Microsoft®

.NET Framework 2.0 程序设计

微软公司 著

人民邮电出版社
POSTS & TELECOM PRESS

CD-ROM

.NET Framework 2.0
程序设计

微软公司　著

人民邮电出版社
北京

图书在版编目（CIP）数据

.NET Framework 2.0程序设计 / 微软公司著. -- 北京 : 人民邮电出版社, 2010.1（2014.6 重印）
ISBN 978-7-115-22191-9

Ⅰ. ①N… Ⅱ. ①微… Ⅲ. ①计算机网络－程序设计 Ⅳ. ①TP393.09

中国版本图书馆CIP数据核字(2010)第005033号

内 容 提 要

目前软件开发行业中应用最广泛的平台之一就 Microsoft .NET Framework。.NET Framework 为各种应用开发提供了丰富的类库资源。开发者使用这些类库，可以快速地开发应用程序。本课程的目标就在于培养能熟练使用.NET Framework 类库的程序设计人员。他们可以使用 VB.NET 或者 C#开发基于.NET Framework 的应用程序。

全书共分为十七章。分别从各个方面向学生介绍了.NET Framework 的各个部分和技术要点。同时介绍了如何使用 VB.NET 和 C#进行开发和实现。第一章简要介绍了 Microsoft .NET Framework。第二章介绍了公共语言运行库和公共类型系统。第三章介绍了.NET Framework 中有关托管代码的编译和执行的相关知识。第四章介绍了事件和委托的概念及其用法。从第五章开始，本书详细介绍了.NET Framwwork2.0 中的各种类库和它们的使用场合。

通过本课程的学习，学生可以掌握如何使用.NET Framework2.0 提供的功能设计应用程序。

.NET Framework 2.0 程序设计

◆ 著　　微软公司
责任编辑　刘　浩
◆ 人民邮电出版社出版发行　　北京市崇文区夕照寺街 14 号
邮编　100061　　电子函件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
北京鑫正大印刷有限公司印刷
◆ 开本：787×1092　1/16
印张：37.5　　2010 年 1 月第 1 版
字数：914 千字　　2014 年 6 月北京第 2 次印刷

ISBN 978-7-115-22191-9

定价：127.00 元（附光盘）

读者服务热线：(010)67132692　印装质量热线：(010)67129223
反盗版热线：(010)67171154

编 委 会

前　言

目前软件开发行业中应用最广泛的平台之一就 Microsoft .NET Framework。.NET Framework 为各种应用开发提供了丰富的类库资源。开发者使用这些类库，可以快速地开发应用程序。本课程的目标就在于培养能熟练使用.NET Framework 类库的程序设计人员。他们可以使用 VB.NET 或者 C#开发基于.NET Framework 的应用程序。

通过本课程的学习，学生可以掌握如何使用.NET Framework2.0 提供的功能设计应用程序：

具体技能包括：

- 能够理解托管执行环境的工作原理。
- 能够进行程序的部署与版本的有效控制。
- 能够使用字符串的各种不同类型的方法及.NET Framework 的集合和泛型。
- 了解委托，多路委托，事件的概念。
- 了解.NET 如何进行内存和资源的管理。
- 了解如何使用 GDI+进行图形操作。
- 能够操作流，并能使用 Reader 和 Writer 的方法进行对于文本的操作。
- 能够实现序列化。
- 了解线程和异步编程。
- 了解托管代码与非托管代码之间的互操作及在 COM 对象中调用.NET 对象的方法。
- 了解属性的概念并能定义自定义属性。
- 了解加密的原理并使用相应的类进行加密。
- 了解创建全球化应用程序的过程。
- 了解代码访问安全性的实现原理。
- 能够开发服务应用程序，并熟练使用邮件。

