.3	输出与平台无关的 EOL 字符	394
12.4	操作目录属性	396
12.5	重命名目录	398
12.6	使用通配符查找目录或文件	399
12.7	获取目录树	403
12.8	解析路径	405
12.9	解析环境变量中的路径	407
12.10	运行并与控制台工具交互	408
12.11	锁定文件的子段	410
12.12	等待文件系统中要发生的动作	412
12.13	比较两个可执行模块的版本信息	415
12.14	查询系统上所有驱动器的信息	417
12.15	压缩及解压缩文件	419

第 13 章 反射	425
13.0 简介	425
13.1 列出引用的程序集	425
13.2 列出导出的类型	427
13.3 找出重写方法	428
13.4 找出程序集中的成员	433
13.5 确定并获取程序集中的嵌套类型	434
13.6 显示类型的继承层次结构	435
13.7 找出类型的子类	437
13.8 找出程序集中所有可序列化的类型	438
13.9 动态调用成员	440
13.10 确定类型或方法是否是泛型	443
13.11 访问局部变量信息	444
13.12 生成泛型类型	446
 第 14 章 Web	 448
14.0 简介	448
14.1 将 IP 地址转换为主机名	448
14.2 将主机名转换为 IP 地址	449
14.3 解析 URI	450
14.4 处理 Web 服务器错误	453
14.5 与 Web 服务器通信	455
14.6 通过代理服务器	456
14.7 从 URL 中获取 HTML	457
14.8 使用 Web 浏览器控件	458
14.9 将数据库表与缓存绑定	460
14.10 通过编程预先构建一个 ASP.NET 网站	462
14.11 为 Web 对数据进行转义与取消转义	464
14.12 使用 UriBuilder 类	466
14.13 检查并修改 Web 应用程序配置	468
14.14 在用 HTTP 时使用缓存的结果以使性能更快	470
14.15 检测 Web 服务器的自定义错误页	472

第 15 章 XML	476
15.0 简介	476
15.1 以文档顺序读取并访问 XML 数据	476
15.2 读取 Web 上的 XML	480
15.3 查询 XML 文档的内容	481
15.4 检验 XML	485
15.5 通过编程生成 XML 文档	489
15.6 检测对 XML 文档的修改	492
15.7 处理 XML 字符串中的非法字符	494
15.8 XML 的转换	497
15.9 分割 XML 文档	502
15.10 合并 XML 文档	507
15.11 不重载验证修改过的 XML 文件	512
15.12 扩展转换	515
15.13 从已有 XML 文件中成批获得用户的模式	519
15.14 向转换传递参数	520
 第 16 章 联网	 525
16.0 简介	525
16.1 编写一个 TCP 服务器端	525
16.2 编写一个 TCP 客户端	530
16.3 模拟表单执行	532
16.4 通过 HTTP 传送数据	535
16.5 使用命名管道进行通信	538
16.6 以编程方式发送 Ping	544
16.7 使用 SMTP 服务发送 SMTP 邮件	545
16.8 使用套接字扫描机器的端口	550
16.9 使用当前的互联网连接设置	554
16.10 使用 FTP 传输文件	559
 第 17 章 安全性	 562
17.0 简介	562
17.1 控制对本地程序集中类型的访问	562

17.2	加密 / 解密字符串	570
17.3	加密和解密文件	573
17.4	清理密码信息	577
17.5	验证字符串保持未破坏的流传输	579
17.6	安全存储数据	582
17.7	保证安全断言的安全	588
17.8	验证程序集已经具备特定许可	590
17.9	最小化程序集的攻击面	591
17.10	获得安全 / 审计信息	592
17.11	授权 / 撤销对文件或注册表键的访问	596
17.12	使用安全字符串保护字符串数据	599
17.13	保护流数据	601
17.14	加密 web.config 信息	609
17.15	获得抛出 SecurityException 的完整理由	611
17.16	实现安全的 Unicode 编码	613
17.17	获得一个更安全的文件句柄	614
第 18 章	线程与同步	616
18.0	简介	616
18.1	创建每个线程的静态字段	616
18.2	向类成员提供线程安全的访问	618
18.3	防止沉默的线程终止	623
18.4	同步委托完成的通知	625
18.5	秘密存储特定线程的数据	627
18.6	使用信号量实现资源的多重访问	630
18.7	使用 Mutex 同步多个进程	634
18.8	使用事件协调线程	643
18.9	为事件获得命名权	645
18.10	在线程间执行原子操作	648
18.11	优化主读访问	649
第 19 章	工具箱	660
19.0	简介	660
19.1	处理操作系统关机、电源管理或用户会话修改	660

