



Paul Kimmel 著 崔斌 齐明霞 译

Visual Basic .Net Power Coding 中文版



适用于
Visual Basic
.NET 2003



华中科技大学出版社
<http://press.hust.edu.cn>

Visual Basic .NET Power Coding 中文版

适用于 Visual Basic .NET 2003

Paul Kimmel 著

崔斌 齐明霞 译

Visual Basic. NET Power Coding 中文版

Visual Basic. NET Power Coding

Paul Kimmel

Copyright 2002 Addison Wesley Longman, Inc

Simplified Chinese Copyright 2003 of the first publication by Huazhong Science and Technology University Press and Pearson Education North Asia Limited

All rights Reserved.

Published by arrangement with Pearson Education North Asia Limited, a Pearson Education Company.

版权所有,翻印必究。

本书封面贴有华中科技大学出版社激光防伪标签,无标签者不得销售。

图书在版编目(CIP)数据

Visual Basic. NET Power Coding 中文版/(美)Paul Kimmel 著;崔斌 齐明霞 译
武汉:华中科技大学出版社,2004年10月
ISBN 7-5609-3278-9

I. V...

I. ①P... ②崔... ③齐...

II. Basic 语言-程序设计

IV. TP312

责任编辑:彭保林 曾光

封面设计:潘群

责任校对:吴晗

责任印制:熊庆玉

出版发行:华中科技大学出版社 (武昌喻家山 邮编:430074 电话:87557437)

<http://press.hust.edu.cn>

录排:北京搜获科技有限公司

印刷:湖北新华印务有限公司

开本:787×1092 1/16

印张:33.25 插页:2 字数:685 000

版次:2004年10月第1版

印次:2004年10月第1次印刷

ISBN 7-5609-3278-9/TP·541

定价:60.00元

(本书若有印装质量问题,请向出版社发行部调换)

前 言

A journey of a thousand miles begins with a single step.^①

引 言

大家读到这本书时恰好是笔者研究.NET 的第 3 个年头。在看了一些有关.NET 的书籍后,笔者试图了解读者接触.NET 时会对哪些内容感兴趣。因此,在构思和编写这本书的过程中,笔者费了一些心思。众所周知,.NET 已经诞生了一段时间,相信有不少读者正准备对 VB .NET 作更为详细、深入的研究。

本书读者对象

本书的读者对象是有扎实基础并希望有突破性提高的专业程序员。在阅读本书前,笔者假设您已经读过 VB .NET 的入门性书籍,并涉猎过一些更深层次的书籍,比如“Visual Basic .NET Unleashed”(Kimmel 2002b),而且现在还准备做更深入的学习。

本书基础性、介绍性的内容相对较少,如果您还不了解怎样编写循环语句、条件语句以及有关函数、子过程的编写规则,那么请您先放下这本书,回过头去看一看那些入门性书籍。等您确定不再需要了解上述那些基础性内容之后,那么再回到这本书上来。

如果您是有一定项目经验的编程人员,本书对您会非常有用,请继续阅读本书。

本书内容

第 1 章为初学者章节。从第 2 章开始,一直到第 18 章,主要介绍 VB .NET 的某些高级主题,这些内容可以帮助读者解决一些具有挑战性的难题,而这些难题的解决方案在帮助文档或者其他书籍中是不易找到的。

本书前几章重点介绍继承和委托等主题,因为即使是有经验的 VB 程序员,也可能在这些方面感到迷茫。毕竟继承和委托在 VB 6 中都不存在,并且委托是.NET 所独有的内

① 千里之行,始于足下。

容。

反射

第4章将深入探讨反射。如果您听说过运行时类型信息(Run Time Type Information, RTTI),那么您可以把反射看成是增强后的 RTTI。到目前为止,反射的全部作用还没有被完全挖掘出来,而此前已被挖掘和发现出来的那些已令人感到吃惊。

例如,.NET 的代码在被实时编译和运行之前会被转换为中间语言(Intermediate Language, IL)代码;而且.NET 在一定程度上模仿了 Java 的字节码模型。VB .NET 支持运行时生成新的类型,并直接将其转换成中间语言,随后实时创建这种类型的实例。本书会向您介绍如何利用反射生成中间语言,并且提供了在应用部署后扩展代码的最基本的方法。

程序集

VB.NET 中通过将元数据添加到程序集的方式避免了“DLL hell”的情况。目前认为,程序集是指带有相关扩展信息的应用程序,它极大地降低了使用注册表和全球性唯一标识(GUID)的必要性。