本书适用对象

本书主要面向面向希望熟练使用.NET Framework 类库的读者。

在学习本课程以前，需具备下列先修课程中的应用操作技能：

- 《Visual C# 2005 程序设计语言》或《Visual Basic 2005 程序设计语言》；
- 《数据库基础》
- 《数据访问技术——ADO.NET 2.0》
- 《Web 应用开发——ASP.NET 2.0》或《Windows 应用开发》

本书结构

本教材的设计策略是遵循由浅入深，循序渐进的策略进行设计的。了解本书的设计策略，对于使用本书教学会有所帮助。

全书共分为十章，第一章介绍了数据库的基础知识和 SQL 语言。这章是现代数据库技术的基础知识，为后续章节的学习打下基础。

第二章从数据访问模型入手，介绍了数据访问模型的基本概念以及主流数据访问模型的发展简史，并在此基础上重点阐述了 ADO.NET 2.0 对象模型及其特征和新特性。

第三章和第四章主要围绕着 ADO.NET 在连接环境中如何连接数据源，并从数据源中获取数据，介绍了相应的对象，方法和操作。

第五章到第七章主要围绕着 ADO.NET 中非常重要的断开连接的环境，介绍了 DataSet 和类型化 DataSet 的概念，并阐述了如何通过 DataAdapter 实现在连接模型和断开连接的模型之间的桥梁作用。

第八章介绍了如何对 DataSet 对象中的数据进行排序、搜索和筛选的各种对象和工具，同时引入了 DataView 对象来使得排序和筛选更加方便。

第九章则阐述了 ADO.NET 中的事务这个概念。第十章主要以 ADO.NET 与 XML 之间的关系为主，介绍了 ADO.NET 操作 XML 的方法。学完这章可以掌握使用 XML 文档和 ADO.NET 之间移动数据的方法，在实际运用中能够同时使用两种技术（ADO.NET 和 XML）编写数据库应用程序。

教学参考资料

获得微软授权使用该教材进行教学的教师，除了可以获得上述材料外，还可以从微软公司获得如下教学资料。

教参	章节教参与教学大纲
实验答案	实验的具体操作步骤
虚拟机	配置好的虚拟机实验环境，用于进行实验和课堂演示
实验室安装指南	包括在实验室部署虚拟机的指南
PowerPoint 和多媒体	用于进行课堂演示
案例	本书随附实用程序训练案例

目　录

第 1 章　.NET Framework 2.0 简介

作为一名 C#或者 VB.NET 程序员，你可能经常会在程序中引用别的类库文件，例如编写 Windows 窗体程序时，无论使用 C#还是 VB.NET，都需要添加对 System.Windows.Forms.dll 文件的引用。不仅如此，你还会发现无论是 VB.NET 或者 C#，它们虽然语法和结构不同，但是它们所调用的系统类库是一样的，例如在 C#中有 System.Collections.ArrayList 类，在 VB.NET 中也有 System.Collections.ArrayList 类，而且在这两种语言中这个类提供的功能完全相同。诸如以上种种事实让我们隐隐约约感觉到了一件事情，那就是在 VB.NET 和 C#之下、操作系统之上有一个共同的运行平台，为这些语言提供了统一的类库和功能调用。作为一名开发人员，你一直在使用这个平台，但是这个平台是什么呢？那就是.NET Framework。本章将带你了解.NET Framework 的基本结构、基本功能和工作方式。

在本章中可以学习以下内容。

- 了解什么是.NET Framework。
- 了解.NET Framework 的构成。
- 知道常见的.NET Framework 应用程序种类。

1.1　.NET Framework 概述

.NET Framework 是用于代码编译和执行的集成托管环境，换句话说，它管理着应用程序运行的方方面面，包括程序首次运行的编译、为程序分配内存以存储数据和指令、对应用程序授予或拒绝相应的权限、启动并管理应用程序执行，并且管理剩余内存的再分配。由于所有的.NET 应用程序都在.NET Framework 上执行，所以开发人员只需考虑如何与.NET Framework 打交道，而不必关心.NET Framework 底层的实现。.NET Framework 由公共语言运行库（Common Language Runtime，CLR）和.NET Framework 类库两个主要组件组成。

