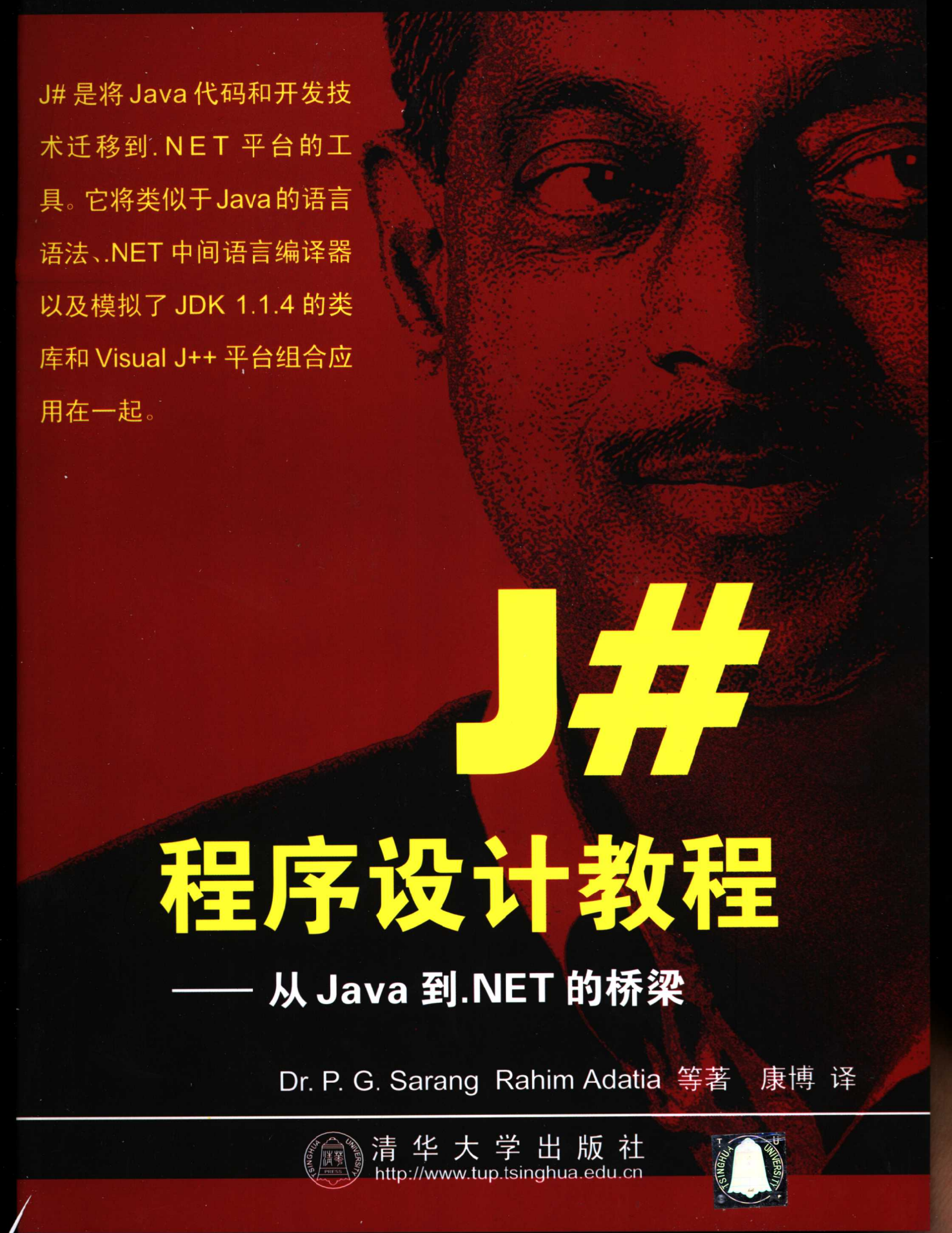
PROGRAMMER TO PROGRAMMER™



J# 是将 Java 代码和开发技术迁移到.NET 平台的工具。它将类似于 Java 的语言语法、.NET 中间语言编译器以及模拟了 JDK 1.1.4 的类库和 Visual J++ 平台组合应用在一起。