另一个很酷的技术是通过 HTTP 连接动态加载程序集的能力,这意味着通过模拟瘦客户端浏览器模型能够实现可自动部署和更新的 Windows 程序。

与基于 Web Forms 的应用程序相比,基于 Windows Forms 的应用程序提供了更丰富的客户端实现技术,瘦客户端程序设计中程序集的采用将实现 Web 和 Windows 两种开发技术的融合。

本书介绍了如何使用程序集元数据,以及如何实现瘦客户端 Windows 应用程序,这种应用程序能够通过 Web 进行部署并在无人干预的情况下自动更新。

多线程

很多时候我们必须使用多线程提供的功能。本书将介绍怎样使用同步进程、异步进程、线程池以及线程类来执行安全的协作式多线程任务,以及如何利用 Windows Form 控件实现。

VB .NET 中的多线程就像是军火库里所需的巡航导弹,要学会安全、专业化地使用线程。

COM Interop

COM 领域有非常庞大的代码。Microsoft 一直还没有放弃 COM, 您又何必要放弃它呢? 微软甚至在 Visual Studio .NET(VS.NET)中都使用了 COM, 这部分内容可以参考插件管理器(Add-Ins Manager)。

COM Interop 允许在 .NET 中使用 COM 组件, 也可以在基于 COM 的应用程序中使用 .NET 代码。通过本书, 将能学到如何在 VB.NET 中使用 COM Interop。

Remoting (远程处理)

在朝着开放标准迈进的过程中, Microsoft 提出一些新的办法来解决现存的问题。Remoting 支持分布式环境下解决方案的管理。第 8 章将介绍如何在分布式项目中串行化对象以实现远程技术。

创建组件

以前, 为 Visual Basic 创建高级组件时往往要用到 ActiveX 和 C++ 编译器。而 VB .NET 自身就可以创建针对 VB .NET 的专业组件。

通过第 9 章的学习, 您可以了解创建用户控件、自定义控件和服务器控件的方法。书中还提供了几个例子来演示如何创建和测试控件, 以及将控件添加到 VS .NET 中的具体细节。

ADO.NET

.NET 的一项革新性的成果是 ADO.NET 技术, .NET 遵循了 Web 应用程序所必需的断开连接的数据模型。ADO.NET 的断开连接的特征由 XML DataSet 支持, 而不是原来的 Recordset。

DataSet 基于 XML, 它要求您重新考虑构建数据库、客户端/服务器以及 Web 应用程序的方式。本书介绍了如何使用 Dataset 和处理断开连接的数据, 以及如何使用 XML 和 XML Schema (XML 模式, XSD)来连接到任何位置的任意类型的数据。

Web 服务

Web 服务是最令人兴奋的新技术之一。Web 服务提供了从世界上任何一个角落都可以调用的代码。Web 服务使用的是开放标准协议(例如 SOAP 和 XML), 它允许任何联网

的计算机向其他计算机请求服务。

在阅读本书相关章节的时候，您会了解到有关 XML、SOAP、Web 服务和 UDDI 的知识。本书将提供大量实例来描述这些技术(以及缩写词，例如 UDDI)。

ASP.NET, 调试和安全

令人兴奋的新事物还包括 ASP.NET。ASP.NET 模型使创建 Web 应用程序的方法与创建 Windows 应用程序的方法非常相似。

在编写本书的时候，笔者正好在使用 ASP.NET 来策划一个企业级解决方案。在策划的过程中，笔者学会了如何最好地实现、调试以及确保 Web 应用程序的安全。所有这些心得和经验笔者都写入了这本书里面。

此外，用户还会学习到关于 Web Forms 和服务器控件的内容，以及怎样用缓存和 XML 串行化来管理状态、如何将 Web 应用程序连接到数据库、还有如何使用 .NET 中的策略管理和新的安全特性(security attribute)等这些技术。

读完本书之后，您将会认可这样一个观点：ASP.NET, 调试和安全，是 VB .NET 的重要组成部分。

获得源代码的地方

您可以从 <http://www.softconcepts.com> 下载所有的源代码。笔者就像一个为孩子制作新鞋的鞋匠，一有时间笔者就会来更新这个站点，偶尔内容还会来来回回地变化。请根据主页上的源代码链接来寻找本书的源代码。如果您有什么疑问或者信息反馈，请给笔者发 E-mail，笔者的邮件地址是：pkimmel@softconcepts.com。

