

VB.NET 高级编程

Fred Barwell

等著

Richard Blair

康 博 译

清华大学出版社

(京) 新登字 158 号

北京市版权局著作权合同登记号: 01-2001-4318

内 容 简 介

Visual Basic.NET 是 Visual Basic 的最新版本,也是 Microsoft 的 .NET 战略中的主要组成部分。与以前的 VB 版本相比,VB.NET 有了质的飞跃,变得更为灵活也更为强大。Microsoft 在 VB.NET 中融入了许多新功能。其中非常重要的就是完全实现了面向对象的编程,并极大地增强了 Web 的设计功能。

本书主要介绍了 .NET Framework 和公共语言运行时(CLR)、应用继承从基类中派生类、利用 Try...Catch 结构处理错误、开发多线程应用程序、COM 和 .NET 组件的互操作性,以及创建定制控件、使用 ADO.NET 和 XML 访问数据、构建 Web 应用程序,以及创建和使用 Web 服务等内容。

本书适用于有一定经验的 Visual Basic 开发人员。通过本书程序员可以深入理解 VB.NET,并且利用 VB.NET 的强大功能构建实用的应用程序。

Fred Barwell,Richard Blair:Professional VB.NET

EISBN: 1-861004-97-4

Copyright©2001 by Wrox Press Ltd.

Authorized translation from the English language edition published by Wrox Press Ltd.

All rights reserved.For sale in the People's Republic of China only.

Chinese simplified language edition published by Tsinghua University Press.

本书中文简体字版由清华大学出版社和英国乐思出版公司合作出版。未经出版者书面许可,不得以任何方式复制或抄袭本书内容。

版权所有,翻印必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

书 名: VB.NET 高级编程

作 者: Fred Barwell Richard Blair 等著 康博 译

出 版 者: 清华大学出版社(北京清华大学学研大厦,邮编 100084)

<http://www.tup.tsinghua.edu.cn>

责任编辑: 于平 李阳

印 刷 者: 清华大学印刷厂

发 行 者: 新华书店总店北京发行所

开 本: 787×1092 1/16 印 张: 51.25 字 数: 1311 千字

版 次: 2002 年 3 月第 1 版 2002 年 3 月第 1 次印刷

书 号: ISBN 7-302-05236-0/TP·3081

印 数: 0001~5000

定 价: 88.00 元

出版者的话

近年来，国内计算机类图书出版业得到了空前的发展，面向初级用户的应用类软件图书铺天盖地，但是真正有深度和内涵的高端图书不多。已经掌握计算机和网络基础知识的人们，尤其是 IT 专业人士迫切需要“阳春白雪”。IT 图书市场呼唤精品！

为了满足这种市场需求，清华大学出版社从世界出版业知名品牌 Wrox 出版公司引进了受到无数 IT 专业人士青睐，被奉为 IT 出版界经典之作的 Professional 系列丛书。这套讲述最新编程技术与开发环境的高级编程丛书，从头到尾都贯穿了 Wrox 出版公司“由程序员为程序员而著(Programmer to Programmer)”的出版理念，每一本书无不是出自软件大师之手。实际上，Wrox 公司的图书作者都是世界顶级 IT 公司(如 Microsoft, IBM, Oracle 以及 HP 等)的资深程序员，他们的作品既深入研究编程机理，传授最新编程技术，又站在程序员的角度，指导程序员拓展编程思路，学习实用开发技巧，从而风靡世界各地，被 IT 专业人士和程序员视为职业生涯中的必读之作。

作为国内 IT 出版社中最知名品牌，清华大学出版社与 Wrox 公司合作引进了这套 Professional 系列，然后迅速组织了一批相关领域的知名专家学者进行翻译，经过编辑人员认真细致的加工后，现陆续奉献给广大读者。

读者可以从 www.wrox.com 网站下载所需的源代码并获得相关的技术支持。同时，也欢迎广大读者参与 p2p.wrox.com 网站上的在线讨论，与世界各地的编程人员交流读书感受和编程体验。

序 言

.NET 旨在提供一种新的环境，利用这种环境，可以开发几乎所有在 Windows（可能的话，将来在其他平台）上运行的应用程序，并且 VB.NET 可能会成为使用这种架构的最受欢迎的开发工具。VB.NET 是一种 .NET 兼容语言，除了继承原因之外，几乎与新型 C# 语言拥有相同的技术功能，并且 .NET 兼容 C++。利用 VB.NET，可以开发动态 Web 网页、分布式应用程序的组件、数据库访问组件或传统的 Windows 桌面应用程序。

