光盘内容

- 本书实例源文件

全新升级

Visual C# 2005 基础与实例教程

郝春强　池同柱　编著

本书是畅销书《C#基础与实例教程》一书的升级版

- **内容全面** 涵盖Visual C# 2005所有知识点，并特别提供C#编码规范
- **实例丰富** 近百个经典C#编程案例，深入诠释C#语言编程的重点和难点
- **零起点** 无需任何编程语言基础，零起点学习C#编程
- **入门快捷** 采用知识讲解+实例练习+课后习题的教学方式，在边学边练中快速掌握C#语言

中国电力出版社
CHINA ELECTRIC POWER PRESS
中国电力出版社
www.infopower.com.cn

Visual C# 2005 基础与实例教程

郝春强　池同柱　编著

中国电力出版社
www.infopower.com.cn

内 容 提 要

.NET 框架与 C# 语言近年来取得了长足的发展，尤其是在企业管理软件领域得到了广泛的应用。本书是畅销书《C# 基础与实例教程》一书的全新升级版，完全针对 C# 2.0 编写，从 .NET 与 C# 的基本概念开始，循序渐进、由浅入深地介绍了使用 C# 语言进行应用程序开发的方方面面。全书共 12 章，内容囊括了 .NET 框架与 C# 语言的基本概念、Visual Studio 2005 集成开发环境的使用、C# 语言基础和面向对象技术、窗体编程相关技术、程序的调试和异常处理、文件操作、数据库编程、网络编程和 Web 服务等。

本书的附录中还提供了《C# 编码规范》，对于一个开发人员来说，养成良好的编程习惯和遵循一定的编码规则是非常重要的。

本书语言简洁，实例丰富，浅显易懂，非常适合 C# 初中级读者学习，还可作为本、专科学生学习计算机编程语言的教科书。

图书在版编目（CIP）数据

Visual C# 2005基础与实例教程 / 郝春强，池同柱编著. —北京：中国电力出版社，2007.12
ISBN 978-7-5083-6133-8

Ⅰ. V… Ⅱ. ①郝…②池… Ⅲ. C语言 - 程序设计 - 教材 Ⅳ. TP312

中国版本图书馆 CIP 数据核字（2007）第 157561 号

责任编辑：马首鳌
责任校对：崔燕菊
责任印制：李文志

书　　名：Visual C# 2005 基础与实例教程
编　　著：郝春强　池同柱
出版发行：中国电力出版社
　　地址：北京市三里河路 6 号　邮政编码：100044
　　电话：（010）68362602　　传真：（010）68316497
印　　刷：北京市同江印刷厂
开本尺寸：185mm × 260mm　　印　张：22　　字　数：536 千字
书　　号：ISBN 978-7-5083-6133-8
版　　次：2007 年 12 月北京第 1 版
印　　次：2007 年 12 月第 1 次印刷
印　　数：0001—4000 册
定　　价：35.00 元（1CD）

前　言

自 2005 年 8 月《C#基础与实例教程》一书上市以来，得到了广大读者的好评。而这期间，.NET 技术与 C#语言本身也得到了长足的发展，尤其是在企业管理软件领域得到了广泛而深入的应用。.NET 平台与 C#语言成为众多软件公司的首选开发平台与编程语言，越来越多的技术人员希望能尽快地了解并掌握.NET 与 C#的最新技术成果。

鉴于此，有必要对本书第一版进行一次全面修订，并补充最新技术发展的内容。

本书第一版是针对.NET Framework 1.1、C# 1.1 和 Visual Studio 2003 编写的，而本版则是针对.NET Framework 2.0、C# 2.0 和 Visual Studio 2005 编写的。

本版沿袭了上一版的写作风格和章节安排，相对于第一版，本版修订和扩充了以下内容。

1．纳入 C# 2.0 语言的新特性

本版在 C#语言基础和面向对象的 C#等章节中纳入了 C# 2.0 新增的许多新的语言特性，例如对泛型的支持、可空类型、部分类和静态类等概念。

2．Visual Studio 2005 集成开发环境的介绍

Visual Studio 2005集成开发环境相对于Visual Studio 2003而言更加强大与高效。在Visual Studio 2005中，提供了一个开发者期待已久的可视化类设计器。类设计器以类关系图的形式提供了一种可视的设计界面，可用于设计、查看Visual Studio项目中的类和其他类型。Visual Studio 2005还提供了重构的功能，通过重构可以方便快捷地进行重写方法、重命名标识符、重构参数，以及实现接口和抽象类等。