公共语言运行库可视为管理代码执行的环境。它介于操作系统和应用程序之间，提供了代码编译、内存分配、线程管理以及垃圾回收之类的核心服务。它还强制实施了严格的类型安全检查，并通过强制实施代码访问安全来确保代码在安全的环境中执行。

.NET Framework 提供了一整套很有用且可重用的类型，这套类型是面向对象的且完全可扩展的，它简化了.NET Framework 应用程序的开发。

.NET 应用程序运行在公共语言运行库之上，并且它还可以使用类库中它所需要的任何部分，如图 1-1 所示。

仅使用.NET Framework 公共语言运行库提供的功能的代码称为托管代码。如图 1-2 所示，可以完全依赖于公共语言运行库以及.NET Framework 类库的相关部分，仅使用托管代码构建应用程序。当然，也可以通过组合托管代码和非托管代码来构建应用程序，这时两种代码

根据需要进行交互。第二种选择（如图 1-2 的右半部分所示）对于现有应用程序尤其重要。现在创建的大部分新的 Windows 应用程序都是用托管代码构建的，但是使用托管代码扩展.NET 之前就已存在的应用程序同样很有用处。例如，企业可以重用.NET 问世之前开发的程序，使用它提供的功能（非托管代码），同时使用公共语言运行库提供的功能（托管代码）。当然，也可以像.NET Framework 问世以前一样，完全使用非托管代码创建 Windows 应用程序，.NET Framework 并不是非用不可。

