

Forth Edition

CLR via C#

(第4版)

(美) Jeffrey Richter 著
周靖 译



- ★ 权威力作全新全面升级
- ★ 名著名译再显经典魅力
- ★ 根据Microsoft .NET Framework 4.5和Visual C# 2012全面更新
- ★ 聚焦于Framework Class Library(FCL)的核心类型
- ★ 清楚阐述多核编程、泛型、线程处理等基本概念



联袂推荐

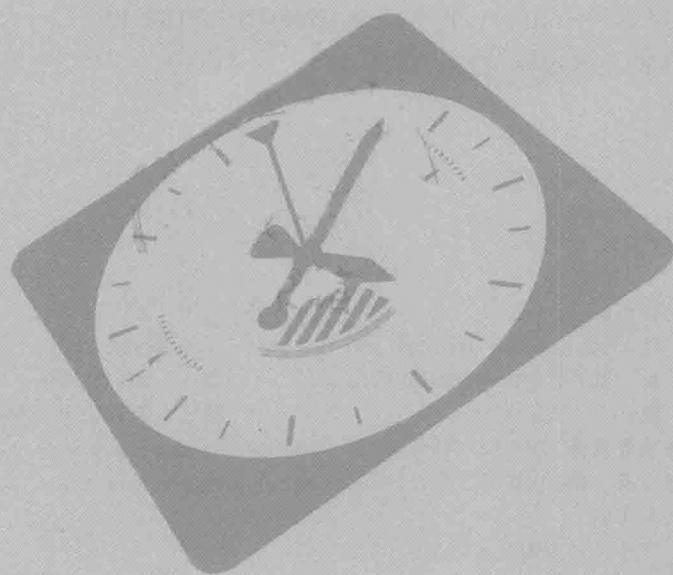


清华大学出版社

CLR via C#

(第4版)

(美) Jeffrey Richter 著
周靖 译



清华大学出版社
北京

内 容 简 介

本书针对 CLR 和 .NET Framework 4.5 进行深入、全面的探讨,并结合实例介绍了如何利用它们进行设计、开发和调试。全书 5 部分共 29 章。第 I 部分介绍 CLR 基础,第 II 部分解释如何设计类型,第 III 部分介绍基本类型,第 IV 部分以核心机制为主题,第 V 部分重点介绍线程处理。

通过本书的阅读,读者可以掌握 CLR 和 .NET Framework 的精髓,轻松、高效地创建高性能应用程序。

© 2014 Tsinghua University Press Limited

Authorized translation of the English edition of Microsoft CLR via C#, 4th Edition.

Copyright © 2012 by Jeffrey Richter. 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,未经出版者许可,不得以任何方式复制或抄袭本书的任何部分。

北京市版权局著作权合同登记号 图字: 01-2013-3404

版权所有,翻印必究。举报电话: 010-62782989 13701121933

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。

图书在版编目(CIP)数据

CLR via C#: 第 4 版/(美)迈克尔·费舍尔(Michael Fisher)著,周靖译. —北京:清华大学出版社,2015

ISBN 978-7-302-38097-7

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

中国版本图书馆 CIP 数据核字(2015)第 221083 号



责任编辑:文开琪

装帧设计:杨玉兰

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

印 刷 者:清华大学印刷厂

装 订 者:三河市新茂装订有限公司

经 销:全国新华书店

开 本:185mm×260mm 印 张:46.75 插 页:1 字 数:1021 千字

版 次:2003 年 11 月第 1 版 2015 年 1 月第 4 版 印 次:2015 年 1 月第 1 次印刷

印 数:1~3500

定 价:109.00 元

产品编号:050691-01

序 言

大家好，我们又见面了。谁预见到了今天啊？哈哈，我就预见到了！一旦步入婚姻的殿堂，就相当于过上了“土拨鼠日”。如果还没有看过那部电影，就去看看吧。看了之后，就会明白为什么自己老是犯同样的错误。当 Jeff 说他不再写书的时候，我就知道这是一个“瘾君子”开的空头支票。Jeff 不可能停止写书。就在今天，我们还在说起他“绝对”不可能写的另一本书呢(实际情况是，有一章已经在写了)。写书已深入到他骨子里面去了。千里马生来就是要奔跑的，Jeff 生来就是要写作的。