3．对新控件的介绍

.NET Framework 2.0 提供了全新的菜单控件、工具栏控件和状态栏控件，通过它们可以轻松创建出类似 Word 风格的菜单、工具栏和状态栏。此外，新提供的 DataGridView 控件用于替换原 DataGrid 控件，更加灵活与强大。

4．新增打印相关内容

介绍如何通过.NET 提供的类和控件，编写具有打印功能的应用程序。

5．新增应用程序部署相关内容

Visual Studio 2005为部署基于Windows的应用程序还提供了两种不同的策略：使用ClickOnce技术发布应用程序，或使用Windows Installer技术通过传统安装来部署应用程序。介绍了ClickOnce部署和Windows Installer部署的特点，并通过实例讲解如何制作Windows Installer安装包和以ClickOnce方式发布应用程序。

6．更多实例

本版包含了更多的应用实例。在数据库一章中新增了一个简单的三层应用程序实例，在 Web 服务一章中新增了如何通过调用 Web 服务获取天气预报的实例。

本书特色

（1）本书定位于快速入门级，阅读本书前不需要读者具有任何C#方面的基础知识，甚至可以是对编程技术一无所知的新手。因此，本书非常适合作为自学教材。

（2）本书通过清晰的概念介绍和大量的代码示例相结合的方式，来介绍使用C#语言进行程序设计的基础与方法，并且书中所有的示例代码均位于本书的配套光盘中，便于读者在学习本书的同时查看和运行示例。

（3）本书每章的最后都给出了一些思考与练习题，通过对这些问题的思考以及编码实践，可以进一步明确概念和掌握编程技巧。在本书的附录B中，提供了所有思考题的参考答案，对于编程练习题，本书的配套光盘中给出了示例程序，通过参考研习这些示例，对快速提高编程水平很有帮助。这也是本书的一个特色，即配套光盘不是对书中内容的重复，而是书与盘互为补充，这就使得本书的内容更充实。

（4）本书的附录A中提供了作者精心编写的《C#编码规范》，这在同类书中还非常少见。对于一个开发人员，养成良好的编程习惯和遵循一定的编码规则是非常重要的。

（5）本书试图让读者在学习C#语言的同时，能够掌握面向对象编程技术的一般思想和方法。面向对象技术是当前程序设计的主流方法，通过C#这种现代的完全面向对象编程语言的学习，可以为以后深入学习面向对象技术打下一个坚实的基础。