为了将 VB 集成到 .NET Framework 中，已经添加了许多新的功能以便创建 VB.NET。总体上来说，这些变化是如此之广，以致 VB.NET 可以看作是一种新的语言，而不是一种升级。然而，这些变化有必要为开发人员提供他们寻求的功能：真正的 OOP、更加易于部署、更好的互操作性，以及开发传统和 Web 应用程序的固有环境。

在本书中，我们实际上从头至尾讲述了首次推出的 VB.NET：开始，我们讨论了 .NET Framework；结尾，我们讨论了创建和部署 Windows 和 Web Services，以及所涉及安全性的最佳实践。在它们之间，我们讨论了从数据库访问到与诸如 XML 的其他技术集成的一切事物，并且详细地介绍了新功能。您将会看到，VB.NET 已经表现为一种强大、但是仍然易于使用的语言，使您定位桌面就像定位 Internet 一样容易。

本书适用的读者对象

由于 VB.NET 仍然在发展之中，所以，本书主要针对有经验的 Visual Basic 开发人员。特别值得一提的是，本书是针对需要过渡到 VB.NET，并且想了解从《VB.NET 程序设计教程》中不能了解的更详细、更新信息的开发人员。尽管 .NET Framework 为 Internet 编程提供了新型而强大的方式，但是，本书并非为 Web 开发人员专门编写，他们应该参考《ASP.NET 高级编程》。

使用本书需要的条件

为了完全而充分地利用本书，应该具有 VS.NET 的企业版或专业版。有几章还需要其他软件，例如，为了完成 17 章和 18 章中的示例，需要 SQL Server。此外，还需要 MSMQ 来进行第 17 章的排队事务处理，它在 Windows 2000 Server 和 Windows 2000 Professional 中都可以找到。

本书的内容

第 1 章开始试图解释 .NET 的重要性，它对应用程序开发的改变有多大。为了让大家理解为什么需要 .NET，当前开发技术的缺陷在哪些方面，我们首先讨论包括 COM 和 DNA 的体系结构模式，然后讲述 .NET 如何纠正这些技术中的缺点，其中包括公共语言运行时(Common Language Runtime, CLR)。

第 2 章简短地讲述 .NET 对 Visual Basic 的改变。它从 IDE 开始，然后讲解数据和应用程序类型、对象功能、错误处理，以及其他的语法和附加的变化。



第 3 章，我们才第一次讨论 VB.NET。该章只为创建 VB.NET 应用程序提供了简短的介绍，因为在此处提及的议题将在以后的章节中详细讨论。

第 4 章集中讲述 IDE(Integrated Development Environment, 集成开发环境)，将描述和检验 IDE 中操作的各种窗口和菜单选项。在该语境中，有必要讨论 Windows Forms、Web Forms 和在其他章节将详细讨论的几种其他技术。

第 5 章检查了 .NET 平台的核心——公共语言运行时。CLR 负责管理执行为 .NET 平台编译的代码——详细地说，就是版本变化和部署、内存管理、跨语言集成、元数据和 IL 反汇编器。

第 6 章将讨论在 Visual Basic 中所使用的许多类型，尤其是基础的 Object 类。本章的主要目标是让您熟悉值和引用类型，帮助 VB6 背景的开发人员理解 VB.NET 中定义变量的方式的主要差异。

第 7 章的主题是名称空间。该章介绍了名称空间及其继承性结构，名称空间的解释和某些常用的名称空间。此外，它还介绍了如何创建新的名称空间，以及在工程中如何导入名称空间，如何给名称空间起别名。

第 8 章讲述了 Windows Forms，并且主要集中讨论窗体和内置控件。它还讨论了哪些是新的，哪些对于 Visual Basic 前一版本已经改变了，并且还讨论了 System.Windows.Forms 名称空间。在这一章，包括了 VB.NET 中对 Windows Forms 的所有改进，以及当更新现有应用程序或编写新的应用程序时开发人员如何利用这些窗体。

第 9 章通过详细讨论 CLR 异常处理程序和新的 Try...Catch...Finally 结构，描述了 VB.NET 中错误处理的原理。除此之外，还描述了受管理的和不受管理代码之间的错误处理，错误和跟踪的记录，以及如何利用这些方法获取程序工作方式的反馈。

