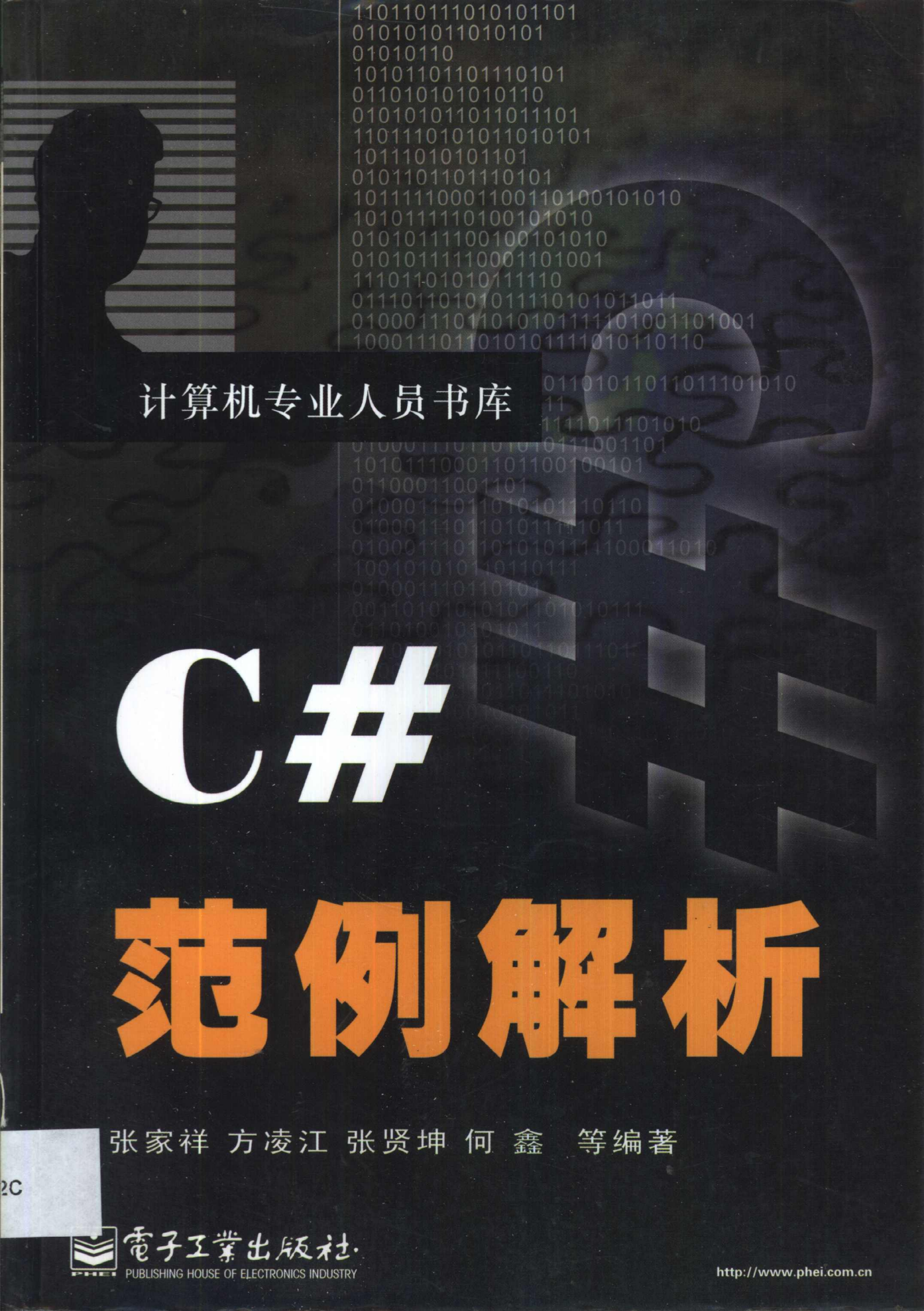




110110111010101101
010101011010101
01010110
10101101101110101
011010101010110
01010101101101101
1101110101011010101
10111010101101
0101101101110101
10111110001100110100101010
1010111110100101010
01010111100100101010
01010111110001101001
11101101010111110
01110110101011110101011011
0100011101101010111101001101001
1000111011010101111010110110

