

Visual C++ .NET

A Managed Code Approach for Experienced Programmers

Visual C++ .NET

高级编程

包含
228 个 LIVE-CODE™
示例

介绍——
非托管代码

- Visual Studio® .NET IDE
- .NET Framework 类库
- 类 / 对象 / 接口
- 程序集 / 命名空间
- 继承性 / 多态性
- Web 服务 / ATL Server
- 数据库 / ADO.NET/SQL
- 异常 / 图形 / 多媒体
- GUI/Windows® 窗体 / 控件
- XML/XSLT™ / 文件和流
- 互操作性服务
- 多线程 / 网络连接

(美) Harvey M. Deitel 等著
Paul J. Deitel
郭凯 蔡飞 译



清华大学出版社

TP312
2015

Visual C++ .NET 高级编程

(美) Harvey M. Deitel 等著
Paul J. Deitel

郭凯 蔡飞 译

清华大学出版社

北 京

Simplified Chinese edition copyright © 2004 by PEARSON EDUCATION ASIA LIMITED and TSINGHUA UNIVERSITY PRESS.

Original English language title from Proprietor's edition of the Work.

Original English language title: Visual C++ .NET: A Managed Code Approach for Experienced Programmers, by Harvey M. Deitel, Paul J. Deitel et al., Copyright © 2003

EISBN: 0-13-045821-X

All Rights Reserved.

Published by arrangement with the original publisher, Pearson Education, Inc., publishing as Pearson Education.

This edition is authorized for sale only in the People's Republic of China (excluding the Special Administrative Region of Hong Kong and Macao).

本书中文简体翻译版由培生教育出版集团授权给清华大学出版社在中国境内(不包括中国香港、澳门特别行政区)出版发行。

北京市版权局著作权合同登记号 图字: 01-2003-4722

版权所有,翻印必究。举报电话: 010-62782989 13901104297 13801310933

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。

图书在版编目(CIP)数据

Visual C++ .NET 高级编程/(美)迪特尔(Deitel,H.M.)等著; 郭凯, 蔡飞译. —北京: 清华大学出版社, 2004.7
书名原文: Visual C++ .NET A Managed Code Approach for Experienced Programmers
ISBN 7-302-08634-6

I. V… II. ①迪…②郭…③蔡… III. Visual C++ .NET—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2004) 第 044451 号

出版者: 清华大学出版社 地址: 北京清华大学学研大厦
<http://www.tup.com.cn> 邮编: 100084
社总机: 010-62770175 客户服务: 010-62776969

组稿编辑: 曹康

文稿编辑: 侯彧

封面设计: 康博

版式设计: 康博

印刷者: 北京市清华园胶印厂

装订者: 三河市新茂装订有限公司

发行者: 新华书店总店北京发行所

开本: 185×260 印张: 94.5 字数: 2418 千字

版次: 2004 年 7 月第 1 版 2004 年 7 月第 1 次印刷

书号: ISBN 7-302-08634-6/TP·6189

印数: 1~3000

定价: 158.00 元

前 言

欢迎使用 Visual C++ .NET，欢迎来到使用 Visual Studio® .NET 和 .NET 平台的 Windows、Internet 和 Web 编程世界！本书是新版 Deitel™ Developer 系列之一，为软件开发人员和 IT 专业人员提供了最前沿的计算技术。

Microsoft 为 .NET 平台所极力推荐的 C++ 托管扩展(MC++)可以增强 Visual C++。MC++ 可以为托管编程提供强有力的支持。利用 MC++，程序员可以方便地访问 .NET Framework 所提供的强大的类，并且还能受益于其自动垃圾收集功能。Visual C++ .NET 为程序员提供了一些重要的功能，例如面向对象的编程、图形、图形用户界面(GUI)组件、异常处理、多线程、多媒体(包括音频、图像、动画和视频)、文件处理、预包装数据结构、数据库处理、网络、Web 服务与分布式计算【注：由于 C++ 与 Visual C++ 具体的实现方式不一样，因此术语上的表示会有所区别。本书中的 C++ 一般指标准 C++。C++ 托管扩展(MC++)一般指托管 C++ 的功能或使用了这些功能的代码。如谈到 Microsoft 的产品或编译器(用来编写托管和非托管代码)，我们将使用术语 Visual C++ .NET】。

.NET 平台为软件开发和部署提供了强大的功能，例如语言和平台无关性。例如，开发人员使用任何一种 .NET 语言(MC++、C# 或者 Visual Basic .NET)编写的代码可以为同一种软件产品提供组件。由于 .NET 应用程序可以驻留在多个平台上并且可以跨平台通信，因此 .NET 不仅有语言上的互操作性功能，还可以扩展程序的可移植性。这样在创建和应用 Web 服务(一种通过 Internet 为客户提供功能的应用程序)时就非常方便。Microsoft 合并到 .NET 平台中的功能能够提高程序员的开发效率并减少开发时间。

目前 Visual C++ .NET 中没有在其他 Visual Studio 产品(比如 Visual C# .NET)中都包含的 Windows 窗体设计器。通过自动生成代码，这种窗体设计器可为程序员简化创建 GUI 的过程。基于这个原因，本书中的所有 GUI 都是利用 C# 创建的，然后自动生成的代码可以转换为 MC++。对于 Visual C++ .NET 来说，Microsoft 目前所使用的 Windows 窗体设计器是 β 版本。本书的后续版本中将会使用公开发布的 GUI 工具。