计 划

从第 1 章开始，笔者就希望本书的每一页都具有可读性。笔者注意到大部分读者都很忙，可能没有时间坐下来一次读数百页的内容。

这本书里面有许多新的理论，同时也有很多代码，这些代码有助于您理解那些理论。本书的每一章都只有一个主题，这一章的所有内容都将围绕着这个主题来开展和组织。所以如果您要寻找一些具体问题的答案，您可以在同一个地方找到所有的资料。

笔者为所写的内容负责，笔者会尽力去提供最精确和最有用的信息。不管您有什么疑问，您都可以随时给笔者发邮件，笔者的邮件地址是：pkimmel@softconcepts.com。

祝您阅读愉快！

关于作者

Paul Kimmel 是 Software Conceptions 公司的创始人。Paul 从事于开发面向对象软件的时间已经超过了十年。他编写了很多关于面向对象程序设计和 .NET 方面的书籍，包括为 McGraw-Hill/Osborne 编写的“Advanced C# Programming”和 Sams 的“Visual Basic .NET Unleashed”。他是“Windows Developer”杂志每月的专栏作家，codeguru.com 的栏目“Visual Basic Today”每双月的撰稿人，也是 InformIT 的长期撰稿人。

在编写本书的同时，Paul 在俄勒冈州的波特兰帮助构建一个企业级 ASP.NET 应用。在北美的任何地方他都可以帮忙设计和实现应用软件，可以通过 pkimmel@softconcepts.com 与之联系。

Paul 同他的妻子 Lori、还有孩子们——Trevor、Doug、Alex 和 Noah——一起居住在密歇根州的 Okemos。Okemos 是一个很安静的小社区，社区靠近美丽的密歇根州立大学校园，这里的居民都平易近人、通情达理。

关于技术审校人员

John P. Cottrell 在 1982 年开始利用 Basic 进行程序设计，并且从那以后就用各种语言编写了很多应用软件。在 1996 年开始用 Visual Basic 4 进行程序设计，并且在 2000 年 6 月开始使用 VB.NET。John 生活在乔治亚州的 Sugar Hill，空余时他喜欢编写程序、搜集硬币以及和他的妻子 Adelle、三岁的女儿 Katy 一起消遣娱乐。

Lowell Mauer 作为程序员从事数据处理工作已经超过了 23 年，他是一名教师，同时也是一名顾问。他在纽约的 Brooklyn 大学和新泽西的 Montclair 州立大学教授过程序设计。他开发并且销售出多个用 Visual Basic 编写的软件，其中包括一个基于 SQL Server 数据库的系统，该系统用于处理私人高尔夫课程的预约。Lowell 目前在纽约担任高级顾问。

目 录

第 I 部分 高级语言基础

第 1 章 基本语言构造(2)	2.2.8 访问修饰符.....(31)
1.1 引言.....(2)	2.3 继承与聚集.....(32)
1.2 变量声明.....(2)	2.3.1 Adapter 模式.....(33)
1.3 值类型和引用类型.....(3)	2.3.2 创建一个 UserControl 形状.....(33)
1.3.1 结构.....(4)	2.4 定义接口.....(35)
1.3.2 类.....(5)	2.4.1 基本原则.....(35)
1.3.3 值类型、引用类型和内存.....(6)	2.4.2 定义一个接口.....(36)
1.3.4 装箱和拆箱.....(8)	2.5 实现接口.....(37)
1.4 定义结构和类.....(9)	2.5.1 添加接口属性.....(39)
1.4.1 添加字段.....(9)	2.5.2 添加接口事件.....(40)
1.4.2 添加属性.....(10)	2.6 继承接口.....(41)
1.4.3 添加方法.....(11)	2.7 多接口继承.....(43)
1.4.4 使用事件.....(15)	2.8 比较抽象类和接口.....(44)
1.4.5 使用访问修饰符.....(17)	2.9 小结.....(45)
1.5 理解面向对象思想.....(18)	第 3 章 委托(46)
1.5.1 对象、类和元类.....(18)	3.1 引言.....(46)
1.5.2 信息隐藏和封装.....(19)	3.2 实现事件处理程序.....(46)
1.5.3 关联、聚集和继承.....(19)	3.2.1 使用窗体设计器.....(47)
1.5.4 多态.....(19)	3.2.2 使用代码编辑器.....(49)
1.6 中间语言.....(20)	3.2.3 Handles 子句.....(50)
1.7 小结.....(21)	3.2.4 EventHandler 类.....(51)
第 2 章 继承和接口(22)	3.2.5 sender 参数.....(51)
2.1 引言.....(22)	3.2.6 用单一处理程序处理多个事件.....(52)
2.2 继承类.....(22)	3.2.7 实现多响应.....(53)
2.2.1 继承术语.....(22)	3.3 使用 WithEvents 语句.....(53)
2.2.2 Inherits 语句.....(23)	3.4 添加和删除事件处理程序.....(54)
2.2.3 NotInheritable 修饰符.....(24)	3.4.1 AddHandler 语句.....(54)
2.2.4 抽象类.....(24)	3.4.2 RemoveHandler 语句.....(55)
2.2.5 重写方法和属性.....(26)	3.4.3 在运行时指派事件处理程序.....(55)
2.2.6 替代方法和属性.....(29)	3.5 在类、结构和接口中声明事件.....(59)
2.2.7 重载 Sub New 构造函数.....(30)	3.6 委托的概念.....(61)
	3.6.1 为什么委托如此重要.....(61)

