

欧阳炜昊 主编
李庆 副主编

程序员突击

- 由浅入深、循序渐进，真正**全面掌握**系统开发技术。
- 以真实的**项目开发**与实现为主线，对各个设计元素进行深入讲解。
- 精选多个**典型案例**，并结合基础知识对系统进行分析、设计与编程实现。

- 对**主流框架**进行详细分析并整合至真实项目中。
- 强大的**技术支持**，读者可在学赛网社区“书评在线”版块中与作者进行直接交流。
- 丰富的**程序资源**，读者也可以在希赛网下载中心或学赛网下载中心下载本书所有实例的源代码和实战项目的源代码。

Visual C#

2008 原理与系统开发

程序员突击系列

清华大学出版社

程序员突击——Visual C# 2008 原理 与系统开发

欧阳炜昊 主编

李 庆 副主编

清华大学出版社

北 京

内 容 简 介

本书集编程思想与具体实践于一体，集中介绍了如何利用 Visual Studio 2008 中的 C#语言编写在 Windows 环境下运行的应用程序。在内容编排上，遵循循序渐进的原则，从 C#基础到 Windows 高级编程都一一作了讲解，每个部分都设置了相应的实例。

全书分 4 篇共 14 章，内容涉及 C#的开发环境及基本语法，面向对象，常用控件，窗体应用，定制控件，文件操作，数据库访问，报表和打印，Web 服务，调试、测试与部署，还给出了一个综合项目案例。全书内容翔实，深入浅出，实例经典，可以帮助读者快速了解并掌握使用 C#开发应用程序所需的知识和技能，提高利用 C#开发 Windows 应用程序的能力。

本书面向初中级读者，适合各大专院校、软件开发培训机构作为 .NET 开发的教材，也可作为相关软件开发人员的参考用书。

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

程序员突击——Visual C# 2008 原理与系统开发/欧阳炜昊主编. —北京：清华大学出版社，2009.8

ISBN 978-7-302-18354-9

I. 程… II. 欧… III. C 语言-程序设计 IV. TP312

中国版本图书馆 CIP 数据核字（2008）第 120361 号

责任编辑：涂 荣

封面设计：刘 超

版式设计：侯哲芬

责任校对：姜 彦

责任印制：李红英

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 装 者：北京国马印刷厂

经 销：全国新华书店

开 本：185×260 印 张：22.75 字 数：526 千字

版 次：2009 年 8 月第 1 版 印 次：2009 年 8 月第 1 次印刷

印 数：1~4000

定 价：36.00 元

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：030408-01

前 言

C#是一门简单易学的面向对象编程语言，其语法风格源自 C/C++家族，融合了 Visual Basic 的高效开发和 C/C++的强大功能，是微软的.NET 平台主流开发语言。

本书集编程思想与具体实践于一体，集中介绍了如何利用 Visual Studio 2008 中的 C#语言编写在 Windows 环境下运行的应用程序。在内容编排上，遵循循序渐进的原则，从 C#基础到 Windows 高级编程都一一作了讲解，每个部分都设置了相应的实例。

通过本书的学习，读者可以快速了解并掌握使用 C#开发应用程序所需的各种知识和技能，提高利用 C#开发 Windows 应用程序的能力。

一、章节内容介绍

全书分为 4 篇共 14 章。

□ 第一篇“步入 Visual C# 2008 开发的殿堂”：包括第 1 章和第 2 章。

第 1 章介绍 .NET 平台及 .NET 框架的基本概念，并通过创建一个简单的 Hello World 项目对 Visual Studio 2008 中 C#集成开发环境作简单介绍。

第 2 章讲解面向对象程序设计的一些基本概念，如类、继承、多态、委托和事件等，让大家理解 OOP 编程内涵以及一些 C#编程技巧。

□ 第二篇“苦练基本功”：包括第 3~12 章共 10 章。

第 3 章首先通过一个启动界面程序实例向读者介绍 Windows 程序的编程机制，然后又通过一个实例介绍如何使用 .NET 集成开发环境进行 Windows 应用程序开发，最后对 Windows 窗体类和对象作比较详细的介绍，其中包括窗体的常见属性、方法和事件。

