



Deitel 编程金典：开发者系列

PEARSON

Prentice
Hall

C# 高级程序员指南

C# for Experienced Programmers

**Contains
230 LIVE-CODE™
Examples**

- Visual Studio® .NET
- Classes/Data Abstraction
- Assemblies/Namespace
- Exception Handling
- GUI/Windows® Forms
- Databases/ADO .NET/SQL
- Files and Streams/XML
- ASP .NET and Web Services
- Networking
- Mobile Web Development

Introducing
Microsoft® Mobile Internet Toolkit

[美] H.M.Deitel, P.J.Deitel, J.A.Listfield, T.R.Nieto, C.H.Vaeger, M.Zlatkina 著

周靖 姜昊 龙劲松 译



清华大学出版社

Deitel 编程金典：开发者系列

C# 高级程序员指南

[美] H. M. Deitel, P. J. Deitel, J. A. Listfield, 著
T. R. Nieto, C. H. Yaeger, M. Zlatkina

周 靖 姜 昊 龙劲松 译

清华大学出版社

北 京

内 容 简 介

本书由全球知名教育专家 H. M. Deitell 领头编写。书中采用 Deitel 独创的“活代码”教学方式,深入讲解和探索了 Microsoft 的 C# 语言,展示了重要的 C# 概念。全书包含 230 个活代码程序,共 26 000 行代码,这些示例程序已经通过完整测试,并配备清楚的语法标识、详细的逐行解释和输出结果。值得一提的是书中包含的编程技巧(共 402 条),这些技巧可帮助读者养成良好的编程习惯,构建出更易于移植、易于重用并具有优良性能的应用程序。

本书适合具备 C++、Visual Basic 和 Java 等高级语言背景的程序员阅读。

EISBN: 0-13-046133-4

C# for Experienced Programmers

H. M. Deitel, P. J. Deitel, J. A. Listfield, T. R. Nieto, C. H. Yaeger, M. Zlatkina

Copyright © 2003 by Pearson Education, Inc.

Original English language edition published by Pearson Education, Inc.

All right reserved.

For sale and distribution in the People's Republic of China exclusively (except Taiwan, Hong Kong SAR and Macau SAR).仅限于中华人民共和国境内(不包括中国香港、澳门特别行政区和中国台湾地区)销售发行。

本书封面贴有 Pearson Education (培生教育出版集团)激光防伪标签,无标签者不得销售。

北京市版权局著作权合同登记号:图字 01-2002-5752 号

图书在版编目(CIP)数据

C# 高级程序员指南 / (美)迪特尔等著;周靖,姜昊,龙劲松译. —北京:清华大学出版社,2003
(Deitel 编程金典:开发者系列)

书名原文: C# for Experienced Programmers

ISBN 7-302-06467-9

I. C... II. ①迪... ②周... ③姜... ④龙... III. C#语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2003)第 020490 号

出 版 者:清华大学出版社(北京清华大学学研大厦,邮编 100084)

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

<http://www.tup.tsinghua.edu.cn>

责任编辑:文开棋

印 刷 者:清华大学印刷厂

发 行 者:新华书店总店北京发行所

开 本:787×1092 1/16 印张:62.5 字数:2570 千字

版 次:2003 年 4 月第 1 版 2003 年 4 月第 1 次印刷

书 号:ISBN 7-302-06467-9/TP·4864

印 数:0001~3500

定 价:118.00 元

前 言

欢迎进入 C# 利用 Visual Studio .NET 和 .NET 平台实现 Windows、因特网和万维网编程的世界！本书是新的《Deitel 开发者系列》的第二本书，这个系列将为软件开发者和 IT 专家讲解领先于时代的计算技术。

C#（读作“C-sharp”）是微软为其 .NET 平台开发的最新语言，它为程序员提供了一些非常重要的特性，比如面向对象编程、图形、图形用户界面 (GUI) 组件、异常处理、多线程、多媒体（声音、图形、动画和视频）、文件处理、预封装数据结构、数据库处理、基于因特网和万维网的多层应用程序开发、联网、Web 服务和分布式计算等。这种语言尤其适用于实现与 Windows 应用程序无缝集成的因特网和万维网应用程序。

.NET 平台为软件的开发和部署提供了强大的功能，其中包括语言 and 平台无关性。例如，开发者可采用任何一种或几种 .NET 语言（比如 C#，Visual Basic .NET 和 Visual C++ .NET）来编写代码，并将其用作同一个软件产品的组件。除了提供语言的独立性之外，.NET 还增强了程序的可移植性，它允许 .NET 应用程序驻留在多个平台上，并允许它们在这些平台之间通信。这样一来，便简化了 Web 服务的创建及使用，后者是通过因特网将功能揭示给客户端的应用程序。

.NET 平台允许将基于 Web 的应用程序发布到消费类电子设备，比如无线电话和个人数字助理 (PDA)，还能发布到桌面计算机。微软集成到 .NET 平台中的功能有效提高了程序员的生产力，并缩短了开发时间。

本书面向的读者

本书言简意赅地介绍了最新的编程技术，适合有经验的程序员阅读。

书中展示了大量完整的、能实际工作的 C# 程序，并采用程序运行时的实际屏幕截图，展示了它们的输入及输出。这是 H. M. Deitel 独创的“活代码” (Live-Code) 教学方法——所有概念都采用完整的工作程序来展示。书中的源代码都可从 www.deitel.com 下载。

在阅读本书的过程中，如果想同我们联系，请发一封电子邮件到 deitel@deitel.com，我们来信必复。另外，请访问我们的网站 www.deitel.com，www.prenhall.com/deitel，以及 www.InformIT.com/deitel，我们会经常发布更新、勘误、FAQ 等。发送电子邮件时，不要忘了写明书籍标题和版本号。我们衷心希望您能通过本书学好 C#。

本书特色

本书具有以下特色：

- **语法结构突出显示。**本书采用 5 种不同的语法突出显示技术来强调 C# 编程元素，具体同 Visual Studio .NET 采用的方式非常相似。语法结构突出显示的约定如下：

注释

关键字

字面值

错误和 ASP .NET 指令

文本、类、方法和变量名

- **“代码清洗”。**这是我们杜撰的术语，是指对书中程序进行格式化的过程，目的是使它们有一个精心注释的、开放式的布局。代码划分为小的、进行了良好编档的区域。这样可显著提高代码的可读性——这正是我们的一个非常重要的目标，尤其是考虑到本书包含约 26 000 行代码，总计 230 个完整的活代码程序。