3.6.2 定义委托.....(61)	5.5 创建自定义特性.....(104)
3.7 研究现有的委托类型.....(62)	5.5.1 特性使用说明.....(105)
3.8 适用于多线程的委托.....(63)	5.5.2 继承特性类.....(106)
3.9 小结.....(63)	5.5.3 定义位置参数.....(106)
第4章 反射.....(64)	5.5.4 定义命名参数.....(106)
4.1 引言.....(64)	5.6 反射特性.....(107)
4.2 隐式后期绑定.....(64)	5.7 发放特性到 IL.....(112)
4.3 运行时发现类型信息.....(66)	5.8 使用 CodeDom 类发放特性.....(115)
4.4 加载程序集.....(67)	5.9 特性和声明安全.....(118)
4.4.1 加载程序集.....(67)	5.10 小结.....(118)
4.4.2 调用被反射的方法.....(68)	第6章 多线程.....(120)
4.4.3 从 Type 对象创建实例.....(69)	6.1 引言.....(120)
4.5 讨论 Binder 类.....(72)	6.2 初识 Timer 控件.....(120)
4.6 使用 DefaultMemberAttribute.....(73)	6.3 比较同步和异步行为.....(121)
4.7 反射成员.....(74)	6.3.1 编写同步行为.....(122)
4.7.1 反射方法.....(74)	6.3.2 编写异步行为.....(123)
4.7.2 反射参数.....(75)	6.3.3 完成异步调用.....(124)
4.7.3 反射任何成员.....(76)	6.4 .NET Framework 的异步处理.....(126)
4.7.4 反射属性.....(76)	6.4.1 异步调用 XML Web 服务.....(126)
4.7.5 反射字段.....(77)	6.4.2 异步调用文件操作.....(128)
4.7.6 反射事件.....(80)	6.5 用线程进行程序设计.....(129)
4.8 反射的实际应用.....(81)	6.5.1 使用 ThreadPool 类进行 多线程设计.....(129)
4.9 反射自定义特性.....(83)	6.5.2 使用 Thread 类进行 多线程设计.....(134)
4.10 理解反射和安全.....(84)	6.6 Windows Forms 中的多线程.....(136)
4.11 在运行时发放 IL 代码.....(84)	6.7 小结.....(139)
4.11.1 快速查看正则表达式.....(86)	第 II 部分 解决方案建立
4.11.2 已编译正则表达式.....(86)	第7章 COM 互操作.....(142)
4.11.3 动态发放类型.....(87)	7.1 引言.....(142)
4.11.4 动态发放已编译正则表达式.....(90)	7.2 从.NET 代码中调用 COM.....(142)
4.12 小结.....(91)	7.2.1 创建一个 COM 对象的测试程序.....(143)
第5章 特性.....(92)	7.2.2 将 COM 二进制 文件导入.NET.....(145)
5.1 引言.....(92)	7.2.3 使用早期绑定的 COM 对象.....(146)
5.2 应用特性.....(92)	7.2.4 垃圾收集器.....(149)
5.2.1 研究特性约定.....(93)	
5.2.2 将特性应用到实体.....(94)	
5.3 使用程序集特性.....(97)	
5.4 使用程序集特性创建一个 【关于】对话框.....(98)	

