Broadview
www.broadview.com.cn

项目实践精解丛书

国家信息专业技术人才知识更新工程（“653工程”）指定参考用书

项目实践精解：C#核心技术应用开发

梁立新　编著
亚思晟科技　审校

光盘内包含书中实例代码，项目案例的源代码、部署和运行

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

项目实践精解丛书

国家信息专业技术人才知识更新工程（“653工程”）指定参考用书

项目实践精解：C#核心技术应用开发

梁立新　编著
亚思晟科技　审校

電子工業出版社
Publishing House of Electronics Industry
北京·BEIJING

内 容 简 介

本书是一本融合项目实践与开发思想于一体的书。它的特色是以项目实践作为主线贯穿其中。书中提供了一个完整的 SuperVCD 项目，通过该项目可使读者快速掌握 C#核心技术，包括熟练掌握 C#基础语法，涵盖变量、表达式、流程控制和数组等；重点掌握面向对象的基本概念，着重理解封装、继承和多态的概念；熟练掌握面向对象的高级特性，包括静态、接口、集合、委托与代理等概念；熟练使用 Windows 窗体编程技术开发可视化用户界面；熟练编写多线程程序，熟练运用多线程的数据共享机制；熟练使用不同的流处理不同数据的类型；熟练编写基于 TCP/UDP 的服务器与客户端程序等。

本书作者有多年软件开发和教学经验，并且有很多学生目前正在从事C#开发工作；因此，作者清楚C#核心技术的合理学习路线，以及在学习过程中的注意事项。

本书适合作为 C#核心技术的培训教材或自学教材，同时也适合作为 C#开发人员的参考书。

图书在版编目（CIP）数据

项目实践精解：C#核心技术应用开发 / 梁立新编著. — 北京：电子工业出版社，2010.9
（项目实践精解丛书）
ISBN 978-7-121-11424-3

Ⅰ. ①项… Ⅱ. ①梁… Ⅲ. ①C 语言—程序设计 Ⅳ.①TP312

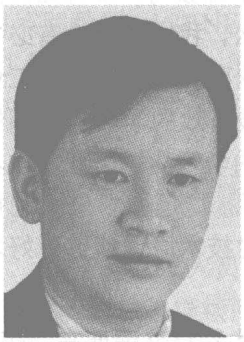
中国版本图书馆 CIP 数据核字（2010）第 140066 号

责任编辑：李云静
印　　刷：北京天宇星印刷厂
装　　订：三河市皇庄路通装订厂
出版发行：电子工业出版社
　　北京市海淀区万寿路 173 信箱　邮编 100036
开　　本：787×1092　1/16　印张：28.75　字数：736 千字
印　　次：2010 年 9 月第 1 次印刷
印　　数：4000 册　　定　　价：59.00 元（含光盘 1 张）

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888。
质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。
服务热线：（010）88258888。

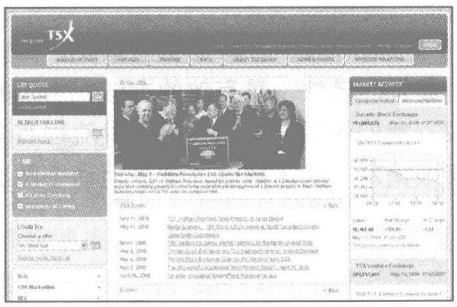
作者简介

梁立新

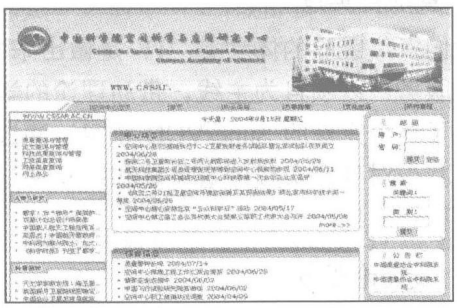


外籍软件专家，具有十多年专业软件开发、架构设计和项目管理的经验，擅长面向对象的分析设计、开发和管理。毕业于中国科学技术大学，获硕士学位。之后留学于美国，获伊利诺依理工大学硕士学位。曾先后工作于美国华尔街咨询服务公司和加拿大多伦多证券交易所，担任高级软件设计师。参与设计建设了美国著名银行 JP Morgan 网上人力资源系统和加拿大最大的证券交易中心 Toronto Stock Exchange 股票交易系统和市场数据传输及分析系统。回国后，创办北京亚思晟商务科技有限公司，设计和开发了中科院空间中心电子政务系统、网上企业财务中心管理系统及 eBiz 企业 ERP 管理系统等，同时从事高端 IT 教育培训、实训体系研发和咨询工作。

