

● Microsoft

微软新软件技术集锦

# 微软.NET平台 软件技术

—— 用 C# 设计各类应用程序的技术

庄成三（等） 编著

微软新软件技术集锦

开发 Web 和 Windows 应用程序统一平台

新的优秀的面向对象语言 C#

互联网新一代语言 XML 和 Web 服务技术



电子科技大学出版社

DIANZIKEJIDAXUECHUBANSHE

TP312  
2071

# 微软.NET 平台软件技术

——用 C# 设计各类应用程序的技术

庄成三（等） 编著

江苏工业学院图书馆  
藏书章

电子科技大学出版社

## 内 容 提 要

本书系统地由浅入深地讲解 C#程序设计。全书共有 15 章,主要内容有:虚函数和多态、接口和事件、异常处理及多线程等技术,在 .NET 中设计 Windows 应用程序,应用 ADO.NET 存取数据库,ASP.NET 及设计 Web 应用程序,XML 应用基础,Web 服务原理和设计,以及软件部署、配置和安全性等技术。

本书内容丰富,概念清晰精辟,程序可靠,实用性强。读者学习本书,将尽快掌握 .NET 平台设计各类软件,包括 C/S 或三层结构的 Windows 应用、Web 应用和 Web 服务的技术。本书可作为高校软件技术课程的教材,也是自学或培训软件人员的首选用书。

### 图书在版编目(CIP)数据

微软.NET 平台软件技术:用 C#设计各类应用程序的技术/庄成三(等)编著. —成都:电子科技大学出版社,2003.8  
ISBN 7-81094-045-7

I. 微… II. 庄… III. C 语言-程序设计  
IV. TP312

中国版本图书馆 CIP 数据核字(2003)第 074827 号

## 微软.NET 平台软件技术 ——用 C#设计各类应用程序的技术

庄成三(等) 编著

---

出 版:电子科技大学出版社 (成都建设北路二段四号,邮编:610054)

责任编辑:徐守铭

发 行:新华书店经销

印 刷:郫县犀浦印刷厂

开 本:787mm×1092mm 1/16 印张:33.25 字数:815 千字

版 次:2003 年 9 月第一版

印 次:2003 年 9 月第一次印刷

书 号:ISBN 7-81094-045-7/TP·30

印 数:1—2000 册

定 价:49.50 元

---

本书如有缺页、破损、装订错误,请寄回印刷厂调换。

# 编著者的话

.NET 是微软公司最近推出的新的软件开发平台，综合体现了软件技术的最新发展，在美国一公布，就迅速地得到推广，占领了巨大的市场，很快在我国也得到了广泛运用。微软的平台是当前世界上的两类主流软件开发平台之一，国内外所有选择了微软平台的公司、单位、研究所的软件人员都面临学习这个新平台技术的任务，否则将会因此被淘汰。可以预料，在今后，这些单位对新聘用的软件人员的大多数，第一位要求将会是掌握.NET 软件开发平台的技术，人才市场对掌握.NET 软件开发平台的技术的人员的需求一定会越来越多。凡用微软平台的软件人员，掌握.NET 技术将会成为第一需要。

当前国内市面上已出版了很多关于.NET 软件技术的书籍，说明了这门技术有多么热门。但是，这些书籍，要么一般介绍.NET 技术，很粗浅地讲解 C# 等知识，要么要求读者具有 VB, Com+ 及 ASP 等知识才能读懂。显然缺乏一本既比较系统地由浅入深地介绍.NET 基本技术，同时又包括作为.NET 平台上的软件开发人员需要掌握的基本内容的书，既适合初学，又适合致力于成为.NET 程序员或从事.NET 平台软件开发工作者学习的书。我们在高校教学和在软件企业指导设计商品化软件时感到广大学生和软件人员需要这样的书，这便是我们决定编写这本书的初衷。

无论是学习 C# 语言，还是学习.NET 中 ASP、ADO 等技术，目的都是用它设计应用软件，最终解决实际的问题。本书紧紧围绕这个目标精心选择和组织教材内容。我们不是就语言讲语言，就某软件技术讲技术，而是系统地讲解设计应用软件需要的知识和技术。编写的指导思想和具体方法在许多方面和已有的同类教材或技术书籍不同。对较难理解和掌握的重要内容，比如虚函数，我们从例子开始，一次一次逐渐加深读者的理解。对异常处理，还有事件等这些较难的内容，也是多次覆盖，从学习简单的控制台程序设计直到学习 Web 程序设计，反复学习巩固，使读者逐步加深印象到最后掌握它，而不是安排一个专门章节来讲解。从开始介绍程序设计，我们就注意逐步介绍面向对象的程序设计思想，而不是等到学习设计类才开始。书中尽量用简单明白的方式表示比较复杂的概念或问题，将复杂问题在读者面前简单化，这也是本书一直追求的

一个目标。我们的目标总是让读者用有限时间掌握用.NET 技术开发面向市场的实际软件的方法，使读者读了这本书，感到书中学的多数知识在实际软件工作中用得上，实际工作能力有所提高。