7.2.5 使用后期绑定的 COM 对象.....(149)	8.7.2 异步 Remoting.....(198)
7.2.6 在.NET 中使用 COM 事件.....(150)	8.7.3 Remoting 安全问题.....(198)
7.3 从 COM 调用.NET 代码.....(152)	8.7.4 为服务器选择一个宿主.....(198)
7.3.1 创建一个测试.NET 类库.....(153)	8.8 小结.....(199)
7.3.2 将.NET 类型发布给 COM.....(154)	第 9 章 创建自定义组件.....(200)
7.3.3 在 VB6 中调用.NET 程序集.....(155)	9.1 引言.....(200)
7.3.4 将.NET 委托发布给 COM.....(156)	9.2 实现自定义组件.....(200)
7.3.5 应用 Interop 特性.....(159)	9.3 实现自定义 Windows 控件.....(203)
7.3.6 让导出到 COM 变得简单.....(160)	9.3.1 定义正则表达式
7.4 理解 COM Interop 中的错误处理.....(161)	TextBox 控件.....(203)
7.5 将 ActiveX 控件导入.NET.....(162)	9.3.2 测试自定义组件.....(205)
7.6 调试可协作的组件.....(163)	9.4 将控件添加到工具箱.....(206)
7.6.1 使用.NET 宿主调试 COM 库.....(163)	9.4.1 将位图与控件关联.....(208)
7.6.2 使用基于 COM 的宿主	9.4.2 将文档添加到工具箱.....(209)
调试.NET 库.....(164)	9.5 实现自定义 Windows 用户控件.....(210)
7.7 附加主题.....(164)	9.5.1 表面化成员属性.....(211)
7.8 小结.....(165)	9.5.2 将数据绑定到自定义用户控件.....(213)
第 8 章 Remoting.....(166)	9.6 测试控件特性.....(215)
8.1 引言.....(166)	9.6.1 应用 EditorBrowsable
8.2 理解.NET Remoting.....(166)	Attribute.....(215)
8.3 按引用编组对象.....(168)	9.6.2 应用 DesignerSerialization
8.3.1 Hello, Remote World!.....(168)	VisibilityAttribute.....(216)
8.3.2 让客户和服务器对话.....(172)	9.6.3 应用BrowsableAttribute.....(217)
8.3.3 使用服务器激活对象.....(174)	9.7 使用 UTypeEditor 类.....(218)
8.3.4 使用客户激活对象.....(176)	9.7.1 将类型编辑器与控件关联.....(221)
8.4 按值编组对象.....(180)	9.7.2 应用默认值.....(222)
8.4.1 使用按值传递类.....(180)	9.7.3 将控件属性分类.....(222)
8.4.2 使用值传递对象	9.8 实现类型转换.....(223)
修改客户代码.....(183)	9.8.1 实现 IConvertible 接口.....(223)
8.4.3 实现 ISerializable.....(184)	9.8.2 实现类型转换器.....(229)
8.4.4 比较按引用对象和	9.8.3 将类型转换器与可转换
按值传递对象.....(188)	类型相关联.....(233)
8.5 写入事件日志.....(188)	9.8.4 使用 IConvertible 行为.....(233)
8.6 处理远程事件.....(189)	9.8.5 使用类型转换行为.....(234)
8.6.1 理解远程事件行为.....(189)	9.9 实现扩展的提供者.....(237)
8.6.2 调用远程事件.....(189)	9.10 创建 Windows 控件设计器.....(240)
8.7 其他 Remoting 主题.....(197)	9.10.1 将设计器应用到控件.....(242)
8.7.1 管理远程对象的生存期.....(197)	9.10.2 测试控件设计器.....(243)