Jeff 太有规律了。他就是离不开硬盘里那些小小的 0 和 1。忽视它们是不可能的。凌晨 3 点，我们睡梦正酣的时候，Jeff 的生物钟就在催促他起床了(巧合的是，我们 4 岁大的小儿子也恰好在这个时候爬到我们的床上。爷儿俩的行为模式我都理解不了啊)。一股神秘的力量促使 Jeff 的大脑自动释放出解决方案、头脑风暴和臭虫之类的东西，迫使他跑到办公室把这些东西从脑袋里倒腾出来。而我们呢，则安心地翻个身继续呼呼大睡，知道 Jeff 会解决那些问题——就像一个神秘的网络超级英雄，防止线程又成为薄弱环节^①。

但积累这些知识供自己使用，这对 Jeff 来说远远不够。好东西不该独享。所以必须把它们传播开来，必须把它们写下来。知识就像电波，有心人能接收得到。这就是他为你所做的，亲爱的读者，是他热爱微软技术的明证。

本书还有另一层意义在里面。Jeff 每次绕着太阳“公转”一圈，都会变得更老一些。经过多年的积累，他也学着“向后看”了。由于看事情的方式变得更成熟，所以他重写了讲反射那一章。或许你也应该跟他一起回顾一下这个主题。可以学到怎样让代码自个儿询问关于代码的事儿，进而更深入地思考为什么反射要那样工作。穿上便服，找一把舒服的皮椅子坐下，花些时间想想自己的代码以及它们生命中更深层次的意义。

本书还讲了一样有趣的东西，就是异步/等待^②。和我老公以前鼓捣过一阵子的 **AsyncEnumerator** 相比，这个东西显然进步了不少。哎，我还以为今后离不开它了呢！事实上，虽然他跟我讲了好多次 **AsyncEnumerator**，但这个东西根本就没有在我的脑子里“阻塞”嘛！于是我窃想，如果知道什么是 **enumerator** 的话，也许就能明白他讲的是啥了。于是我查了一下维基百科，发现 **enumerator** 是人口普查员的意思。这一章难道是讲人口普查

① saving the thread from becoming just another loose end, 直译就是“防止线头儿又松了”。——译注

② **async** 和 **await** 是 C# 的两个关键字，允许用顺序编程模型执行异步操作。——译注

员怎样协调工作的事儿？太浪费纳税人的钱了吧！不过，我相信它在计算机里面的意义比我查到的好。Jeff 和微软的团队一起工作，将异步/等待打磨得很完美。你现在通过这本书就能舒舒服服地享受他们的成果了。我建议你好好读一下。嗯，要顺着读。^①

本书的另一个重头戏是我感觉最兴奋的。希望你们都来看看关于 WinRT 的内容。这个术语太书呆子气了，我的理解就是“马上为我无敌帅气的平板搞一些很酷的应用出来！”你猜得没错，新的 Windows Runtime 就是围绕无敌帅气的触摸屏展开的。孩子喜欢小鸟飞向小猪，我则喜欢跟鲜花有关的东西，而你完全可以用平板做其他事情。没有做不到，只有想不到！去折腾出一些“奇思妙想非常牛掰”(Wonderful Innovative Nifty Really Touchy, WinRT)的东西出来。就当是为了我，好好看看这一章。否则的话，我对 Jeff 和他无休止的写作事业可能真的会失去耐心，会把他关到一间只有针头线脑^②而且没有电的小黑屋里面。你们程序员看着办吧，是用 WinRT 写一些很酷的应用，还是再也没有 Jeff 的新书看！

总之，在你们的力挺之下，Jeff 的又一部大作诞生了。我们的家庭貌似又可以回归正常状态了。但真的正常吗？或许他不停写书才是真的正常吧。

让我们耐心等待下一本书的神秘召唤。

Kristin Trace(Jeffrey 的妻子)

2012 年 10 月



救命！Jeff 要被关小黑屋了！

① Kristin 用“sequentially”一词来吐槽顺序编程模型。——译注

② Kristin 又在吐槽“thread”了。——译注



前言

1999年10月，Microsoft的一些人首次向我展示了Microsoft .NET Framework、公共语言运行时(CLR)和C#编程语言。看到这一切时，我惊呆了，我知道我写软件的方式要发生非常大的变化了。他们请我为团队做一些顾问工作，我当即同意了。刚开始，我以为.NET Framework是Win32 API和COM上的一个抽象层。但随着我投入越来越多的时间研究，我意识到它是一个更宏伟的项目。某种程度上，它是自己的操作系统。有自己的内存管理器，自己的安全系统，自己的文件加载器，自己的错误处理机制，自己的应用程序隔离边界(AppDomain)、自己的线程处理模型等。本书解释了所有这些主题，帮你为这个平台高效地设计和实现应用程序和组件。