本书最重要部分的内容完全由撰写者根据本人经验编写而成，其中例题由撰写者精心策划，独立完成了程序设计，并都已经在机上通过。从参考书选择的例题也都全部在机上通过。同时考虑了初学者和读者自学的需要，对例题的解释比较细。本书不仅介绍程序设计技术，也兼顾工具使用和排错、配置和部署等内容（这也就稍稍扩大了篇幅）。

本书主要内容有：**C#**程序设计，包括了虚函数和多态、接口和事件、异常处理、文件操作及多线程等内容，**Form** 类及控件的应用和 **Windows** 应用程序设计，**ADO.NET** 的应用和存取数据库的方法，**Web** 窗口和控件的应用及 **Web** 应用程序设计，**XML** 应用基础，**Web** 服务原理和设计，应用软件配置和安装，以及应用程序的安全等等。经过这些内容的学习，将引导读者由浅入深，一步一步，由实践到抽象，逐渐掌握用 **C#**语言和.NET 框架类库设计单层的、**C/S** 结构或三层结构的 **Windows** 应用程序，**B/S** 结构的 **Web** 应用程序，以及最新的 **Web** 服务的技术。

本书第一章介绍.NET 的特点，中间用到的一些术语较深，初学者可跳过，待学完全书后再回头来学。第二章和第三章介绍了几个.NET 环境的不同类型程序的简单例子，使读者对什么是程序以及.NET 环境开发的应用程序有个总的印象。

第四章讲述 **C#**程序设计的基本知识，也是所有程序设计共有的基本知识，就是简单数据类型、其上的操作和控制流程，以及最常用的两种聚集机制——结构和数组。在讲解数据类型时就开始向读者介绍面向对象的知识，在讲解控制流程时就开始讲述异常处理的概念和技术。

第五章和第六章学习 **C#**面向对象程序设计。第五章学习基本的内容，类和类的成员的定义，包括属性、索引符、操作符的重载等内容。第六章学习继承和多态，着重讲解虚函数的用法，此外还包括接口、委托和事件的概念和用法。这章还讨论了 **C#**程序设计的一些常用的较高级的技术，包括类型转换、文件操作、字符串处理的进一步知识、数组列表、集合和词典等，并进一步学习异常处理技术和多线程技术。

第七章讲解 **Windows** 应用程序设计的技术。从介绍事件驱动程序的特点出发，明白设计 **Windows** 应用程序的特点和用到的方法。然后依次学习这些方法，

从输入、输出、用户交互界面设计，最后到交互式绘图的技术。在这当中，反复回顾了虚函数应用，事件和异常处理等程序设计中较难的内容。

第八章讲解用 ADO.NET 存取数据库技术基础，为了方便现在还没有具备数据库知识的读者，先从程序设计员角度简略介绍数据库以及和数据库接口的基本知识，然后围绕存取数据库经常完成的操作讲解 ADO.NET 的几个常用类的用法。最后简单介绍数据绑定的概念和方法，以及利用 VS 的工具设计存取数据库的程序的方法。

第九章介绍了 Windows 应用程序部署的方法。第十章学习用 ASP.NET 开发 Web 应用程序的基本方法。为了方便没有具备 ASP 的知识，甚至对国际互联网也不熟悉的读者，这章开始便扼要介绍了 Web、Web 应用程序和 ASP 的知识。然后，介绍了 .NET 环境 Web 应用程序工作原理和设计 Web 应用程序的常用技术，包括事件处理器设计的有关技术问题，常用服务器控件的用法，ASP.NET 的核心对象的用法等。最后介绍了 Web 应用程序中的数据绑定的方法。

第十一章介绍设计用户控件和定制控件的方法以及组件的概念和设计方法，第十二章介绍用 ADO.NET 处理 XML 数据的基本方法。

第十三章介绍 Web 应用程序的配置方法和安全性的初步知识。还讲解了在 Global.asax 设计管理状态的代码方法和应用有关的事件的用法。

第十四章和第十五章学习 Web 服务的基础知识。第十四章学习 Web 服务的原理，第十五章学习设计 Web 服务和调用 Web 服务的方法。

本书全部围绕教会读者用 .NET 的技术设计出社会和经济生活中实际需要的软件进行讲述，根据软件开发设计的实际要求，学习 C# 语言和程序设计、.NET 框架类库和工具应用的技术和方法。重要概念和技术方法都用例子说明，用例子引入，并解释清楚在实际软件开发中的作用。所有的例与讲述的内容紧密结合。例子由简到繁，学到最后，读者将能完成功能十分齐全的应用程序。

如果作教材用，本书可供高校有程序设计基础知识的本科生或研究生 50~70 学时的软件技术课程使用，或缺乏程序设计基础知识的本科生两门各自 40~60 学时的课程（一门可代替 C++ 程序设计课）使用；如果自学或培训班用，以本教材为主结合学习有关计算机软件知识，只要在机上独立测试通过所有例题，并独立完成大多数习题，学员在微软平台上的实际软件开发能力可在半年多的时间达到甚至超过按印度 NIIT 计划近两年学习后（学完 VC）的水平。

本书适宜自学，不要求读者一定要有很多软件知识背景。有使用 Web 经历

较好，有一点程序设计的基本知识更好。但这些都不是必须的，没有这些基础，完全从头开始也能使用本书，只要循序渐进学完本书基本部分，在机上完成全部练习，就一定能达到目的。