第 4 章描述 Windows 窗体一些常用的控件。通过一些实例，让读者了解这些控件的基本使用方法，同时理解如何使用控件来设计 Windows 窗体。

第 5 章通过几个实例介绍 Windows 窗体中提供一些比较高级的功能，比如验证应用程序中用户输入的数据、动态添加和使用控件以及控件的组织排列等。

第 6 章讲述如何根据自己的需要扩展 .NET 框架的功能，如通过组合几个现有 Windows 窗体控件创建复合控件等。

第 7 章的主要内容是 .NET 的 I/O 操作，重点介绍文件系统输入和输出以及 .NET 的文件系统管理。

第 8 章介绍 GDI+绘制基本图形的基础知识和基本操作，读者从中可以了解如何在 Windows 界面上绘制各种图形，如何对图像对象进行加载、显示、裁切、缩放、旋转、扭曲等处理。

第 9 章讲解连接环境下的编程模式，主要介绍连接数据库的 Connection 对象、执行数据库操作的 Command 对象，以及如何执行存储过程、如何创建和使用参数。

第 10 章介绍在非连接环境下访问数据库的模式,描述以数据库为中心的应用程序设计、ADO.NET 架构以及 ADO.NET 和 XML 之间的集成。

第 11 章详细讲述了如何在 Windows 窗体应用程序中实现打印功能以及如何使用 Visual Studio 2008 的 Crystal Report 在 Windows 窗体应用程序中创建报表。

第 12 章系统介绍了在开发应用程序时如何使用调试、测试技术来找出和解决应用程序出现的各种问题,此外还介绍了应用程序的部署。

□ 第三篇“突出重围 项目实战”:即第 13 章。

通过一个综合应用实例——“考试管理系统”项目,针对 C#开发 Windows 应用程序各方面的知识点作系统的回顾与总结,读者从实战演练中可以切身体会 C#的实际应用,并且进一步加深对软件工程知识的认识。

□ 第四篇“高级开发技术——Web 服务”:即第 14 章。

讲述 System.Web.Services 命名空间,以及如何在 C#应用程序中创建和使用 Web Services 这一高级开发技术。

本书深入浅出,并辅以大量的实例说明,适合具有初级编程能力、对 .NET 和 C#有基本的了解,并且希望学到更多关于 Windows 编程的开发人员,同样适合各大专院校、软件开发培训机构作为 .NET 开发的教材,还可作为相关软件开发人员的参考用书。

二、技术支持

希赛是中国领先的互联网技术和 IT 教育公司,在互联网服务、图书出版、人才培养方面,希赛始终保持 IT 业界的领先地位。希赛对国家信息化建设和软件产业化发展具有强烈的使命感,始终坚持利用希赛网(www.csai.cn)强大的平台优势,加强与促进 IT 人士之间的信息交流和共享,实现 IT 价值。“希赛,影响 IT”是全体希赛人不懈努力和追求的目标!

希赛网以希赛顾问团为技术依托,是中国最大的 IT 资源平台。希赛 IT 发展研究中心是希赛公司下属的一个专门从事 IT 咨询、行业发展报告、技术/管理书籍编写的部门,在 IT 技术/管理导向方面具有极高的权威性。在国家权威机构发布的《计算机图书出版市场综述》中,称赞希赛丛书为读者所称道,希赛图书已经逐渐形成品牌,在读者心目中具有良好的形象。

参加本书编写工作的有欧阳炜昊、李庆、王敏、伍荣跃、王春艳、周泉、周进、顿海丽、张爱民、王勇、唐强、谢顺、王永明、左南、张友生、邓子云、黄婧、梁赛、杨花、彭欢。

囿于作者的水平,加之时间仓促,书中的错误和不妥之处在所难免,敬请读者批评指正。真诚地希望本书能对想学习用 C#开发 Windows 应用程序的读者有所帮助。

有关本书的意见反馈和咨询,读者可在学赛网社区(<http://bbs.educity.cn>)“书评在线”版块中与作者进行交流。读者也可以在希赛网下载中心(<http://data.csai.cn>)或学赛网下载中心(<http://data.educity.cn>)下载本书所有实例的源代码和实战项目的源代码。