计算机专业人员书库

C#

范例解析

张家祥 方凌江 张贤坤 何鑫 等编著



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>

2C

525

TP312C

2329

计算机专业人员书库

C # 范例解析

张家祥 方凌江 张贤坤 何 鑫 等编著

電子工業出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书概括介绍了 .NET 的来龙去脉和特点,较详细地叙述了 C# 语言的语法规则,包括 C# 的特点、C# 的数据类型和变量、名字空间、接口、类、结构、不安全代码、语句、异常与异常处理及 C# 与 COM 的互操作。重点讲解如何利用 C# 进行实例编程,其中包括 C# 编程的一些小技巧,用 C# 创建 Windows 应用程序,C# 开发网络程序,C# 与 XML 结合编程及 C# 对注册表的控制编程等。另外,本书还给出了相关内容的附录,包括专业英语名词、C# 编译选项和 C# 编译错误。

本书条理清楚地讲解了在 .NET 上的 C# 编程,使理论和实例相结合,是计算机开发人员和有关专业的大专院校师生学习 C# 的一本很有参考价值的书。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究。

图书在版编目(CIP)数据

C# 范例解析/张家祥等编著. —北京:电子工业出版社,2002.7

(计算机专业人员书库)

ISBN 7-5053-7413-3

I. C… II. 张… III. C 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2002)第 036908 号

责任编辑:朱沐红 特约编辑:蔡祖瑛

印 刷:北京人卫印刷厂

出版发行:电子工业出版社 <http://www.phei.com.cn>

北京市海淀区万寿路 173 信箱 邮编 100036

经 销:各地新华书店

开 本:787×1092 1/16 印张:20.25 字数:515 千字

版 次:2002 年 7 月第 1 版 2002 年 7 月第 1 次印刷

印 数:6 000 册 定价:29.00 元

凡购买电子工业出版社的图书,如有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系。
联系电话:(010)68279077

前 言

曾几何时, IT 业步入网络时代, Internet 已经深入人心。自从 Sun 公司于 1995 年正式推出面向对象的开发语言 Java, 并提出跨平台、跨语言的概念后, Java 就逐渐成为企业级应用系统开发的首选工具, 而且使得越来越多的基于 C/C++ 的应用开发人员转向了从事基于 Java 的应用开发。Java 的先进思想使其在软件开发领域中大有山雨欲来风满楼之势。

Microsoft 公司另辟蹊径, 决定推出其进军互联网的计划 .NET 和该计划中旗帜性的开发语言 C#。

.NET 是一项非常庞大的计划, 也是该公司今后几年发展的战略核心。“在任何时间、任何地点, 采用相应的设备以获取所需的信息”的梦想并非一朝一夕就能实现的。Visual Studio.NET 是 Microsoft.NET 的技术开发平台, 其重要性可见端倪, 而 C# 就集成在 Visual Studio.NET 中。C# 这一新的开发语言是由 Anders Hejlsberg 主持开发的, Anders Hejlsberg 曾是 Borland 公司的首席研发设计师, 他在 Borland 任职期间开发了著名的 Turbo Pascal 语言, 并在其基础上开发了面向对象的应用开发工具 Delphi。Delphi 目前仍是 Borland 公司的最重要的旗帜性产品, 并已成为广大开发人员最喜欢的 RAD 应用开发工具之一。

Microsoft 公司对 C# 的定义是“C# 是一种类型安全的、现代的、简单的并由 C 和 C++ 衍生出来的面向对象的编程语言, 它牢牢根植于 C 和 C++ 语言之上, 并可立即被 C 和 C++ 的使用者所熟悉。C# 的目的就是综合 Visual BASIC 的高生产率和 C++ 的行动力。”这个定义是恰如其分的。

当然, 脱离了 .NET 平台而单纯去讲 C# 是没有太大意义的, 所以必须先了解一些 .NET 平台开发的基础知识, 才能更深入了解 C# 的特性。

.NET 框架的核心是基于 XML 的跨平台跨语言的信息交换, 基于此种平台将十分有助于开发因特网分布式应用程序。

基于 .NET 的 Web 应用开发主要使用 ASP.NET。ASP.NET 可以使用任何一种 .NET 支持的语言编写, 而其中 C# 可以提供脚本语言所无法提供的强大功能。使用 C# 编写 ASP.NET 的 Web 应用程序在未来将成为 Windows 平台上 Web 开发的最佳选择。

本书编写的内容是循序渐进的, 共分 4 部分。

第一部分包括第 1 章到第 4 章, 是关于 .NET 和 C# 的初步介绍, 概括介绍 .NET 和 .NET 框架、通用语言运行时、C# 语言的功能和特点, 针对简单的 C# 程序的编辑, 编译和运行的描述以及其与 Java、C++ 程序的比较, 使读者对 C# 程序有一个初步的了解。