- **Web 窗体、Web 控件和 ASP .NET。** .NET平台使开发者能创建健壮的、易于扩展的基于 Web 的应用程序、微软的.NET服务器端技术、活动服务器页(ASP) .NET,允许程序员生成 Web 文档以响应用户请求。为使交互式网页成为可能,服务器端程序要处理用户输入 HTML 表单中的信息。ASP .NET是对 ASP 3.0 的一次重大升级,它允许开发者使用.NET强大的面向对象语言来编写基于 Web 的应用程序,这些语言包括 C# 和 Visual Basic .NET 等,而不是只能使用脚本语言。ASP .NET还具有增强的可视化编程功能,类似于为桌面程序生成 Windows 窗体时所采用的。程序员通过将 Web 控件拖放到 Web 窗体上,即可通过可视化的手段创建网页。第 17 章将对这些强大的技术进行介绍。
- **Web 服务和 ASP .NET。**微软的.NET策略将 Internet 和 Web 视为软件开发及部署时的一个整体。利用 Web 服务技术,可通过标准的 Internet 协议和技术,比如超文本传输协议(HTTP)、可扩展标记语言(XML)以及简单对象访问协议(SOAP)等,有效实现信息共享、电子商务以及其他集成。Web 服务允许程序员包装应用程序功能,将 Web 转换成包含了许多可重用软件组件的一个庞大的库。第 18 章将展示一个具体的 Web 服务,它允许用户处理“巨整数”——太大以至于不能包含在 C# 的内建数据类型中的整数。示例中,用户输入两个巨整数,并按相应的按钮来调用 Web 服务。Web 服务则对这两个整数执行加法、减法和比较运算。另外,附录 O 将展示与 Web 服务有关的信息,该附录讨论了为以数据库为中心的应用程序而设计的主流报表程序。Crystal Reports 目前已集成到 Visual Studio .NET,允许将 Web 服务以报表的形式呈现出来。本附录提供了介绍性信息,并指导读者到 Crystal Decisions 网站(www.crystaldecisions.com/netzone)查阅具体的操作过程。
- **面向对象编程。**面向对象编程是用于开发健壮的、可重用软件的最普遍的一种技术。本书全面探讨了 C# 的面向对象编程特性。第 5 章将介绍如何创建类和对象。这些概念将在第 6 章得到进一步扩展,同时还要讨论程序员如何通过“吸收”现有类的功能,快速创建出功能强大的新类。第 7 章则向读者讲授多态性、抽象类、具体类和接口的关键概念。利用它们,可方便地对从属于一个继承层次结构的对象进行处理。
- **XML。**可扩展标记语言(XML)正在引发一场革命,软件开发工业、电子商务社区以及整个.NET平台都在普遍地运用它。由于 XML 是一种与平台无关的技术,用于描述数据和创建标记语言,所以 XML 的数据移植能力较强,可与基于 C# 的可移植应用程序及服务很好地集成。第 15 章将介绍 XML,内容包括基本的 XML 标记,像 DTD 和架构(Schema)这样的技术,它们用于对 XML 文档的内容进行校验。另外,还将解释如何利用文档对象模型(DOM)以程序化的方式处理 XML 文档,以及如何利用可扩展样式表语言转换(XSLT)将 XML 文档转换成其他文档类型。
- **多线程处理。**计算机允许程序员并行(并发)执行多个任务,比如打印文档、从网络下载文件以及在 Web 上冲浪等。利用多线程处理技术,程序员可开发出同时执行多个任务的应用程序。过去,计算机包含了单一的、昂贵的处理器,它的操作系统由所有应用程序共享。如今,处理器变得越来越便宜,所以完全能生产出价格低的、多个处理器并行工作的计算机。像这样的计算机称为“多处理器计算机”。另一方面,多线程处理同时适用于单处理器和多处理器计算机。.NET的多线程处理能力使平台及其相关技术能更好地处理如今高级的、以多媒体为中心的、以数据库为中心的、基于网络的、基于多处理器的分布式应用程序。
- **ADO .NET。**数据库保存着大量信息,个人和组织必须访问这些信息才能高效地进行商业运作。随着 Microsoft 的 ActiveX 数据对象(ADO)技术的发展,ADO .NET揭示了如何通过一种新途径来生成与数据库进行交互的应用程序。ADO .NET使用 XML 和一个增强的对象模型,针对大规模的、可扩展的、面向关键任务的多层应用程序,为开发者提供了访问和操纵它们的工具。第 16 章将介绍如何使用 ADO .NET和结构查询语言(SQL)来操纵数据库。
- **无线部署。**据预测,全球约有 10 亿人正在使用移动设备,比如无线电话和 PDA 等,而且这一数量正在快速地增长。为了简化创建供移动设备使用的内容,Microsoft 提供了 Mobile Internet Toolkit (MIT)。MIT 基于 ASP .NET,允许使用 Visual Studio .NET的面向对象语言来创建无线内容。这样创建的程序可兼容于大量设备,并可根据设备的类型(比如无线电话和 PDA)显示不同的内容。