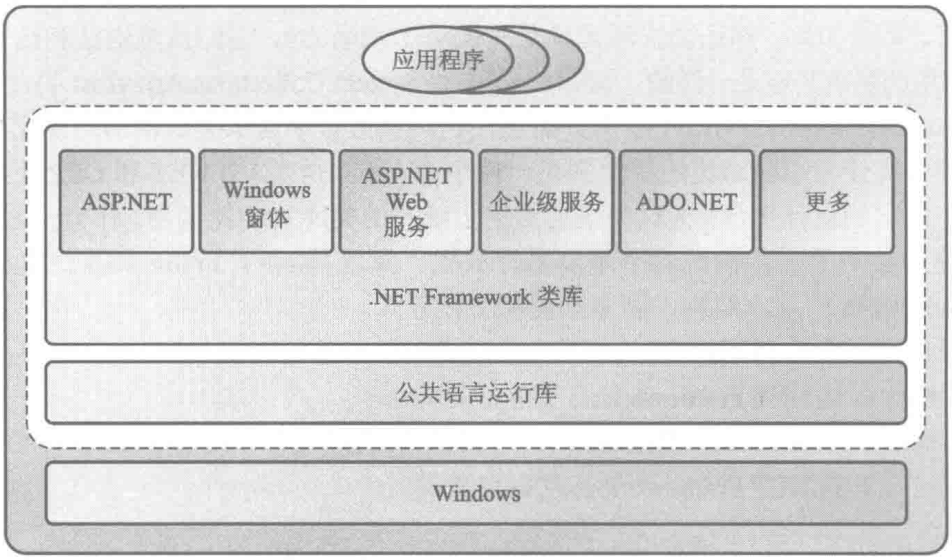

图 1-1　.NET Framework 结构

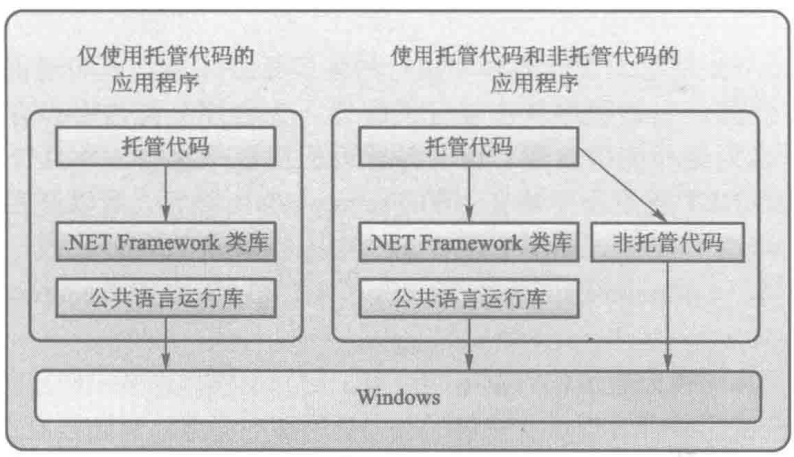

图 1-2　托管代码和非托管代码

托管代码通常是面向对象的，因此它所创建和使用的对象称为托管对象。一个托管对象可使用和继承另一个托管对象，即使两个对象是用不同的语言编写的。这是.NET Framework 类库成为开发代码的有效基础的关键所在：即以任何基于公共语言运行库的语言编写的对象都可继承和使用该类库的代码。例如，用户完全可以使用 C#编写一个类，在 VB.NET 中继承这个类或者直接使用这个类。由于公共语言运行库是.NET Framework 中所有其他内容的基础，因此我们将从理解公共语言运行库入手来深入了解.NET Framework。

1.2 .NET Framework 的结构

在图 1-1 中，使用.NET Framework 编写的每个应用程序都依赖于公共语言运行库。公共语言运行库提供了一组通用数据类型，这些类型充当 C#、VB.NET 以及任何其他针对.NET Framework 的语言的基础。例如，VB.NET 中提供了 Integer 类型，C#中提供了 System.Int32 类型，它们都与公共语言运行库中的 System.Int32 类型相对应。由于无论开发人员选择何种语言这一基础都是相同的，因此开发人员将看到更为一致的环境。

1.2.1 公共语言运行库

请先思考一下我们以前所学的编程语言，如 VB.NET 和 C#，它们通常是如何定义的。每种语言一般都有自己独特的语法、自己的控制结构集、独特的数据类型集和自己的类继承概念等多种内容。例如，VB.NET 使用换行结束当前语句，C#中则使用“;”来结束当前语句。目前对于现代编程语言所应提供的多数功能，人们已达成了广泛的共识。虽然开发人员对于编程语言的语法不能达成一致意见，有的人喜欢花括号，而有的人则不喜欢，但是对于语言应提供的语义仍然存在着广泛的共识。既然如此，为什么不为这些语义定义一种标准实现，然后允许使用不同的语法来表达这些语义呢？

公共语言运行库利用了不同编程语言的相似性，抽象出了通用类型系统（Common Type System，CTS）。通用类型系统构成了.NET Framework 的公共语言运行库的基础，它定义了.NET Framework 中的所有的数据类型，并提供了面向对象的模型以及各种语言需要遵循的标准。简单来说，无论我们用 C#还是 VB.NET 进行开发，它们中的所有数据在编译后都转换成了 CTS 中定义的类型，这就意味着.NET 代码中的全部数据最终都以相同的数据类型存储。但是公共语言运行库不涉及语法。编程语言的表示形式是怎样的？它是包含花括号、冒号还是其他什么符号？这些问题完全由编程语言设计者决定。公共语言运行库本身为编程语言设计者提供了构建工作要基于的统一语义集。

公共语言运行库不是以任何特定编程语言定义的。相反，它的功能主要是从现有流行的编程语言中派生的，如 C++、.NET 之前的 VB 版本以及 Java。现在，Microsoft 提供了多种基于公共语言运行库的语言，包括 VB.NET、C#、经过改进的 C++和 JScript.NET 以及后来增加的 Visual J#。第三方也提供了构建于公共语言运行库的语言。