我写这本书是2012年10月，距离首次接触.NET Framework和C#正好13年。13年来，我以Microsoft顾问身份开发过各式各样的应用程序，为.NET Framework本身也贡献良多。作为我自己公司Wintellect(<http://Wintellect.com>)的合伙人，我还要为大量客户工作，帮他们设计、调试、优化软件以及解决使用.NET Framework时遇到的问题。正是因为有了这些资历，所以我才知道如何用.NET Framework进行高效率编程。贯穿本书所有主题，你都会看到我的经验之谈。

本书面向的读者

本书旨在解释如何为.NET Framework开发应用程序和可重用的类。具体地说，我要解释CLR的工作原理及其提供的功能，还要讨论Framework Class Library(FCL)的各个部分。没有一本书能完整地解释FCL——其中含有数以千计的类型，而且这个数字正在以惊人速度增长。所以，我准备将重点放在每个开发人员都需要注意的核心类型上面。另外，虽然不会专门讲Windows窗体、Windows Presentation Foundation(WPF)、Microsoft Silverlight、XML Web服务、Web窗体、Microsoft ASP.NET MVC、Windows Store应用等，但本书描述的技术适用于所有这些应用程序类型。

本书围绕Microsoft Visual Studio 2012/2013，.NET Framework 4.5.x和C# 5.0展开。由于Microsoft在发布这些技术的新版本时，会试图保持很大程度的向后兼容性，所以本书描述的许多内容也适合之前的版本。所有示例代码都用C#编程语言写成。但由于CLR可由许多编程语言使用，所以本书内容也适合非C#程序员。



注意 本书代码可从 Wintellect 网站下载: <http://wintellect.com/Resource-CLR-Via-CSharp-Fourth-Edition>, 也可从译者博客下载: <http://transbot.blog.163.com>。

我和我的编辑进行了艰苦卓绝的工作, 试图为你提供最准确、最新、最深入、最容易阅读和理解、没有错误的信息。但是, 即便有如此完美的团队协作, 疏漏和错误也在所难免。如果你发现了本书的任何错误或者想提出一些建设性的意见, 请发送邮件到 JeffreyR@Wintellect.com。

致谢

没有别人的帮助和技术援助, 我是不可能写好这本书的。尤其要感谢我的家人。写好一本书所投入的时间和精力无法衡量。我只知道, 没有我的妻子 Kristin 和两个儿子 Aidan 和 Grant 的支持, 根本不可能有这本书的面世。多少次想花些时间一家人小聚, 都因为本书而放弃。现在, 本书总算告一段落, 终于有时间做自己喜欢做的事情了。

本书的修订得到了一些“高人”的协助。.NET Framework 团体队的一些人(其中许多都是我的朋友)审阅了其中的章节, 我和他们进行了许多发人深省的对话。Christophe Nasarre 参与了我几本书的出版, 在审阅本书并确保我能以最恰当的方式来表达的过程中, 表现出了非凡的才能。他对本书的品质有至关重要的影响。和往常一样, 我和 Microsoft Press 的团队进行了令人愉快的合作。特别感谢 Ben Ryan, Devon Musgrave 和 Carol Dillingham。另外, 感谢 Susie Carr 和 Candace Sinclair 提供的编辑和制作支持。

勘误和支持

我们尽最大努力保证本书的准确性。勘误或更改会添加到以下网页:

<http://www.oreilly.com/catalog/errata.csp?isbn=0790145353665>

<http://go.microsoft.com/fwlink/?Linkid=266601>

如果发现未列出的错误, 可通过相同的网页报告。

如需其他支持, 请致函 Microsoft Press Book Support 部门:

mspinput@microsoft.com

注意, 上述邮件地址不提供产品支持。

最后, 本书中文版的支持(勘误和资源下载)请访问译者博客:

<http://transbot.blog.163.com>



简 明 目 录

第 I 部分 CLR 基础

第 1 章 CLR 的执行模型.....	3
第 2 章 生成、打包、部署和管理应用程序及类型	29
第 3 章 共享程序集和强命名程序集	57

第 II 部分 设计类型

第 4 章 类型基础.....	81
第 5 章 基元类型、引用类型 和值类型	99
第 6 章 类型和成员基础	135
第 7 章 常量和字段	155
第 8 章 方法	161
第 9 章 参数	185
第 10 章 属性	201
第 11 章 事件	221
第 12 章 泛型	233
第 13 章 接口	259

第 III 部分 基本类型

第 14 章 字符、字符串和文本处理	279
第 15 章 枚举类型和位标志	319
第 16 章 数组	329
第 17 章 委托	345

第 18 章 定制特性	371
第 19 章 可空值类型	389

第IV部分 核心机制

第 20 章 异常和状态管理	399
第 21 章 托管堆和垃圾回收	447
第 22 章 CLR 寄宿和 AppDomain	489
第 23 章 程序集加载和反射	515
第 24 章 运行时序列化	541
第 25 章 与 WinRT 组件互操作	569

第V部分 线程处理

第 26 章 线程基础	591
第 27 章 计算限制的异步操作	611
第 28 章 I/O 限制的异步操作	643
第 29 章 基元线程同步构造	669
第 30 章 混合线程同步构造	697

目 录

第 I 部分 CLR 基础

第 1 章 CLR 的执行模型.....3	2.4.2 使用程序集链接器.....45
1.1 将源代码编译成托管模块.....3	2.4.3 为程序集添加资源文件.....46
1.2 将托管模块合并成程序集.....6	2.5 程序集版本资源信息.....47
1.3 加载公共语言运行时.....7	2.6 语言文化.....51
1.4 执行程序集的代码.....10	2.7 简单应用程序部署 (私有部署的程序集).....52
1.4.1 IL 和验证.....15	2.8 简单管理控制(配置).....53
1.4.2 不安全的代码.....16	
1.5 本机代码生成器: NGen.exe.....17	
1.6 Framework 类库.....20	
1.7 通用类型系统.....22	
1.8 公共语言规范.....24	
1.9 与非托管代码的互操作性.....28	
第 2 章 生成、打包、部署和管理应用 程序及类型.....29	第 3 章 共享程序集和强命名程序集.....57
2.1 .NET Framework 部署目标.....29	3.1 两种程序集, 两种部署.....58
2.2 将类型生成到模块中.....31	3.2 为程序集分配强名称.....59
2.3 元数据概述.....33	3.3 全局程序集缓存.....63
2.4 将模块合并成程序集.....39	3.4 在生成的程序集中引用强命名 程序集.....65
2.4.1 使用 Visual Studio IDE 将 程序集添加到项目中.....44	3.5 强命名程序集能防篡改.....66

第 II 部分 设计类型

第 4 章 类型基础.....81	4.4 运行时的相互关系.....90
4.1 所有类型都从 System.Object 派生.....81	第 5 章 基元类型、引用类型 和值类型.....99
4.2 类型转换.....83	5.1 编程语言的基元类型.....99
4.3 命名空间和程序集.....86	

5.2 引用类型和值类型	106	第 9 章 参数	185
5.3 值类型的装箱和拆箱	111	9.1 可选参数和命名参数	185
5.3.1 使用接口更改已装箱值 类型中的字段(以及为什么 不应该这样做)	121	9.1.1 规则和原则	186
5.3.2 对象相等性和同一性	123	9.1.2 DefaultParameterValueAttribute 和 OptionalAttribute	188
5.4 对象哈希码	126	9.2 隐式类型的局部变量	188
5.5 dynamic 基元类型	127	9.3 以传引用的方式向方法传递参数	190
第 6 章 类型和成员基础	135	9.4 向方法传递可变数量的参数	195
6.1 类型的各种成员	135	9.5 参数和返回类型的设计规范	197
6.2 类型的可见性	138	9.6 常量性	198
6.3 成员的可访问性	139	第 10 章 属性	201
6.4 静态类	141	10.1 无参属性	201
6.5 分部类、结构和接口	142	10.1.1 自动实现的属性	204
6.6 组件、多态和版本控制	143	10.1.2 合理定义属性	205
6.6.1 CLR 如何调用虚方法、属性 和事件	145	10.1.3 对象和集合初始化器	208
6.6.2 合理使用类型的可见性和 成员的可访问性	148	10.1.4 匿名类型	209
6.6.3 对类型进行版本控制时的 虚方法的处理	150	10.1.5 System.Tuple 类型	212
第 7 章 常量和字段	155	10.2 有参属性	214
7.1 常量	155	10.3 调用属性访问器方法时的性能	218
7.2 字段	156	10.4 属性访问器的可访问性	219
第 8 章 方法	161	10.5 泛型属性访问器方法	219
8.1 实例构造器和类(引用类型)	161	第 11 章 事件	221
8.2 实例构造器和结构(值类型)	164	11.1 设计要公开事件的类型	222
8.3 类型构造器	167	11.1.1 第一步: 定义类型来容纳 所有需要发送给事件通知 接收者的附加信息	222
8.4 操作符重载方法	170	11.1.2 第二步: 定义事件成员	223
8.5 转换操作符方法	173	11.1.3 第三步: 定义负责引发 事件的方法来通知事件的 登记对象	224
8.6 扩展方法	176	11.1.4 第四步: 定义方法将输入 转化为期望事件	226
8.6.1 规则和原则	178	11.2 编译器如何实现事件	226
8.6.2 用扩展方法扩展各种类型	179	11.3 设计侦听事件的类型	228
8.6.3 ExtensionAttribute 类	181	11.4 显式实现事件	230
8.7 分部方法	181		