第 22 章将介绍无线 Web 应用程序的开发。

- **Visual Studio .NET 调试器**。调试器帮助程序员查找和纠正程序代码里的逻辑错误。附录 C 将解释如何使用关键的调试器特性,比如设置“断点”和“监视”,逐语句和逐过程方法,以及检查方法调用堆栈等。
- **COM(组件对象模型)集成**。在 .NET 问世之前,许多组织将大量时间和金钱花在创建名为“COM 组件”的可重用软件组件上,其中包括用于 Windows 应用程序的 ActiveX 控件和 ActiveX DLL(动态链接库)。附录 G 讨论如何利用 Visual Studio .NET 的一些工具将这些遗留组件集成到 .NET 应用程序中。通过这样的集成,程序员可将现有的、基于 COM 的控件与 .NET 组件结合使用。
- **XML 编档**。对于软件开发过程来说,程序代码的编档是至关重要的,因为不同的程序员通常要在软件的生命期内对一个应用程序进行操作。这样的软件可能有多个版本,而且可能延续数年。如果程序员对软件代码和方法进行了编档,其他接手这个应用程序的程序员便可轻松学习和理解代码的底层逻辑,从而节省时间,并避免误解。针对自动化程序的编档过程,Visual Studio .NET 为 C# 程序员提供了一个 XML 工具。附录 D 将向程序员介绍如何在代码中插入注释,从而生成一个单独的文件来提供代码文档。
- **Unicode**。随着计算机系统的全球化,针对不同国家所讲的本地语言,计算机厂商开发了大量技术支持字符集和特殊符号。某些情况下,甚至会对同一种语言开发不同的表示方法。字符集的差异妨碍了计算机系统的顺利通信。C# 支持 Unicode 标准(Unicode 协会是一家非营利性组织),它在一种字符集中使用独一无二的数值来表示全球大多数语言中的字符和特殊符号。附录 F 将讨论这一标准,简要介绍 Unicode 协会网站(www.unicode.org),并展示了一个 C# 应用程序,它能用多种语言打出“Welcome to Unicode!”。
- **XHTML**。万维网协会(W3C)已明确表示 HTML 是一种过时的技术,不会有进一步的发展。HTML 已被可扩展超文本标记语言(XHTML)取代。这是一种基于 XML 的技术,正在快速成为描述 Web 内容的一种标准。第 15 章将使用 XHTML,附录 J 和附录 K 将对该技术进行简要介绍。这两个附录还将概述 XHTML 的标题、图像、列表、图像地图以及其他特性。另外,附录 H 和附录 J 还将对 HTML 进行简要说明,因为第 17 章和第 18 章使用的 ASP.NET 需要生成 HTML 内容。
- **可访问性**。尽管万维网已成为人们日常生活的一个重要部分,但对于那些残疾人士,要想正常使用万维网,却显得困难重重。尤其是听力和/或视力有缺陷的用户,很难访问具有丰富多媒体内容的网站。为改善这一现状,万维网协会(W3C)制订了“Web 可访问性提案”(WAI)。在这份提案中,提供了一系列指导准则,描述了应该如何设计网站,使其便于残疾人士访问和使用。第 21 章将描述这些准则,并重点介绍一些产品和服务,它们设计用于改善残疾人士的 Web 浏览体验。例如 VoiceXML 和 CallXML——两种基于 XML 的技术——可使视力有缺陷的人更方便地访问基于 Web 的内容。
- **位处理**。计算机采用二进制位(或比特)的形式来处理数据,即一系列的 0 和 1。计算机电路执行的是各种简单的位处理,比如检查一个位的值,设置一个位的值,以及反转一个位(1 变成 0,0 变成 1)等。操作系统、测试仪器、网络软件以及其他多种软件都要求程序通过位处理“直接与硬件通信”。附录 N 概述了 .NET 框架所提供的位处理功能。

教学思路

本书包含的丰富示例已在 Windows 2000 和 Windows XP 上进行了全面测试。本书强调良好的软件工程规范以及程序结构清晰的必要性。我们在全球各地的课堂讲授实用主题,擅长举例说明的授课方式,尽量避免令人费解的术语和语法规则。本书忠实地反映了我们的教学思路。

活代码教学方式

本书包含大量活代码示例。这种风格是我们从事教学和编程书籍写作的最大特色。随着每个新概念的提出,都会同时提供一个完整的、可实际运行的程序。紧接在这个程序之后,则用一个或多个窗口来显示

该程序的输入/输出对话框。我们把这种风格的教学和写作称为活代码(Live-Code)方式。我们用完整的、可实际运行的程序来教授程序语言。浏览这些程序时,整个过程极像亲自在一台计算机上输入并运行它们。大家可从 www.deitel.com(选择 Downloads/Resources 链接)下载书中所有示例代码。其他有用的链接还包括本书勘误表和常见问题解答。

访问万维网

本书(以及我们的其他出版物)所有示例的源代码都可在以下网站下载:

www.deitel.com

www.prenhall.com/deitel

注册过程非常简单、快捷,而且所有下载都是免费的。建议下载所有例子,在阅读书中相应部分时,运行每一个程序。对例子进行修改,可马上看到修改的效果——这是提升编程技术的有效方式。示例的运行环境是 Windows 2000 或 Windows XP 操作系统,而且正在使用 Microsoft 的 Internet 信息服务(IIS)。我们的网站上,除了这些例子之外,还有 IIS 和其他软件的安装指南。注意,这些材料是有版权的。学习时可任意使用,但没有 Prentice Hall 和作者的书面许可,不可以任何形式重新出版它的任何部分。

Visual Studio .NET 是一个产品家族的总称,可从 Microsoft 购买和下载。Visual Studio .NET(内含 C#)提供 4 种不同的版本——Academic、Professional、Enterprise Developer 和 Enterprise Architect。其中,Visual Studio .NET Academic 除包含 Visual Studio .NET Professional 的特性之外,还包括专为学生和教授设计的特性(例如用于编档作业提交的 Assignment Manager 和有助于作业分配、代码示例及其他类似信息通知的 Application Publishing Tools)。

Microsoft 还为各种 .NET 语言提供了独立的产品(Visual C# .NET Standard、Visual C++ .NET Standard 以及 Visual Basic .NET Standard)。每种产品都提供一个集成开发环境(类似于 Visual Studio .NET)和一个编译器,详细说明和订购信息请访问 msdn.microsoft.com/vstudio/howtobuy。

学习目标

每章开头都有一个“学习目标”列表。它告诉读者本章涉及哪些主题,读者学完一章后,还有机会回过头去复查这些学习目标,看看自己是否已达到所有目标。

230 个示例程序,约 26 000 行代码,并附程序输出

我们用大量完整的、能实际运行的程序来讲解 C# 的各种特性。程序的大小不一,从包含几行代码到数百行代码的都有。所有例子都可从我们的网站 www.deitel.com 下载。

665 张插图

包括丰富的图表、线条图和程序输出。

402 条编程提示

我们采用编程提示来帮助读者关注程序开发的重要环节。我们采用以下形式突出显示数百条这样的提示,包括:良好编程习惯、常见编程错误、测试和调试提示、性能提示、可移植性提示、软件工程知识以及界面知识等。这些提示和操作是作者数十年来编程和教学经验的结晶。我们的一个客户(一个数学硕士)告诉我们,她觉得这些“提示”就像数学教科书中的那些公理、定理和推论,牢记这些东西,有助于开发出最优秀的软件。

52 条“良好编程习惯”

良好编程习惯提醒读者关注使程序更清晰、更易理解和更易维护的技术。

131 条“常见编程错误”

开发者在学习一种语言时,通常会犯某些常见错误。把注意力放在这些“常见编程错误”上,有助于避免以后再犯同样的错误。

33 条“测试和调试提示”

我们第一次设计这类提示时,考虑只让它们描述如何揭示 Bug,并从程序中删除它们。但事实上,最终许多提示都描述了 C# 的一些特性,它们有助于从源头杜绝 Bug 进入程序,从而简化了测试与调试过程。

46 条“性能提示”

开发者喜欢不停地对自己的程序进行优化。我们提供了 46 条性能提示,目的是便是指出对程序性能进行改进的可能性——这会使程序运行得更快,并减小它们占用的内存。

15 条“可移植性提示”