欧阳炜昊

目 录

第一篇 步入 Visual C# 2008 开发的殿堂

第 1 章 Visual C# 2008 简介	2
1.1 .NET 平台和.NET 框架概述	2
1.1.1 .NET 平台	2
1.1.2 .NET 框架	3
1.2 Visual C# 2008 开发环境	8
1.2.1 Visual Studio 2008 概述	9
1.2.2 Visual C# 2008 集成开发环境	9
1.3 创建一个简单的 Visual C# Windows 应用程序	13
1.4 小结	15
第 2 章 C#面向对象程序设计	16
2.1 类的概述	16
2.1.1 类的定义	16
2.1.2 创建类实例	17
2.1.3 类成员	17
2.1.4 构造函数	18
2.1.5 方法	19
2.1.6 字段与属性	24
2.1.7 如何使用命名空间来组织类	26
2.2 继承	28
2.3 多态	30
2.3.1 抽象类	31
2.3.2 密封类	32
2.4 接口	33
2.4.1 接口声明	33
2.4.2 接口实现	34
2.5 委托与事件	35
2.5.1 委托	35
2.5.2 事件	37
2.6 小结	38

第二篇 苦练基本功

第 3 章 Windows 窗体	40
3.1 Windows 窗体简介	40
3.2 使用 Windows 窗体	46
3.2.1 创建窗体	46
3.2.2 设置窗体属性	47
3.2.3 调用方法	51
3.2.4 窗体的事件介绍	52
3.3 创建多文档界面应用程序	57
3.3.1 单文档界面和多文档界面应用程序	57
3.3.2 创建多文档界面应用程序	58
3.3.3 父窗体和子窗体的交互	59
3.4 小结	61
第 4 章 使用 Windows 窗体基本控件	62
4.1 使用 Windows 窗体控件	62
4.1.1 根据功能选择 Windows 窗体控件	62
4.1.2 制作登录框	64
4.1.3 Label 控件	67
4.1.4 TextBox 控件	67
4.1.5 Button 控件	68
4.2 选择类控件	69
4.2.1 个人资料填写程序	69
4.2.2 使用 ListBox 控件和 ComboBox 控件	72
4.2.3 使用 CheckBox 和 RadioButton 控件	76
4.2.4 使用 GroupBox 和 Panel 控件	77
4.3 菜单和工具栏	78
4.3.1 创建菜单	78
4.3.2 使用 toolStrip 控件	79
4.3.3 使用 statusStrip 控件	81
4.4 在 Windows 窗体应用程序中使用对话框	82
4.4.1 在 Visual Studio .NET 中选择对话框	82
4.4.2 在应用程序中显示对话框	86
4.4.3 DialogResult 属性	87
4.5 文本编辑	88
4.5.1 RichTextBox 控件简介	88
4.5.2 RichTextBox 的主要属性	89

4.5.3	RichTextBox 控件的主要方法	90
4.5.4	RichTextBox 控件的主要事件	90
4.6	制作写字板	91
4.6.1	程序分析功能	91
4.6.2	控件属性设置	91
4.6.3	程序代码编写	95
4.6.4	程序总结	101
4.7	小结	101
第 5 章	Windows 窗体高级控件	102
5.1	视图列表类控件	102
5.1.1	使用 ImageList 控件	102
5.1.2	ListView 控件	102
5.1.3	TreeView 控件	106
5.1.4	制作员工资料管理器	107
5.2	验证用户输入	111
5.2.1	处理焦点	111
5.2.2	使用 Validating 事件验证控件	112
5.2.3	窗体级别的验证	113
5.2.4	ErrorProvider 控件	114
5.3	在运行时添加控件	115
5.3.1	控件集合	115
5.3.2	在运行时添加控件	116
5.4	控件的组织	116
5.4.1	排列控件	116
5.4.2	设置 Tab 键顺序	117
5.4.3	锚定 Windows 窗体中的控件	117
5.4.4	停靠 Windows 窗体的控件	117
5.5	小结	118
第 6 章	控件的高级应用	119
6.1	创建用户控件	119
6.1.1	创建控件的方式	119
6.1.2	提供并重载控件属性	122
6.1.3	触发并重载控件事件	124
6.1.4	测试控件	125
6.1.5	创建复合控件	125
6.2	为控件添加设计时支持	129
6.2.1	属性的属性 (Property Attribute)	129