第二部分包括第 5 章到第 10 章, 介绍 C# 语言的语法规则, 其中包括数据类型、变量、数组、枚举、名字空间、代表、接口、类、结构、不安全代码、异常、语句及 C# 与 COM 内部互操作。读者学习这一部分后, 可掌握 C# 语言的基础, 为进一步编写 C# 程序做准备。

第三部分包括第 11 章到第 17 章, 介绍 .NET 上的 C# 编程实例, 是本书的重点。其中第 11 章列举了 C# 应用程序的一些技巧; 第 12 章介绍用 C# 创建桌面应用程序; 第 13 章介绍用 C# 开发 WinForm 程序; 第 14 章介绍 C# 与 XML 结合编程; 第 15 章介绍在 ASP.NET 上的 C# 编程; 第 16 章介绍 C# 的 Web 数据库编程; 第 17 章介绍 C# 对 Windows 注册表的

编程。读者通过阅读和上机实践，可以更加深入地掌握 C# 编程。

第四部分是附录，包括附录 A 到附录 C。其中附录 A 是专业英语名词；附录 B 是 C# 编译器选项；附录 C 是 C# 编译错误。这一部分在书末为读者提供便于查阅的资料和工具。

全书在讲述 C# 基本语法规则的同时，还提供了大量的程序实例，相信读者阅读本书会有所收获。读者可以结合本书的实例来学习 C# 的基本知识，并在此基础上设计自己的 C# 程序。

本书的编写人员有张家祥、方凌江、毛全胜、张贤坤、何鑫、刘岩峰、卜先锦、张凤林、魏永森、蔡益朝、张涛、汪文元、李慧、陈光、冯静、张渺和任花梅等，由张一鸣统一定稿。感谢在此书写作过程中给予支持与帮助的各位互联网上的朋友，感谢博士生导师罗雪山教授的支持与关心，感谢电子工业出版社的编辑和工作人员的辛勤劳动。

最后，由于 C# 语言是新兴语言，加之作者水平所限，书中的错误还望广大读者批评指正。

编 者
2002 年 3 月

目 录

第 1 章 令人耳目一新的 .NET	(1)
1.1 认识 .NET	(1)
1.2 .NET 的技术特征和新特性	(3)
1.2.1 .NET 的技术特征	(3)
1.2.2 .NET 的新特性	(4)
1.3 .NET 的组成	(5)
1.4 .NET 与 J2EE 的比较	(7)
1.5 小结	(8)
第 2 章 .NET 框架与 CLR	(9)
2.1 .NET 框架	(9)
2.1.1 .NET 框架的组成	(9)
2.1.2 .NET 中的开发工具和编程语言	(11)
2.1.3 .NET 框架的重大改进	(11)
2.2 通用语言运行时	(12)
2.2.1 通用语言运行时的概念与特点	(12)
2.2.2 通用语言运行时的功能	(13)
2.2.3 通用语言运行时的组成与原理分析	(14)
2.2.4 CLR 与 COM 的比较	(16)
2.3 小结	(17)
第 3 章 C# 语言	(18)
3.1 C# 产生的背景	(18)
3.2 在 .NET 上使用 C# 语言的原因	(18)
3.3 C# 的特点	(21)
3.4 C# 与 Java 比较	(23)
3.4.1 C# 与 Java 的相同之处	(24)
3.4.2 C# 与 Java 的区别	(24)
3.5 小结	(25)
第 4 章 初步体验 C# 编程	(26)
4.1 安装 C# 编译器	(26)
4.1.1 在 Windows 2000 环境中安装 .NET	(26)
4.1.2 在 Windows 98 下安装 C# 编译器	(26)
4.1.3 编译器 csc.exe 的常用命令	(27)
4.2 选择 C# 程序编辑器	(27)
4.3 用 CSC 编译器编译一个 C# 程序	(28)
4.3.1 编辑 C# 程序	(28)