本书在撰写过程中，撰写者指导的研究生给予了很大帮助。特别是刘丽丽和周艳芳同学在机上测试通过全部例题，刘丽丽还完成了第三章、第十章和第十一章、以及第十二章、第十五章部分的最初的稿子，周艳芳完成了第二章、第七章和第八章最初的稿子，李晓伟同学帮助录入第九章，程晓云同学帮助测试个别例题。全书所有章节都由庄成三教授做几次重大修改，反复增删，最后才定稿。我们尽可能争取向读者奉献一本正确、有用而易读易学的技术书，经常为了某个概念解释得更准确而反复上机编程测试数次。但是，由于时间比较紧，我们使用.NET的时间还不长，水平有限，加之工作仍显匆忙，不完善的、甚至错误的地方在所难免，欢迎读者指正，不胜感激。

编著者

2003年8月

# 目 录

## 第一章 .NET 的结构和主要特征

|                       |   |                           |   |
|-----------------------|---|---------------------------|---|
| 1.1 微软的软件开发平台的发展..... | 1 | 1.3 .NET 中应用程序开发特点.....   | 5 |
| 1.2 .NET 框架的主要特征..... | 2 | 1.3.1 Windows 应用程序开发..... | 5 |
| 1.2.1 公共语言运行环境.....   | 3 | 1.3.2 Web 应用程序开发.....     | 5 |
| 1.2.2 NET 框架类库.....   | 4 | 1.3.3 XMLWeb 服务开发.....    | 6 |

## 第二章 用 C# 设计简单应用程序

|                                  |    |                        |    |
|----------------------------------|----|------------------------|----|
| 2.1 C# 语言及其开发环境.....             | 7  | 2.3 用 VS 设计控制台应用程序 ... | 12 |
| 2.1.1 C# 简介.....                 | 7  | 2.4 设计简单的 Windows 应用程序 |    |
| 2.1.2 开发环境 VisualStudio.NET..... | 8  | .....                  | 16 |
| 2.2 不用 VS 开发简单程序的例 ...           | 10 | 习题.....                | 26 |

## 第三章 简单的 Web 应用和 Web 服务

|                          |    |                          |    |
|--------------------------|----|--------------------------|----|
| 3.1 设计 HTML 网页.....      | 28 | 3.4 用 ASP.NET 设计 Web 服务的 |    |
| 3.2 设计 ASP 应用的例子.....    | 31 | 例子.....                  | 38 |
| 3.2.1 ASP 技术概貌.....      | 31 | 3.4.1 设计 Web 服务.....     | 38 |
| 3.2.2 用 ASP 设计网页的例子..... | 31 | 3.4.2 使用 Web 服务.....     | 40 |
| 3.3 设计 ASP.NET 应用程序的例子   |    | 习题.....                  | 41 |
| .....                    | 35 |                          |    |

## 第四章 C# 程序设计基础

|                     |    |                       |    |
|---------------------|----|-----------------------|----|
| 4.1 简单数据类型和表达式..... | 42 | 4.1.3.4 字符串.....      | 46 |
| 4.1.1 数据类型.....     | 42 | 4.1.4 运算符和表达式.....    | 47 |
| 4.1.2 C# 语法特征.....  | 43 | 4.1.4.1 运算符.....      | 48 |
| 4.1.3 简单数据类型.....   | 45 | 4.1.4.2 运算符的优先级.....  | 50 |
| 4.1.3.1 数值类型.....   | 45 | 4.1.4.3 类型转换.....     | 51 |
| 4.1.3.2 字符类型.....   | 46 | 4.1.4.4 常量和常量表达式..... | 54 |
| 4.1.3.3 布尔类型.....   | 46 | 4.2 流程控制.....         | 54 |



|                          |    |                     |    |
|--------------------------|----|---------------------|----|
| 4.2.1 分支结构和分支语句 .....    | 54 | 4.2.5 函数 .....      | 62 |
| 4.2.1.1 if 语句 .....      | 54 | 4.2.6 异常处理 .....    | 63 |
| 4.2.1.2 switch 语句 .....  | 55 | 4.3 复杂数据类型 .....    | 65 |
| 4.2.2 循环结构和循环语句 .....    | 56 | 4.3.1 枚举类型 .....    | 65 |
| 4.2.2.1 Do 循环 .....      | 56 | 4.3.2 结构类型 .....    | 68 |
| 4.2.2.2 While 循环 .....   | 57 | 4.3.3 数组 .....      | 70 |
| 4.2.2.3 For 循环 .....     | 58 | 4.3.3.1 一维数组 .....  | 70 |
| 4.2.3 中断循环 .....         | 59 | 4.3.3.2 多维数组 .....  | 74 |
| 4.2.3.1 有多个出口的循环 .....   | 59 | 4.3.3.3 数组的数组 ..... | 75 |
| 4.2.3.2 终止循环体的一次执行 ..... | 61 | 4.3.4 字符串处理 .....   | 78 |
| 4.2.4 foreach 循环 .....   | 61 | 习题 .....            | 84 |