本书读者对象

Deitel&Associates 有限公司有两种 Visual C++ .NET 出版物，它们面向不同的读者。在 www.deitel.com 上以及本书中所提供的信息可以帮助确定哪一种更适合于您。

本书是 Deitel™ Developer 系列之一，也是 For Experienced Programmers 子系列之一，该子系列面向有经验的软件开发人员，他们一般都不希望介绍太多的基础内容，而是可以直接切入正题深入了解新的技术。本书将深入讨论一些更为高级的主题。

于 2003 年秋季面世的 *Visual C++ .NET How to Program* 是我们的 How to Program 系列之

一，面向各个学院和综合性大学的学生。该书对 Visual C++ .NET 进行了全面介绍，并提供了学习帮助和许多辅助支持。*Visual C++ .NET How to Program* 假定读者具备很少或几乎不具备编程经验。前面几章主要介绍基本的编程原理。该书根据这些基础知识创建了各种逐步复杂和高级的程序，以说明如何利用 Visual C++ .NET 来创建图形用户界面、网络应用程序、多线程应用程序和基于 Web 的应用程序等。

这些 Visual C++ .NET 书籍都提供了许多可立即执行的完整程序，并且描述了在运行程序时的实际输入和输出。这就是我们的标志性 LIVE-CODE™ 方法——我们在完整的工作程序环境中展示所有的概念。每本书的源代码都可以通过 www.deitel.com 免费下载。

请比较 Deitel™ Developer 系列和 How to Program 系列，以确定哪一系列更适合您。您可以在我们的新 Deitel™ BuzzOnline e-mail 时事通讯(www.deitel.net/newsletter/subscribe.html)上进行注册，以便给您提供与出版物有关的信息、公司通告、精彩技术文章的链接、编程技巧、教学技巧、疑难问题和趣闻轶事等。

如果您愿意与我们联系，请将 e-mail 发至 deitel@deitel.com，我们会迅速给您回复。请访问我们的 Web 站点 www.deitel.com、www.prenhall.com/deitel 和 www.InformalT.com/deitel，以获得定期的更新、勘误表、常见问题的解答等。在发送 e-mail 时，请附上书名和版本号。我们诚挚地希望您能使用我们的出版物愉快地学习 Visual C++ .NET。

本书特色

本书有许多特色，具体如下所示：

- **Web 服务。** Microsoft 的 .NET 战略将 Internet 和 Web 也作为软件开发和部署的一部分。Web 服务技术利用标准的 Internet 协议和技术(例如超文本传输协议 HTTP、可扩展标记语言 XML 和简单对象访问协议 SOAP)来实现信息共享、电子商务和其他活动。利用 Web 服务，程序员能以某种方式将应用程序功能打包，从而使得 Web 成为可重用软件组件的一个库。第 17 章“Web 服务”中所提供的 Web 服务可以让用户处理“大整数”——一类由于太大而无法以 .NET 内置数据类型进行保存的整数。在这个例子中，用户可以输入两个大整数，然后单击按钮调用 Web 服务以对这两个整数进行加减以及比较运算。
- **面向对象的编程。** 面向对象编程是在开发健壮可重用的软件时应用最为广泛的技术。本书对 MC++ 的面向对象编程特性进行了详细讨论。第 5 章“基于对象的编程”介绍了如何创建类和对象，这些概念在第 6 章“面向对象的编程：继承性”中得到了扩展，这一章主要是讨论程序员如何“提取”出已有类的功能，从而快速创建出强大的新类【注：本书假定读者具有一些 C++ 指针方面的知识】。
- **XML。** 可扩展标记语言(XML)在软件开发、电子业务、电子商务等方面的应用日益广泛，更是渗透到了整个 .NET 平台中。因为 XML 是一种独立于平台的、可描述数据和创建标记语言的技术，所以 XML 的数据可移植性与基于 MC++ 的应用程序和服务结合得很好。第 15 章“可扩展标记语言”就具体介绍了 XML。在这一章中，我们对 XML 和文档对象模型(DOM)进行了讨论，DOM 通过编程的方式来处理 XML 文档。