第 12 章 泛型.....	233	12.8.4 其他可验证性问题.....	254
12.1 FCL 中的泛型.....	237	第 13 章 接口.....	259
12.2 泛型基础结构.....	238	13.1 类和接口继承.....	259
12.2.1 开放类型和封闭类型.....	239	13.2 定义接口.....	260
12.2.2 泛型类型和继承.....	240	13.3 继承接口.....	261
12.2.3 泛型类型同一性.....	242	13.4 关于调用接口方法的更多探讨.....	263
12.2.4 代码爆炸.....	243	13.5 隐式和显式接口方法实现 (幕后发生的事情).....	264
12.3 泛型接口.....	243	13.6 泛型接口.....	266
12.4 泛型委托.....	244	13.7 泛型和接口约束.....	268
12.5 委托和接口的逆变和协变泛型 类型实参.....	245	13.8 实现多个具有相同方法名和 签名的接口.....	269
12.6 泛型方法.....	247	13.9 用显式接口方法实现来增强 编译时类型安全性.....	270
12.7 泛型和其他成员.....	249	13.10 谨慎使用显式接口方法实现.....	271
12.8 可验证性和约束.....	250	13.11 设计: 基类还是接口.....	274
12.8.1 主要约束.....	252		
12.8.2 次要约束.....	253		
12.8.3 构造器约束.....	254		

第Ⅲ部分 基本类型

第 14 章 字符、字符串和文本处理.....	279	14.4.2 将多个对象格式化成一个 字符串.....	303
14.1 字符.....	279	14.4.3 提供定制格式化器.....	304
14.2 System.String 类型.....	282	14.5 解析字符串来获取对象: Parse.....	306
14.2.1 构造字符串.....	282	14.6 编码: 字符和字节的相互转换.....	308
14.2.2 字符串是不可变的.....	284	14.6.1 字符和字节流的编码和 解码.....	313
14.2.3 比较字符串.....	285	14.6.2 Base-64 字符串编码和 解码.....	314
14.2.4 字符串留用.....	290	14.7 安全字符串.....	315
14.2.5 字符串池.....	293	第 15 章 枚举类型和位标志.....	319
14.2.6 检查字符串中的字符和 文本元素.....	293	15.1 枚举类型.....	319
14.2.7 其他字符串操作.....	295	15.2 位标志.....	324
14.3 高效率构造字符串.....	296	15.3 向枚举类型添加方法.....	328
14.3.1 构造 StringBuilder 对象.....	296	第 16 章 数组.....	329
14.3.2 StringBuilder 的成员.....	297	16.1 初始化数组元素.....	331
14.4 获取对象的字符串表示: ToString.....	299	16.2 数组转型.....	333
14.4.1 指定具体的格式和 语言文化.....	299		