这种提示帮助开发者写出可移植的代码,并解释了 C# 如何实现它的高移植性。

96 条“软件工程知识”

“面向对象编程思维模式”要求我们重新彻底思考构造软件系统的方式。C# 是实现良好软件工程的一种高效语言。在“软件工程知识”中,强调了一些重要的结构和设计问题,它们会影响软件系统——尤其是那些大型系统——的构成方式。

25 条“界面知识”

它强调的是图形用户界面(GUI)的设计规范,可帮助开发者设计出富有吸引力的、对用户友好、但又符合行业标准的图形用户界面。

小结

每章以“小结”结尾,对关键概念进行了回顾和补充。

作者简介

H. M. Deitel 博士;Deitel & Associates 公司主席兼首席战略官,在计算领域已有 41 年的工作经验,无论专业技术还是学校教育,均有非常高的造诣。H. M. Deitel 博士拥有麻省理工学院的 B.S 和 M.S 学位,以及波士顿大学的 Ph.D 学位。他参与过 IBM 和 MIT 的一系列领先于时代的虚拟内存操作系统项目,研究成果如今已广泛用于 UNIX、Windows NT、OS 2 和 Linux 系统。他有 20 年的大学教学经验,而且在和其子 P. J. Deitel 成立 Deitel & Associates 公司之前,一直担任波士顿大学计算机科学系主任的职位。他创作或参与了数十本书,并参与了多媒体产品的开发。多年来,他的作品已被翻译成简体中文、繁体中文、日语、俄语、西班牙语、朝鲜语、法语、波兰语、意大利语、葡萄牙语以及希腊语,畅销全球。H. M. Deitel 博士经常举办国际性技术研讨会,向大型公司、政府部门以及各种军事单位“传经布道”。

P. J. Deitel;Deitel & Associates 公司 CEO 兼首席技术官,毕业于麻省理工学院的斯隆管理学院,主修信息技术。在 Deitel & Associates 公司,他负责面向业内许多知名客户讲授 Java、C、C++、Internet/万维网课程,这些客户包括康柏、Sun、White Sands Missile Range、Rogue Wave Software、Computervision、Stratus、Fidelity、Cambridge Technology Partners、Lucent Technologies、AdraSystems、Entergy、CableData Systems、NASA 肯尼迪太空中心、美国国家大风暴实验室(NSSL)、IBM 以及其他许多公司和机构。他负责为计算机机构联盟波士顿分部讲授 C++ 和 Java 课程。他目前正用 Deitel & Associates、Prentice Hall 以及“美国技术教育网络”(Technology Education Network)联合投资的一笔风险基金,利用卫星技术来提供 Java 课程。他和其父 H. M. Deitel 博士是全球畅销程序语言教科书的作者。

T. R. Nieto;Deitel & Associates 公司产品开发主管,毕业于麻省理工学院,主修工程与计算。在 Deitel & Associates 公司,他负责向业内的一系列客户发布培训课程,这些客户包括 Sun、康柏、EMC、Stratus、Fidelity、NASDAQ、Art Technology、Progress Software、Toys “R” Us、美国国家海洋和天气管理局的运营支持部、喷气推进实验室、Nynex、摩托罗拉、芝加哥联邦储备银行、Banyan、Schlumberger、Notre Dame 大学、NASA、惠普、大量军事基地和其他许多单位。他和两位 Deitel 先生合作出版了多种计算机书籍和多媒体教学光盘。事实上,Deitel & Associates 出版的几乎每一样产品都有他的功劳。

C. H. Yaeger, Deitel & Associates 公司的 Microsoft 软件出版物总监, 毕业于波士顿大学, 3 年内就获得了计算机科学学士学位。Cheryl 参与创作了 Deitel & Associates 的大量出版物, 其中包括《C# How to Program》、《C# A Programmer's Introduction》、《C# for Experienced Programmers》和《Visual Basic .NET for Experienced Programmers》。另外, 他还为 Deitel 其他出版物写过稿, 包括《Perl How to Program》、《Wireless Internet & Mobile Business How to Program》、《Internet and World Wide Web How to Program (第二版)》和《Visual Basic .NET How to Program (第二版)》。

M. Zlatkina: 毕业于布兰代斯大学, 3 年内就获得计算机科学和数学学士学位, 第 4 年获得计算机科学硕士学位。在布兰代斯大学上学期间, 她领导了数据库领域的研究, 并成为一名教学助理。她参与创作了《C# How to Program》、《C# A Programmer's Introduction》和《C# for Experienced Programmers》, 并为另一本 Deitel 出版物写过稿, 即《e-Business & e-Commerce for Managers》。

J. A. Listfield: 哈佛大学应届毕业生, 主修计算机科学。他的毕业论文涉及了计算机图形、网络 and 计算理论, 并有 C, C++, Java, Perl 和 Lisp 的编程经验。Jeff 参与创作了《C# How to Program》、《C# A Programmer's Introduction》和《C# for Experienced Programmers》, 并参与创作了《Perl How to Program》。

万维网协会(W3C)

Deitel & Associates 公司是“万维网协会”(World Wide Web Consortium, W3C)的成员。W3C 成立于 1994 年, 宗旨是“开发标准协议, 适应并推动万维网的发展”。作为 W3C 成员, 我们在 W3C 顾问委员会占有一个席位。我们的顾问委员会代表是 CEO 兼首席技术官(CTO) Paul Deitel 先生。顾问委员会的成员们通常在全球各地举行常务会议, 借此帮助 W3C 制订“战略决策”。通过参加不同的 W3C 的活动组, 成员单位也要负责制订一系列推荐性的 Web 技术标准(比如 HTML、XML 和其他许多标准)。目前, W3C 的成员大多是公司和大型组织。欲了解 W3C 的详情, 并了解如何申请成为 W3C 成员, 请访问 www.w3.org/Consortium/Prospectus/Joining。