- **多线程。**程序员利用计算机可以同时(也就是所说的并发)执行许多任务,例如打印文档、从网上下载文件并且进行网上冲浪等。程序员利用多线程技术可以开发出能够执行并发任务的应用程序。过去,计算机中惟一的处理器很昂贵,它的操作系统需要在所有的应用程序中共享这个处理器。现在处理器越来越便宜,我们可以利用带有多个处理器的计算机来进行并发操作——我们将这种计算机称之为多处理器。多线程在单处理器和多处理器系统中都很有用。.NET 平台的多线程功能有助于平台和与之相关的技术更好地处理多媒体密集的、数据库密集的、基于网络的、基于多处理器的分布式应用程序。第 11 章“多线程”就介绍了这种强有力的功能。
- **ADO.NET。**数据库存储了个人或单位进行商业往来时需要访问的大量信息。随着 Microsoft ADO 技术的发展,新面世的 ADO.NET 可以以新的方式构建与数据库进行交互作用的应用程序。ADO.NET 为开发人员提供了 XML 和经过增强的对象模型,利用这些工具可以在运行可扩展的、任务关键的多层大型应用程序时访问和处理数据库。第 16 章“数据库、SQL 和 ADO.NET”介绍了处理数据库时的 ADO.NET 和 SQL 语言的功能。
- **非托管代码。**非托管代码是指不能使用 CLR(公共语言运行库)所提供服务的代码。CLR 是指 .NET 运行环境,它可以提供一些基本的服务,例如垃圾回收、对 .NET 应用程序进行数组边界检查等。虽然本书重点是介绍托管代码,但第 21~25 章还是对 .NET 中的非托管代码进行了深入阐述。这些章节谈到了利用 Visual C++ .NET 对非托管代码的改变和增删,包括特性化编程(可以让程序员方便地将功能块插入到程序中)、ATL 服务器(可以利用 Visual C++ .NET 创建 Web 应用程序)。后面两章为读者介绍了托管和非托管代码间的 .NET 互操作。Microsoft 花费了很大的心血以确保原来的 Visual C++ 代码能与托管代码共存,而不用再去重写这部分代码。
- **图形用户界面。**本书提供了一些 GUI(图形用户界面)应用程序,以说明 Visual C++ .NET 中的不同主题。GUI 应用程序是显示图形元素(例如用户可以与之交互的按钮工标签)的程序。Visual Studio .NET 中包含了可用于 C# 和 Visual Basic .NET 的内置 GUI 开发工具,但没有用于 Visual C++ .NET 的相应工具。为了创建本书中的图形用户界面,我们将 C# 代码(由 Visual Studio .NET IDE 所创建)转换为 Visual C++ .NET 代码。我们带读者一起检查这些代码,读者就可以用 Visual C++ .NET 创建自定义的图形用户界面了。从而,就可以理解如何创建易懂易用的程序。
- **XHTML™。**万维网协会(W3C)宣称 HTML 将成为不再继续开发的技术。HTML 将被可扩展超文本标记语言(XHTML)——一种将很快成为描述 Web 内容的标准的基于 XML 的技术——所代替。在第 15 章“可扩展性标记语言”中我们提到了 XHTML,并在附录 E“XHTML 介绍:第 1 部分”和附录 F“XHTML 介绍:第 2 部分”里面对此进行了具体介绍。这两个附录中还谈到了这门新兴标记语言的标题、图像、列表、图像映射及其他功能。
- **可访问性。**虽然 Web 已经成为现在人们生活中的重要部分,但该媒体对于残疾人士还是有很多不方便的地方。特别是对于有听觉和视觉缺陷的人来说,访问多媒体资源丰富的 Web 站点还是件困难的事情。为了改变这种现状,W3C 提出了一个 WAI(Web Accessibility Initiative)计划,旨在为有困难的人群访问 Web 站点提供指南。第 20 章“可访问性”就介绍了这些指导原则,

并介绍了可以提高残疾人士的 Web 浏览体验的各种产品和服务。例如，这一章介绍了 VoiceXML 和 CallXML 这两种基于 XML 的技术，以提高基于 Web 内容的可访问性，从而方便有视觉缺陷的人群。

- **Unicode®**。由于计算机系统已经延伸到全世界的范围中，计算机厂商将不同国家的当地语言的字符集和特殊符号用数字表示出来。在有些情况下，同一种语言还有多种不同的表示方式。这种完全不同的字符集就会妨碍计算机系统之间的交流。Visual C++ .NET 运行 Unicode® 标准(由 Unicode® Consortium 这一非营利性组织制定)，该标准所制定的一个字符集为世界上大多数语言中的各个字符以及特殊符号都指定了惟一的数字值。附录 D “Unicode®” 中就讨论了这种标准，并介绍了 Unicode® Consortium 的 Web 站点：www.unicode.org，同时通过一个 MC++ 应用程序用多种语言来显示出 “Welcome to Unicode” 的字样。
- **位操作**。计算机以二进制数字或二进制位(假定值为 1 或 0)来处理数据。计算机电路执行各种简单的位操作，例如检查位的值、设置位的值和颠倒位的值(从 1 到 0 或从 0 到 1)。操作系统、测试设备、网络软件以及许多其他软件都需要通过使用位操作来使程序能够 “与硬件进行直接交流”。附录 I “位操作” 中就介绍了 .NET Framework 所提供的位操作功能。

教学方式

本书包含丰富的示例，这些示例代码已经在 Windows 2000 和 Windows XP 上经过了测试。本书集中讲述优秀的软件工程的原则，强调程序的清晰性。我们在课堂上讲述的是实践，避免使用深奥的术语和繁杂的语法，而代之以实例教学。

实例代码教学方式

本书收集了许多实例代码。这种风格体现了我们一贯的教学原则，也是我们的网络教室和基于 Web 的培训的核心理念。每个新的概念都用一个完整的、能够运行的例子来阐述，辅以程序的输入输出窗口来增加理解。我们把这种教学方式称为 “实例代码方式” (LIVE-CODE™ Approach)。我们使用编程语言来讲授编程语言。阅读书中的示例代码颇像在计算机上输入并运行代码。读者也可以通过我们的网站 www.deitel.com 上的 Downloads/Resources 链接下载本书的全部示例代码。其他链接提供了勘误表和常见问题的答案。

Web 访问

本书及我们的其他书籍的所有源代码都可以从以下 Web 站点下载：

www.deitel.com

www.prenhall.com/deitel

注册一下很方便也很快捷。建议您下载所有的示例，然后在学习本书中相应部分时运行对应的程序。对示例进行修改后，您立即就可以看出这些修改所产生的效果——这样可以有效地提

高您的编程技巧。这些示例中都假定用户的操作系统是 Windows 2000 或 Windows XP，并使用了 Microsoft 的 Internet Information Services(IIS)。如果需要安装 IIS 或其他软件，可从下载示例的 Web 站点中找到设置方法【注：这已经被授予版权，您在学习时可以免费使用，但如果没有 Prentice Hall 和作者的明确许可，您不能转载其中的任何内容】。