19.3	列出加载程序集的进程	667
19.4	使用本地工作站上的消息队列	669
19.5	找出当前框架版本的路径	671
19.6	确定全局程序集缓存 (GAC) 中注册的程序集版本	672
19.7	捕获标准输出流的输出	675
19.8	运行位于自己的 AppDomain 中的代码	676
19.9	确定操作系统和当前操作系统的服务包版本	678
第 20 章 数字与枚举		680
20.0	简介	680
20.1	度与弧度之间的转换	682
20.2	对各种数据类型使用按位求补运算符	682
20.3	将数字从某种基数转换为基数 10	684
20.4	确定一个字符串是否是合法数字	685
20.5	舍入浮点值	685
20.6	选择一种舍入算法	686
20.7	温度计数法之间的转换	687
20.8	安全地执行收缩数字强制转换	688
20.9	将一个枚举值显示为字符串	690
20.10	将纯文本转换为等价的枚举值	692
20.11	测试合法的枚举值	693
20.12	测试合法的标志枚举	695
20.13	使用位掩码中的枚举成员	697
20.14	确定一个或多个枚举标志是否被设置	699
20.15	确定 Decimal 或 Double 数字的整数部分	702

前言

C# 语言面向的是 Microsoft .NET 平台的开发人员，他们应该使用过类 C 的语言，如 C、C++ 或 Java。与用于 Microsoft Windows 平台的 C 或 C++ 的以前版本不同，C# 代码运行在托管执行环境之下。Microsoft 把 C# 描绘为用于 .NET 开发的现代创新型语言，并且继续为其提供像语言级集成查询（Language Integrated Query, LINQ）这样的新特性来强化这一点。C# 3.0 中的新特性在合适时允许更具描述性和功能性的编程风格，同时它还仍然具有优秀的面向对象特性。其主要思想是使用适合于你的问题的编程风格，并且 C# 将支持你的努力。

C# 允许你执行许多类似于 C/C++ 的功能，比如通过指针和运算符重载进行直接内存访问，在 Visual Basic .NET 中不支持它。C# 是一种用于 .NET 的系统级编程语言。在 C# 中仍然可以做极佳的应用程序级的工作，但是它的真正闪光之处体现在你需要构建更贴近 Framework 的代码时。

如果你见过 C#，你可能注意到它在许多方面与 Java 相似；Java 程序员一旦学习了 Framework SDK，他们在 C# 中就会有一种宾至如归的感觉。C# 还非常适合于 Visual Basic .NET 程序员，当他们需要能够更多地控制代码执行的任务并且不想为此编写 C++ 代码时就可以使用 C#。在 Web 上，你将发现非常多的人利用 C# 做一些非常优秀的工作，还可以在一些站点上发现大量的示例代码，如 <http://www.codeplex.com> 和 <http://www.codeproject.com>。

我们基于第一次学习 C# 时遇到的编程问题开始编写本书，并且基于该语言中新的挑战和能力继续扩展它。在这一版中，我们重新编写了许多解决方案的方法，以便利用 LINQ，以及基于 LINQ 及 C# 3.0 中的其他新特性创建全新的解决方案。我们希望本书将帮助你顺利通过一些常见的（以及不那么常见的）陷阱，解决每一个人在刚开始学习一种新语言时将会碰到的一些问题，以及帮助他们处理在开发周期中遇到的一些稍微有点出人意料的事项。我们发现 .NET Framework 类库（FCL）中没有提供这些解决问题的秘诀，即