目 录

第 1 章 .NET和C#概述	1	3.15 小结	66
1.1 概述	1	第 4 章 方法和数组	68
1.2 因特网和万维网发展简史	1	4.1 概述	68
1.3 万维网协会(W3C)	2	4.2 C#中的方法	68
1.4 可扩展标记语言(XML)	2	4.3 方法定义	69
1.5 重要的软件趋势:对象技术	3	4.4 参数提升	73
1.6 Microsoft 的.NET概述	4	4.5 C#命名空间	75
1.7 C#语言	5	4.6 值类型和引用类型	75
1.8 .NET框架和公共语言运行库	5	4.7 传递参数:传值和传引用	76
1.9 本书导读	6	4.8 作用域规则	79
1.10 小结	13	4.9 递归	81
1.11 因特网和万维网资源	14	4.10 方法重载	83
第 2 章 Visual Studio .NET IDE 和C#编程概述	15	4.11 数组	85
2.1 概述	15	4.12 声明和分配数组	86
2.2 Visual Studio .NET集成开发环境(IDE)	15	4.13 向方法传递数组	88
2.3 菜单栏和工具栏	18	4.14 通过传值和传引用来传递数组	88
2.4 Visual Studio .NET窗口	19	4.15 多下标数组	92
2.5 使用帮助	22	4.16 foreach 重复结构	95
2.6 简单程序(一):显示文本和图像	23	4.17 小结	96
2.7 简单程序(二):打印一行文本	30	第 5 章 基于对象的编程	97
2.8 算术运算	36	5.1 概述	97
2.9 做出决定:相等和关系运算符	37	5.2 用类实现 Time 抽象数据类型	98
2.10 小结	41	5.3 类的作用域	103
第 3 章 控制结构	42	5.4 控制对成员的访问	103
3.1 概述	42	5.5 初始化类的对象:构造函数	104
3.2 控制结构	42	5.6 使用重载构造函数	105
3.3 if 选择结构	43	5.7 属性	108
3.4 if/else 选择结构	43	5.8 合成:对象引用用作其他类的实例 变量	114
3.5 while 重复结构	44	5.9 使用 this 引用	117
3.6 赋值运算符	46	5.10 垃圾回收	118
3.7 自增和自减运算符	47	5.11 静态类成员	119
3.8 for 重复结构	47	5.12 常量和只读成员	122
3.9 示例:用 for 结构计算复利	49	5.13 索引器	124
3.10 switch 多选结构	52	5.14 数据抽象和信息隐藏	129
3.11 do/while 重复结构	55	5.15 软件重用	130
3.12 break 语句和 continue 语句	56	5.16 命名空间和程序集	131
3.13 逻辑和条件运算符	58	5.17 类视图和对象浏览器	135
3.14 Windows 应用程序编程概述	61	5.18 小结	136

第 6 章 面向对象的编程	138	9.11 小结	259
6.1 概述	138	第 10 章 图形用户界面概念(二)	261
6.2 基类和派生类	139	10.1 概述	261
6.3 受保护成员和内部成员	140	10.2 菜单	261
6.4 基类和派生类的关系	141	10.3 LinkLabel	268
6.5 案例分析:三级继承层次结构	156	10.4 列表框和带复选框的列表框	271
6.6 派生类中的构造函数和析构函数	159	10.5 组合框	277
6.7 与继承相关的软件工程	161	10.6 TreeView	280
6.8 小结	164	10.7 ListView	284
第 7 章 面向对象编程:多态性	166	10.8 TabControl	290
7.1 概述	166	10.9 多文档界面(MDI)窗口	294
7.2 派生类对象转换为基类对象	166	10.10 可视化继承	301
7.3 类型字段和 switch 语句	171	10.11 用户自定义控件	304
7.4 多态性示例	171	10.12 小结	308
7.5 抽象类和方法	172	第 11 章 多线程	309
7.6 示例(一):接口和实现的继承	173	11.1 概述	309
7.7 密封的类和方法	183	11.2 线程状态:线程的生命周期	310
7.8 示例(二):使用多态性的工资发放系统	186	11.3 线程的优先级和线程调度	311
7.9 示例(三):创建和使用接口	188	11.4 线程同步和类监视器	315
7.10 委托	198	11.5 没有线程同步的生产者/消费者关系	316
7.11 运算符重载	201	11.6 有线程同步的生产者/消费者关系	321
7.12 小结	206	11.7 生产者/消费者关系:循环缓冲区	327
第 8 章 异常处理	207	11.8 小结	336
8.1 概述	207	第 12 章 字符串、字符和正则表达式	338
8.2 异常处理概述	207	12.1 概述	338
8.3 示例:DivideByZeroException	209	12.2 字符和字符串基础	338
8.4 .NET的异常层次结构	213	12.3 String 类的构造函数	339
8.5 finally 块	211	12.4 String 类的索引器、Length 属性和 CopyTo 方法	340
8.6 Exception 属性	219	12.5 字符串的比较	341
8.7 程序员自定义的异常类	223	12.6 String 类的 GetHashCode 方法	344
8.8 用 checked 和 unchecked 运算符处理溢出	226	12.7 在字符串中定位字符和子串	345
8.9 小结	228	12.8 从字符串中提取子串	348
第 9 章 图形用户界面概念(一)	230	12.9 字符串的连接	348
9.1 概述	230	12.10 String 类的方法	349
9.2 Windows 窗体	231	12.11 StringBuilder 类	351
9.3 事件处理模型	233	12.12 StringBuilder 索引器、属性 Length 与 Capacity 以及方法 EnsureCapacity	352
9.4 控件属性和布局	237	12.13 StringBuilder 类的方法 Append 和 AppendFormat	354
9.5 标签、文本框和按钮	239	12.14 StringBuilder 类的方法 Insert、Remove 和 Replace	356
9.6 分组框和面板	241		
9.7 复选框和单选钮	247		
9.8 图片框	253		
9.9 鼠标事件处理	255		
9.10 键盘事件处理	257		