Visual Studio .NET 是可以从 Microsoft 购买或下载的产品之一。Visual Studio .NET 包括了 4 种不同版本的 Visual C++ .NET——Academic、Professional、Enterprise Developer 和 Enterprise Architect。Visual Studio .NET Academic 中包含了 Visual Studio .NET Professional 的功能(为学生和老师所设计的功能除外，例如管理作业提交的 Assignment Manager、发布作业通知的 Application Publishing Tool 以及代码示例等)。

Visual Studio .NET 提供了一个强大的调试器，它结合了先前 Visual Studio 调试器的各方面优点，并且允许程序员调试包含托管代码和非托管代码的应用程序。与 Visual Studio .NET 相关的更多介绍和应用信息，可参见下列网址的内容：

msdn.microsoft.com/library/default.asp?url=/library/en-us/vsdebug/html/vc_debugging_your_application_home_page.asp

Microsoft 还为各种 .NET 语言提供了单独的产品(Visual C++ .NET Standard、Visual C++ .NET Standard 和 Visual Basic .NET Standard)。每种产品都提供了一个集成的开发环境(类似于 Visual Studio .NET)和一个编译器。访问 msdn.microsoft.com/vstudio/howtobuy 可获得相应描述和订购的信息。

目标

每一章的开头都提出了读者学习本章的目标，每学习完一章，读者都可以对照这些目标来确定自己是否已经掌握了本章的内容。

概要

每章的概要可使读者先大致了解本章的内容。根据每章的目标，读者可按照概要来判定最有效的学习步骤。

接近 34, 131 行代码和 228 个示例程序(带有程序输出)

在本书中，我们使用许多有效的完整程序来介绍 Visual C++ .NET 的特性。这些程序有大有小。读者可从我们的 Web 站点，www.deitel.com 中下载所有的示例。

884 幅图

本书中包含丰富的图表，其中包含大量的图、代码和程序输出。

388 个编程技巧

我们在本书中包含了大量的编程技巧，读者可通过这些技巧将注意力集中在程序开发的重要方面。这些编程技巧的介绍形式包括：良好的编程习惯、常见的编程错误、测试和调试技巧、

有关性能的技巧、可移植性技巧、软件工程经验和视觉和感觉经验。这些技巧都是作者根据长时间的编程和教学经验所得出的结论。只有掌握这些技巧，读者才有可能更快速地编写出优秀的软件。

58 个良好的编程习惯

良好的编程习惯提醒用户注意一些技术，这些技术可帮助开发者编写出更清晰、更易懂和更易于维护的程序。

32 个常见的编程错误

向读者提示一些学习该编程语言时经常会犯的错误。预先指出这些常见的编程错误有助于读者减少犯这些错误的可能性。

26 个测试和调试技巧

通过这些技巧，读者可防止程序中产生许多 bug，从而简化测试和调试的过程。

46 个性能技巧

我们在本书中提供了 46 个性能技巧来帮助读者改进程序的性能——使程序运行得更快，或者减少程序所占用的内存。

12 个可移植性技巧

可移植性技巧帮助开发者编写可移植的代码，同时也介绍了 Visual C++ .NET 高度的可移植性。

93 个软件工程的经验

软件工程的经验主要介绍了一些影响构建软件系统(特别是大规模的系统)的体系结构和设计方面的问题。

21 个视觉和感觉的经验

视觉和感觉的经验突出了图形用户界面的一些约定。这些经验有助于读者设计出吸引人的、友好的图形用户界面，这些图形用户界面都严格遵循行业规范。

小结

每章的结尾都包括本章的小结，读者可通过这些小结来回顾和强化关键的概念。

本书概述

在本节中，我们将介绍本书的全部章节和附录。除了每章所要阐述的主题外，一些章节还包含 Internet 和 Web 资源的相关介绍。在这些章节中列出了额外的资源，读者可以通过学习这

些资源来加强自己的 Visual C++ 程序设计知识。

第 1 章——.NET 和 Visual C++® .NET 简介

第 1 章介绍了 Internet、万维网以及各种推动计算机发展的技术(如 XML 和 SOAP)的历史。其中介绍了 Microsoft .NET 计划和 Visual C++ .NET, 包括 Web 服务, 并且探讨了 .NET 对软件开发和软件重用的影响。

第 2 章——Visual Studio® .NET IDE 和 Visual C++ .NET 编程

第 2 章介绍了 Visual Studio .NET, 这是一个允许程序员用标准 C++ 和 C++ 托管扩展(MC++) 创建应用程序的集成开发环境(IDE)。Visual Studio .NET 包含了代码调试和编写工具。本章介绍了 Visual Studio .NET 的特性(包括它的主窗口)并展示了如何编译和运行程序。本章还向读者介绍了 MC++ 中的控制台应用程序编程。每一个概念都会用一个完整的 MC++ 工作程序来说明, 随后用一张或多张屏幕截图来展示程序执行时的实际输入和输出结果。这是我们的实例代码(LIVE-CODE™)方法。我们还讨论了一些基本操作, 如用户如何输入程序数据和编写数学表达式。

第 3 章——控制结构

这一章介绍了结构化编程的原理, 这是一组帮助读者开发清晰的、易理解的和可维护的程序的技术。然后, 本章介绍了如何使用影响语句执行顺序的控制结构。控制结构能产生易理解的、易调试的和易维护的程序。我们讨论了程序控制的三种形式——顺序、选择和循环——即 if/else, while, for, do/while 和 switch 结构。解释了 break、continue 语句和逻辑运算符。在前两章的基础上, 创建了交互式程序(它们可以根据用户提供的输入来改变自己的行为)。