4.3.2	编译运行此程序	(29)
4.3.3	分析语法	(29)
4.3.4	输入和输出	(30)
4.3.5	添加注释	(31)
4.4	在 .NET IDE 中创建简单的 C# 程序	(32)
4.4.1	在 IDE 中创建一个新的工程	(32)
4.4.2	创建 hello.cs 源文件	(33)
4.4.3	生成可执行文件	(35)
4.4.4	运行程序	(35)
4.5	小结	(36)
第 5 章	数据类型与变量	(37)
5.1	数据类型	(37)
5.1.1	值类型	(37)
5.1.2	引用类型	(44)
5.1.3	加框和解框	(49)
5.1.4	统一系统类型	(51)
5.2	变量	(51)
5.2.1	变量的种类	(52)
5.2.2	默认值	(53)
5.2.3	明确的赋值变量	(53)
5.2.4	变量引用	(56)
5.3	小结	(56)
第 6 章	名字空间与接口	(57)
6.1	名字空间	(57)
6.1.1	编译单元	(57)
6.1.2	名字空间声明	(57)
6.1.3	using 指示符	(58)
6.1.4	名字空间成员	(63)
6.2	接口	(63)
6.2.1	接口声明	(63)
6.2.2	接口成员	(64)
6.2.3	完全资格接口成员名字	(68)
6.2.4	接口执行	(68)
6.3	小结	(78)
第 7 章	类与结构	(79)
7.1	类声明	(79)
7.1.1	类修饰符	(79)
7.1.2	类基本描述	(80)
7.1.3	类主体	(81)
7.2	类成员	(82)

7.2.1	继承	(82)
7.2.2	new 修饰符	(83)
7.2.3	访问修饰符	(83)
7.2.4	代表类型	(85)
7.2.5	静态和实例成员	(85)
7.3	常量	(86)
7.4	域	(87)
7.4.1	静态域和实例域	(88)
7.4.2	只读域	(88)
7.4.3	域初始化	(89)
7.4.4	变量初始化	(90)
7.5	转换操作	(91)
7.6	方法	(92)
7.6.1	方法的参数	(92)
7.6.2	静态方法和实例方法	(97)
7.6.3	virtual 方法	(97)
7.6.4	override 方法	(99)
7.6.5	sealed 方法	(101)
7.6.6	abstract 方法	(102)
7.6.7	extern 方法	(103)
7.6.8	方法体	(104)
7.7	属性	(105)
7.7.1	静态属性	(105)
7.7.2	访问标志	(105)
7.8	事件	(107)
7.9	索引	(108)
7.10	实例构造	(110)
7.10.1	构造初始化	(110)
7.10.2	实例变量初始化	(110)
7.10.3	构造执行	(111)
7.10.4	默认构造器	(113)
7.10.5	私有构造器	(113)
7.10.6	析构	(114)
7.11	静态构造器	(114)
7.12	结构	(117)
7.12.1	结构声明	(117)
7.12.2	结构成员	(117)
7.12.3	类和结构的区别	(117)
7.12.4	结构例子	(120)
7.13	小结	(123)

第 8 章 不安全代码与异常	(125)
8.1 不安全内容	(125)
8.2 指针类型	(127)
8.3 固定的和可移动的变量	(128)
8.4 指针转换	(128)
8.5 在表达式中的指针	(129)
8.5.1 指针指向	(129)
8.5.2 指针成员访问	(129)
8.5.3 指针元素访问	(130)
8.5.4 地址操作	(131)
8.5.5 指针增加和减少	(131)
8.5.6 指针算术运算	(131)
8.5.7 指针比较	(132)
8.5.8 sizeof 操作符	(132)
8.6 fixed 语句	(133)
8.7 堆栈分配	(135)
8.8 动态存储定位	(136)
8.9 异常及异常处理	(138)
8.9.1 产生异常的原因	(138)
8.9.2 System.Exception 类	(138)
8.9.3 普通的异常类	(138)
8.10 小结	(139)
第 9 章 语句	(140)
9.1 using 语句	(140)
9.2 结束点和可达性	(141)
9.3 语句块	(143)
9.4 空语句	(143)
9.5 标签语句	(144)
9.6 声明语句	(144)
9.6.1 局部变量声明	(144)
9.6.2 局部常量声明	(145)
9.7 表达式语句	(145)
9.8 选择语句	(145)
9.8.1 if 语句	(145)
9.8.2 switch 语句	(146)
9.9 循环语句	(149)
9.9.1 for 语句	(149)
9.9.2 foreach 语句	(150)
9.9.3 while 语句	(151)
9.9.4 do 语句	(152)