第 10 章，我们讨论了 Windows 控件的创建。特别是，我们讨论如何从其他控件继承，如何建立复合控件，如何根据 Control 类从头开始编写控件。

第 11 章探索了 VB.NET 中类和对象的创建和使用。该章还定义了对象、类、实例、封装、抽象、多态性和继承性。

第 12 章讲述了继承及其在 VB.NET 中如何利用继承。我们将解释支持 VB.NET 中继承性的语法，如何创建简单和抽象的基类，如何创建可以派生其他类的基类（以及创建那些派生的类），以及类的再细分(subclassing)。

第 13 章将深入地探讨面向对象的编程，并且完全定义所列的功能，探讨如何使用这些概念。我们将讨论 4 个主要的面向对象概念（抽象性、封装性、多态性、继承），以及这些概念如何应用到设计和开发中，以创建高效的面向对象的应用程序。

第 14 章集中介绍了装配件和它们在 CLR 中的用法。在这一章里，我们将讨论装配件的结构、装配件所包含的内容，以及装配件所包含的信息。此外，我们还讨论了装配件的各种表现形式及其它们在部署中的作用。

第 15 章，我们将学习线程，并且将看到 .NET Framework 中的各种对象如何让用户开发出多线程的应用程序。我们将讨论如何创建线程，如何将它们与进程关联在一起，以及多任务和多线程之间的差异。

第 16 章介绍了 COM 和 .NET 组件的互操作性，Microsoft 提供了哪些工具用于帮助将这两项技术链接起来。传统的基本 COM 对象可以在 VB.NET 程序中运行，同样在 COM 对象中也

可以运行一些 VB.NET 代码。

第 17 章探索了 .NET Component Services，特别还讨论了事务处理和排队的组件。在此所涉及的是 RegSvcs 工具，它被用来利用 Component Services 注册组件。

第 18 章为了能够建立灵活、快速和可伸缩的数据访问对象和应用程序，集中讨论了所需要了解的 ADO.NET 对象模型。在这一章，我们讲述了 ADO 向 ADO.NET 的转化，还讨论了为了建立对 .NET 应用程序的数据访问而需要快速启动并运行的 ADO.NET 中的主要对象。

第 19 章展示了 .NET Framework 用于 XML 的生成和操作的功能。该章还讲述了 .NET Framework 的 XML 相关的名称空间，并且详细地讨论了这些名称空间所展示的类的子集。这一章还接触了一系列利用 XML，特别是利用 ADO.NET 和 SQL Server 的技术。

第 20 章探讨了 Web Forms 以及如何从它们的使用中受益。本章通过逐渐使用越来越复杂的示例，讲述了如何利用快速应用程序开发(Rapid Application Development)的强大功能开发 Web 应用程序。

第 21 章则围绕着全新形式的 Visual Basic 控件开发、定制 Web 控件进行讨论。它描述了 .NET Framework 中可用的各种形式的自定义的 Web 控件开发，还探讨了 Web 用户的基本结构和子类的控件，以及复合控件、模板控件和一些控件设计时的最佳实践。

第 22 章探索了如何利用 VB.NET 创建和使用 Web Services。该章还讨论 CLR 提供的抽象类以建立和利用 Web Services，并且还讨论了一些支持 Web Services 的技术，然后介绍了 Web Services 中的安全性。最后，该章讲述了使用任何分布式体系结构的缺点和 Web Services 的前景。

第 23 章讨论了在 Windows Services 的生成过程中，如何使用 VB.NET。该章将讨论 Windows Services 的创建、安装、运行和调试。在这部分中，还包括了远程服务的示例和 InstallUtil.exe 实用程序的使用。

第 24 章中，将学习 Visual Studio.NET 和 CLR 在部署时，为我们提供的功能和内容。该章将让您理解可用的部署选项的基础知识。

第 25 章研究为了注重安全性而由 .NET 提供的附加工具和功能。在该章，我们将讨论 Caspol.Exe 和 Permview.exe，以及它们如何帮助建立和维护安全策略。我们还描述了 System.Security.Permissions 名称空间，以及它与管理权限的关系。最后，我们将讨论三种类型的安全性——基于角色的、基于代码访问的，以及基于身份的。我们还探索它们与代码的权限处理段的关系，以及它们如何用于代码的权限处理段中。