9.11 使用默认属性.....(243)	11.6 使用 DataTable 和 DataView 类.....(273)
9.12 实现自定义 Web 控件和自定义 Web 用户控件.....(244)	11.7 定义数据库关系.....(274)
9.13 小结.....(245)	11.8 使用 Command 对象.....(276)
第 10 章 .NET 中自动更新	11.9 使用 Command Builder 生成 SQL 语句.....(277)
的智能客户(246)	11.10 更新 DataSet.....(278)
10.1 引言.....(246)	11.11 给 DataSet 添加数据.....(280)
10.2 实现“Hello, World!” 瘦客户.....(247)	11.12 对 DataSet 排序和筛选.....(281)
10.3 智能客户和服务器的预先配置.....(250)	11.12.1 对 DataView 筛选和排序.....(281)
10.3.1 连接到 Internet.....(250)	11.12.2 通过 DataTable 对列 过滤和排序.....(283)
10.3.2 安装 .NET Framework.....(250)	11.13 小结.....(284)
10.3.3 添加受信任站点清单内容.....(251)	第 12 章 高级 ADO.NET(285)
10.3.4 为应用程序程序 集调整安全策略.....(252)	12.1 引言.....(285)
10.3.5 针对智能客户程序集进行 服务器预先配置.....(253)	12.2 更新 DataView.....(285)
10.4 考虑通用应用程序加载器.....(254)	12.3 利用存储过程编程.....(291)
10.5 创建 Microsoft 安装文件 管理安全策略.....(255)	12.3.1 调用存储过程.....(292)
10.5.1 定义新的代码组.....(255)	12.3.2 定义 SQL Server 连接字符串.....(294)
10.5.2 以编程的方式管理 代码组和权限.....(257)	12.3.3 使用输入参数.....(294)
10.5.3 使用安装程序管理权限.....(260)	12.3.4 使用输出参数.....(296)
10.6 处理 COM 组件.....(263)	12.4 在 Visual Studio .NET 中调试存储过程.....(304)
10.7 其他相关概念.....(263)	12.4.1 配置 MSDE 以进行 存储过程调试.....(304)
10.8 小结.....(264)	12.4.2 配置 SQL Server 进行 存储过程调试.....(306)
第 11 章 ADO.NET 数据库编程(265)	12.5 使用事务.....(306)
11.1 引言.....(265)	12.6 创建类型化 DataSet.....(309)
11.2 ADO.NET 基础.....(266)	12.6.1 定义 XML 架构.....(310)
11.3 定义一个数据库连接.....(267)	12.6.2 生成类型化 DataSet.....(312)
11.3.1 定义一个连接字符串.....(267)	12.6.3 使用类型化 DataSet 对象编程.....(314)
11.3.2 存储连接字符串.....(268)	12.7 串行化 DataSet.....(315)
11.3.3 读取连接字符串.....(270)	12.8 使用 ADO.NET 接口编程.....(317)
11.3.4 测试连接.....(270)	12.9 小结.....(318)
11.4 使用 Adapter 填充 DataSet 对象.....(271)	第 III 部分 Web 编程
11.5 使用 DataReader 类.....(272)	第 13 章 创建 Web 服务(320)

13.1 引言	(320)	14.4 编写使用 DataSet	
13.2 查找 Web 服务	(320)	对象的 Web 服务	(352)
13.3 使用现有的 Web 服务	(321)	14.4.1 将对象序列化为	
13.3.1 为项目添加 Web 引用	(322)	XML DataSet	(353)
13.3.2 声明 Web 服务类的实例	(323)	14.4.2 了解返回 DataSet 的	
13.3.3 调用 Web 方法	(323)	Web 方法的代理类	(354)
13.3.4 浏览 Web 服务代理类	(324)	14.5 将代理类修改为返回胖对象	(357)
13.4 创建 Web 服务应用程序	(326)	14.6 返回强类型集合	(357)
13.4.1 应用 WebService 特性	(328)	14.6.1 回顾强类型集合	(358)
13.4.2 编写 Web 方法	(328)	14.6.2 序列化强类型集合	(361)
13.5 调试和测试 Web 服务	(328)	14.6.3 从 Web 服务返回集合	(363)
13.5.1 使用集成的调试器测试		14.7 异步调用 Web 服务	(364)
Web 服务	(328)	14.8 小结	(366)
13.5.2 不使用集成的调试器			
测试 Web 服务	(330)		
13.5.3 使用【生成和浏览】菜单测试			
Web 服务	(330)		
13.6 部署 Web 服务	(330)	第 15 章 构建 ASP.NET Web	
13.6.1 创建 Web 应用程序目录	(331)	应用程序	(367)
13.6.2 复制基本文件	(332)	15.1 引言	(367)
13.6.3 创建 .disco 文件	(332)	15.2 设计屏幕布局	(367)
13.6.4 添加 Web.config 文件	(333)	15.2.1 管理具有表的控件位置	(368)
13.7 理解 XML Web 服务和安全	(333)	15.2.2 使用级联样式表确保一致性	(369)
13.7.1 浏览默认的 Web.config 文件	(334)	15.2.3 使用文档大纲	(371)
13.7.2 在文件或目录级别上使用		15.2.4 在外部样式表中编写样式块	(371)
Windows 身份验证	(336)	15.2.5 可视化生成外部样式表	(373)
13.8 小结	(340)	15.2.6 理解样式的级联应用程序	(373)
第 14 章 高级 Web 服务	(341)	15.2.7 将外部样式表链接	
14.1 引言	(341)	到一个页面	(374)
14.2 从 Web 服务返回简单的数据	(341)	15.2.8 将样式应用到控件	(374)
14.2.1 实现埃拉托色尼筛选	(342)	15.2.9 以编程的方式修改属性	(374)
14.2.2 实现 IsPrime Web 服务	(343)	15.3 使用用户控件创建呈现	(375)
14.3 从 Web 服务返回复杂的数据	(345)	15.3.1 创建基本的用户控件布局	(376)
14.3.1 实现一个复杂类型	(345)	15.3.2 创建导航链接	(377)
14.3.2 实现 Web 服务	(348)	15.3.3 将用户控件添加到页面	(381)
14.3.3 引用 Web 服务	(349)	15.3.4 动态加载用户控件	(381)
14.3.4 浏览代理类	(349)	15.3.5 将 Web 页面转换	
14.3.5 实现 Web 服务使用者	(350)	为用户控件	(383)
		15.4 处理应用程序级事件	(385)
		15.5 缓存对象	(387)
		15.5.1 使用 HttpSessionState	
		类	(387)