12.15	Char 方法	359	16.2	关系数据库模型	527
12.16	模拟扑克洗牌和发牌程序	361	16.3	关系数据库概述:Books 数据库	528
12.17	正则表达式和 Regex 类	365	16.4	结构化查询语言	532
12.18	小结	373	16.5	ADO.NET 对象模型	544
第 13 章	图形和多媒体	374	16.6	ADO.NET 编程:从数据库中提取信息	544
13.1	概述	374	16.7	ADO.NET 编程:修改数据库	552
13.2	图形上下文和图形对象	375	16.8	读写 XML 文档	558
13.3	颜色处理	376	16.9	小结	560
13.4	字体处理	382	第 17 章	ASP.NET, Web 窗体和 Web 控件	562
13.5	绘制线段、矩形和椭圆	386	17.1	概述	562
13.6	绘制弧线	388	17.2	简单的 HTTP 事务	562
13.7	绘制多边形和折线	390	17.3	系统结构	564
13.8	高级图形功能	394	17.4	创建和运行简单的 Web 窗体示例	565
13.9	多媒体简介	399	17.5	Web 控件	575
13.10	载入、显示和缩放图像	399	17.6	会话跟踪	591
13.11	让一连串图像动起来	401	17.7	案例分析(一):网上留言簿	606
13.12	Windows Media Player	411	17.8	案例分析(二):在 ASP.NET 中连接	612
13.13	Microsoft Agent	413	17.9	跟踪调试	623
13.14	小结	424	17.10	小结	625
第 14 章	文件和流	426	17.11	因特网和万维网资源	626
14.1	概述	426	第 18 章	ASP.NET 和 Web 服务	628
14.2	数据的层次结构	426	18.1	概述	628
14.3	文件和流	428	18.2	Web 服务	629
14.4	File 类和 Directory 类	428	18.3	简单对象访问协议(SOAP)和 Web	631
14.5	创建顺序访问文件	436	18.4	发布和使用 Web 服务	632
14.6	从顺序访问文件中读取数据	446	18.5	Web 服务的会话跟踪	644
14.7	随机访问文件	455	18.6	使用 Web 窗体和 Web 服务	654
14.8	创建随机访问文件	458	18.7	案例分析:一个 Web 服务(气温信息	660
14.9	向随机访问文件随机写入数据	461	18.8	Web 服务中的用户定义类型	668
14.10	从随机访问文件中顺序读取数据	465	18.9	小结	676
14.11	案例分析:交易处理程序	469	第 19 章	联网:流套接字和数据文报	677
14.12	小结	486	19.1	概述	677
第 15 章	XML 概述	487	19.2	使用流套接字创建简单的服务器	678
15.1	概述	487	19.3	使用流套接字创建简单的客户端	679
15.2	XML 文档	487	19.4	用流套接字连接进行客户端/	679
15.3	XML 命名空间	490	19.5	用数据文报进行无连接的客	687
15.4	文档对象模型	493		端/服务器通信	
15.5	文档类型定义、架构和验证	507			
15.6	XSL 和 XslTransform	517			
15.7	Microsoft 的 BizTalk	522			
15.8	小结	524			
15.9	因特网和万维网资源	525			
第 16 章	数据库、SQL 和 ADO.NET	527			
16.1	概述	527			

19.6 使用多线程服务器的客户端/ 服务器 Tic-Tac-Toe 游戏	691	附录 A 运算符优先顺序表	829
19.7 小结	703	附录 B 数值系统	831
第 20 章 数据结构和集合	704	B.1 概述	831
20.1 概述	704	B.2 将二进制数简写为八进制和十六进 制数	833
20.2 自引用类	704	B.3 将八进制数和十六进制数转换为 二进制数	834
20.3 链表	705	B.4 将二进制、八进制或十六进制转换 为十进制	834
20.4 堆栈	714	B.5 将十进制转换为二进制、八进制或 十六进制	835
20.5 队列	718	B.6 负的二进制数:2 的补值记号法	836
20.6 树	721	B.7 小结	837
20.7 集合类	733	附录 C Visual Studio .NET调试器	838
20.8 小结	748	C.1 概述	838
第 21 章 可访问性	750	C.2 断点	839
21.1 概述	750	C.3 检查数据	841
21.2 规章和资源	750	C.4 程序控制	843
21.3 Web 可访问性促进会	751	C.5 其他方法调试功能	845
21.4 为图像提供备用内容	752	C.6 其他类调试功能	848
21.5 突出结构以增强可读性	753	C.7 小结	850
21.6 Visual Studio .NET的可访问性	753	附录 D 在 Visual Studio .NET中生成文档	852
21.7 C#的可访问性	758	D.1 概述	852
21.8 XHTML 表格的可访问性	762	D.2 文档注释	852
21.9 XHTML 框架的可访问性	765	D.3 编写 C# 源代码文档	853
21.10 XML 的可访问性	766	D.4 创建注释 Web 页	859
21.11 使用 VoiceXML 的语音合成和识别	766	D.5 创建 XML 文档文件	861
21.12 CallXML	772	D.6 小结	865
21.13 JAWS for Windows	776	附录 E ASCII 字符集	867
21.14 其他可访问性工具	776	附录 F Unicode	868
21.15 Microsoft Windows 2000 的可访 问性	777	F.1 概述	868
21.16 小结	786	F.2 Unicode 转换格式	868
21.17 因特网和万维网资源	786	F.3 字符和字形	869
第 22 章 Mobile Internet Toolkit	789	F.4 Unicode 的优点和缺点	870
22.1 概述	789	F.5 Unicode 协会网站	870
22.2 客户端设备	789	F.6 使用 Unicode	871
22.3 Mobile Internet Toolkit 和移动 Web 窗体	790	F.7 字符范围	873
22.4 高级移动 Web 窗体控件	802	F.8 小结	874
22.5 示例:Deitel 无线门户 Web 站点	808	附录 G COM 集成	875
22.6 利用样式表和模板设计与设备无关 的 Web 站点	812	G.1 概述	875
22.7 通过移动应用程序享受 Web 服务	823	G.2 ActiveX 集成	875
22.8 小结	827	G.3 DLL 集成	879
22.9 因特网和万维网资源	828	G.4 小结	882
		G.5 因特网和万维网资源	883

附录 H HTML 4.0 概述(一)	884	J. 7 图像	931
H. 1 概述	884	J. 8 特殊字符和更多的换行	935
H. 2 标记语言	884	J. 9 无序列表	936
H. 3 编辑 HTML	884	J. 10 嵌套和顺序列表	937
H. 4 常用元素	885	J. 11 小结	940
H. 5 标题	887	J. 12 因特网和万维网资源	940
H. 6 添加链接	888	附录 K XHTML 概述(二)	942
H. 7 图像	890	K. 1 概述	942
H. 8 特殊字符和更多的换行	892	K. 2 简单的 XHTML 表格	942
H. 9 无序列表	894	K. 3 较复杂的 XHTML 表格和格式化	944
H. 10 嵌套和顺序列表	895	K. 4 简单的 XHTML 表单	946
H. 11 小结	898	K. 5 较复杂的 XHTML 表单	949
H. 12 因特网和万维网资源	898	K. 6 内部链接	955
附录 I HTML 4.0 概述(二)	900	K. 7 创建和使用图像地图	958
I. 1 概述	900	K. 8 meta 元素	960
I. 2 简单的 HTML 表格	900	K. 9 frameset 元素	961
I. 3 普通的 HTML 表格和格式化	902	K. 10 嵌套 frameset	964
I. 4 简单的 HTML 表单	904	K. 11 小结	965
I. 5 更复杂的 HTML 表单	906	K. 12 因特网和万维网资源	966
I. 6 内部链接	912	附录 L HTML/XHTML 特殊字符	967
I. 7 创建和使用图像地图	915	附录 M HTML/XHTML 颜色	968
I. 8 <meta>标记	917	附录 N 位处理	970
I. 9 frameset 元素	918	N. 1 概述	970
I. 10 嵌套式帧集	920	N. 2 位处理和按位运算符	970
I. 11 小结	922	N. 3 BitArray 类	980
I. 12 因特网和万维网资源	923	N. 4 小结	982
附录 J XHTML 概述(一)	924	附录 O Crystal Reports for Visual Studio .NET	983
J. 1 概述	924	O. 1 概述	983
J. 2 编辑 XHTML	924	O. 2 Crystal Reports 网上资源	983
J. 3 第一个 XHTML 示例	924	O. 3 Crystal Reports 和 Visual Studio .NET	983
J. 4 W3C XHTML 校验服务	926		
J. 5 标题	927		
J. 6 链接	929		