## 第五章 C# 面向对象编程基础

|                            |    |                               |     |
|----------------------------|----|-------------------------------|-----|
| 5.1 类和对象的定义 .....          | 85 | 5.3.2.3 参数数组 .....            | 96  |
| 5.1.1 C# 的面向对象程序设计特征 ..... | 85 | 5.3.2.4 Main() 函数和控制台参数 ..... | 97  |
| 5.1.2 类的定义 .....           | 86 | 5.3.3 名字的作用域和名字空间 .....       | 97  |
| 5.1.3 构造函数和创建类的实例 .....    | 90 | 5.4 属性和索引符 .....              | 98  |
| 5.1.4 析构函数 .....           | 91 | 5.4.1 属性 .....                | 98  |
| 5.2 数据成员的特性 .....          | 92 | 5.4.2 索引符 .....               | 103 |
| 5.2.1 值类型和引用类型 .....       | 92 | 5.5 操作符的重载 .....              | 104 |
| 5.2.2 数据成员的访问特性 .....      | 93 | 5.6 结构和类 .....                | 106 |
| 5.2.3 静态数据成员和只读数据成员 .....  | 93 | 5.7 类库项目 .....                | 107 |
| 5.3 定义方法和调用方法 .....        | 94 | 5.7.1 增加类到项目 .....            | 107 |
| 5.3.1 方法 .....             | 94 | 5.7.2 类库项目 .....              | 108 |
| 5.3.1.1 定义方法 .....         | 94 | 5.8 预处理指令和.NET 属性 .....       | 110 |
| 5.3.1.2 静态方法 .....         | 95 | 5.8.1 预处理指令 .....             | 110 |
| 5.3.2 调用方法 .....           | 95 | 5.8.2 .NET 属性 .....           | 110 |
| 5.3.2.1 函数的参数和参数传递 .....   | 95 | 习题 .....                      | 111 |
| 5.3.2.2 输入和输出参数 .....      | 96 |                               |     |

## 第六章 C# 面向对象编程进一步学习

|                                   |     |                        |     |
|-----------------------------------|-----|------------------------|-----|
| 6.1 继承和继承的应用 .....                | 112 | 6.2.1 虚函数和多态 .....     | 119 |
| 6.1.1 派生类的定义 .....                | 112 | 6.2.2 抽象函数和抽象类 .....   | 121 |
| 6.1.2 派生类的构造函数 .....              | 118 | 6.2.3 密封类和密封方法 .....   | 122 |
| 6.1.3 protect 和 private 修饰符 ..... | 119 | 6.2.4 object 类 .....   | 122 |
| 6.2 虚函数、多态和抽象类 .....              | 119 | 6.2.5 应用继承和多态的例子 ..... | 122 |

|                                |     |                                     |     |
|--------------------------------|-----|-------------------------------------|-----|
| 6.3 接口.....                    | 130 | 6.6.2.3 读字节流文件 .....                | 150 |
| 6.3.1 接口的定义 .....              | 131 | 6.6.2.4 向字节流文件写 .....               | 151 |
| 6.3.2 实现接口 .....               | 132 | 6.6.3 用 StreamWriter 和 StreamReader |     |
| 6.3.3 接口的应用 .....              | 133 | 类读写文件 .....                         | 152 |
| 6.4 委托和事件.....                 | 134 | 6.6.4 异步存取文件 .....                  | 154 |
| 6.4.1 委托 .....                 | 134 | 6.7 几个常用的基类.....                    | 157 |
| 6.4.1.1 为什么用委托 .....           | 134 | 6.7.1 字符串处理 .....                   | 157 |
| 6.4.1.2 委托的定义 .....            | 135 | 6.7.1.1 string 类 .....              | 157 |
| 6.4.2 事件 .....                 | 137 | 6.7.1.2 StringBuilder 类的应用 .....    | 158 |
| 6.4.2.1 事件的概念 .....            | 137 | 6.7.1.3 格式化字符串 .....                | 158 |
| 6.4.2.2 使用事件的例子 .....          | 139 | 6.7.2 数组列表 .....                    | 159 |
| 6.4.2.3 定义事件的实例 .....          | 140 | 6.7.3 集合 .....                      | 161 |
| 6.4.3 C# 设计的应用程序的结构 .....      | 144 | 6.7.4 词典 .....                      | 163 |
| 6.5 类型转换.....                  | 144 | 6.8 异常处理.....                       | 164 |
| 6.5.1 封箱和拆箱 .....              | 144 | 6.8.1 异常类 .....                     | 164 |
| 6.5.2 is 运算符和 as 运算符 .....     | 145 | 6.8.2 嵌套的 try 块 .....               | 166 |
| 6.5.3 深度复制 .....               | 145 | 6.9 多线程.....                        | 171 |
| 6.5.4 定义转换函数 .....             | 147 | 6.9.1 什么是多线程 .....                  | 171 |
| 6.6 I/O 和文件操作 .....            | 148 | 6.9.2 C# 多线程设计 .....                | 172 |
| 6.6.1 控制台输入/输出 .....           | 148 | 6.9.2.1 类 Thread .....              | 172 |
| 6.6.2 用 FileStream 类读写文件 ..... | 149 | 6.9.2.2 多线程程序的例子 .....              | 173 |
| 6.6.2.1 FileStream 类对象创建 ..... | 149 | 6.9.3 线程同步 .....                    | 174 |
| 6.6.2.2 文件当前指针 .....           | 150 | 习题 .....                            | 174 |