致谢

再次诚挚感谢我的父母家人，本书写作时，正值我的新家装修之际，是他们为我分忧解难，才使得我能有更多的写作时间。

感谢孙恋、骆成凤、徐文军、李易珉、桂笛、滕石欣等人一直以来对我的关心与支持。

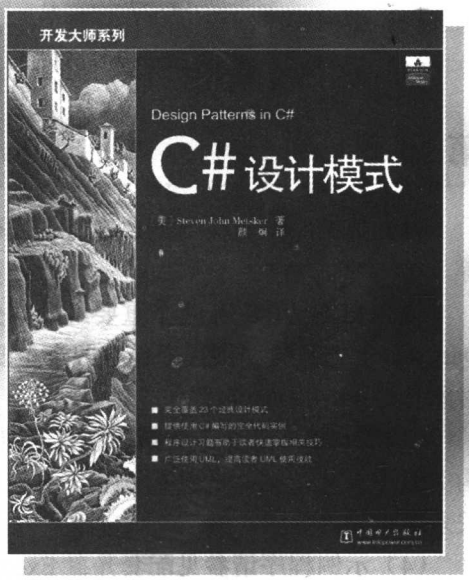
作者　郝春强

2007年6月9日于水木天成小区

中国电力出版社 C# 精品图书

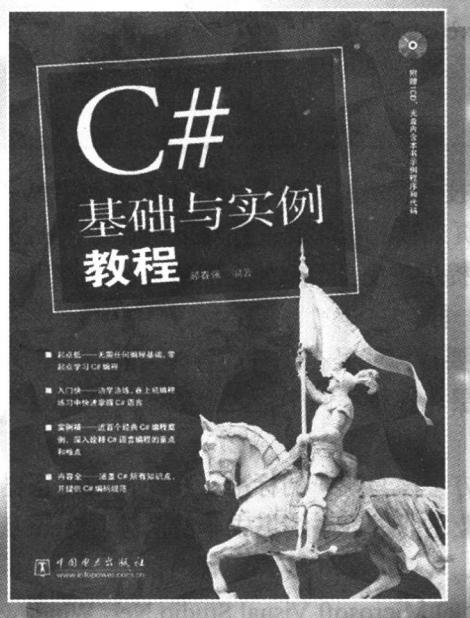
C#经典实例（第二版）

定价:128.00 元
书号:978-7-5083-5152-0



C#设计模式

定价:42.00 元
书号:7-5083-3492-2



C#基础与实例教程

定价:35.00 元
书号:7-5083-3284-9

目　录

第 4 章 面向对象的 C#

第 5 章 Windows 应用程序

第 6 章 基本控件的使用

第 1 章　.NET 与 C#

微软本身为了保持其领先地位，总是愿意抛弃旧的方式，积极创造新的方法。这方面很好的一个例子就是在 20 世纪 80 年代，微软把公司前途的赌注放在了图形界面上。尽管当时很多人认为老的字符界面已经非常完美，但是微软坚持认为图形界面要好得多，它创造了 Windows 这个伟大的产品，从此个人电脑（PC）走进了千家万户，深刻地改变了我们的工作方式甚至生活方式，时至今日，我们几乎无法想象，没有 Windows，世界将会怎样？

2000 年左右，微软又开始了另一个宏伟的战略，这个战略在很多场合被微软的高层和 20 世纪 80 年代的图形界面战略相提并论，这个战略就是.NET。.NET 是微软面向未来互联网的战略，微软把公司未来发展的赌注放在了.NET，如同 20 世纪 80 年代微软在图形界面上下的赌注一样。

.NET 战略所引发的重大变革同样在起初遭到各种非议和怀疑，但最终同样被证明是微软的又一次伟大创造。历经数年的发展，.NET Framework 从 1.0 到 1.1，再到 2.0 再到目前最新的 3.0，.NET 已经发展成为构建企业应用程序最重要的平台之一。

本章作为全书的第一章，将从宏观角度介绍.NET 战略、.NET 平台以及 C# 语言。对于初次接触.NET 的读者来讲，建立对.NET 和 C# 的全局认识至关重要。

1.1　什么是.NET

关于.NET 最被人经常提到的问题是：“什么是.NET？”

微软董事长兼首席软件设计师比尔·盖茨这样回答：“.NET 是指连接信息、人群、系统和设备的软件。”

微软原总裁兼首席执行官鲍尔默说：“.NET 代表了一个集合、一个环境、一个可以作为平台支持下一代 Internet 的可编程结构。”

上述两种回答简明扼要地表述了.NET 的外在特征，从中初步可以得出.NET 的出现将带来这样的变化：软件将使不同的计算机以不同的方式相互交流，人们使用互联网的方式将与我们过去五六年使用互联网的方式大不相同。

在这里我们通过对.NET 四个关键特性的阐述，来进一步了解.NET 的概貌。

- .NET面向软件服务
- .NET依存于XML
- 融合多种设备和平台
- 新一代的人机界面

1．.NET 面向软件服务

今天的软件产品仅仅是一张或数张光盘，用户购买软件，亲自安装、管理和维护。软件服务则是来自因特网的服务，它替用户安装、更新和跟踪软件，这些软件的执行可能会

跨越处于不同地理位置的不同机器，同时用户的资料等各种数据也是存储在网络机器上的而不是本地。这些就是软件产品和软件服务的不同之处。

伴随着被称为第三次 IT 革命的 Web 服务（Web Service）技术的出现以及 ASP（应用服务提供）产业的兴起，软件正逐渐从产品形式向服务形式转化，这是软件未来发展的趋势。.NET 正是为这一趋势所努力的成果。

Web 服务允许应用程序通过 Internet 进行通信和共享数据，而不管采用的操作系统、设备或编程语言是否相同。

.NET 是 Microsoft XML Web 服务平台，它提供创建 XML Web 服务并将这些服务集成在一起所需要的功能。

2．.NET 战略依存于 XML

XML 是一种格式，它让数据容易理解并具有相当的灵活性。XML 是下一代产品的关键组成因素。微软的.NET 战略是依存于 XML 的，就像微软以前的产品依赖于图形界面一样。微软致力于把 XML 变成整个业界的标准，而微软.NET 战略的实施也许会成为最好的 XML 的实施案例，就像过去 Windows 是图形用户界面最好的实施案例一样。