15.5.2 使用 HttpSessionState 类.....(397)

15.5.3 使用 Cache 类.....(401)

15.5.4 使用 cookie.....(402)

15.5.5 启用和使用数据视图.....(402)

15.5.6 缓存 Web 页面.....(403)

15.5.7 缓存部分 Web 页面.....(404)

15.6 使用具有 XML 的动态接口.....(404)

15.6.1 使用 XML 数据.....(404)

15.6.2 实现一个 XSL 文档.....(405)

15.6.3 为 XSL 格式的 XML 文档
定义一个缓存依赖项.....(407)

15.7 保护具有窗体身份验证的
Web 应用程序.....(408)

15.8 小结.....(410)

第 16 章 组合使用 ADO.NET

和 ASP.NET.....(411)

16.1 引言.....(411)

16.2 连接到数据库.....(411)

16.2.1 在 Web.config 文件
中定义连接字符串.....(411)

16.2.2 读取配置设置.....(412)

16.2.3 通过使用接口
定义 Database 类.....(413)

16.2.4 编写测试代码.....(414)

16.2.5 使用 ConditionalAttribute.....(415)

16.3 使用 DataView 类.....(416)

16.4 将数据绑定到单值 Web 控件.....(419)

16.4.1 使用 DataBindings 对话框.....(420)

16.4.2 在 HTML 编辑器中定义
一个数据绑定.....(424)

16.5 将数据绑定到多值 Web 控件.....(425)

16.6 使用 DataGrid 控件分页和分类.....(427)

16.6.1 将 DataGrid 控件用于分页.....(427)

16.6.2 将 DataGrid 控件用来排序.....(428)

16.7 使用 DataList 控件
来重复复合控件.....(433)

16.7.1 定义复合用户控件.....(433)

16.7.2 定义具有 DataList 控件的
Web 页面.....(435)

16.8 将绑定列转换为模板列.....(438)

16.9 管理到服务器的往返.....(439)

16.10 小结.....(440)

第 IV 部分 调试与管理

第 17 章 调试.NET.....(442)

17.1 引言.....(442)

17.2 显示【调试】窗口.....(442)

17.2.1 使用【局部变量】和
【自动窗口】.....(453)

17.2.2 使用【快速监视】窗口.....(454)

17.2.3 使用【监视】窗口.....(455)

17.2.4 使用【内存】窗口.....(456)

17.2.5 使用【调用堆栈】窗口.....(456)

17.2.6 使用【反汇编】窗口.....(456)

17.2.7 使用【立即】/【命令】
模式窗口.....(457)

17.2.8 使用【输出】窗口.....(458)

17.3 管理断点.....(458)

17.4 使用编辑并继续行为.....(459)

17.5 调试、断言与跟踪.....(460)

17.5.1 使用 Debug 类.....(460)

17.5.2 使用 Debug.Assert 方法.....(461)

17.5.3 使用 Trace 类.....(462)

17.6 使用跟踪侦听器编程.....(463)

17.6.1 创建 FileStream 跟踪侦听器.....(464)

17.6.2 把跟踪信息记入事件日志.....(465)

17.6.3 创建自定义跟踪侦听器.....(466)

17.7 使用 Boolean 开关管理调试代码.....(468)

17.7.1 在应用程序配置文件
中定义一个开关.....(469)

17.7.2 在应用程序中使用
Boolean 开关.....(469)