客户支持

我们已经尽最大努力使本书准确、实用，但是实际的情况是本书事实上对您所起的作用。请让我们知道您对本书的评价，即可以向 feedback@wrox.com 发邮件。但是，请一定要在您的信中注明本书的书名。

源代码

本书的源代码可以从 <http://www.wrox.com> 下载。



勘误表

我们已经尽力确保本书中没有错误。但是，“人孰能无过？”，我们认为必须告诉您已经发现和改正的错误。本书的勘误表在<http://www.wrox.com>。如果您发现了还未报告的错误，请告诉我们。

电子邮件支持

如果您希望直接就本书的问题向对本书知之甚多的专家咨询，那么，就向 support@wrox.com 发电子邮件，在电子邮件的“主题”(Subject)字段，加上本书的名称和 ISBN 的最后 4 位号码。典型的电子邮件应该包括下列内容：

- 在“主题”字段加上书的名称、ISBN 的最后 4 位数字和问题所在的页码。
- 在邮件的正文中加上您的姓名、联系信息和问题。

我们不会发给您垃圾邮件。我们只需要详细的情况以节省您的宝贵时间和我们的时间。当您发送电子邮件时，它会直接到达以下支持链：

- 客户支持——您的消息会传送到我们的客户支持人员，他们是阅读信息的第一人。他们有常见问题的文件，会迅速回答一般性的问题。他们回答关于本书和网站的一般性问题。

- 编辑——更深的问题会转发到负责本书的技术编辑处。他（或）她具有编程或特殊产品的经验，能够回答某个主题的详细技术问题。一旦已经解决了某个问题，该编辑会更新网站上的勘误表。

- 作者——最后，在编辑都不能回答问题的情况下（这种情况很少出现），这些问题将转发到作者。我们试图保护作者不要从写作中分心，但是，我们也很愿意将特殊的问题转发给他们。所有的 Wrox 作者帮助支持他们的书籍。他们对客户和编辑回复电子邮件，所有的读者都会从中受益。

P2P.WROX.COM

为了获得作者和同行的支持，请加入 VB.NET 邮件列表中。我们的独特系统在邮件列表、论坛和新闻组中提供 **programmer to programmer support**——所有都在一对一的电子邮件系统。要确信您的查询不仅正在被支持专家检查，而且被许多 Wrox 作者和我们邮件列表中的许多行业专家检查。在 p2p.wrox.com，不仅在您阅读本书时，而且在您开发自己的应用程序时，您将会发现支持您的 Visual Basic 编程员的许多不同列表。

为什么这个系统提供最佳的支持

您可加入该邮件列表中，也可以每周分类接收它们。如果您没有时间或设备接收该邮件列表，那么您可以搜索我们的在线文档。垃圾邮件和广告邮件会被删除，您自己的电子邮件地址会被独特的 Lyris 系统保护起来。任何加入或退出列表的查询，或者一般性的列表，都应发送到 listsupport@p2p.wrox.com。

目 录

第 1 章 需要 Microsoft.NET 的原因	1
1.1 我们现在使用的工具有什么问题	1
1.1.1 DNA 编程模式	2
1.1.2 COM 的限制	4
1.1.3 VB6 用于 DNA 应用程序开发的局限	5
1.1.4 基于 DNA 的 Internet 开发的其他局限	6
1.1.5 Windows API 的缺点——需要抽象操作系统	7
1.2 解决方案——Microsoft.NET	8
1.3 Microsoft .NET 的结构	9
1.3.1 基础——公共语言运行时	10
1.3.2 下一层——.NET Class Framework	11
1.3.3 用户界面和程序接口	12
1.3.4 XML 作为.NET 的“元语言”	12
1.4 Microsoft.NET 如何达到它的目标	13
1.4.1 简化软件开发	13
1.4.2 高度分布的系统	13
1.4.3 Web 上的更好用户界面	14
1.4.4 简化部署	14
1.4.5 对各种语言的支持	14
1.4.6 平台的可扩展性	15
1.4.7 未来的可移植性	15
1.5 COM 的角色	15
1.6 DNA 的角色	16
1.7 其他优点	16
1.8 对 Visual Basic 的影响	16
1.9 避免混淆——.NET 企业服务器的角色	17
1.10 小结	17
第 2 章 VB.NET 中的新特性概述	18
2.1 新的 IDE 特性	19
2.1.1 工具箱	20
2.1.2 Solution Explorer	20
2.1.3 类视图	20