XML 是.NET 的基础与灵魂，上述的软件服务和将要提到的融合多种设备和平台目标的实现，都离不开 XML 技术的支持。

3．融合多种设备和平台

在.NET 之前，软件是围绕一个系统写的，软件工程师当时是考虑一个系统而不是考虑用户来开发软件的。如果用户换一台 PC 的话，他们要做很多的工作才能把文档以及其他信息转移到另一台 PC 上；如果他们想用另外一种终端工作，比如一种先进的电话或者手持便携设备，就必须运行一些协同软件以便让这两种不同的装置一起工作。

.NET 的出发点是不把系统当作关键因素。诚然，会有不同的系统，但是它们应该能够自然地协同工作。在服务器层面，不把某个应用单纯地看作是在一种服务器上的一种应用，而是认为这个应用可以使用很多的服务器，并且能够自动地利用多个服务器带来的扩展的、更强的功能。以人为本的理念保证了由此产生的生产力和可靠性会超越大型机时代或者是 UNIX 时代的最好的应用，它所带来的巨大的可扩展性使得我们有很大的余地，这样，只要不断把新系统加入进来，我们就有了更大的能力。

Microsoft .NET 的基本理念是：不再关注单个的网站和与 Internet 连接的单个设备，而是要让计算机群、相关设备和服务协同工作，提供更加广泛和丰富的解决方案以及服务，而不是像现在一样成为一座座信息孤岛。人们将能够控制何种信息、在何时、以何种方式传送给自己。.NET 的目标是把计算和通信带入一个丰富、合作和互动的环境中，远远胜过今天的单向网络。

在一些地方，这已经成为现实，比如说为 Windows 平台设置的用于交易的 TPCC 基准，它的功效更为强大，同时性能价格比更加优越。

4．新一代的人机界面

.NET 是一个巨大的变化，它不仅是编程方面的巨大变化，也是用户界面方面的一个巨大变化。微软认为用户界面还可以更加自然，就是说我们坐在电脑（当然还可能是其他设备）前浏览信息的时候，不仅能够使用键盘，还能够使用一支笔来手写，也就是说电脑有手写识别的功能，还可能用声音来操作，就是说电脑还有语音识别功能。我们所需要的信

息将展示在屏幕上，极高的分辨率使得屏幕的可读性非常强，即使是一个比较长的电子邮件也不需要打印。这些都是.NET 战略所推动的。

计算模式从终端主机时代、字符 PC 时代、GUI 时代发展到当前主流的 Internet 浏览器时代，今天的 Internet 与以前主机工作模式有许多相似之处，信息被储存在中央服务器内，而用户的所有操作都要依靠它们。在目前的技术环境下，网站之间相互传递有意义的信息，或者合作提供更广泛和更深层次的服务，是十分困难的。而用户要获得个性化的网络体验，或者得到“私人信息空间”来对网络信息进行编辑、分析和共享，则更加困难。现在，人们还在适应技术，但微软认为技术应当以人为本。包括 XML 和 SOAP（简单访问对象协议）在内的新行业标准将信息解放出来，使它们能够被重新组织、调整和编程，然后以任何可能的方式、在任何设备和系统上显示出来。以这些标准为基础的平台，将控制信息的权利重新交还给需要这些信息的人们。.NET 完全是为了实现这一目标而设计出来的，是微软公司提出的下一代互联网构想。

.NET 是微软的一个重要转折点。微软的产品在全球的应用愈来愈广泛，以服务为主的 MSN 也发展得越来越好，.NET 将以这些成功为基础。实现.NET 是一个较长的过程，类似从 MS－DOS 到 Windows 的转变。微软将继续提供和支持现有的平台和应用软件，包括不含.NET 技术的平台。但是，经过长时间的努力，微软的产品和服务将最终转化改进成订购式服务，通过 Internet 送货。

1.2 .NET 平台

.NET 是一种战略，.NET 平台则是实现这一战略的技术基础。它包括设备、基础设施、积木块服务、框架和工具等几个部分，这些部分所组成的一个垂直结构就是.NET 平台，如图 1.1 所示。