17.8 把应用程序事件记入日志.....(470)

17.8.1 发送信息到事件日志.....(471)

17.8.2 定义一个自定义事件日志.....(472)

17.8.3 发送调试和跟踪信息到 自定义事件日志..... (472)	18.6.1 什么是 Demand? (495)
17.9 使用性能计数器 (473)	18.6.2 要求打印权限..... (496)
17.10 使用 Process 类..... (475)	18.6.3 要求读取注册表的权限..... (497)
17.11 附加到一个运行的进程 (476)	18.7 使用代码访问安全性的断言..... (497)
17.12 调试 Windows 应用程序 (477)	18.7.1 把程序装入沙箱 以进行测试..... (498)
17.13 调试 Web 应用程序 (477)	18.7.2 示范 Assert 动作的行为..... (499)
17.14 调试多语言程序 (478)	18.8 其他安全性动作简介 (502)
17.15 补充话题..... (481)	18.9 总则..... (503)
17.15.1 查看自动化调试对象模型..... (481)	18.10 小结..... (504)
17.15.2 查看可用的调试器..... (481)	
17.16 小结..... (482)	
第 18 章 代码访问安全性..... (483)	附录 A 迁移 Visual Basic 6 应用 程序到 Visual Basic.NET (505)
18.1 引言..... (483)	A.1 引言..... (505)
18.2 代码访问安全性的含义 (484)	A.2 在迁移前 (505)
18.3 防御式编程..... (485)	A.2.1 是否进行迁移..... (506)
18.4 管理安全策略..... (486)	A.2.2 您想进行迁移..... (506)
18.4.1 安全策略的级别..... (486)	A.3 .NET 中不支持的 Visual Basic 6 特征..... (507)
18.4.2 修改安全策略..... (487)	A.4 迁移 Visual Basic 6 Windows 应用程序..... (508)
18.4.3 许可运行下载的程序集..... (490)	A.4.1 迁移示例客户应用程序..... (508)
18.5 声明性安全性与强制性 安全性的比较..... (491)	A.4.2 解决迁移错误..... (510)
18.5.1 利用声明性安全性..... (494)	A.5 迁移 Visual Basic 6 ASP Web 应用程序..... (512)
18.5.2 利用强制性安全性..... (495)	A.6 小结..... (513)
18.6 使用代码访问安全性的要求 (495)	

第 I 部分 高级语言基础

第 1 章 基本语言构造

第 2 章 继承和接口

第 3 章 委托

第 4 章 反射

第 5 章 特性

第 6 章 多线程

第 1 章 基本语言构造

Willst du immer weiter schweifen?

Sieh, das Güte liegt so nah.^①

—Goethe

1.1 引 言

阅读本章可以快速了解 Visual Basic .NET(VB .NET)的工作原理。这些内容有助于理解一些重要的基础知识。.NET 有自己的一些独特特征,即使曾用 Visual Basic 6(VB6)以及其他语言编写过程序,但您也可能没有碰到过这些特征。

.NET 中的类型都是 Object 类的子类,但是,有些类型具有类中基本类型的优秀特征。本章将介绍值类型和引用类型之间的差异,也回顾了新型面向对象语言的特征(类),这是必要的基础知识。还讨论了非确定性的对象清理和垃圾收集的方法,以及如何查看 VB .NET 编译器生成的中间语言(Intermediate Language, IL)代码等内容。

理解 VB .NET 的上述特征有助于快速深入到更高级的概念。

1.2 变 量 声 明

VB6 中声明变量使用 dim 语句,而 VB .NET 继续支持 dim 关键字。它们的区别在于,VB .NET 允许并提倡在同一行代码中定义变量、创建实例并提供初始值。

声明并创建一个实例称为“实例化”,下面是几个声明变量和实例化对象的示例。

```
Dim I As Integer = 5
Dim S As String = "Welcome to Valhalla Tower Material Defender!"
Dim ADate As DateTime = DateTime.Now
Dim Objects As New Object() {1, 2, DateTime.Now, "Some Text"}
Dim Log As EventLog = New EventLog()
```

第 1 条语句声明了一个初始值为 5 的整型变量 I; 第 2 条语句声明一个字符串变量并将其初始化为“Welcome to Valhalla Tower Material Defender!”; 第 3 条语句声明了一个 DateTime 结构,并用当前的日期和时间对其进行初始化。第 4 条语句较复杂,它声明了

① 还在因为找不到方向继续犹豫不前吗?看,美好的一切就在不远处。