第 4 章——函数和数组

函数是程序员创建的一个代码段, 可以从一个程序的多个点调用它。大型的程序可以分成若干交互的类, 每一个类包含若干函数——这种方式有时称为“分而治之”策略(“divide and conquer” strategy)。程序可以分成许多以简明方法交互的简单组件。我们讨论了如何创建能够进行数据输入、执行计算和返回输出的函数, 并且介绍了递归函数(能够调用自己的函数)和函数重载(允许多个函数拥有相同的名称)。我们分别创建了两个求平方值的函数, 一个针对整数, 一个针对浮点数。我们用这两个函数解释了函数重载。本章还介绍了本书的第一个数据结构——数组。数据结构对大规模数据的存储、分类、查找和处理是至关重要的。数组是一组相关的数据项, 程序员可直接访问它的任何一个元素。程序员能够创建一个包含 100 个元素的数组, 并根据它们在数组中的位置来访问这些元素, 这样做要优于创建 100 个单独的变量并以某些方式将它们联系起来。我们讨论了如何声明和分配托管数组, 并利用前面章节中的技术将数组传递给函数。第 3 章提供了讨论数组的重要背景, 因为循环结构可用于遍历数组中的元素。这些概念的组合可以帮助读者创建高度结构化和组织合理的程序。我们还讨论了多维数组(矩形数组和嵌套数组)。多维数组可以用来存储数据表。

第 5 章——基于对象的编程

第 5 章介绍了对象和类。面向对象技术使软件开发进步迅速, 它允许程序员创建可重用的

软件组件。对象允许程序员以一种自然的、直观的方式组织程序。本章介绍了基于对象的程序设计基础，如封装、数据抽象和抽象数据类型(ADT)。这些技术隐藏了组件的细节，从而程序员只需注意“大的框架”。我们创建了 Time 类，它以标准通用的格式显示时间。我们展示了如何用程序集、命名空间和动态链接库(DLL, dynamic-link-library)来创建可重用的软件组件。我们创建了类和名称空间，并且讨论了特性和 const 关键字。本章为后两章介绍面向对象的程序设计打下了基础。

第 6 章——面向对象的编程：继承性

在本章中，讨论了继承性——这是一种通过继承已有类(基类)的特性和方法来创建新类(称为派生类)的软件重用形式。继承的类(也就是派生类)还可以有额外的特性和方法。我们展示了如何通过寻找对象类的共性来减少构建大型软件系统所需的工作量。本章给出了查找三层继承关系(Point、Circle 和 Cylinder 类)共性的具体示例，并且通过分析这个具体示例来说明软件重用和良好的编程技术。我们讨论了面向对象程序设计的软件工程的好处。本章还介绍了重要的面向对象程序设计概念，如创建类和扩展类。

第 7 章——面向对象的编程：多态性

第 7 章继续介绍面向对象的程序设计。我们讨论了多态程序设计和它的优点。多态性允许以一般的方式来对待类，同样的函数调用可以根据不同接收信息的对象而产生不同的行为(如，发送给鸟和鱼的“移动”信息会导致截然不同的运动类型——鸟在飞而鱼在游)。除了以一般的方式来对待现有的类，多态性还允许将新类很容易地加入到一个系统中。我们需要识别在何种情况下多态性是非常有用的。我们用一个工资系统示例来说明多态性——这个系统根据雇员类型来决定每个雇员的薪水(老板：付固定薪水；计时工人：按小时付工资；销售人员：基本工资加销售提成；计件工人：按生产的每件产品付酬)。这些程序设计技术和前面章节中介绍的技术让程序员能够创建可扩展和重用的软件组件。

第 8 章——异常处理

从构建任务级和商业级应用程序的观点来看，异常处理是 Visual C++ .NET 的一个重要内容。用户可能输入不正确的数据，数据可能被破坏，客户端可能访问不存在的纪录或受限制的纪录。一个除以零的简单错误就会导致计算程序崩溃，但是如果类似的错误出现在飞机的导航系统上(并且该飞机正在飞行)会出现什么情况呢？在一些情况下，程序故障将带来灾难性的结果。程序员需要知道如何辨认出软件组件中可能会产生的错误(异常)并进行有效的处理，从而使程序不会崩溃并能够处理问题和继续执行。利用其他程序员编写的可重用软件组件来构建软件系统的程序员必须要处理这些组件可能会“抛出”的异常情况。本章详细介绍了 MC++ 异常、异常处理的中断模型、抛出和捕获异常以及 FCL 类 Exception。

第 9 章——图形用户界面概念：第 1 部分

第 9 章介绍了设计图形用户界面的技术。讨论了如何用 Windows Form 控件(如标签、按钮、文本框和图形框)来创建用户界面。还介绍了事件，事件是由程序向一个或多个对象发送的消息，指示某个动作已经发生。事件通常用于指明用户与 GUI 控件间的相互作用，但也可以发出程序内部动作的信号。我们概括了事件处理并讨论了如何处理控件、键盘和鼠标事件。本章还介绍

了一些帮助编程者创建视觉上吸引人的、组织有序的和一致的 GUI 的技巧。

第 10 章——图形用户界面概念：第 2 部分

第 10 章介绍了更为复杂的 GUI 组件，包括菜单、链接标签、面板、列表框、组合框和标签控件。本章介绍了多文档界面(MDI)的程序设计，它允许在一个单独的 GUI 中同时打开多个文档(如窗体)。我们介绍了可视化继承，它使程序员能够将本章中介绍的 GUI 概念和第 6 章中介绍的面向对象概念结合起来开发用户界面，所开发的界面可以被其他程序员使用和扩展。本章的结尾讨论了如何创建用户自定义的控件。