## 第七章 用 WindowsForms 设计 Windows 应用

|                       |     |                            |     |
|-----------------------|-----|----------------------------|-----|
| 7.1 事件驱动程序的特点.....    | 177 | 7.7 使用对话框.....             | 217 |
| 7.2 简单的用户界面设计.....    | 180 | 7.8 GDI+绘图初步 .....         | 220 |
| 7.3 在窗口上输出.....       | 189 | 7.8.1 坐标和定位的基本结构 .....     | 221 |
| 7.3.1 输出文本 .....      | 190 | 7.8.2 两个参数类 .....          | 221 |
| 7.3.2 输出图形 .....      | 191 | 7.8.3 颜色、画刷和画笔 .....       | 223 |
| 7.3.3 输出图像 .....      | 192 | 7.8.4 字体和字体系列 .....        | 224 |
| 7.4 接收窗口上的输入.....     | 198 | 7.8.5 应用 GDI 绘图功能的实例 ..... | 225 |
| 7.5 控件的使用和数据输入.....   | 203 | 7.9 实例.....                | 231 |
| 7.6 菜单、工具栏和 MDI ..... | 211 | 7.9.1 图形编辑的实例 .....        | 231 |
| 7.6.1 菜单 .....        | 211 | 7.9.2 交互式图形编辑的实例 .....     | 236 |
| 7.6.2 工具栏 .....       | 214 | 习题 .....                   | 244 |
| 7.6.3 SDI 和 MDI ..... | 215 |                            |     |

## 第八章 应用 ADO.NET 存取数据库的

|                                     |     |                                           |     |
|-------------------------------------|-----|-------------------------------------------|-----|
| 8.1 应用软件的结构和数据库接口 .....             | 248 | 8.7.5 用 DataSet 和 DataAdapter 更新数据 .....  | 276 |
| 8.1.1 两层和三层结构 .....                 | 248 | 8.7.5.1 DataReader 更新数据库的一些技术问题 .....     | 276 |
| 8.1.2 数据库和数据库接口 .....               | 249 | 8.7.5.2 更新数据库实例 .....                     | 277 |
| 8.2 简单的存取数据库的应用实例 .....             | 252 | 8.7.5.3 插入记录 .....                        | 280 |
| 8.3 ADO 的主要类 .....                  | 255 | 8.7.5.4 修改记录 .....                        | 281 |
| 8.3.1 数据提供者的类 .....                 | 256 | 8.7.5.5 删除记录 .....                        | 281 |
| 8.3.2 用户对象类 .....                   | 257 | 8.7.5.6 输出到 XML 文档 .....                  | 282 |
| 8.4 连接数据库系统 .....                   | 257 | 8.7.5.7 完整的代码 .....                       | 282 |
| 8.4.1 连接数据库系统 .....                 | 257 | 8.7.5.8 应用 SqlCommandBuilder 的例 .....     | 286 |
| 8.4.2 事务的独立性 .....                  | 259 | 8.8 数据绑定 .....                            | 288 |
| 8.5 用命令对象存取数据库 .....                | 260 | 8.8.1 DataGrid 控件数据绑定的例子 .....            | 288 |
| 8.5.1 命令对象 .....                    | 260 | 8.8.2 绑定到数组 .....                         | 291 |
| 8.5.2 用命令对象更新数据库 .....              | 261 | 8.8.3 绑定到 DataTable 和 DataView .....      | 291 |
| 8.5.2.1 更新记录 .....                  | 261 | 8.8.4 绑定到 DataSet 和 DataViewManager ..... | 293 |
| 8.5.2.2 插入记录 .....                  | 263 | 8.8.5 简单绑定 .....                          | 294 |
| 8.5.2.3 删除记录 .....                  | 264 | 8.9 用 VS 创建数据库存取程序的方法 .....               | 294 |
| 8.5.3 用命令对象做简单的查询 .....             | 264 | 8.9.1 建立数据库连接 .....                       | 294 |
| 8.6 用 DataReader 读出数据 .....         | 265 | 8.9.2 操纵数据库 .....                         | 296 |
| 8.7 DataSet 和 DataAdapter 的应用 ..... | 269 | 8.9.3 选择数据 .....                          | 296 |
| 8.7.1 数据表 DataTable .....           | 269 | 8.9.4 定义 DataSet 对象 .....                 | 298 |
| 8.7.1.1 列 DataColumn .....          | 269 | 8.9.5 定义 DataGrid .....                   | 298 |
| 8.7.1.2 数据行 DataRow .....           | 271 | 8.9.6 执行程序 .....                          | 299 |
| 8.7.2 数据联系 DataRelation .....       | 272 | 习题 .....                                  | 301 |
| 8.7.3 数据约束 .....                    | 273 |                                           |     |
| 8.7.4 生成 DataTable 的数据模式的方法 .....   | 275 |                                           |     |
| 8.7.4.1 运行时生成模式 .....               | 275 |                                           |     |
| 8.7.4.2 代码生成模式 .....                | 275 |                                           |     |

## 第九章 Windows 应用程序的部署

|                    |     |                                    |     |
|--------------------|-----|------------------------------------|-----|
| 9.1 VS 的部署项目 ..... | 303 | 9.1.2 Microsoft Windows 安装程序 ..... | 304 |
| 9.1.1 部署项目 .....   | 303 | 9.2 创建安装程序软件包 .....                | 305 |