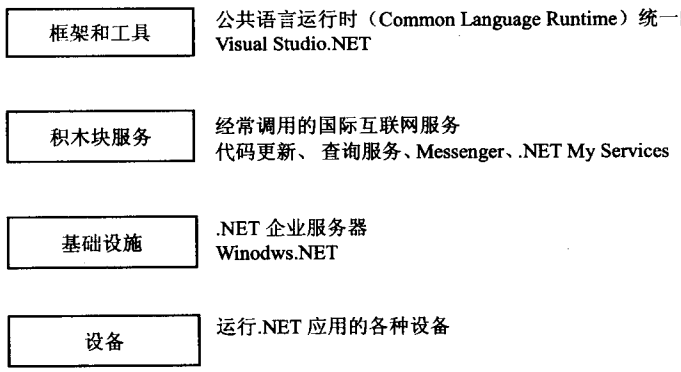


图 1.1 .NET 平台

1．设备

我们注意到处于.NET 平台底层的硬件部分使用了术语“设备”，而不再是“服务器”、“PC”等字眼。这也是我们上面所提到的.NET 融合多种设备和平台理念的体现。这些设备可能是电脑也可能是手持设备，如手机等。

2．基础设施

基础设施包括构建企业应用所必须的操作系统和各种服务器，如数据库服务器等。

Windows.NET 是融入了.NET 技术的 Windows，它将紧密地整合.NET 的一系列核心构造模块，为数字媒体及应用间协同工作提供支持，是微软公司的下一代 Windows 桌面平台。

在微软宣称的“第三代互联网”中，.NET 企业服务器是企业集成和管理所有基于 Web 的各种应用的基础，它提供企业未来开展电子商务的高可靠性、高性能、高可伸缩性以及高可管理性。.NET 企业服务器的构成异常庞大和复杂，目前共包括 8 个各司其职的服务器，如表 1.1 所示。

表 1.1 .NET 企业服务器一栏表

服务器名称	功能描述
Application Center 2000	部署和管理基于 Windows 2000 之上的 Web 应用
Biztalk Server 2000	用于企业间交换商务信息
Commerce Server 2000	用于快速创建在线电子商务
Exchange 2000	提供基于 Windows 2000 的通信和协作功能
Host Integration Server 2000	为主机系统的组件集成提供方便
Internet Security & Acceleration Server 2000	主要解决企业应用安全性和可管理性的问题
Mobile Information 2001 Server	为移动解决方案提供可靠而具伸缩性的平台
Sql Server 2000	提供完全的数据库和数据分析与数据挖掘解决方案

3．积木块服务

积木块服务（Building Block Services）是.NET 平台中的核心网络服务集合，它主要包括以下几个组成部分：

（1）Internet XML 通信，使 Web 站点变成灵活的服务来交换和处理数据，Internet XML 数据空间，为 Web 应用提供安全的和可编程的 XML 存储空间。

（2）Internet 动态更新，为快速开发和动态配置应用提供服务。

（3）Internet 日程安排，集成工作、社会和私人的日历。

（4）Internet 身份认证，提供从口令、钱包到生理数据等多级身份认证手段。此外，还有 Internet 目录服务和 Internet 即时信息传递等服务。

4．框架和工具

.NET 框架（.NET Framework）处于.NET 平台的上层，它是.NET 平台的核心部分，是在.NET 平台上进行开发的基础，下一节我们将专门讨论.NET 框架。

.NET 平台上最好的开发工具当数微软发布的 Visual Studio.NET，它是 Visual Studio 6.0 的下一代产品，目前它的最新版本是 Visual Studio 2005，关于 Visual Studio 2005 的使用我们将在第二章中进行详细介绍。

1.3 .NET 框架

.NET 框架是.NET 平台的编程模型，是创建、部署和运行 Web 服务及其他应用程序的一个环境。

1.3.1 .NET 框架的演化

任何技术的发展都是具有连续性的，.NET 也不例外。尽管.NET 框架富有革命性，但它同样是对过去技术的一种继承与发展，图 1.2 展示了.NET 框架的演化过程。