J# 程序设计教程

—— 从 Java 到.NET 的桥梁

Dr. P. G. Sarang Rahim Adatia 等著　康博 译

清华大学出版社
http://www.tup.tsinghua.edu.cn

J#程序设计教程

——从 Java 到.NET 的桥梁

Dr.P.G.Sarang
Rahim Adatia 等著

康　博　译

清华大学出版社

(京) 新登字 158 号
北京市版权局著作权合同登记号：01-2002-3208

内 容 简 介

J#是 Microsoft 推出的一种新型.NET 语言。它其实是 J++的升级版本，帮助 Visual J++开发人员和 Java 开发人员轻松转向.NET 平台。本书介绍了 J#的语法、它所提供的类库以及 J#与 Visual J++和 Java 语言的关系。通过本书还可以学习如何使用 J#创建 Windows 应用程序、ASP.NET Web 应用程序和 Web 服务。本书最后讨论了 J++和 Java 向 J#的迁移技术。

本书适用于希望在.NET 平台中开发应用程序的 Visual J++开发人员和 Java 开发人员，以及希望学习这种.NET 语言的其他读者。

Dr.P.G.Sarang Rahim Adatia et al:J#
EISBN:1-861007-01-9

图书在版编目(CIP)数据

J#程序设计教程——从 Java 到.NET 的桥梁/(美)沙朗，(美)艾德逊著；康博译.
—北京：清华大学出版社，2002
ISBN 7-302-05725-7

Ⅰ.J… Ⅱ.①沙… ②艾…③康… Ⅲ.JAVA 语言—程序设计 Ⅳ.TP312

中国版本图书馆 CIP 数据核字(2002)第 059142 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.tsinghua.edu.cn
责任编辑：夏兆彦
印 刷 者：北京昌平环球印刷厂
发 行 者：新华书店总店北京发行所
开　　本：787×1092 1/16　**印张**：16　**字数**：379 千字
版　　次：2002 年 8 月第 1 版　2002 年 8 月第 1 次印刷
书　　号：ISBN 7-302-05725-7/TP・3377
印　　数：0001～4000
定　　价：32.00 元

出版者的话

近年来，国内计算机类图书出版业得到了空前的发展，面向初级用户的应用类软件图书铺天盖地，但是真正有深度和内涵的高端图书不多。已经掌握计算机和网络基础知识的人们，尤其是 IT 专业人士迫切需要“阳春白雪”。IT 图书市场呼唤精品！

为了满足这种市场需求，清华大学出版社从世界出版业知名品牌 Wrox 出版公司引进了受到无数 IT 专业人士青睐，被奉为 IT 出版界经典之作的 Professional 系列丛书。这套讲述最新编程技术与开发环境的高级编程丛书，从头到尾都贯穿了 Wrox 出版公司“由程序员为程序员而著(Programmer to Programmer)”的出版理念，每一本书无不是出自软件大师之手。实际上，Wrox 公司的图书作者都是世界顶级 IT 公司(如 Microsoft，IBM，Oracle 以及 HP 等)的资深程序员，他们的作品既深入研究编程机理，传授最新编程技术，又站在程序员的角度，指导程序员拓展编程思路，学习实用开发技巧，从而风靡世界各地，被 IT 专业人士和程序员视为职业生涯中的必读之作。

为了保证该系列丛书的质量，清华大学出版社迅速组织了一批位于 IT 开发领域前沿的专家学者进行翻译，经过编辑人员的进一步加工整理后，现陆续奉献给广大读者。

读者可以从 www.wrox.com 网站下载所需的源代码并获得相关的技术支持。同时，也欢迎广大读者参与 p2p.wrox.com 网站上的在线讨论，与世界各地的编程人员交流读书感受和编程体验。

目　　录

第1章　Visual J# .NET

首先要说明的是 J#读作“J Sharp”。接下来我们讨论 J# 和 Microsoft .NET 的根源，它们出现的原因，以及如何使用这门新的.NET 语言。在本章，我们将先讨论 J#是如何适应 Microsoft .NET 体系结构的。同时，我们将把.NET 与 Sun Microsystems J2EE 作一个简单的比较。接着看一下编译时和运行时组件、语言语法、J#类库，讨论 J#与 Java 和 Visual J++的关系。最后介绍 Visual Studio .NET 的功能，以及 J#是如何与之集成，从而为 J#应用程序创建一个完善的开发环境。