## 第十一章 控件和组件设计

|                                  |     |                                               |     |
|----------------------------------|-----|-----------------------------------------------|-----|
| 11.1 设计 Windows 应用程序的用户控件 .....  | 386 | 11.5.1 组件 .....                               | 411 |
| 11.1.1 创建用户控件 .....              | 386 | 11.5.2 .NET 装配件 .....                         | 412 |
| 11.1.2 添加属性 .....                | 387 | 11.5.2.1 .NET 装配件和.NETFrame-<br>work 类库 ..... | 412 |
| 11.1.3 添加事件处理程序 .....            | 389 | 11.5.2.2 装配件的结构 .....                         | 412 |
| 11.1.4 调试用户控件 .....              | 390 | 11.5.3 创建装配件 .....                            | 413 |
| 11.1.5 扩展 LabelTextbox 控件 .....  | 391 | 11.5.3.1 创建装配件 .....                          | 413 |
| 11.1.5.1 添加更多属性 .....            | 391 | 11.5.3.2 使用装配件 .....                          | 414 |
| 11.1.5.2 添加更多的事件处理程序<br>.....    | 392 | 11.5.4 查看装配件 .....                            | 416 |
| 11.2 为 Windows 应用程序定制控件<br>..... | 394 | 11.5.4.1 将 ILDASM 作为外部工具<br>添加到 VS 中 .....    | 416 |
| 11.2.1 定制控件 .....                | 394 | 11.5.4.2 使用 ildasm .....                      | 416 |
| 11.2.2 部署和使用定制控件 .....           | 398 | 11.5.4.3 清单 .....                             | 418 |
| 11.3 设计 Web 应用程序的用户控件<br>.....   | 403 | 11.5.5 装配件的.NET 属性 .....                      | 419 |
| 11.4 为 Web 应用程序定制服务器控<br>件 ..... | 408 | 11.5.5.1 AssemblyInfo.cs .....                | 420 |
| 11.5 .NET 装配件设计 .....            | 411 | 11.5.5.2 版本属性 .....                           | 421 |
|                                  |     | 11.5.5.3 版本兼容性 .....                          | 421 |
|                                  |     | 11.5.5.4 反射 .....                             | 422 |
|                                  |     | 11.5.5.5 私有和共享装配件 .....                       | 424 |
|                                  |     | 习题 .....                                      | 425 |

## 第十二章 用 ADO.NET 处理 XML

|                                    |     |                                          |     |
|------------------------------------|-----|------------------------------------------|-----|
| 12.1 XML 基础 .....                  | 426 | 12.2 Dataset 数据组织 .....                  | 435 |
| 12.1.1 什么是 XML .....               | 426 | 12.3 Dataset 读和写 XML 文档<br>.....         | 436 |
| 12.1.2 XML 的模式定义 .....             | 427 | 12.3.1 将 XML 文档读入 DataSet ...            | 436 |
| 12.1.2.1 文档类型定义(DTD) .....         | 428 | 12.3.2 从 DataSet 中写 XML 文档 ...           | 438 |
| 12.1.2.2 XMLSchema .....           | 430 | 12.4 得到 XML 数据模式的方法<br>.....             | 441 |
| 12.1.3 XML 文档的应用程序接口 ...           | 431 | 12.4.1 通过 ReadXml 使用模式 .....             | 441 |
| 12.1.3.1 文档对象模型(DOM) .....         | 431 | 12.4.2 利用单独的模式 .....                     | 444 |
| 12.1.3.2 简单 API(SAX) .....         | 432 | 12.5 XMLDOCUMENT 类 .....                 | 445 |
| 12.1.4 XSL 和 Xpath .....           | 432 | 12.5.1 用 XmlDocument 类装载 XML<br>文档 ..... | 446 |
| 12.1.4.1 扩展样式表单转换语言(XSL)<br>.....  | 432 | 12.5.2 用 XmlDocument 类保存 XML<br>.....    | 446 |
| 12.1.4.2 Xpath .....               | 433 |                                          |     |
| 12.1.5 .NET 中与 XML 相关的类简介<br>..... | 434 |                                          |     |

|                                         |     |
|-----------------------------------------|-----|
| 12.5.3 用 XmlDocument 类创建节点 .....        | 448 |
| 12.5.4 用 XmlDocument 在 XML 文档插入节点 ..... | 449 |

|                              |     |
|------------------------------|-----|
| 12.6 XMLDATADOCUMENT 类 ..... | 450 |
| 习题 .....                     | 454 |