第 11 章——多线程

我们希望应用程序能做很多的事。希望从 Internet 上下载文件、听音乐、打印文件和浏览 Web——所有这些都是同时进行的。为了达到这个目的，我们需要称为多线程的技术，该技术允许应用程序能够并发执行多个任务。MC++允许程序员访问 FCL 提供的多线程类，而程序员不需要知道其中的复杂细节。与其他不具有多线程特性的语言相比，MC++能更好地处理复杂的多媒体应用程序、网络应用程序和多处理器应用程序。本章概括了 FCL 中的线程类，并介绍了线程、线程生命周期、时间片、线程调度和优先级。我们分析了生产者-消费者的关系、线程同步和环状缓冲区(circular buffer)。本章为创建令人难忘的多线程程序打下了基础，这些程序都是客户端所需要的。

第 12 章——字符串，字符和正则表达式

在本章中，我们讨论了字、句和字符集合的处理。在 MC++中，字符串(字符的集合)是 String *类型的对象。这也是 MC++强调面向对象程序设计的另一个优点。String *类型的对象包含了复制、查找、提取子串和连接字符串的方法。我们介绍了 StringBuilder 类，该类定义了类似字符串的对象，这些对象在初始化后可以被修改。我们创建了一个洗牌和发牌的模拟程序作为字符串的一个有趣示例。还讨论了正则表达式，这是一个查找和处理文本的强有力工具。

第 13 章——图形和多媒体

在这一章中，讨论了 GDI+(图形设备界面 GDI 的一种扩展)，它是提供 .NET 应用程序使用的图形特性的 Windows 服务。GDI+的扩展图形能力能够更直观和有趣地创建和使用程序。我们讨论了 Visual C++ .NET 中对图形对象和颜色控件的处理。还讨论了如何画圆弧、多边形和其他形状。本章还解释了如何使用各种笔和刷来创建颜色效果，并且使用了一个演示渐变色填充和纹理的示例。还介绍了一种将纯文本应用程序调整为令人赏心悦目的应用程序的技术，即便是新手也能轻而易举地编写出这种程序。本章的第 2 部分重点介绍了音频、视频和语音技术。我们讨论了将声音、视频和动画人物插入程序中(主要通过现有的音频和视频片断)。您会发现将多媒体加入 Visual C++ .NET 程序中是一件很容易的事。本章介绍了一种可以将交互动画人物插入程序中的技术，称为 Microsoft Agent。每一个人物都允许用户和应用程序以一种更为自然的人类通信技术(如语言)来进行交互。代理人物对鼠标和键盘事件做出响应，并且说话和倾听(也就是说，它们支持语音合成和语音识别)。由于具有了这些能力，应用程序可对用户说话，并对他们的语言命令及时做出响应。

第 14 章——文件和流

设想一个不能将数据存入文件的程序。一旦程序被关闭，所有程序执行的工作将会永远消失。因此，对开发商务应用程序的程序员而言，本章是一个重点。我们介绍了用 FCL 类来输入和输出数据。本章通过一个具体示例来解释这些概念。这是一个允许用户从文件中读取银行账户信息，并且将银行账户信息写入文件中的示例。我们介绍了有助于执行输入、输出操作的 FCL 类和方法——它们展示了面向对象程序设计和重用类的能力。我们讨论了顺序文件、随机存取文件和缓冲的优点。本章为第 18 章中的内容打下了基础。

第 15 章——可扩展标记语言

可扩展标记语言(XML)起源于 1986 年成为行业标准的 SGML(Standard Generalized Markup Language)。尽管 SGML 用于在全球范围内发布应用程序，但鉴于它的实际大小和复杂性，SGML 未能融入主流的程序设计团体中。XML 是为了让类似于 SGML 的技术可被更广泛使用而做的一个尝试。W3C 所创建的 XML 以可移植的格式来描述数据。XML 与诸如 HTML 的标记语言在概念上就不相同。诸如 HTML 的标记语言仅仅描述了如何在一个浏览器上显示信息。XML 是一种为任何类型的信息创建标记语言的技术。文件作者使用 XML 创建全新的标记语言，并使用这种标记语言来描述特殊的数据类型，包括数学公式，化学分子结构，音乐和配方等。用 XML 创建的标记语言包括 XHTML(Extensible HyperText Markup Language，用于 Web 内容)、MathML(用于数学领域)、VoiceXMLTM(用于语音)、SMILTM(Synchronized Multimedia Integration Language，用于多媒体演示)、CML(Chemical Markup Language，用于化学领域)和 XBRL(Extensible Business Reporting Language，用于金融数据交换)。XML 的可扩展性使它成为今天工业领域中最重要技术之一，它被集成到几乎所有的领域中。公司和个人会经常发现 XML 的新用途。在这一章里，我们举例说明了使用 XML 标记数据的基础知识，还演示了 XML 衍生出的几种标记语言，如 XML 模式(用于检查 XML 文件的语法)和 XSLT(Extensible Stylesheet Language Translations，用于将 XML 文件的数据转换为其他纯文本格式，如 XHTML)(我们为不熟悉 XHTML 的读者准备了附录 E 和 F，这两个附录详细介绍了 XHTML)。