1.1　背景知识

为了使本书简洁明了，我将不深入讨论企业软件的发展史——它是如何从哑终端访问和集中式大型机迁移到客户机-服务器体系结构，接着又发展为分布式计算构架的。我们假定读者已经对该主题非常熟悉，可以直接学习 Web 服务，为什么 Web 服务令人振奋，以及如何用.NET 来使用它。

如今，人们都在讨论 Web 服务，就好像它是企业数据处理技术的圣器一样。Web 服务并不是一个新的概念；它其实由来已久。目前，大部分的 Web 开发都是以大量孤立站点的形式来进行的，每个站点都是单独开发的，解决的是一些类似的问题，如配置、身份验证和通信。对于企业来说，实现服务的更好方法是能够重用或结合使用从别处可得的以软件服务形式出现的功能。这一概念通常被称之为“软件就是服务”或简称“Web 服务”。从概念上看，这类似于 LEGO(积木玩具)，其中每个单独的部分都表示可以通过 Web 访问的可重用软件服务的组件。企业可以使用这些以软件服务形式出现的单独部分，把它们作为一个新服务结构中的子组件，这样就无需总是重新构建软件服务。要完成这项工作，组合组件必须知道如何互相通信和互相“拼接”(类似于 LEGO 块的插座和插头)。

在过去，我们曾多次尝试过以 CORBA (Common Object Request Broker Architecture)、IIOP (Internet Interoperability Protocol)、Java RMI (Remote Method Invocation)和 Microsoft DCOM (Distributed Component Object Model)等形式来实现，而且也获得了一定的成功。然而 Web 服务有希望通过采用开放标准和协议，使整个行业步入一个新台阶。

Web 服务只是实现定位透明的分布式计算技术的最新术语。这意味着可以部署、

集成各种不同的服务，使它们能相互作用，而不必知道这些服务的定位。随着可扩展标记语言(XML)、简单对象访问协议(SOAP)、通用说明、发现和集成(UDDI)和 Web 服务描述语言(WSDL)的出现及广泛运用，计算机行业已经日臻成熟，企业能够设计和实现这类服务，见图 1-1。

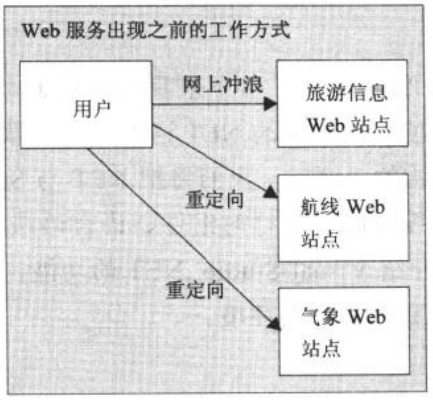


图 1-1

图 1-1 所示为许多公司普遍采用的一种服务方式，其中相关的服务间没有合作关系，它只是让客户导航到各个不同的站点。然而，如果使用了 Web 服务，旅行社就可以把一个公司的旅游预定服务和另一个公司的气象服务结合起来，提供一个有用的新服务，如图 1-2 所示。该模式还可以得到扩展，例如把预定服务与一个公司的信用卡验证服务联系起来，或是与另一公司的通知服务联系起来，向客户通知预定情况或是其他更新消息(如航班取消和延迟)。在这种情况下，开发人员不需要重新实现这些单独的服务，而是可以利用其他公司已实现的服务。