9.10 跳转语句	(153)
9.10.1 break 语句	(154)
9.10.2 continue 语句	(154)
9.10.3 goto 语句	(155)
9.10.4 return 语句	(156)
9.10.5 throw 语句	(156)
9.11 try 语句	(157)
9.12 checked 和 unchecked 语句	(159)
9.12.1 对编译器设置溢出校验	(160)
9.12.2 语句溢出校验	(160)
9.13 lock 语句	(161)
9.14 小结	(162)
第 10 章 C# 与 COM 互操作	(163)
10.1 属性	(163)
10.1.1 ComImport 属性	(163)
10.1.2 StructOffset 属性	(164)
10.1.3 ComRegisterFunction 属性	(164)
10.1.4 ComSourceInterfaces 属性	(164)
10.1.5 ComUnregisterFunction 属性	(165)
10.1.6 ComVisible 属性	(165)
10.1.7 DispId 属性	(165)
10.1.8 DllImport 属性	(166)
10.1.9 FieldOffset 属性	(167)
10.1.10 Guid 属性	(167)
10.1.11 HasDefaultInterface 属性	(167)
10.1.12 ImportedFromTypeLib 属性	(168)
10.1.13 In 和 Out 属性	(168)
10.1.14 IndexerName 属性	(169)
10.1.15 InterfaceType 属性	(169)
10.1.16 MarshalAs 属性	(169)
10.1.17 NoIDispatch 属性	(170)
10.1.18 PreserveSig 属性	(170)
10.1.19 StructLayout 属性	(171)
10.1.20 TypeLibFunc 属性	(171)
10.1.21 TypeLibType 属性	(171)
10.1.22 TypeLibVar 属性	(172)
10.2 C# 和 COM 互操作的例子: 支持枚举	(172)
10.3 小结	(175)
第 11 章 C# 应用的一些小技巧	(176)
11.1 在 C# 中生成安装程序	(176)

11.2	上传文件到服务器	(177)
11.3	C# 对线程的处理	(178)
11.4	用 C# 编写时钟程序	(180)
11.5	其他的小技巧	(182)
11.5.1	用 C# 获取主机的 IP 地址	(182)
11.5.2	用 C# 调用 MessageBox	(183)
11.5.3	用 C# 调用 Windows API	(183)
11.5.4	C# 对光标的操作	(183)
11.5.5	C# 对进程的获取	(183)
11.5.6	在 C# 中运行其他程序	(184)
11.5.7	用 C# 打开声音文件	(184)
11.5.8	从控制台读取数值	(184)
11.5.9	C# 中使用 OpenFileDialog 类	(185)
11.6	小结	(185)
第 12 章	用 C# 开发桌面程序	(186)
12.1	用 C# 实现清除文件夹	(186)
12.1.1	程序设计与分析	(186)
12.1.2	程序源代码及运行	(188)
12.2	用 C# 开发 DLL 程序	(189)
12.2.1	程序设计	(189)
12.2.2	编译生成 DLL 文件	(190)
12.2.3	创建简单的客户端程序	(190)
12.3	C# 开发 Windows 程序	(191)
12.3.1	创建应用程序框架	(191)
12.3.2	消息处理	(193)
12.3.3	创建 C# 类库	(195)
12.4	小结	(198)
第 13 章	用 C# 开发 WinForm 程序	(199)
13.1	WinForm 是什么	(199)
13.2	WinForm 的使用	(200)
13.3	WinForm 设计器的布局	(201)
13.4	C# 的 WinForm 编程	(204)
13.4.1	用 C# 创建普通的 WinForm	(204)
13.4.2	用 C# 创建特殊的 WinForm	(205)
13.4.3	在 WinForm 中加入一个标签	(207)
13.5	WinForm 与 GDI+	(208)
13.6	使用 WinForm 访问系统底层	(211)
13.7	小结	(212)
第 14 章	C# 与 XML 结合编程	(213)
14.1	关于 XML	(213)