## 第十三章 Web 应用程序的配置和安全

|                                  |     |
|----------------------------------|-----|
| 13.1 Web 应用程序和 Global.asax ..... | 456 |
| 13.1.1 Global.asax 文件的指令 .....   | 456 |
| 13.1.2 应用程序状态管理 .....            | 456 |
| 13.1.2.1 管理用户会话状态 .....          | 457 |
| 13.1.2.2 管理应用程序状态 .....          | 457 |
| 13.1.2.3 管理应用程序瞬时状态 .....        | 457 |
| 13.1.2 Application(应用程序)事件 ..... | 458 |
| 13.2 Web 应用程序的配置 .....           | 460 |
| 13.2.1 配置文件的格式 .....             | 461 |

|                                    |     |
|------------------------------------|-----|
| 13.2.2 配置项处理程序 .....               | 462 |
| 13.2.3 配置设置 .....                  | 462 |
| 13.3 Web 应用程序的安全性 .....            | 464 |
| 13.3.1 IIS 和 Windows 提供的安全措施 ..... | 465 |
| 13.3.2 ASP.NET 提供的安全措施 .....       | 465 |
| 13.3.2.1 应用 Windows 内置验证 .....     | 466 |
| 13.3.2.2 应用基于表单的验证 .....           | 466 |
| 习题 .....                           | 469 |

## 第十四章 Web 服务基础

|                                            |     |
|--------------------------------------------|-----|
| 14.1 Web 服务 .....                          | 470 |
| 14.1.1 什么是 Web 服务 .....                    | 470 |
| 14.1.2 Web 服务模型和所用技术 .....                 | 471 |
| 14.1.2.1 Web 服务模型 .....                    | 471 |
| 14.1.2.2 实现 Web 服务的主要技术 .....              | 471 |
| 14.2 SOAP .....                            | 472 |
| 14.2.1 SOAP 信封(SOAPenvelop) .....          | 473 |
| 14.2.2 SOAP 编码规则(SOAPencoding-rules) ..... | 475 |
| 14.2.2.1 默认数据类型 .....                      | 476 |
| 14.2.2.2 复杂数据类型 .....                      | 476 |
| 14.2.3 SOAPRPC 表示和 SOAP 绑定 .....           | 478 |
| 14.3 Web 服务描述 .....                        | 479 |
| 14.3.1 WSDL(WebServiceDescription-         |     |

|                                        |     |
|----------------------------------------|-----|
| Language) .....                        | 479 |
| 14.3.2 生成 Web 服务的 WSDL 文件 .....        | 481 |
| 14.3.3 对 WSDL 文件的解释 .....              | 486 |
| 14.3.3.1 定义部分 .....                    | 486 |
| 14.3.3.2 类型部分 .....                    | 487 |
| 14.3.3.3 消息(Message)部分 .....           | 487 |
| 14.3.3.4 端口类型部分 .....                  | 487 |
| 14.3.3.5 绑定部分 .....                    | 487 |
| 14.3.3.6 服务部分 .....                    | 488 |
| 14.4 Web 服务发现机制 .....                  | 488 |
| 14.4.1 UDDI 简介 .....                   | 488 |
| 14.4.2 UDDI 提供的 Web 服务 .....           | 489 |
| 14.4.3 使用 ASP.NETSDK 搜索 UDDI 注册库 ..... | 490 |
| 习题 .....                               | 491 |

## 第十五章 用 ASP.NET 设计的 Web 服务

|                                    |     |                                         |     |
|------------------------------------|-----|-----------------------------------------|-----|
| 15.1 直接创建 Web 服务 .....             | 492 | 15.2.2.2 .NET 属性 WebService .....       | 503 |
| 15.1.1 存取应用程序设置的类 .....            | 492 | 15.2.2.3 .NET 属性 WebMethod .....        | 505 |
| 15.1.2 将 C# 应用程序类设置为 Web 服务 .....  | 494 | 15.2.3 继承 WebService 类实现 Web 服务讨论 ..... | 509 |
| 15.1.3 部署 Web 服务 .....             | 495 | 15.3 调用 Web 服务的方法 .....                 | 510 |
| 15.1.4 测试 Web 服务 .....             | 495 | 15.3.1 用 WSDL 创建代理类 .....               | 511 |
| 15.1.5 用装配件建立 Web 服务 .....         | 498 | 15.3.2 用 VS 创建代理类 .....                 | 515 |
| 15.2 用 VS 创建 Web 服务 .....          | 499 | 15.3.3 同步调用和异步调用 .....                  | 517 |
| 15.2.1 VS 建立 Web 服务的方法 .....       | 499 | 习题 .....                                | 518 |
| 15.2.2 Web 服务常用页面指令和 .NET 属性 ..... | 502 |                                         |     |
| 15.2.2.1 页面指令 WebService ...       | 502 |                                         |     |

# 第一章

## .NET 的结构和主要特征

### 1.1 微软的软件开发平台的发展

.NET 是微软公司新的计算平台。计算平台也可以理解为一种软件开发的环境，它由计算机操作系统和一批支持应用程序开发的实用程序和工具软件组成，全面地支持软件设计开发人员的工作。

微软最早的平台是 MS-DOS，它由 MS-DOS 操作系统和上面的命令解释程序，若干语言编译和目标运行库，若干编辑调试等工具软件组成。这个计算平台比较简单，是字符界面，使用不方便，在上面开发软件的效率不高。