2.1.4	属性窗口	20
2.1.5	代码窗口	20
2.2	数据类型的变化	21
2.2.1	布尔类型	21
2.2.2	整型	22
2.2.3	字符串	22
2.2.4	变量	22
2.3	应用程序类型	23
2.3.1	Windows 应用程序	23
2.3.2	Web 应用程序	23
2.3.3	Web 服务	24
2.4	新对象能力	24
2.5	结构化错误处理	25
2.6	语法的不兼容性	26
2.6.1	属性	26
2.6.2	用户定义的类型	26
2.6.3	作用域	27
2.6.4	ByVal	27
2.6.5	GoSub 和 Return	27
2.7	其他变化	28
2.7.1	圆括号和子例程	28
2.7.2	没有 Set	28
2.7.3	Return	28
2.7.4	DefType	29
2.8	小结	29
第 3 章	VB.NET 程序的初步讨论	30
3.1	创建范例应用程序	30
3.1.1	New Project 对话框	30
3.1.2	Solution Explorer	32
3.1.3	Properties 窗口	33
3.2	HelloWorld 范例分析	34
3.2.1	选项卡与 MDI 界面	34
3.2.2	作为类的窗体	34
3.2.3	代码区域	35
3.3	AssemblyInfo.vb 文件	41
3.3.1	名称空间和 Imports 关键字	42
3.3.2	装配件属性标志	43

3.4 增强范例应用程序	44
3.4.1 访问工具框	44
3.4.2 添加控件	45
3.4.3 添加事件处理程序	45
3.4.4 代码清单	46
3.5 进入 Hello World 范例	49
3.5.1 建立配置	49
3.5.2 建立我们的应用程序	51
3.5.3 运行应用程序	52
3.6 小结	52
第 4 章 Visual Studio.NET 集成开发环境	53
4.1 Visual Studio.NET 的 IDE	53
4.2 解决方案与工程	54
4.3 Visual Studio.NET IDE 的概述	54
4.3.1 配置文件	55
4.3.2 选择 Help 选项	55
4.3.3 Start Page	55
4.3.4 存放新闻和信息的位置	56
4.3.5 选项卡式窗口	58
4.3.6 非停靠窗口	65
4.3.7 AutoHide	66
4.3.8 Server Explorer	66
4.3.9 完成其他工程类型时的差异	69
4.4 Visual Studio.NET 中的工具栏	72
4.5 Visual Studio.NET 中的菜单结构	73
4.5.1 File 菜单	73
4.5.2 Edit 菜单	73
4.5.3 View 菜单	73
4.5.4 Project 菜单	73
4.5.5 Build 菜单	74
4.5.6 Debug 菜单	74
4.5.7 Tools 菜单	74
4.5.8 Window 菜单	75
4.5.9 Help 菜单	75
4.5.10 与 VB6 相比的其他变化	75
4.6 Visual Studio 中的宏	75
4.7 新型开发工具	77



4.8 XML、XSD Schemas 和 XSLT 文件的可视化编辑器	78
4.9 跨工程和进程的调试	78
4.10 扩展的 IntelliSense	79
4.11 小结	79
第 5 章 公共语言运行时	80
5.1 .NET 应用程序的概述	80
5.1.1 装配件	81
5.1.2 模块	81
5.1.3 类型	82
5.2 版本转换和部署	82
5.2.1 对于版本的更好支持	82
5.2.2 更好的部署	83
5.3 内存管理	83
5.3.1 更好的无用单元收集功能	83
5.3.2 对象的更快的内存分配	87
5.3.3 GC 优化	87
5.4 跨语言的集成	90
5.4.1 公共类型系统	90
5.4.2 引用和值类型	91
5.4.3 自定义类型	93
5.4.4 值类型的装箱和拆箱	98
5.5 元数据	99
5.5.1 对元数据的更好支持	100
5.5.2 属性标志	100
5.5.3 Reflection API	102
5.6 IL Disassembler	102
5.7 小结	103
第 6 章 变量和类型	104
6.1 值类型(结构)	104
6.1.1 布尔型	106
6.1.2 整型	108
6.1.3 小数	110
6.1.4 Char 和 Byte	112
6.1.5 DateTime	113
6.2 原始类型	113
6.3 引用类型(类)	115
6.3.1 对象的概述	115