14.1.1	XML 是什么	(213)
14.1.2	XML 的语法	(214)
14.1.3	如何使用 XML	(215)
14.2	C# 对 XML 文件处理	(216)
14.2.1	浏览 XML 文件内容	(216)
14.2.2	往 XML 文件中添加内容	(217)
14.2.3	删除 XML 文件内容	(218)
14.3	用 C# 与 XML 创建客户浏览菜单程序	(219)
14.3.1	程序设计	(219)
14.3.2	程序代码分析	(219)
14.3.3	编译 C# 文件	(225)
14.4	小结	(225)
第 15 章	在 ASP.NET 上的 C# 编程	(227)
15.1	ASP.NET 简介	(227)
15.1.1	ASP.NET 与 ASP 的联系和区别	(227)
15.1.2	ASP.NET 的配置	(227)
15.1.3	ASP.NET 的内置对象	(228)
15.1.4	ASP.NET 的特点	(229)
15.2	ASP.NET 与 C# 编程	(231)
15.2.1	用 ASP.NET 实现域名查询	(231)
15.2.2	用 C# 实现发送 E-mail	(232)
15.2.3	用 C# 编程实现显示事件日志记录	(234)
15.2.4	在 ASP.NET 中用 C# 实现文件处理	(239)
15.3	小结	(243)
第 16 章	C# 与 Web 数据库编程	(244)
16.1	C# 与 ADO.NET 结合编程	(244)
16.1.1	ADO.NET 是什么	(244)
16.1.2	ADO.NET 核心组件	(245)
16.1.3	ADO.NET 的对象	(247)
16.2	用 C# 添加 Access 数据库的记录	(248)
16.2.1	程序运行的环境及数据库设置	(249)
16.2.2	程序设计	(249)
16.2.3	程序运行和源代码	(250)
16.3	用 C# 的 SQL 编程	(257)
16.3.1	存取 ODBC 数据库	(257)
16.3.2	对记录的维护	(258)
16.3.3	编写简单的留言本	(258)
16.3.4	使用 XML 进行信息处理	(261)
16.4	小结	(261)
第 17 章	用 C# 编程操作注册表	(262)

17.1 有关注册表的 C# 编程基础	(262)
17.1.1 Windows 注册表	(262)
17.1.2 Registry 及 RegistryKey 类	(263)
17.2 生成注册表的键及键值	(264)
17.3 读取注册表的主键和键值	(265)
17.3.1 准备工作	(265)
17.3.2 程序设计	(265)
17.3.3 源程序代码	(267)
17.4 删除注册表中键及键值	(269)
17.4.1 删除键和键值所用的方法	(269)
17.4.2 程序设计与实现	(270)
17.5 小结	(271)
附录 A 专业英语名词	(272)
附录 B C# 编译器选项	(279)
B.1 输出文件相关选项	(279)
B.2 输入文件相关选项	(279)
B.3 资源相关选项	(279)
B.4 代码生成相关选项	(279)
B.5 错误和警告相关选项	(280)
B.6 语言相关选项	(280)
B.7 其他方面的选项	(280)
B.8 增强的选项	(280)
附录 C C# 编译错误	(281)

第 1 章 令人耳目一新的 .NET

人们已进入了互联网时代。近来，一个新的名词“ .NET（读做 dot net）”频频出现，逐步进入日新月异的 IT 生活。2000 年 6 月 22 日，微软公司正式对外宣布其 .NET 战略，并确定每年为这个新的战略直接投入 40 亿美元的研发费用。 .NET 这个曾经模糊不定的庞大体系迅速成为业界瞩目的焦点。2000 年 9 月，微软公司在旧金山发布了 Enterprise 2000。同月，微软原总裁兼首席执行官鲍尔默来到中国就“下一代互联网”主题进行演讲，在中国掀起了一股“ .NET 旋风”。

2000 年 11 月，微软在 COMDEX 计算机大展上发布了 Visual Studio .NET 软件，并展示了其 .NET 发展战略的框架体系和开发工具的相关特性，全面加速了微软以 .NET 技术进军市场的步伐。

随着 .NET 步伐的加快，身为中国的 IT 界人员，要审时度势，认清 .NET 计划的本质和发展趋势。

1.1 认识 .NET

.NET 曾经一度在微软内部称做“下一代视窗服务”（NGWS），最终确定为 .NET 名称。这不仅仅是微软惯用的市场化手段（为了朗朗上口和便于用户的识别），而且是把这个包含创新性的概念转变成为一个集全商业模型、全新技术模型于一体的统一框架。

.NET 的产生和发展离不开 IT 业界数年来围绕着因特网的发展、起落和沉浮。微软在这一领域中曾经落后过，比如网络浏览器，但它一直在奋起直追。激烈的业界竞争和不断的官司缠身使其必须提出一个崭新的战略和概念来重新树立自己的信心和软件业界的地位，这就是 .NET 计划。图 1.1 是 .NET 的功能和连接示意图。