后来，微软开发了 Windows 系列的操作系统，随着软件工程技术的发展，又开发出了许多支持新的软件开发技术的工具，组成了新的环境，这就是 Windows 平台。Windows 是图形用户界面的多用户程序环境。在 Windows 平台除能开发在字符界面和单用户环境运行的应用软件外，还能开发在图形用户界面（简称 GUI）和多用户环境运行的软件。微软在 Windows 上推出了许多工具，制定了支持 Windows 上应用开发接口的标准 Win32API。应用程序要调用 Windows 操作系统功能及其扩展，比如多媒体等功能等都通过标准的 Win32API 完成。Win32API 加上一些工具，比如资源管理、编辑、编译和调试工具等，组成了开发 Windows 应用程序平台的基础。随选择的语言为 Visual Basic (VB)，Visual C (VC)，Visual Foxpro (VF) 等不同，有不同的开发环境，不同的设计开发工具和不同的设计开发方法。VB 和 VF 都有所见即所得的用户界面设计工具，大大减少了设计用户界面的工作量（这部分软件的代码在普通 Mis 等一大类应用的代码中占 50%~70%甚至更多比例）。在 VC 中，微软又推出了 MFC（基础类库），将 Win32API 的功能包装成类的层次，以支持面向对象编程。在微软的这些软件开发环境中，VC 功能最强大，可方便地直接调用 Win32API，但缺乏所见即所得的界面设计工具，使用不够方便。VB 和 VF 使用方便，用得很广，但由于受各自语言功能的局限，不适合设计较大型、要求较高的软件。

20 世纪 90 年代互联网的应用迅速推广，为适应开发互联网上的应用程序（又叫做 Web 应用程序）的需要，微软推出了新的平台，称为 Active 平台（Active Platform）。这个平台企图将开发 Windows 应用程序的平台和开发互联网上应用程序的平台统一为一个平台（实际上并没有完全统一），平台关键的技术是 Active X 控件（服务器端称为组件）和 Active X 脚本。Active X 控件和服务端运行的组件是用 VC 或 VB 遵从一定标准设计的模块，它



们和 Active X 脚本都可以嵌入网页在 Web 浏览器执行或在 Web 服务器上执行。服务器端的组件和脚本技术在开发 Web 应用程序时特别有用,通常称为 ASP 技术。同时,微软还将他支持的几种主要的编程语言:VC,VB,VF 和 VJ (Visual J++) 的平台绑在一起,加上 ASP 技术设计互联网上应用程序的工具 Visual InterDev,称为 Visual Studio (简称 VS)。同时,微软先后又加入了对开发分布式处理程序的支持和事务管理等功能,制订了 COM 和 DCOM 标准,将支持分布式的 Windows 应用程序和 Web 应用程序的平台统称为 COM+。从这里可以看出,随着技术进步,微软的软件平台越来越复杂,越来越难学习和掌握,越来越不好用。以 COM+ 平台为例,应用开发人员常需要掌握多个工具,在不同的工具中切换。比如,当设计 Active X 控件时,需要用 VC 的 ATL 工具,设计 Windows 应用时最好用 VB,或 MFC,设计 Web 应用时需用 Visual InterDev 和 ASP,仅这一点,造成了很大不便。因此,迫切需要一种简单而专业化的语言,一种简单而专业化的平台,便于软件开发人员能轻松地编写优秀的软件,这就是 C# 和 .NET。

.NET 框架是微软继推出 COM+ 平台后推出的最新的平台。用 .NET 框架取代 COM+ 后,原 COM+ 下开发的应用仍可运行,在这个意义上似乎继承了 COM+,但实质上 .NET 框架技术和 COM+ 完全不同,是一次十分彻底地变化,是全新的技术,比较 COM+ 有很大的改进和很多优势。

## 1.2 .NET 框架的主要特征

.NET 框架同样适用于开发 Web 应用程序和 Windows 应用程序。和 COM+ 仅在形式上将开发这两类不同应用的不同环境的工具绑在一起不同,.NET 框架中可用同一种语言(如 C#),用同一套支持类库,即 .NET 框架类库开发 Web 应用和 Windows 应用这两类不同应用。而且,.NET 框架中可用的任何语言,无论 C#,VB 和 Jscript 都用同一套支持类库。

Web 应用环境的一大特征是高度分布式和用户数量伸缩很大。.NET 框架简化了在高度分布式的 Web 环境中的应用程序开发,提供一个一致的分布式的面向对象的编程环境,设计出的对象代码可在本地存储和执行,也可在本地执行但代码在 Internet 上分布,或是在本地存储代码在远程执行。

针对互联网安全的要求,.NET 框架提供一个保证安全执行代码的 .NET 框架的执行环境。和 COM+ 相比,.NET 框架解决了 DLL 版本冲突引起的问题,提供了版本冲突最小化的代码执行环境。

COM+ 中脚本解释执行引起了执行效率低的问题。相比,.NET 框架中嵌入网页的代码是编译后执行,消除了脚本解释环境引起的性能问题。

最吸引开发人员的特点是 .NET 框架提供的 Web Forms 等类(Class)使软件开发人员能像开发 Windows 应用一样开发 Web 应用,大大提高了开发 Web 应用的工作效率,还使多数开发人员在开发 Windows 应用中积累的经验在面对类型大不相同 Web 应用时仍有用。

.NET 框架有两个主要组成部分:公共语言运行环境(Common Language Runtime)和 .NET 框架类库(基类库),其上支持 ADO.NET,ASP.NET,XML 和 Windows Form 等,可采用符合公共语言规范(CLS)的任何语言进行程序设计。