6.2.2 内建在.NET 框架中的设计时支持选项	129
6.3 控件许可	131
6.3.1 文件许可	132
6.3.2 启用控件许可	133
6.3.3 LicFileLicenseProvider 在.NET 中的工作流程	135
6.4 小结	135
第 7 章 文件操作	136
7.1 流	136
7.1.1 .NET 流的体系结构	137
7.1.2 Stream 类	137
7.2 Reader 和 Writer	139
7.3 基本文件 I/O	142
7.3.1 FileStream 类	143
7.3.2 File 和 FileInfo 类	144
7.3.3 Directory 和 DirectoryInfo 类	145
7.3.4 FileSystemWatcher	146
7.3.5 独立存储	148
7.4 小结	149
第 8 章 图形图像处理	150
8.1 GDI+简介	150
8.2 GDI+对象	151
8.2.1 Graphics 对象	152
8.2.2 Graphics 对象的方法	152
8.2.3 创建和使用 Pen、Brush 对象	153
8.3 绘制基本图形	155
8.3.1 绘制直线和矩形	155
8.3.2 绘制椭圆和弧线	157
8.3.3 绘制多边形	158
8.3.4 画刷和填充的形状	159
8.4 图像处理	162
8.4.1 加载和显示图片	162
8.4.2 裁切和缩放图像	163
8.4.3 旋转、反射和扭曲图像	164
8.4.4 创建缩放图像	165
8.5 文本显示	166
8.5.1 使用 Font 对象	166
8.5.2 使用 DrawString 方法绘制文本	167

8.6 小结	169
第 9 章 连接环境下的 ADO.NET 数据库	170
9.1 ADO.NET 架构	170
9.2 在连接环境中使用 ADO.NET	172
9.3 连接数据库	173
9.3.1 .NET 数据提供程序	174
9.3.2 定义连接	176
9.3.3 打开和关闭连接	179
9.4 执行数据库操作	180
9.4.1 使用连接环境下应用程序的对象模型	180
9.4.2 创建 Command 对象	181
9.4.3 执行 Command 对象	184
9.5 使用存储过程	189
9.5.1 Command 对象参数	189
9.5.2 为 Command 对象创建参数	190
9.5.3 实例分析	193
9.6 小结	195
第 10 章 非连接环境下访问数据库	196
10.1 在非连接环境中使用 ADO.NET	196
10.1.1 ADO.NET 对象模型	196
10.1.2 在非连接环境中使用 ADO.NET	197
10.2 使用数据集	198
10.2.1 构建 DataSet 和 DataTable	198
10.2.2 更改 DataTable 中的数据	201
10.2.3 DataView 对象	204
10.3 使用数据集访问和修改数据	208
10.3.1 DataAdapter 类	208
10.3.2 使用 DataAdapter 填充 DataSet	212
10.3.3 配置 DataAdapter 更新后台数据源	216
10.4 DataSet 与 XML	221
10.4.1 创建 XSD 架构	221
10.4.2 将架构和数据加载到 DataSet 对象中	225
10.4.3 将 DataSet 写入 XML	227
10.5 数据绑定技术	231
10.5.1 为 Windows 控件绑定数据	231
10.5.2 ADO.NET 2.0 中的新数据绑定技术	232
10.6 小结	238

第 11 章 报表和打印	239
11.1 .NET 打印概述	239
11.1.1 .NET Windows 窗体应用程序中的打印操作	239
11.1.2 PrintDocument 对象	241
11.1.3 PrintPage 事件和 PrintPageEventArgs 对象	242
11.1.4 在 Windows 窗体应用程序中启用打印	244
11.2 使用打印常用对话框	245
11.2.1 使用 PrintPreviewDialog 控件	246
11.2.2 使用 PageSetupDialog 控件	247
11.2.3 使用 PrintDialog 控件	249
11.3 使用 Crystal Report 创建报表	250
11.3.1 Crystal Report	251
11.3.2 使用 Crystal Report 创建和格式化报表	251
11.4 使用 Report 创建报表	255
11.4.1 Report 报表的创建	255
11.4.2 使用 ReportViewer 控件查看器查看报表	257
11.4.3 ReportViewer 工具栏	258
11.4.4 如何为 ReportViewer 报表添加数据源	260
11.5 小结	264
第 12 章 调试、测试与部署	265
12.1 应用程序调试	265
12.1.1 错误分类	265
12.1.2 调试命令	266
12.2 应用程序测试	269
12.2.1 测试概述	270
12.2.2 测试方法	272
12.3 打包和部署.NET 应用程序	275
12.3.1 基本概念	275
12.3.2 Windows Installer 安装项目的组件	275
12.4 小结	281