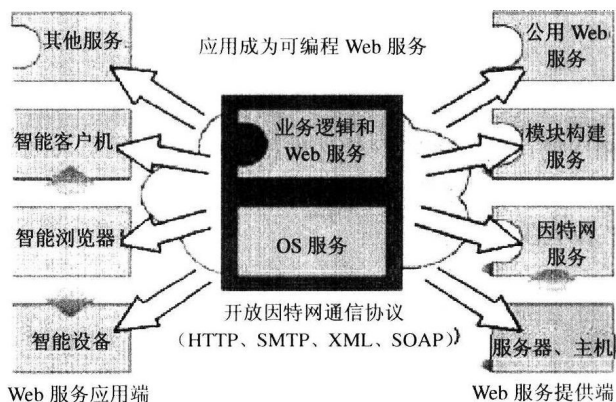


图 1.1 .NET 的连接和功能示意

对于 .NET 到底是什么这个问题,有着各种各样的说法。其中身为首席执行官的鲍尔默应该最能代表微软公司的观点,他说:“Microsoft .NET 代表了一个集合、一个环境、一个可以用做平台支持下一代因特网的可编程结构。”这句话基本上简明扼要地表述了 .NET 的基本特性。

首先, .NET 是一个环境。这是一个理想化的未来互联网环境,微软的构想是一个“不再关注单个网站、单个设备与因特网相连的互联网环境,而是要让所有的计算机群、相关设备和服务商协同工作”的网络计算环境。简而言之,互联网提供的服务,要能够完成更高层次的自动化处理。未来的互联网,应该以一个整体服务的形式展现在最终用户面前,用户只需要知道自己想要什么,而不需要一步步地在网上搜索、操作来达到自己的目的。这是一种美好的理想,但确实也是今后国际互联网的发展趋势。

.NET 谋求的是一种理想的互联网环境。要搭建这样一种互联网环境,首先需要解决的问题是针对现有因特网的缺陷,来设计和创造一种下一代因特网结构。这种结构不是物理网络层次上的拓扑结构,而是面向软件和应用层次的一种有别于浏览器只能静态浏览的可编程因特网软件结构。因此把 .NET 定位为可以用做平台支持下一代因特网的可编程结构。

.NET 的最终目的就是让用户在任何地方、任何时间,以及利用任何设备都能访问他们所需要的信息、文件和程序。用户不需要知道这些东西存在什么地方,甚至连如何获得等具体细节都不必知道。他们只需发出请求,然后只管接收就是了,而所有后台的复杂性是完全被屏蔽起来的。所以对于企业的程序开发人员来说,他们并不需要管理复杂的平台以及协调各种分布应用。

.NET 的体系结构如图 1.2 所示。

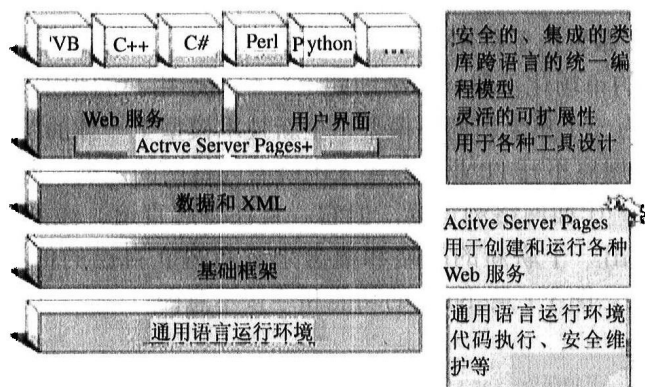


图 1.2 .NET 的体系结构

有了理想目标和相应的可编程因特网软件结构,要具体实现 .NET 这样一个协同计算环境必然还需要一系列的软件产品支撑,因此微软的 .NET 包括一个产品的集合,如 Visual Studio .NET, ASP .NET 和 ADO .NET 等。

1.2 .NET 的技术特征和新特性

1.2.1 .NET 的技术特征

.NET 具有 4 个重要特点, 分别是: 基于 XML 的共同语言, 定制服务软件, 融合多平台和设备以及新一代的人机界面。这 4 个特点基本上覆盖了 .NET 的技术特征。

1. 基于 XML 的共同语言

XML 是从 SGML 语言演化而来的一种标记性语言。用做元语言, 它可以定义不同种类应用的数据交换语言。在 .NET 体系结构中, XML 用做一种应用间无缝接合的手段, 用于多种应用之间的数据采集与合并, 用于不同应用之间的互操作和协同工作。具体而言, .NET 通过 XML 语言定义了简单对象访问协议 (SOAP)、Web 服务描述语言 (WSDL) 和 Web 服务发现协议 (DISCO)。SOAP 协议提供了在无中心分布环境中使用 XML 交换结构化有类型数据的简单轻便的机制。WSDL 协议定义了服务描述文档的结构, 如类型、消息、端口类型、端口和服务本身。DISCO 协议定义了如何从资源或者资源集合中提取服务描述文档与相关服务发现算法等。