6.3.2	对象	117
6.3.3	字符串	119
6.3.4	DBNull	122
6.3.5	数组	123
6.3.6	Collections	125
6.4	显式选项和严格选项	127
6.4.1	显式转换	128
6.5	参数传递	130
6.6	装箱	132
6.7	撤销的关键字和函数	133
6.8	小结	135
第 7 章	名称空间	136
7.1	名称空间的定义	136
7.1.1	名称空间 Namespace_Sampler 和引用	141
7.1.2	通用的名称空间	143
7.2	导入名称空间并为其命名别名	145
7.2.1	导入名称空间	145
7.2.2	为名称空间命名别名	148
7.3	创建名称空间	151
7.4	小结	161
第 8 章	Windows Forms	162
8.1	System.Windows.Forms 名称空间	162
8.2	窗体是类	163
8.3	设计窗体	165
8.3.1	非用户界面控件不在窗体中出现	165
8.3.2	设计时间网格	165
8.3.3	设置启动窗体	165
8.3.4	窗体边框	166
8.3.5	总在最上面	167
8.3.6	启动位置	168
8.3.7	窗体的不透明性(透明性)	168
8.3.8	可视化继承	168
8.3.9	MDI Forms	169
8.3.10	窗体尺寸限制的设置	171
8.3.11	滚动窗体	171
8.4	运行时的窗体	171
8.5	控件	172



8.5.1 所有控件的普遍设计特征	173
8.5.2 数据输入确认和错误提供器	179
8.5.3 菜单	182
8.5.4 工具栏	186
8.5.5 DataGrid	188
8.5.6 公共对话框	190
8.5.7 拖和放	193
8.5.8 标准 Windows.Forms 控件的总结	195
8.5.9 撤销控件	199
8.5.10 使用 ActiveX 控件	200
8.5.11 其他方便的编程技巧	200
8.6 小结	201
第 9 章 错误处理	202
9.1 CLR 异常处理	202
9.2 标准化错误处理	203
9.3 On Error 语句	203
9.4 异常处理程序	205
9.4.1 Try...Catch...Finally	205
9.4.2 异常的属性和方法	209
9.4.3 受管理和不受管理代码之间的错误处理	216
9.5 错误记录	217
9.5.1 事件日志	218
9.5.2 写入追踪文件	221
9.6 调试与测定性能	223
9.7 小结	227
第 10 章 创建 Windows 控件	229
10.1 控件的源	229
10.1.1 内置控件	229
10.1.2 现有 ActiveX 控件	229
10.1.3 第三方控件	230
10.1.4 自定义控件	230
10.2 在 .NET 中开发自定义控件的 3 种方法	230
10.2.1 从另一个控件继承	230
10.2.2 建立复合控件	231
10.2.3 根据 Control 类从头编写控件	231
10.3 控件创建的基类	231
10.3.1 控件类	231

10.3.2	UserControl 类	233
10.4	从其他控件继承	234
10.4.1	进程的概述	234
10.4.2	例子 1 – 一个仅能输入数值的文本框	234
10.4.3	在自定义控件中改变逻辑处理	237
10.4.4	调试自定义控件	237
10.4.5	为自定义控件增加额外的逻辑处理	238
10.4.6	为自定义控件创建属性	238
10.4.7	为继承控件定义自定义事件	241
10.4.8	例子 2 – 限制选择项数量的复选列表框	241
10.5	复合 UserControl	246
10.5.1	复合 UserControl 的例子	246
10.5.2	如何进行调整大小工作	247
10.5.3	为控件设置最小尺寸	247
10.5.4	显示子控件的属性	248
10.5.5	逐步完成例子	248
10.6	从头建立控件	252
10.6.1	用 GDI+显示自定义控件	252
10.6.2	该控件的完整实现	257
10.7	小结	257
第 11 章	对象语法概述	258
11.1	面向对象的术语	259
11.1.1	对象、类和实例	259
11.1.2	对象的组成	260
11.2	使用对象	264
11.2.1	对象声明和对象实例	264
11.2.2	对象引用	266
11.2.3	间接引用对象	268
11.2.4	前期及后期绑定	268
11.3	创建类	271
11.4	构造函数方法	288
11.5	高级概念	294
11.5.1	高级方法	294
11.5.2	类和组件	306
11.6	.NET 属性	310
11.7	小结	311