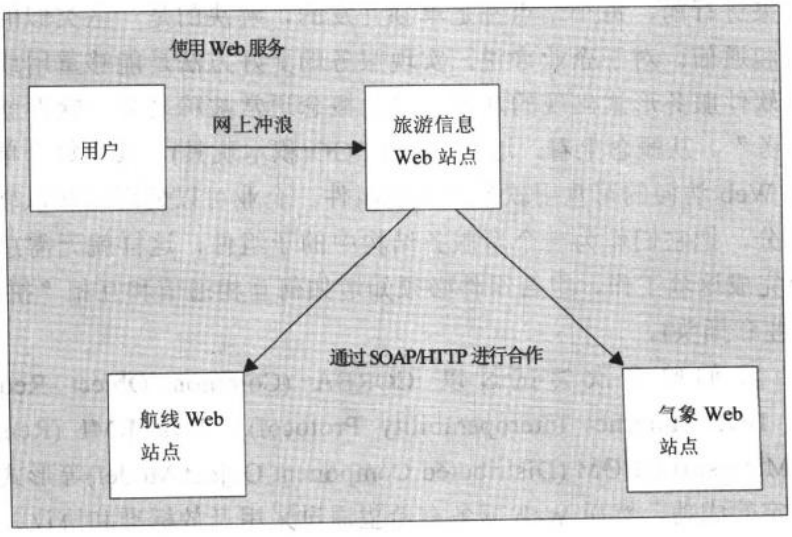


图 1-2

目前市场上，实现 Web 服务的主要有两大阵营：一个是 Sun Microsystems J2EE(与最初的 Sun ONE Open Net Environment 是一起的)，而另一个是 Microsoft .NET。2000 年 7 月，在佛罗里达举行的 Professional Developers Conference (PDC)上，Microsoft 宣布了它的“软件就是服务”的思想，并向世人展示了.NET 战略，声称它改变了软件开发的方向。

1.2　.NET 论述

Microsoft .NET 战略的主要目的是重新定义软件开发、部署和管理的方式。Microsoft 所拟定的崇高目标是希望它能像从 Windows 3.1 转变成 Windows 95 那样，对计算机行业产生深远的影响。

Microsoft 目前所采用的方法是致力于突破 DNA/ASP/COM 的局限性，例如开发所需的高级专业知识，缺乏版本和安装控制能力，与各种组件类型相互作用的功能有限。通过重新定义开发和运行环境，创建一个新的部署机制，采用开放标准，Microsoft 使得 IT 部门在选择开发商和软件商时有了更大的余地。

支持 XML 的.NET，不仅可以很容易地开发和部署服务和应用程序，而且可以使这些服务和应用程序驻留在各种平台上，且与它们相互作用。这和过去相比是个巨大的飞跃，因为过去如果选择采用 Microsoft 方法，那么通常要求所有的组件都是 Microsoft Windows 特有的组件。

尽管 Microsoft 通常把目前的.NET 打包成一个整体进行销售，但它实际上由 4 个部分组成(见图 1-3)。

图　1-3

这 4 个部分构成了.NET 战略。然而，这些部分是相互独立的，它们可以很好地单独工作。

1.2.1　.NET Framework

.NET Framework的主要目标之一是提供一个多语言共用的基础平台。在.NET以前，开发人员要在多语言间共享代码是很困难的，因为这些语言派生于不同的库，且在不同的运行环境中执行。例如，VB应用程序构建于VB Forms上，而用C++编写的Windows应用程序通常构建于MFC上(Microsoft Foundation Classes)。

.NET架构通过定义一个.NET运行环境合并了各种模型，其中有如图1-4所示的公共语言运行库(CLR)。CLR是个类似于Sun JVM(Java虚拟机)的虚拟机。然而，CLR执行的不是Java字节码，而是Microsoft中间语言(MSIL)代码。