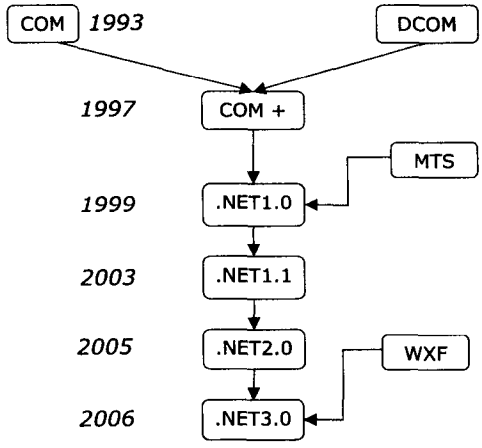


图 1.2 .NET 框架的演化

1.3.2 .NET 框架体系结构

.NET 框架体系结构如图 1.3 所示，它由以下四个主要部分组成：

- 公共语言运行时（Common Language Runtime，CLR）。
- 统一类库（Base Class Library）。
- ADO.NET。
- 活动服务器页面（ASP.NET）。

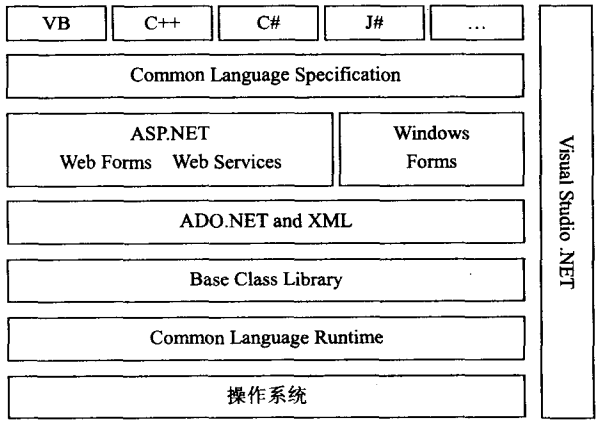


图 1.3 NET 框架体系结构

1. 公共语言运行时（CLR）

公共语言运行时是 .NET 框架应用程序的执行引擎。该名称不能准确反映它的全部功

能。实际上，公共语言运行时在组件的开发及运行过程中，都扮演着非常重要的角色。在组件运行过程中，运行时负责管理内存分配、启动或删除线程和进程、实施安全性策略，同时满足当前组件对其他组件的需求。在开发阶段，运行时的作用有些变化，与 COM 相比，运行时的自动化程度大为提高（比如可自动执行内存管理），因而开发人员的工作变得非常轻松，尤其是映射功能将锐减开发人员将业务逻辑程序转化成可复用组件的代码编写量。对编程语言而言，运行时这个概念并不新奇。实际上每种编程语言都有自己的运行时。Visual Basic 开发系统具有最为明显的运行时（名为 VBRUN），Visual C++跟 Visual FoxPro、JScript、SmallTalk、Perl、Python 和 Java 一样有一个运行时，即 MSVCRT。.NET 框架的关键作用在于，它提供了一个跨编程语言的统一编程环境，这也是它能独树一帜的根本原因。

2．统一的编程类库

.NET 框架为开发人员提供了一个统一、面向对象、层次化、可扩展的类库集（API）。如今，C++开发人员使用的是 Microsoft 基类（MFC）库，Java 开发人员使用的是 Java 基类库，而 Visual Basic 用户使用的又是 Visual Basic API 集。.NET 框架统一了微软当前的各种不同类框架。这样，开发人员无需学习多种框架就能顺利编程。远不止于此的是，通过创建跨编程语言的公共 API 集，.NET 框架可实现跨语言继承性、错误处理功能和调试功能。实际上，从 JScript 到 C++的所有编程语言，都是相互等同的，开发人员可以自由选择理想的编程语言。

3．ADO.NET

在开始设计 .NET框架时，Microsoft就以此为契机重新设计了数据访问模型。Microsoft 没有进一步扩展ADO，而是决定设计一个新的数据访问框架，但保留了ADO。Microsoft 根据其成功的ADO对象模型经验设计了ADO.NET。ADO.NET满足了ADO无法满足的三个重要需求：提供了断开的数据访问模型，这对 Web 环境至关重要；提供了与XML的紧密集成；还提供了与.NET框架的无缝集成（例如，兼容基类库类型系统）。

4．活动服务器页面(ASP.NET)