1.2.2 .NET Framework 类库

.NET Framework 类库提供了一整套通用功能的标准代码，这些代码包含了开发人员可用于简化其开发工作的类和其他类型的库。例如，System.Windows.Forms 命名空间包含了构建 Windows 窗体及其所使用控件的所有类。虽然这些类本身都是用 C#编写的，但是以任何.NET 语言编写的应用程序都可以使用.NET Framework 类库中的代码，例如，用 C#、VB.NET、C++或.NET Framework 所支持的任何其他语言编写的代码，都可创建这些类的实例并调用其

方法。如果公共语言运行库中的类是可继承的，这些代码还可以从公共语言运行库中的类继承。而且.NET Framework 类库支持跨语言的继承和调试，比如可以在 C#里面继承一个用 VB.NET 定义的类，或使用一段用 VB.NET 编写的代码，这样在 C#程序里一样可以调试 VB.NET 程序。

.NET Framework 类库的内容组织为命名空间树。命名空间是执行相关功能的类型（如类和接口）的逻辑组织单位。例如，System.Windows.Forms 命名空间包含了构建 Windows 窗体及其所使用控件的所有类。每个命名空间还可以包含其他命名空间。图 1-3 所示为.NET Framework 类库命名空间树的极小一部分。

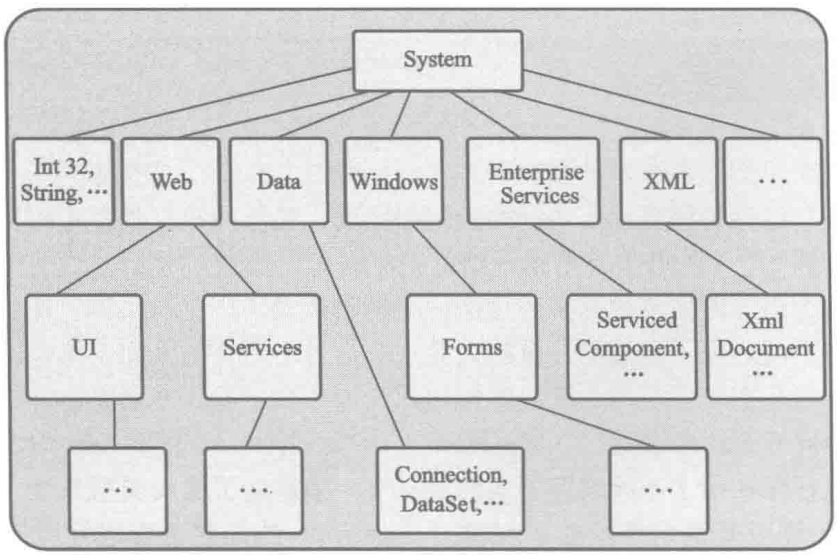

图 1-3 .NET Framework 类库命名空间树

下面简要介绍上图所列出的命名空间。

- System：树的根，此命名空间包含.NET Framework 类库中所有其他命名空间。System 还包含了公共语言运行库以及基于公共语言运行库构建的编程语言所使用的核心数据类型。这些类型包括整型的多种变体、字符串类型以及很多其他类型。
- System.Web：此命名空间包含对于创建 Web 应用程序有用的类型，并且与很多命名空间一样，它有下级命名空间。例如，开发人员可使用 System.Web.UI 中的类型来构建 ASP.NET 浏览器应用程序，而 System.Web.Services 中的类型用于构建 ASP.NET Web 服务应用程序。
- System.Data：此命名空间中的类型构成了 ADO.NET。例如，Connection 类用于建立与数据库管理系统（DBMS）的连接，而 DataSet 类的实例用于缓存对该 DBMS 所发出的查询的结果。
- System.Windows.Forms：此命名空间中的类型组成了 Windows 窗体，它们用于构建 Windows 图形用户界面（GUI）。以任何编程语言编写的.NET Framework 应用程序已不再依赖于特定于语言的机制，如 C++中早期的 Microsoft 基础类（MFC），而是使用 System.Windows.Forms 这一通用类型集来构建 Windows 图形用户界面。
- System.EnterpriseServices：此命名空间中的类型提供了企业级应用程序所需要的某