第 16 章——数据库、SQL 和 ADO .NET

数据存储和访问是创建功能强大的软件应用程序所不可缺少的。本章讨论了 .NET 对数据库操作的支持。今天，最流行的数据库系统是关系数据库。在本章中，我们介绍了关系数据库中执行查询操作的结构化查询语言(SQL, Structured Query Language)，并且还介绍了 ActiveX Data Objects(ADO.NET)——能够让 .NET 应用程序访问和操作数据库的 ADO 的扩展。ADO.NET 允许数据作为 XML 导出，它能够让使用 ADO.NET 的应用程序与能够理解 XML 的各种程序进行通信。我们向读者展示了如何创建数据库连接以及如何使用 ADO .NET 类来查询数据库。

第 17 章——Web 服务

前面的章节介绍了如何创建仅在用户计算机上执行的应用程序。在这一章中，介绍了 Web 服务。Web 服务是一些能够通过 Internet、内部网和外部网向客户端开放服务(方法)的程序。Web 服务允许不同平台上的服务无缝地相互作用，从而提供了增强的软件重用能力。我们讨论了 .NET Web 服务的基础知识和相关技术，包括简单对象访问协议(SOAP)和 ASP(动态服务器页，

Active Server Pages).NET。本章提供了一个处理巨大整数(一直到 100 位)的 Web 服务的有趣示例。还给出了一个演示会话跟踪的 21 点游戏(Black Jack)应用程序。会话跟踪是一种使应用程序能够“识别”用户的个性化形式。我们在本章结尾讨论了 Microsoft 的全局 XML Web 服务体系结构(GXA)，这是一系列向 Web 服务开发者提供了额外能力的规范。

第 18 章——网络连接：流式套接字和数据报

第 18 章介绍了流式网络连接的基础技术。说明了流式套接字如何使程序员可以隐藏许多网络连接的细节。通过利用套接字，程序员进行网络连接将如同读写文件般简单。我们还介绍了数据报。通过利用数据报，可以在程序间发送信息包。每一个包都包含了接收方的地址，它们被发送到网络上，而网络则将包沿着某一条路线发送到目的地。本章的例子注重于应用程序间的通信。一个例子演示了用流式套接字在两个 MC++ 程序间通信。另一个相似的例子则在两个应用程序间发送数据报。我们还展示了如何创建能在多个客户端间并行通信的多线程服务器应用程序。在这个客户端/服务器的井字游戏(tic-tac-toe game)中，服务器维持着游戏的状态，而客户端则通过和服务器通信来玩这个游戏。

第 19 章——数据结构和数据集合

本章介绍了将数据排列成链表、堆栈、队列和树的集合体。从排列元素到记录函数调用，每一个数据结构都有用于多种应用程序的属性。我们讨论了如何构建每一种数据结构。这对构思有用的类是非常有价值的经验。另外，我们还涉及了 FCL 中预建的集合类。这些类储存数据集或集合，并提供了排列、插入、删除和检索数据项的功能。不同的集合类用不同的方法存储数据。本章注重于 Array、ArrayList、Stack 和 Hashtable 类，并讨论了每一个类的细节。如果可能，MC++ 程序员应该使用 FCL 来寻找合适的数据结构，而不要自己构造这些数据结构。本章强化了在第 5~7 章中讨论的许多面向对象技术，包括类、继承和组合(composition)。

第 20 章——可访问性

万维网给残疾人带来了挑战。多媒体网站很难由文本阅读器和程序进行解释；因此，听力和视力受损者在浏览这些网站时会遇到困难。为了改善这种情况，W3C 发布了 Web 可访问性措施(WAI, Web Accessibility Initiative)，该措施提供了让残疾人能够访问网站的指导方针。本章叙述了这些方针，如使用 headers 元素来使网页阅读器更容易访问表，使用 img 元素的 alt 特性来描述图像，以及联合使用 XHTML 和 CSS 以确保网页可在所有显示器上显示或被所有的阅读器读取。我们举例说明了 Visual Studio .NET、Internet Explorer 6.0 和 Windows XP 中主要的可访问性特性。还介绍了 VoiceXML 和 CallXML 这两种增强网页内容可访问性的技术。VoiceXML 帮助视力损伤者通过语音合成和语音识别来访问万维网。CallXML 允许视力损伤者通过电话访问网页内容。

第 21 章——Visual C++ .NET 的非托管代码介绍

本章介绍了 Visual C++ .NET 中非托管方面的知识。强调了托管代码和非托管代码的区别，讨论了非托管代码如何适应 Microsoft.NET 策略，并且推断了非托管代码可能的未来。另外，还讨论了新的统一事件模型(Unified Event Model)，该模型提供了处理 Visual C++ .NET 事件的简化语法。本章介绍了对非托管代码的改进，即增加了对 ANSI/ISO 标准的兼容性，以及对本

地代码宽字符和 64 位处理的支持。本章有一个例子展示了新编译器的特点，即提供了程序进入本地代码调试模式的运行时检查。学习本章(以及 22~25 章)需要具备标准 C++(包括指针)的知识。

第 22 章——ATL/COM 中的特性化编程

特性化程序设计是 Visual C++ .NET 的一个新特性，它在 COM 开发领域中可用于简化通用程序设计任务。在这一章中，讨论了特性如何简化 COM 和 ATL。本章是针对具备旧 COM 和 ATL 知识的开发者，并且用来说明 COM 中特性化程序设计的功能。尽管特性能简化 COM 开发，但对 COM 基础知识的可靠理解是正确使用特性所必需的。本章的例子示范了基本 COM 组件的创建和 COM 中的统一事件模型，并且包含了一个演示特性化 ATL ActiveX 控件的案例研究。