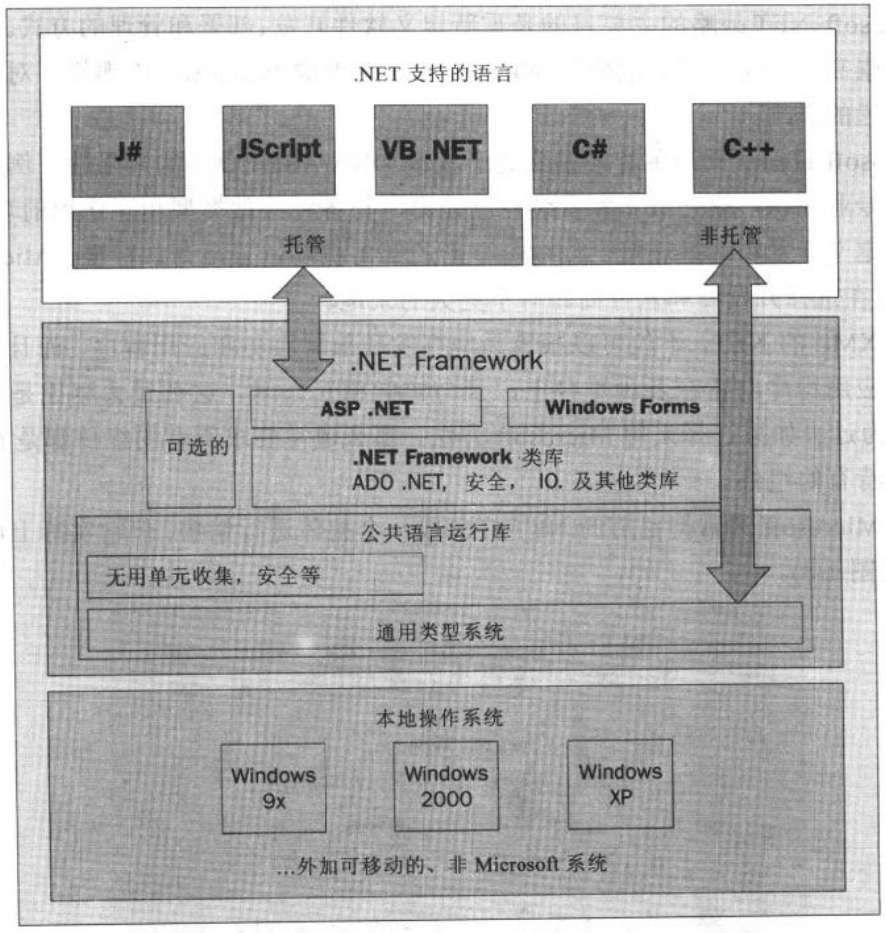


图　1-4

在编译源代码时，通常都是把它编译为专用于某种操作系统和处理器的二进制代码。这样一个产品就需要多种发布版本。例如，某个公司需要分别为Windows 2000/Intel和Linux/MIPS机器开发和编译产品，生成多种基代码，测试套件，最终形成完全不同

的版本。MSIL 是个中间语言，因为所有的.NET 语言都是被编译成这个通用语言，而不是直接编译为专用于某种操作系统和处理器的二进制代码。在运行时，CLR 执行 MSIL 并把它编译成本地机器代码。MSIL 可以被看作是 CLR 的机器代码。因此，通过创建这样一个通用语言，用任何一种.NET 语言编写的代码都可以在安装了.NET CLR 的机器上运行。尽管目前 CLR 只能在 Windows 平台上使用，但从理论上讲，MSIL 可以在安装了 CLR 的任何机器上工作。

CLR 不仅可以在运行时把 MSIL 编译成专用于本地处理器的指令(有时候被称为 Just-In-Time 或 JIT 编译)，它还对安全性、无用单元收集(内存管理)等负责。而把这一功能委托给 CLR 的代码叫作托管代码，它是.NET 中的标准。然而，有时需要访问操作系统的低级功能，不能受 CLR 的控制(可以使用 C++创建这类代码，即非托管代码)。

除了 CLR 外，.NET 运行库还包括了一个综合性的类库，所有的.NET 语言都是构建在这个类库上的。为了把某种语言变成适用于.NET 的语言，必须把它编译成 CLR 兼容的类型。通过把这些.NET 语言构建于这些公共类上，开发人员就可以使用任何一种.NET 语言创建的组件。

语言的互用会造成已有的 Microsoft 语言的浪费。例如，目前的编译器会把用 VB 和 C++编写的代码直接编译成本地可执行的代码，而不是编译成 MSIL，这样它们就不能利用.NET 运行库。为了解决这一问题，Microsoft 提供了一个新的 IDE，即 Visual Studio .NET (VS.NET)，它支持新的.NET 语言，如 VB.NET、Visual C++.NET、JScript .NET、C#和现在的 J#，这些语言都是被编译成 MSIL 的。

顾名思义，VB.NET 和 C++.NET 分别来源于 VB 和 C++。为了在新的 IDE 中编译已有的代码，需要对代码作一些修改。例如在 VC++中，如果开发人员想让 CLR 管理内存分配，那么必须添加名为托管指令的额外代码(有时叫作托管 C++代码)，尽管没有使用托管指令的 C++代码仍会被编译为 MSIL。