第三篇 突出重围 项目实战

第 13 章 综合应用实例——考试管理系统	284
13.1 系统总体结构设计	284
13.1.1 系统功能设计	284
13.1.2 系统数据库设计	285

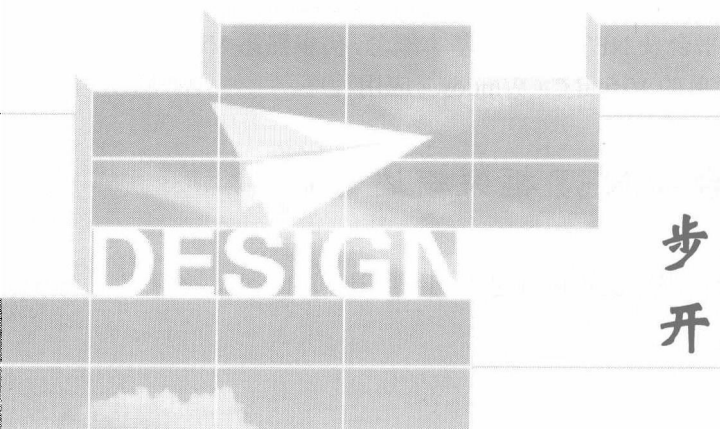
13.2 公共类	288
13.2.1 主要功能	288
13.2.2 具体要求	288
13.2.3 注意事项	289
13.2.4 主要代码段	289
13.3 “考生管理”模块	291
13.3.1 功能	291
13.3.2 数据流程	291
13.3.3 具体界面设计	291
13.3.4 具体要求和注意事项	292
13.3.5 所运用的知识点	292
13.3.6 主要代码段	292
13.4 “试卷管理”模块	300
13.4.1 功能	300
13.4.2 数据流程	300
13.4.3 具体界面设计	300
13.4.4 具体要求和注意事项	301
13.4.5 所运用的知识点	301
13.4.6 主要代码段	302
13.5 学生登录模块	312
13.5.1 功能	312
13.5.2 数据流程	312
13.5.3 具体界面设计	313
13.5.4 具体要求和注意事项	313
13.5.5 所运用的知识点	313
13.5.6 主要代码段	314
13.6 考生考试控制模块	316
13.6.1 功能	317
13.6.2 具体要求	317
13.6.3 具体界面设计	317
13.6.4 注意事项	317
13.6.5 所运用的知识点	318
13.6.6 主要代码段	318
13.7 教师考试控制模块	323
13.7.1 功能	323
13.7.2 具体界面设计	323
13.7.3 所运用的知识点	323
13.7.4 主要代码段	324

13.8 小结	325
---------------	-----

第四篇 高级开发技术——Web 服务

第 14 章 构建和使用 Web 服务	328
14.1 XML Web Services 概述	328
14.1.1 什么是 XML Web Services	328
14.1.2 XML Web Services 的特点	329
14.1.3 使用 XML Web Services 的原因	330
14.1.4 查找 XML Web Services	332
14.2 通过 HTTP 调用 XML Web Services	334
14.3 通过代理调用 XML Web Services	335
14.3.1 使用代理调用 XML Web Services	336
14.3.2 在 Windows 应用程序代码中调用 XML Web Services 方法	339
14.3.3 XML Web Services 错误处理	340
14.3.4 使用 WSDL 工具生成代理代码	341
14.4 创建 XML Web Services	343
14.4.1 创建 XML Web Services	343
14.4.2 XML Web Services 的代码	344
14.4.3 XML Web Services 文档	346
14.4.4 浏览 Web Service	347
14.5 部署 Web Service	349
14.6 综合实例	350
14.7 小结	352