16.3 所有数组都隐式派生自 System.Array	335	17.7.3 简化语法 3: 局部变量不需要 手动包装到类中即可传给 回调方法.....	364
16.4 所有数组都隐式实现 IEnumerable, ICollection 和 IList.....	336	17.8 委托和反射	367
16.5 数组的传递和返回	337	第 18 章 定制特性	371
16.6 创建下限非零的数组	338	18.1 使用定制特性	371
16.7 数组的内部工作原理	339	18.2 定义自己的特性类	374
16.8 不安全的数组访问和固定大小的 数组	342	18.3 特性构造器和字段/属性数据类型	377
第 17 章 委托	345	18.4 检测定制特性	378
17.1 初识委托	345	18.5 两个特性实例的相互匹配	382
17.2 用委托回调静态方法	347	18.6 检测定制特性时不创建从 Attribute 派生的对象.....	384
17.3 用委托回调实例方法	349	18.7 条件特性类	387
17.4 委托揭秘	349	第 19 章 可空值类型	389
17.5 用委托回调多个方法(委托链).....	353	19.1 C#对可空值类型的支持	391
17.5.1 C#对委托链的支持	356	19.2 C#的空接合操作符	393
17.5.2 取得对委托链调用的 更多控制.....	357	19.3 CLR 对可空值类型的特殊支持	394
17.6 委托定义不要太多(泛型委托).....	359	19.3.1 可空值类型的装箱	394
17.7 C#为委托提供的简化语法.....	360	19.3.2 可空值类型的拆箱	395
17.7.1 简化语法 1: 不需要构造 委托对象.....	360	19.3.3 通过可空值类型 调用 GetType.....	395
17.7.2 简化语法 2: 不需要定义 回调方法(lambda 表达式)	361	19.3.4 通过可空值类型调用接口 方法	395

第IV部分 核心机制

第 20 章 异常和状态管理	399	20.8 设计规范和最佳实践	422
20.1 定义“异常”	399	20.8.1 善用 finally 块	423
20.2 异常处理机制	401	20.8.2 不要什么都捕捉	424
20.2.1 try 块	402	20.8.3 得体地从异常中恢复	425
20.2.2 catch 块	402	20.8.4 发生不可恢复的异常时回滚 部分完成的操作——维持 状态.....	426
20.2.3 finally 块	404	20.8.5 隐藏实现细节来维系协定	427
20.3 System.Exception 类	407	20.9 未处理的异常	429
20.4 FCL 定义的异常类	410	20.10 对异常进行调试	433
20.5 抛出异常	412	20.11 异常处理的性能问题	435
20.6 定义自己的异常类	413	20.12 约束执行区域(CER).....	438
20.7 用可靠性换取开发效率	415		

20.13 代码协定	441	22.7 高级宿主控制	509
第 21 章 托管堆和垃圾回收	447	22.7.1 使用托管代码管理 CLR	509
21.1 托管堆基础	447	22.7.2 写健壮的宿主应用程序	510
21.1.1 从托管堆分配资源	448	22.7.3 宿主如何拿回它的线程	511
21.1.2 垃圾回收算法	449	第 23 章 程序集加载和反射	515
21.1.3 垃圾回收和调试	451	23.1 程序集加载	516
21.2 代: 提升性能	454	23.2 使用反射构建动态可扩展应用 程序	520
21.2.1 垃圾回收触发条件	458	23.3 反射的性能	521
21.2.2 大对象	459	23.3.1 发现程序集中定义的类型	522
21.2.3 垃圾回收模式	459	23.3.2 类型对象的准确含义	522
21.2.4 强制垃圾回收	462	23.3.3 构建 Exception 派生类型的 层次结构	524
21.2.5 监视应用程序的内存使用	463	23.3.4 构造类型的实例	525
21.3 使用需要特殊清理的类型	464	23.4 设计支持加载项的应用程序	527
21.3.1 使用包装了本机资源的 类型	470	23.5 使用反射发现类型的成员	529
21.3.2 一个有趣的依赖性问题	474	23.5.1 发现类型的成员	530
21.3.3 GC 为本机资源提供的 其他功能	475	23.5.2 调用类型的成员	533
21.3.4 终结的内部工作原理	479	23.5.3 使用绑定句柄减少进程的 内存消耗	537
21.3.5 手动监视和控制对象的 生存期	481	第 24 章 运行时序列化	541
第 22 章 CLR 寄宿和 AppDomain	489	24.1 序列化/反序列化快速入门	542
22.1 CLR 寄宿	489	24.2 使类型可序列化	546
22.2 AppDomain	491	24.3 控制序列化和反序列化	548
跨越 AppDomain 边界访问对象	494	24.4 格式化器如何序列化类型实例	551
22.3 卸载 AppDomain	504	24.5 控制序列化/反序列化的数据	552
22.4 监视 AppDomain	505	24.6 流上下文	558
22.5 AppDomain FirstChance 异常通知	507	24.7 类型序列化为不同类型以及对象 反序列化为不同对象	559
22.6 宿主如何使用 AppDomain	507	24.8 序列化代理	562
22.6.1 可执行应用程序	507	24.9 反序列化对象时重写程序集/类型	566
22.6.2 Microsoft Silverlight 富 Internet 应用程序	508	第 25 章 与 WinRT 组件互操作	569
22.6.3 Microsoft ASP.NET 和 XML Web 服务应用程序	508	25.1 CLR 投射与 WinRT 组件类型系统 规则	571
22.6.4 Microsoft SQL Server	509		
22.6.5 更多的用法只局限于 想象力	509		