如果已有的代码是使用 VB 编写的，那么它就不容易被编译成 MSIL。不作任何改变的 VB 代码将不能被编译成 MSIL；我们需要对代码作一些修改，使之变成 VB.NET 代码，这样它就可以在.NET Framework 中工作了。在过去，VB 的新版本都是向后兼容的，而对于 VB.NET(以前叫作 VB 7.0)和 VB 6.0 却并不是这样。遗憾的是，学习 VB.NET 是件相当困难的事(尽管益处多多)。VB.NET 的新功能体现在创建对象，改进处理技术和确定线程等。要进一步了解 VB.NET，可参见《Visual Basic.NET 入门经典》(清华大学出版社引进并出版)。

表 1-1 所示为部分新.NET 语言与旧版本语言的对应关系。

表 1-1

.NET 语 言	相应的旧版本语言
Visual Basic .NET (VB.NET)	Visual Basic (VB)
托管或非托管的 Visual C++	Visual C++
ASP.NET	ASP (有时指典型的 ASP)
JScript.NET	JScript
C#	C#是种新语言
Visual J#	Visual J++

1.2.2 C#

C#是第一种.NET 所特有的编程语言，它用于优化.NET Framework 开发。从语法上看，C#像是派生于 Java，由于它是个完全面向对象的类型安全的语言(通过在编译时检查其类型，防止出现不相关的类型)，允许多接口继承但只能实现单一继承，它可以被编译成中间语言(这里指 MSIL)但不是本地机器代码。

C#允许开发人员使用 C++ (或 Java) 编程技术，但它更像 Java，从语法中删除了指针，通过引用传递对象以及提供了类型安全的语法。它还允许开发人员利用.NET Framework，把它编译成 MSIL，把系统级的维护工作如内存管理和自动无用单元收集委托给 CLR。Visual Studio .NET 中包含了 C#。

1.2.3 .NET 企业服务器

.NET 企业服务器是新一代的 Microsoft 企业服务器集合，包括 Windows Server、BizTalk Server、Exchange Server 和 SQL Server 等等。这些服务器将为部署核心企业应用程序所需的各种服务提供后端实现。

.NET 服务器资源管理器这个名称引起了很多混乱，它暗示着使用.NET 实现了服务器，而事实并非如此。.NET 企业服务器当前版本并没有使用.NET 实现，但利用 XML Web 服务可以在很大程度上与.NET 结构相互作用。在将来的更新产品中，部分服务器会使用托管代码实现(可以在 CLR 中执行)。例如，Commerce Server 将使用 ASP.NET，SQL Server 允许把存储过程编写为托管代码，更有可能的是使用 SOAP 和 XML Web 服务来提供数据。预计将来发行的所有.NET 企业服务器都是使用.NET Framework 编写的。

1.2.4 构建模块化的.NET 服务

.NET 战略的最后组成部分是一组用于身份验证、授权等的 Web 服务组件。读者可能已听说过以代号 HailStorm 出现的服务，其最新正式名称为.NET My Services。可能有人还见过把这些服务运用到 Microsoft 自己的站点上，例如 Hotmail、Passport 和 MSN Messenger，而现在它们已进行了无缝集成。这些服务较好地演示了.NET 的开发功能，尽管这些服务不是.NET Framework 开发所必需的。要了解 HailStorm 的更多内容，可参阅 Early Adopter Hailstorm(Wrox Press，ISBN：1-861006-08-X)一书。

1.2.5 比较.NET 和 J2EE

Microsoft .NET 与 Sun J2EE 是相对的，因为它提供了一个完整的包—— 一个新的开发环境，支持类库和对开放标准的支持，从而可以创建核心企业应用程序，见表 1-2。

表 1-2

特 征	.NET	J2EE	说 明
开发语言	C#、VB.NET、 J#和任何其他可以转换为 MSIL 的语言	目前只有 Java	这是.NET 的长处，因为它允许进行跨语言开发
公共运行环境	用于执行和管理 MSIL 代码的 CLR	用于执行和管理 Java 字节码的 JVM	目前，MSIL 只能在 Windows(惟一带有 CLR 的平台)上执行，但从理论上讲，它应该可以在所有平台上使用。.NET CLR 与 JVM 不一样，它允许存在非托管代码
支持的类	.NET Framework 类	Java 的核心 API (J2SE)	.NET 提供了 XML 处理功能作为构架的一部分
GUI 组件	Windows Forms Web Forms	Java Swing	.NET 提供了与 VS.NET IDE 完全集成的 Web 组件在各种 Java IDE 中可获得类似的组件，但它们都不是作为标准