第一篇



步入 Visual C# 2008 开发的殿堂

- ➔ 要想成为一名程序员，首先就要了解所要掌握的编程语言的开发环境，以及语言语法，从而步入开发的殿堂。
- ➔ 本篇包括两章——第 1 章 Visual C# 2008 简介；第 2 章 C#面向对象程序设计。
Visual C# 2008 是目前最新的 C#语言开发环境，通过本篇的学习，读者应当掌握.NET 平台及.NET 框架的基本概念，初步熟悉 Visual C# 2008 的集成开发环境；能够理解类、继承、多态、接口、委托和事件的基本概念，并能够运用 Visual C# 2008 编写简单的应用程序。

第 1 章 Visual C# 2008 简介

Visual Studio 2008 是一套完整的开发工具集,用于生成 ASP.NET Web 应用程序、XML Web Services、桌面应用程序和移动应用程序。本章将针对 Visual Studio 2008 中 Visual C# 2008 集成开发环境作一简单介绍。

完成本章的学习后,读者应掌握.NET 平台及.NET 框架的基本概念,初步熟悉 Visual C# 2008 的集成开发环境,并能够创建一个简单的 Visual C# Windows 应用程序。

1.1 .NET 平台和.NET 框架概述

随着网络技术的不断发展,单机的软件程序已难以满足人们网络计算的需求,各种各样的网络计算模式应运而生。

1.1.1 .NET 平台

.NET 平台为开发人员提供了多种核心的技术和服务,使用这些技术和服务,能够对基于 Web 的应用程序开发进行简化。

1. .NET 平台组件

(1) 开发工具

Visual Studio 2008 和.NET 框架(.NET Framework)为开发人员提供了一整套的解决方案。利用 Visual Studio 2008 提供的一整套高效实用、功能强大的开发工具,可以极大地简化 ASP Web 应用程序和 XML Web Services 的创建、部署和升级。用.NET 开发环境所创建的 ASP Web 应用程序和 XML Web Services 具有安全性、可伸缩性以及高可用性的特点。

(2) 设备

设备包括个人计算机、便携式计算机、工作站、电话机、手持计算机、Tablet PC 以及游戏控制台等。这些智能化设备可以访问 XML Web Services 和数据,而无须考虑正在使用的设备类型、数量以及它们所处的位置。

(3) 用户体验

.NET 体验(.NET Experiences)是一些使用 XML Web Services 的应用程序,它们提供了一种集成而有效的方法,使得用户可以从 Internet 或是从独立的应用程序中快捷、方便地获得海量信息。

(4) 服务器组件

通过提供对 XML Web Services 的支持,.NET Enterprise Server 家族提高了系统、应

用程序以及其他合作伙伴的集成速度。由于支持 XML，企业可以在先前系统的基础上继续开发，而不是替换这些系统。例如，Microsoft Host Integration Server 提供了对大型机的简单访问，Microsoft BizTalk Server 则提供了现有数据格式和 XML 数据格式之间的自动转换功能。

（5）XML Web Services

通过使用 XML Web Services，应用程序可以共享数据并调用其他应用程序，而无须考虑这些应用程序的创建方式、运行该应用程序的操作系统类型或平台类型，以及访问这些应用程序的设备。

2. .NET 平台的优化

针对开发人员，.NET 进行了以下优化。

（1）加快应用程序的开发

在创建应用程序时，开发人员可以针对特定问题，自由选择最合适的编程语言。

（2）优良的可靠性

.NET 平台具有分布式计算的优势。CLR 提供了一个可管理的执行环境，通过该环境可有效地避免内存泄漏、访问冲突以及版本控制等问题。此外，.NET 框架强化了类型安全、代码共享以及应用程序隔离等功能，从而确保任何应用程序都不会影响或非法调用其他应用程序。

（3）基于 Web 标准

通过使用 XML，可以消除数据共享和软件集成中的障碍。简单对象访问协议（Simple Object Access Protocol, SOAP）是一种基于 XML 的消息传输技术，万维网联盟（World Wide Web Consortium, W3C）对该协议进行了标准化，该标准制定了使用 XML Web Services、将 XML Web Services 集成到应用程序以及在 XML Web Services 之间进行通信的所有必要规则。