第 23 章——ATL 服务器 Web 应用程序

在这一章中，讨论了如何用 ATL 服务器创建基于 Web 的应用程序。这种技术可以提供与本地代码执行相结合的快速 Web 应用程序开发。ATL 服务器是一种用本地上下文创建动态 Web 内容的技术。服务器响应文件(SRF, Server Response File)使用与 HTML 标识相结合的替代标签(或模板)提供了动态内容。ATL 服务器 Web 应用程序用本地 C++生成的 HTML 动态替换 SRF 模板。本章介绍了许多有趣的例子，其中包含了一个在线来宾登记簿应用程序，以及一个允许用户在数据库中查寻特定作者出版物的多层且大量使用数据库的应用程序。

第 24 章——托管和非托管代码的互操作性

本章主要介绍了托管代码和非托管代码的区别，以及用于克服这些区别的技术，从而使代码上下文互操作。与其他的 .NET 语言不同，Visual C++ .NET 允许在同一个应用程序中并行执行托管代码和非托管代码。这种可能性由把执行分离到一个托管和非托管的上下文与管理上下文间所有通信而产生。Visual C++ .NET 的互操作性特性可以分成三种技术，IJW(It Just Works)编译器设计目标、平台调用服务(Platform Invoke Service)和 COM 互操作性服务(COM Interop Service)。本章讨论了 IJW 和平台调用服务，而第 25 章讨论了 COM 的互操作性特性。本章包含了一组用来示范通用互操作性使用的例子，如使用 Win32 API 函数和模板类。

第 25 章——COM 互操作性服务

这一章概括了 Visual C++ .NET 提供的服务，这些服务允许 COM 和 .NET 体系结构间的互操作。从数据类型的表示和对象的创建到各自组件库的描述和布局，COM 和 .NET 组件体系结构均是原理上完全不同的技术。COM 互操作性服务使 .NET 程序员能够绕过这些困难，并使两种组件体系结构能无缝地相互通信。本章示范了类型库和程序集转换，讨论了运行库可调用包装器(RCW, Runtime Callable Wrapper)和 COM 可调用包装器(CCW, COM Callable Wrapper)，并提供了展示 .NET 中 COM 对象和 COM 中 .NET 对象的例子。

附录 A——运算符优先级表

本附录列出了 MC++ 的运算符和它们的优先级。

附录 B——数字系统

本附录解释了二进制、八进制、十进制和十六进制数字系统，回顾了在这些基数间的数字转换，并阐述了在每一种基数下的数学运算。

附录 C——ASCII 字符集

本附录包含了一个表格，表格中是 128 个 ASCII(American Standard Code for Information Interchange, 美国标准信息交换码)符号(按字母顺序排列)和对应的整数值。

附录 D——Unicode®

本附录介绍了统一编码标准(Unicode Standard)。这是一种将惟一的数字值分配给世界上大部分语言的字符的编码方案。我们列举了使用 Unicode 编码打印出几种语言的欢迎信息的应用程序。

附录 E 和 F——XHTML 介绍：第 1 部分和第 2 部分

在这两个附录中，介绍了可扩展超文本标记语言(XHTML)。XHTML 是一种替代 HTML 的新兴 W3C 技术，它是描述 Web 内容的主要方式。作为以 XML 为基础的语言，XHTML 比 HTML 更稳健，并且更具可扩展性。XHTML 包含了 HTML 的大部分元素和特性——这些是这两个附录的主要内容。附录 E 和 F 是为不知道 XHTML 或在学习第 15 章“可扩展标记语言”和第 20 章“可访问性”之前希望回顾 XHTML 的读者准备的。

附录 G——XHTML 特殊字符

本附录提供了许多经常使用的 XHTML 特殊字符，称为字符实体引用(character entity references)。

附录 H——XHTML 颜色

本附录列出了常用的 XHTML 颜色名称和对应的十六进制值。

附录 I——位操作

本附录讨论了 Visual C++ .NET 强大的位操作能力。这有助于程序更简洁地处理位串、合并或拆分单个的位和更紧凑地存储信息。这些能力是低级汇编语言的特点，并受到编写系统软件(如操作系统和网络软件)的程序员的高度重视。

目 录

第 1 章 .NET 和 Visual C++[®] .NET 简介	1
1.1 引言	1
1.2 Internet 和万维网的历史	1
1.3 万维网协会(W3C)	3
1.4 可扩展标记语言	3
1.5 主要的软件开发趋势：面向对象技术	4
1.6 Microsoft .NET 简介	5
1.7 Visual C++ .NET	7
1.8 .NET Framework 和公共语言运行库	9
1.9 小结	10
1.10 Internet 和 Web 资源	13
1.11 尾注	13
第 2 章 Visual Studio[®] .NET IDE 和 Visual C++ .NET 编程	15
2.1 引言	16
2.2 Visual Studio .NET 集成开发环境概述	16
2.3 菜单栏和工具栏	19
2.4 Visual Studio .NET 窗口	20
2.4.1 Solution Explorer	20
2.4.2 Properties 窗口	21
2.5 使用帮助	22
2.6 创建一个简单的控制台应用程序	23
2.7 一个简单的程序：显示一行文本	26
2.8 算术运算	30
2.9 进行判断：等值运算符和关系运算符	31
2.10 小结	36
2.11 尾注	37
第 3 章 控制结构	38
3.1 引言	38
3.2 控制结构	39
3.3 if 选择结构	40
3.4 if/else 选择结构	41