ASP.NET 不仅仅是 Active Server Page（ASP）的下一个版本，它还提供了一个统一的 Web 开发模型，其中包括开发人员生成企业级 Web 应用程序所需的各种服务。ASP.NET 的语法在很大程度上与 ASP 兼容，同时它还提供了一种新的编程模型和结构，可生成伸缩性和稳定性更好的应用程序，并提供更好的安全保护。可以通过在现有 ASP 应用程序中逐渐添加 ASP.NET 功能，随时增强 ASP 应用程序的功能。

ASP.NET 是一个已编译的、基于 .NET 环境的、可以用任何与 .NET 兼容语言（包括 Visual Basic .NET、C# 和 JScript .NET）创建的应用程序。另外，任何 ASP.NET 应用程序都可以使用整个 .NET 框架，开发人员可以方便地获得.NET 技术的优点，其中包括托管的公共语言运行时环境、类型安全以及继承等。

ASP. NET 是使用.NET 框架提供的编程类库构建而成的，它提供了 Web 应用程序模型，该模型由一组控件和一个基本结构组成。有了它，Web 应用程序的构建变得非常容易。开发人员可以直接使用 ASP. NET 控件集，ASP.NET 还提供一些基本结构服务（诸如会话状态管理和进程重启服务），这些服务大大减少了开发人员要编写的代码量，并使应用程序的可靠性得到大幅度提高。ASP.NET 还允许开发人员将软件作为一项服务（即 Web 服务）来提供。通过使用 ASP.NET Web 服务功能，ASP.NET 开发人员只需进行简单的业务逻辑编程，

而由 ASP.NET 基本结构负责通过简单对象访问协议（SOAP）来提供服务。

1.3.3 .NET 框架编程模型

使用 .NET 框架编程不同于使用 Win32 API 编程，正如 Windows 编程与 DOS 编程大相径庭一样。每一个 API 都是某一个时代的产品，DOS API 是 20 世纪 80 年代早期的产品；Windows API 是 20 世纪 80 年代中期的产品；而 .NET API 是 20 世纪 90 年代后期的产品。

.NET 框架编程模型和传统编程模型有所不同，如图 1.4 所示。

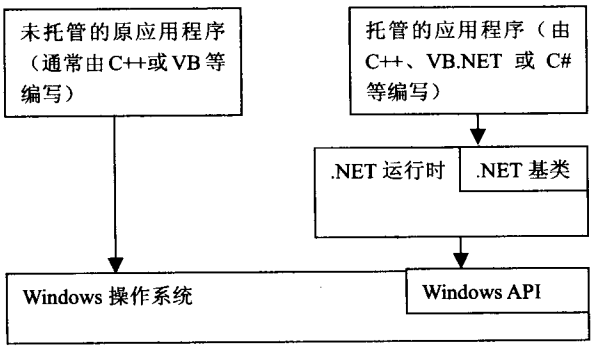


图 1.4 .NET 框架编程模型

传统的编程模型是上层应用直接依附在操作系统之上。例如我们在 Windows 中编程，程序运行于 Windows 之上。但在.NET 平台中，.NET 框架位于操作系统与上层应用之间，上层应用创建于.NET 框架之上，同时也运行于.NET 框架之上，而不像过去一样直接运行在操作系统之上。正是在操作系统和应用程序之间有了.NET 框架，才使得应用程序的平台无关性成为可能。

.NET 框架是Microsoft为开发应用程序创建的一个富有革命性的新环境。这句话最有趣的地方是它的含糊不清，但这是有原因的。注意这句话没有说“在Windows操作系统上开发应用程序”。尽管.NET Framework发布的第一个版本是运行在Windows操作系统上，但是以后将推出可运行在其他操作系统上的版本，这些操作系统包括FreeBSD、Linux和Macintosh，甚至个人数字助手类设备。这就是说.NET具有平台独立性，是可移植的，这的确是一个很大的突破。我们知道微软为了保护其Windows操作系统的利益，向来在平台独立问题上非常保守，其开发的产品都只能运行中Windows环境中，这一直为人们所诟病。

要进一步理解.NET 框架编程模型，还需要认识两个新的概念：MSIL 和 JIT。下一节将对这两个概念作详细介绍。

1.3.4 .NET 程序的编译与运行

1．MSIL 和 JIT

在编译使用.NET 框架创建的代码时，不是立即创建操作系统特定的本机代码，而是把代码编译为微软中间语言（Microsoft Intermediate Language，MSIL）代码，这些 MSIL 代码