.NET 具有数据库访问功能，这使得开发人员可以在其应用程序中访问与开放式数据库互连（ODBC）相兼容的数据源。

1.1.2 .NET 框架

.NET 框架是构成 .NET 平台核心部分的一组技术，其设计目的是为了简化 Internet 广泛分布式环境中的应用程序开发过程，为创建和运行 .NET 应用程序提供了必要的编译和运行基础。

通常认为框架是一种用于帮助开发应用程序的类库集，但 .NET 框架不仅仅是一种类库集。迄今为止，已经有多种编程语言的编译器使用了 .NET 框架。运用这些编程语言可以创建各种各样的应用程序，小至在手持设备上运行的小组件，大至大型的 ASP.NET 应用程序。NET 框架为这些应用程序提供一个可共享的基本平台，该平台可以监视应用程序的执行，使得用不同编程语言所编写的软件可以安全地使用类和其他数据类型。

.NET 框架旨在实现下列目标：

- ❑ 提供一个一致的面向对象的编程环境，而无论对象代码是在本地存储和执行，还是在本地执行但在 Internet 上分布，或者是在远程执行的。
- ❑ 提供一个将软件部署和版本控制冲突最小化的代码执行环境。
- ❑ 提供一个可提高代码（包括由未知的或不完全受信任的第三方创建的代码）执行安全性的代码执行环境。
- ❑ 提供一个可消除脚本环境或解释环境的性能问题的代码执行环境。
- ❑ 使开发人员的经验在面对类型迥异的应用程序（如基于 Windows 的应用程序和基于 Web 的应用程序）时保持一致。
- ❑ 按照工业标准生成所有通信，以确保基于 .NET 框架的代码可与任何其他代码集成。

.NET 框架包含公共语言运行库（Common Language Runtime, CLR）和 .NET 框架类库两个主要组件。此外，.NET 框架还提供一些运行库宿主，包括 Windows 窗体和 ASP.NET，它们可直接使用运行库为管理的代码提供可扩展的服务器端环境。默认情况下，.NET 框架随 Windows Server 2003 操作系统一同安装。

1. .NET Framework 环境

.NET 框架具有两个主要组件：公共语言运行库和 .NET 框架类库。公共语言运行库是 .NET 框架的基础。可以将运行库看作一个在执行时管理代码的代理，它提供内存管理、线程管理和远程处理等核心服务。事实上，代码管理的概念是运行库的基本原则。以运行库为目标的代码称为托管代码（Managed Code），而不以运行库为目标的代码称为非托管代码。公共语言运行库也称为“托管环境”，在这个托管环境中将自动提供诸如垃圾回收和安全性等通用服务。.NET 框架的另一个主要组件是类库。.NET 框架类库是面向对象的可重用类的广泛集合，开发人员可以使用该类库开发多种应用程序，这些应用程序包括传统的命令行或图形用户界面（GUI）应用程序，也包括基于 ASP.NET 所提供的创新的应用程序（如 Web 窗体和 XML Web Services）。

.NET 框架可由非托管组件承载，这些组件将公共语言运行库加载到它们的进程中并启动托管代码的执行，从而创建一个可以同时利用托管和非托管功能的软件环境。.NET 框架不但提供若干个运行库宿主，而且还支持第三方运行库宿主的开发。

例如，ASP.NET 承载运行库可以为托管代码提供可伸缩的服务器端环境，ASP.NET 直接使用运行库以启用 ASP.NET 应用程序和 XML Web Services。

Internet Explorer 是承载运行库（以 MIME 类型扩展的形式）的非托管应用程序的一个示例。使用 Internet Explorer 承载运行库，用户可以在 HTML 文档中嵌入托管组件或 Windows 窗体控件。以这种方式承载运行库使得托管移动代码（类似于 Microsoft ActiveX 控件）成为可能，不过它需要进行重大改进（如不完全受信任的执行和独立的文件存储），而这种改进只有托管代码才能提供。

图 1-1 对公共语言运行库、类库和应用程序之间以及和整个系统之间的关系进行了描述，同时还显示了托管代码如何在更大的结构内运行。