第 12 章 继承和接口	312
12.1 继承	312
12.2 多接口	358
12.2.1 对象接口	359
12.2.2 辅助接口	360
12.3 小结	366
第 13 章 对象和组件的应用	367
13.1 抽象性	367
13.2 封装性	370
13.3 多态性	373
13.3.1 方法签名	373
13.3.2 实现多态性	373
13.4 继承	383
13.4.1 何时使用继承	383
13.4.2 继承和多接口	387
13.4.3 进一步探讨	392
13.4.4 容易被破坏的基类问题	392
13.5 小结	395
第 14 章 装配件	397
14.1 清单	397
14.1.1 标识部分	399
14.1.2 被引用的装配件	400
14.2 装配件和部署	401
14.2.1 应用程序私有装配件	401
14.2.2 共享装配件	402
14.3 版本问题	404
14.3.1 应用程序隔离	404
14.3.2 并行执行	405
14.3.3 自描述	405
14.3.4 版本策略	405
14.3.5 配置文件	406
14.4 小结	410
第 15 章 线程	411
15.1 线程的含义	411
15.1.1 进程与线程	412
15.1.2 何时使用线程	415

15.2	线程范例	416
15.2.1	创建 ThreadWorker	417
15.2.2	调用 SpinUp	418
15.3	同步	419
15.3.1	中断、等待状态和传送信号	420
15.3.2	同步的过程	423
15.3.3	死锁	439
15.4	线程池	439
15.4.1	Worker 对象	441
15.4.2	创建 Worker 对象	443
15.5	使用线程池改进 ManualResetEvent.WaitOne	450
15.6	小结	456
第 16 章	使用传统的 COM 和接口	457
16.1	COM 简介	457
16.2	现在所使用的 COM 和 .NET	458
16.2.1	传统组件	458
16.2.2	.NET 应用程序	461
16.2.3	调试	464
16.2.4	直接使用 TlbImp	464
16.2.5	后期绑定	466
16.3	ActiveX 控件	471
16.3.1	传统 ActiveX 控件	471
16.3.2	另一个 .NET 应用程序	473
16.3.3	再次调试	474
16.4	在 COM 中使用 .NET 组件	475
16.4.1	.NET 组件	475
16.4.2	RegAsm	477
16.4.3	TlbExp	480
16.5	小结	480
第 17 章	组件服务	481
17.1	事务处理	481
17.2	事务处理组件	483
17.3	事务处理的其他方面	497
17.3.1	人工事务处理	497
17.3.2	Just In Time	498
17.3.3	对象池	498
17.3.4	其他情况	498



17.4	排队组件	499
17.4.1	排队组件的范例	499
17.4.2	使用排队组件进行事务处理	504
17.5	小结	506
第 18 章	用 ADO.NET 访问数据	507
18.1	ADO.NET 体系结构	508
18.1.1	ADO.NET 组件	508
18.1.2	ADO 和 ADO.NET 之间的区别	510
18.2	.NET 数据提供者	513
18.2.1	Connection 对象	513
18.2.2	Command 对象	514
18.2.3	DataReader 对象	515
18.2.4	DataAdapter 对象	517
18.2.5	SQL Server .NET 数据提供者	520
18.2.6	OLE DB .NET 数据提供者	520
18.3	DataSet 组件	521
18.3.1	DataTableCollection	521
18.3.2	DataRelationCollection	521
18.3.3	ExtendedProperties	521
18.3.4	创建和使用 DataSet 对象	522
18.3.5	强类型化的 DataSet 对象	527
18.3.6	DataSet 对象和 XML	527
18.3.7	ADO.NET DataTable 对象	530
18.4	ADO.NET 中的连接池	531
18.5	创建 Data Access 组件	532
18.5.1	构造函数	533
18.5.2	属性	534
18.5.3	存储过程 XML 结构	535
18.5.4	方法	535
18.6	使用 DataSet 对象绑定 DataGrids	547
18.7	小结	549
第 19 章	在 VB.NET 中使用 XML	550
19.1	XML 简介	551
19.2	XML 串行化	552
19.3	System.Xml 文档支持	558
19.4	XML 流类型分析程序	558
19.4.1	写入 XML 流	559