第 1 章 .NET和C#概述

学习目标

- 了解因特网和万维网的历史
- 熟悉万维网协会(W3C)
- 了解可扩展标记语言(XML)的定义及其重要性
- 理解对象技术对软件开发的影响
- 理解 Microsoft 的.NET战略
- 简介本书其余各章

1.1 概述

欢迎进入C#的世界!我们经过艰苦努力,希望为程序员提供有关C#语言和.NET平台的最精确、最全面的信息。希望本书不管在信息量、趣味性还是挑战性上,能让大家称心如意。本章介绍了因特网和万维网的历史,并介绍了 Microsoft 的.NET战略。本章最后对全书其余各章内容进行了概括。

1.2 因特网和万维网发展简史

20 世纪 60 年代末期,在伊利诺大学(Urbana-Champaign),ARPA(美国国防部高级研究项目局)展示了一份蓝图,打算将 ARPA 出资赞助的几十所大学和研究机构的主计算机系统连成一个网络。在当时看来,通信线路的速度是非常快的 56 Kbps(1 Kbps 等于每秒 1024 位)。当时,大多数人(只有极少人才能接触到网络技术)通过电话线的连接速度最高只能达到每秒 110 位!哈佛大学的研究人员开始谈论以后如何“跨越全国”,同犹它州立大学的“超级计算机”Univac 1108 通信,以加快本校计算机图形研究的速度。当然,其他大量可能的、令人心驰神往的应用也会浮出水面。在人们的印象中,学术研究似乎立即能向前跨出一大步。这次会议后不久,ARPA 果然很快着手这个网络的建设,而且人们很快开始把这个网络称为 ARPAnet(阿帕网)——它是今天的因特网的始祖。

不过,网络建成后,同计划稍有偏差的是,尽管 ARPAnet 确实允许研究人员相互共享计算机,但事实证明,它最主要的用途却是进行快速和简便的通信——“电子邮件”(E-mail)。在如今的因特网上,电子邮件同样是非常重要的一项应用。使用 E-mail,全球数亿人之间的通信变得异常简便和高效。

网络被设计成无需中心控制。也就是,即使网络的一部分瘫痪,剩下的部分仍能通过其他路径,将数据包从发送端传递到接收端。

在 ARPAnet 上,用来通信的协议(即一系列规则的统称)是 TCP(传输控制协议)。TCP 保证消息从发送端正确传递到接收端,而且收到的消息肯定是原封未动的。

世界各地与因特网的这个雏形并行发展的还有大多数单位自行开发的网络。这些网络要么在一个单位内部使用,要么在单位之间使用。那时,许多网络硬件和软件都开始出现。不过,由此也带来了一个挑战,即如何使这些不同的产品相互间通信。ARPA 为此开发了 IP(网际协议),创造出了一个真正的“网间网”,即今天因特网的基本结构。两种协议现在统称为“TCP/IP”。

最初,因特网只限于在各个大学和研究机构中使用。不久,军方成为它最大的用户。最后,政府决定向商业领域开放因特网。当时,许多研究机构及军事单位的开发人员还为政府的这项决定而顾虑重重——他们担心随着用户数量的增加,网络的响应速度会变慢。

事实却和人们想象的相反。商业用户很快便意识到,通过更有效率地利用因特网,能改进公司的运作,

为客户提供更新、更好的服务。为此,他们投入了大量资金,不断地对因特网进行开发与改进。与此同时,不管在商业电缆公司之间,还是在硬件及软件开发与制造公司之间,都展开了激烈的竞争,以满足不断增长的对基础结构的需求。结果便是,因特网的“带宽”(指通信线路的信息负载能力)得到了极大提高,同时硬件成本却在急剧下降。过去的 10 年间,因特网在许多国家的经济发展中都充当了重要的角色。

万维网(World Wide Web)使计算机用户能在因特网上查找并查看基于多媒体的、涉及几乎任何主题的文档。更妙的是,在这些文档中,可同时包含文字、图形、动画、声音以及视频信息。虽然因特网问世于 30 年前,但万维网的问世其实是在不久以前。1989 年,CERN(欧洲粒子物理实验室)的 Tim Berners-Lee 着手开发一项技术,目的是通过超链接的文本文档来共享信息。Berners-Lee 将他的发明命名为“超文本标记语言”(HTML),它基于成熟的“标准通用标记语言”(SGML)——这是商业数据交换的一项标准。另外,他还开发了相应的通信协议,为新的超文本信息系统建立了一个基本框架。这个系统就称为万维网。

毫无疑问,历史学家将会称因特网和万维网为人类最重要的、意义最深刻的少数几项发明之一。过去,大多数计算机都只能采用“单机”模式运行——相互间不能连接。今天却可轻易地同全球数亿台计算机通信。因特网和万维网将计算和通信技术合并到一起,加快并简化了我们的工作。它们使许多人能快速、方便地访问信息;它们使个人及小公司能向全世界展示自己;它们还改变了商业规则,并在一定程度上改变了人们的日常生活。

1.3 万维网协会(W3C)

1994 年 10 月, Tim Berners-Lee 创建了一个组织,名为万维网协会(W3C),致力于为万维网开发非专利的、可互操作的技术。W3C 的主要目标之一是使 Web 能被普遍访问——无论用户是否身有残疾,是否使用同一种语言以及是否具有同样的文化背景。

W3C 还是一个标准化组织,它有 3 大“根据地”——麻省理工学院(MIT)、法国国家信息与自动化研究院(INRIA)以及日本庆应(Keio)大学,另外还有 400 多名成员,其中包括 Deitel & Associates 公司。成员为 W3C 提供主要资金,并帮助协会制定战略决策。欲知 W3C 的详情,请访问 www.w3.org。