25.2 框架投射	575
25.2.1 从.NET 代码中调用异步 WinRT API	575
25.2.2 WinRT 流和.NET 流之间的 互操作	579

25.2.3 在 CLR 和 WinRT 之间传输 数据块	580
25.3 用 C# 定义 WinRT 组件	583

第V部分 线程处理

第26章 线程基础

26.1 Windows 为什么要支持线程	591
26.2 线程开销	592
26.3 停止疯狂	595
26.4 CPU 发展趋势	597
26.5 CLR 线程和 Windows 线程	598
26.6 使用专用线程执行异步的计算 限制操作	599
26.7 使用线程的理由	601
26.8 线程调度和优先级	603
26.9 前台线程和后台线程	608
26.10 继续学习	609

第27章 计算限制的异步操作

27.1 CLR 线程池基础	612
27.2 执行简单的计算限制操作	612
27.3 执行上下文	614
27.4 协作式取消和超时	615
27.5 任务	619
27.5.1 等待任务完成并获取结果	620
27.5.2 取消任务	622
27.5.3 任务完成时自动启动 新任务	623
27.5.4 任务可以启动子任务	625
27.5.5 任务内部揭秘	625
27.5.6 任务工厂	627
27.5.7 任务调度器	628
27.6 Parallel 的静态 For, ForEach 和 Invoke 方法	630
27.7 并行语言集成查询(PLINQ)	634
27.8 执行定时计算限制操作	636

27.9 线程池如何管理线程	639
27.9.1 设置线程池限制	639
27.9.2 如何管理工作线程	640

第28章 I/O 限制的异步操作

28.1 Windows 如何执行 I/O 操作	643
28.2 C# 的异步函数	647
28.3 编译器如何将异步函数转换成 状态机	649
28.4 异步函数扩展性	653
28.5 异步函数和事件处理程序	655
28.6 FCL 的异步函数	656
28.7 异步函数和异常处理	658
28.8 异步函数的其他功能	658
28.9 应用程序及其线程处理模型	661
28.10 以异步方式实现服务器	663
28.11 取消 I/O 操作	664
28.12 有的 I/O 操作必须同步进行	665
28.13 I/O 请求优先级	666

第29章 基元线程同步构造

29.1 类库和线程安全	671
29.2 基元用户模式和内核模式构造	672
29.3 用户模式构造	673
29.3.1 易变构造	674
29.3.2 互锁构造	678
29.3.3 实现简单的自旋锁	682
29.3.4 Interlocked Anything 模式	685
29.4 内核模式构造	687
29.4.1 Event 构造	691
29.4.2 Semaphore 构造	693
29.4.3 Mutex 构造	694

第 30 章 混合线程同步构造.....	697	30.3.5 CountdownEvent 类.....	711
30.1 一个简单的混合锁	697	30.3.6 Barrier 类	711
30.2 自旋、线程所有权和递归	699	30.3.7 线程同步构造小结	712
30.3 FCL 中的混合构造	701	30.4 著名的双检锁技术	713
30.3.1 ManualResetEventSlim 类和		30.5 条件变量模式	717
SemaphoreSlim 类	701	30.6 异步的同步构造	719
30.3.2 Monitor 类和同步块	701	30.7 并发集合类	723
30.3.3 ReaderWriterLockSlim 类	706	译者后记	727
30.3.4 OneManyLock 类	709		



第 I 部分 CLR 基础

- ▶ 第 1 章 CLR 的执行模型
- ▶ 第 2 章 生成、打包、部署和管理应用程序及类型
- ▶ 第 3 章 共享程序集和强命名程序集