2. 定制服务软件

微软首席执行官史蒂夫·鲍尔默在谈到软件服务时说: “今天的软件产品仅仅是一张光盘, 用户购买软件, 亲自安装、管理和维护。但是软件服务是来自因特网的服务, 它替用户安装、更新和跟踪这些软件, 并让它们和用户一起在不同的机器间漫游。它为用户存储自己的信息和参考资料。这些就是软件和软件服务各自不同的风格。”

图 1.3 示意 .NET 的定制服务软件的技术特征。

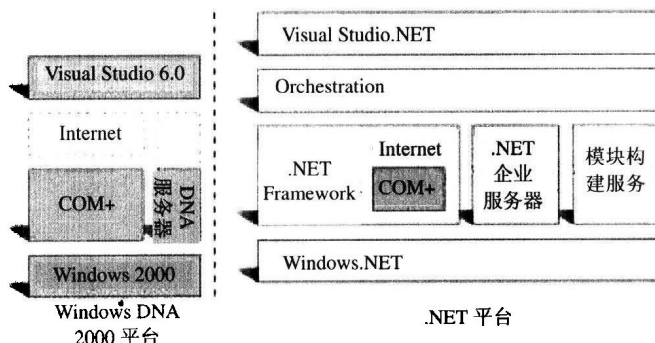


图 1.3 .NET 的技术特征之一：定制服务软件

Orchestration 可视化编程工具产生基于 XML 的 XLANG 代码, 它和 BizTalk 服务器、.NET Framework 以及 Visual Studio.NET 都曾是的 Windows DNA 2000 战略的重要部分。

伴随着 ASP 产业的兴起, 软件正逐渐从产品形式向服务形式转化, 这是整个 IT 行业的大势所趋。在 .NET 中, 最终的软件应用是以 Web 服务的形式出现并在因特网发布的。Web 服务是一种经过包装的可以在 Web 上发布的组件。.NET 通过 WSDL 协议来描述和发布这

种 Web 服务信息，通过 DISCO 协议来查找相关的服务，通过 SOAP 协议进行相关的简单对象传递和调用。

微软的 .NET 战略意味着：微软公司以及在微软平台上的开发者将会制造服务，而不是制造软件。在未来几年之内，微软将陆续发布有关 .NET 的平台和工具，用于在因特网上开发 Web 服务。那时，工作在 .NET 上的用户、开发人员和 IT 工作人员都不再购买软件，安装软件和维护软件。取而代之的是，他们将定制服务，软件会自动安装，所有的维护和升级也会通过互联网进行。

3. 融合多平台和设备

随着因特网逐渐成为一个信息和数据的中心，各种设备和服务已经或正在接入和融入因特网，成为其中的一部分。.NET 谋求与各种因特网接入设备和平台的一体化，主要关注无线设备和家庭网络设备及相关的软件 and 平台。

4. 新一代的人机界面

.NET 的新一代人机界面特征主要体现在“智能与互动”两个方面，包括通过自然语音、视觉、手写等多种模式的输入和表现方法，基于 XML 的可编辑复合信息架构——通用画布，个性化的信息代理服务及使机器能够更好地进行自动处理的智能标记等技术。

.NET 的平台及框架是基于微软软件工业基础的又一次升级和演化。然而，还是要尽力保证 Windows 系统及系列产品和 .NET 能够融为一体，尽量在微软公司原有的软件资产基础上，使 .NET 继续成为因特网的中心。

1.2.2 .NET 的新特性

.NET 是一种全新的技术，因此，.NET 中包括了很多新特性。这里只列出一些比较重要的特性。.NET 新特性包括一致的编程模式，简化的编程模式，运行于多个平台，支持多语言的综合，自动资源管理，一致的出错处理方式，安全性和 XML 语言的引入等。

1. 一致的编程模式

在 .NET 环境中，所有的应用程序都采用通用的面向对象编程模式，不再像 Windows 环境中那样，既有 DLL 函数也有 COM 对象。

2. 简化的编程模式

这也许是令开发人员最欢欣鼓舞的消息，在 .NET 环境下，由于 CLR 的作用，在进行编程时不再需要掌握 GUIDs, Iunknown 和 AddRef 等令人头疼的 COM 知识了。

3. 运行于多个平台

对于任何操作平台，只要支持 .NET 运行时均可以运行 .NET 应用程序。现在所有的 Windows 平台均可以实现这一点。在将来甚至在非 Windows 操作系统上也可以实现这一点。

4. 支持多语言的综合

按照 COM 的原理，代码重用是建立在二进制代码的级别上的。在 .NET 环境下，代码