由 W3C 标准化的 Web 技术称为“建议规范”(Recommendations)。最新的 W3C 建议规范包括“可扩展超文本标记语言”(XHTML),用于创建 Web 内容;“层叠样式表”(CSS),用于描述如何格式化内容;以及“可扩展标记语言”(XML),用于创建标记语言。“建议推荐”不是实际的软件产品,只是一种定义技术的职能、语法及规则的文档。文档要成为一项 W3C 建议规范,必须经历 3 个主要阶段:“工作草案”(Working Draft),它规定了一种有待改进的草案;候选建议规范(Candidate Recommendation),这是文档的一个稳定版本,业界可开始实施;以及“提案”(Proposed Recommendation),它被认为已经成熟(即已经开始实施,并经过了一段时间的测试),准备好成为正式的 W3C 建议规范。欲详细了解 W3C 建议规范的评审历程,请参见以下网址处的 6.2 节“The W3C Recommendation track”:

www.w3.org/Consortium/Process/Process-19991111/process.html#RecsCR

1.4 可扩展标记语言(XML)

随着 Web 的迅速普及,HTML 的局限日趋明显。HTML 缺乏扩展性(即更改或添加特性的能力),大大地束缚了开发者。另外,由于定义模糊,滋生了大量有错误的 HTML。为解决这些问题,W3C 为 HTML 增添了有限的扩展性。但这只是一种临时性解决方案。显然,人们需要一种标准的、具有完全扩展能力的以及结构严谨的语言。为此,W3C 开发了 XML。XML 综合了其父语言 SGML(标准通用标记语言)的功能与扩展性,并根据 Web 社区的需求进行了相应的简化。

XML 的一项重要特征便是“数据独立性”,即内容与其呈现方式是分开的。由于 XML 文档只是描述数据,所以任何应用程序都能轻松处理这样的文档。鉴于此,许多软件开发者在自己的应用程序中集成 XML,以增强 Web 功能和互操作性。XML 的灵活性和强大功能使其特别适合作为客户机/服务器系统的中间层,因为这种系统必须与大量客户机进行交互。以前只能由服务器来完成的大量处理现在可由任何能

处理文本的应用程序来进行。如此一来,便减轻了服务器和网络通信的负担,使 Web 变得更快、更有效。

XML 不仅适用于 Web 应用程序。XML 正在逐渐应用于数据库——由于 XML 文档的特殊结构,使其可与数据库应用程序方便地集成。随着应用程序变得越来越“Web 化”,XML 有望成为数据表示的统一技术。运用了 XML 的所有应用程序相互间都能通信,只要它们能理解对方的 XML 标记——或称为“词汇表”。

简单对象访问协议(SOAP)是用于在因特网上传递数据(标记为 XML)的一种技术。它最初由 Microsoft 和 DevelopMentor 开发,目前是 W3C 的一份工作草案,目的是提供一个用于表示应用程序语义的、并对数据进行编码和包装的基本框架。Microsoft 的 .NET(详见 1.6 节和 1.7 节)使用 XML 和 SOAP 在因特网上标记和传输数据。XML 和 SOAP 都是 .NET 的核心,使软件组件能够相互操作(换言之,相互间可方便地通信)。SOAP 得到了许多平台的支持,因其基础是 XML 和 HTTP。第 15 章将详细讨论 XML,第 18 章将详细讨论 SOAP。

1.5 重要的软件趋势:对象技术

什么是对象技术,它们有什么特别之处?对象技术是一种打包方案,帮助我们创建有意义的软件单元。这些单元都比较大,集中于特定的应用领域,其中包括日期对象、时间对象、支票对象、发票对象、声音对象、视频对象、文件对象、记录对象等。事实上,几乎任何一个名词都可表示成一个对象。对象有“属性”(比如颜色、尺寸和重量),并可执行“动作”(即“行为”,比如移动、睡觉和绘画)。类表示相关对象的分组。例如,所有汽车都从属于 car 类,尽管各辆汽车的制造商、型号、颜色和可选配件是不同的。一个类指定了它的对象的常规格式;即一个类的对象具有通用的属性和动作。

我们生活在一个对象的世界,环顾四周便可体会到这一点。汽车、飞机、人、动物、建筑、红绿灯、电梯等,一切都是对象。在面向对象的语言问世之前,所有过程化语言(如 FORTRAN, Pascal, Basic 和 C)都将重点放在动作(动词)上,而不是放在对象上(名词)。程序员生活在一个对象的世界,却不得使用计算机,主要用动词编程。这种思维模式的强制转变,造成了编写程序的不便。但现在,由于可选用流行的面向对象语言(比如 C++, Java 和 C# 等),程序员可继续生活在一个充满着形形色色对象的世界,并在使用计算机时,采用自然的、面向对象的方式进行编程。这意味着他们能采用由接触世界而直接感知到的方式编写程序。与过程式编程相比,这样做显得更自然,而且能显著提高工作效率。

过程化编程的一个主要问题在于,创建的程序单元不能方便、有效地对应于现实世界的实体,因此难以重复使用。程序员经常要为不同的项目编写和改写类似的软件。重复性劳动浪费了大量宝贵时间和金钱。采用对象技术,只要设计得当,软件实体(称为“对象”)一经创建,就可重用于未来的项目中。采用由可重用组件构成的库,可减少实现特定种类的系统所需的精力(不必在新项目中从头实现这些功能)。C# 程序员使用的是 .NET 框架类库(FCL)——详情参见 1.8 节。

有公司报告,在采用了面向对象编程技术之后,软件重用所带来的效益并不大。相反,他们发现采用了面向对象编程技术后,生成的软件更易理解、更易管理,以及更易维护、修改和调试。之所以会出现这一现象,是基于一个公认的事实:根据估计,软件成本的 80% 和原先开发软件时付出的努力无关,而是在软件开发成功之后的生命期内,同后期的升级及维护工作紧密联系在一起。面向对象使程序员能对软件的细节进行抽象,并重点放在“大局”上。他们不必关心细节,只需关心对象的行为和交互。如果在地图上显示出每一棵树、每一幢房子以及每一条车道,这样的地图肯定是很难以辨认的,甚至根本无法使用。如果删除这些细节,只保留基本信息(路),地图会变得更易理解。同样地,划分为对象的一个程序容易理解、修改和升级,因其隐藏了这些细节。显然,面向对象编程至少在下一个 10 年间,会是一种关键的编程方法。

软件工程知识 1.1 使用“组装方法”来创建程序。尽量避免重复别人已经做过的工作。尽可能地利用现有的构建单元——这正是“软件重用”的主旨,也是面向对象编程的精髓所在。

注意:我们将在正文中列出许多这样的**软件工程知识**,指出会影响和改进一个软件系统(尤其是大型软件系统)总体结构及质量的关键概念。另外,我们还会强调一些**良好编程习惯**(帮助大家采用更清晰、更容