参与设计建设的项目展示

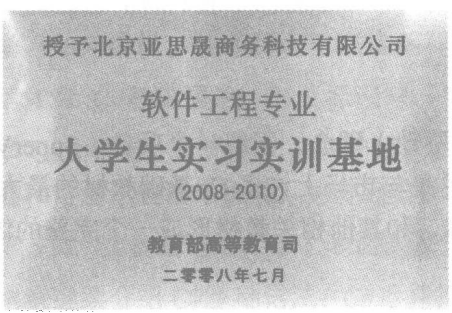


项目开发：Toronto Stock Exchange 股票交易系统



项目开发：中国科学院空间中心电子政务系统

回国后创业的亚思晟商务科技有限公司风貌



教育部授牌

公司实训环境

前　言

学习任何技术都要从基础开始，这本书就是讲解 C#核心技术的基础教材。读者如果简单调研一下就不难发现，市面上最多的就是这类 C#基础类教材。那么，我们还有没有必要再多写一本没有特色、枯燥乏味的书呢？答案当然是否定的！作者从事软件设计开发工作十多年，最近才萌发了写一本好书的想法。因为作者希望推广一种最有效的学习与培训的捷径，这就是 Project-driven training，也就是用项目实践来带动理论的学习。基于此，作者围绕一个项目（SuperVCD）来贯穿 C#核心技术各个模块的理论讲解。这是本书最大的特色！通过项目实践，可以对技术应用有明确的目的性（为什么学），对技术原理能够更好地融会贯通（学什么），也可以更好地检验学习效果（学得怎样）。

本书的内容以作者在亚思晟高端 IT 实训过程中的项目实践为基本素材整理而来，同时结合作者多年项目设计开发经验，它具有良好的实践性和可操作性，并具有具体化、通俗化的特点。

本书特点

1. 重项目实践

我们多年经验的体会是“IT 是做出来的，不是想出来的”，理论虽然重要，但一定要为实践服务！以项目为主线，带动理论的学习是最好、最快、最有效的方法！本书的特色是提供了一个完整的 SuperVCD 项目。通过此书，我们希望读者对项目开发流程有个整体了解，减少对项目实践的盲目感和神秘感，能够根据本书的体系循序渐进地动手做出自己的真实项目来！

2. 重理论要点

本书是以项目实践为主线，着重讲解 C#核心技术理论中最重要、最精华的部分，以及它们之间的融会贯通，这就是本书的特色！读者首先通过项目把握整体概貌，之后深入局部细节，系统学习理论；然后不断优化和扩展细节，完善整体框架和改进项目。

既有整体框架，又有重点技术。一书在手，思路清晰，项目无忧！

为什么选择这本书

本书基于全新 Project-driven training（项目驱动）理念，围绕一个项目（SuperVCD）来贯穿 C#核心技术开发各个模块的理论讲解，这是与市场上许多 C#基础教材的最大区别。另外，此书是《项目实践精解》系列教材的一本，和其他相关教材形成一个完整的体系。

本书与《项目实践精解》系列中其他图书的区别及联系

“万丈高楼平地起”，好的开头等于成功的一半！这些都说明打基础的重要性。这本书就是介绍C#核心技术基础的。它是系列丛书《项目实践精解》其中的一本，也是学习《项目实践精解：ASP.NET应用开发——基于ASP.NET、C#和ADO.NET的三层架构案例分析》和《项目实践精解：IT项目的面向对象分析设计、开发及管理——基于.NET平台的ERP系统案例分析》的基础。

本书的组织结构和学习指南

常常有开发人员和学员问我，什么是学习C#的最佳途径？我的回答是“多做，以项目驱动”。学员又继续问我，那学习理论就不重要了吗？我的回答是“当然也重要， 但同样需要以项目驱动”。为什么呢，试想一下，当你面对上百页的参考书或者文档时，是否会茫然不知所措，不知如何开始，不知彼此谁更重要？所以只有以项目驱动才能更好地明确重点和明确主线，才能更好地分配时间和精力，这样才是最有效的。

本书是围绕一个完整的SuperVCD项目来组织和设计学习C#核心技术的。

第1章 C#核心技术概述

首先介绍.NET平台以及C#历史、现状和特点；之后引入C#核心技术中最重要的面向对象基本概念，包括封装、继承和多态；最后概述C#核心技术体系结构，包括核心部分以及应用部分等。

第2章 SuperVCD项目概述

作者希望推广一种最有效的学习与培训的捷径，这就是Project-driven training，也就是用项目实践来带动理论的学习。第2章重点介绍一个完整的项目（SuperVCD）。在此提供了完整的需求分析、结构分析和运行指南。

接下来具体介绍C#核心技术开发和编码。

第3章 C#启动

学习C#开发的第一步是熟悉C#开发环境，包括.NET开发平台和C#开发工具；在此基础上讲解C#应用程序开发的基本概念，包括C#程序的基本结构，定义类、对象、方法等；以及学习如何编辑、编译和运行C#应用程序。

第4章 C#基础语法（一）标识符、关键字及数据类型

学习C#语言，首先要学习基础语法。本章介绍标识符（Identifier）、关键字（Keyword）及变量和常量这些基本元素；另外还介绍C#的数据类型，包括值类型和引用类型。

第5章 C#基础语法（二）表达式及流程控制

本章继续介绍C#基础语法：C#运算符；表达式运算，包括运算符的优先次序和数据类

型转换；以及流程控制，包括顺序流程、分支流程和循环流程。

第 6 章 C#基础语法（三）数组

数组的概念也是 C#语言中的一个重要组成部分。本章介绍数组的声明、生成和定义，数组的使用，以及数组对象的高级功能。

第 7 章 C#面向对象核心语法

在前几章的基础之上，现在进入本书重点内容的学习，也就是面向对象的重要概念。我们在第 7 章介绍面向对象核心语法，包括封装（C#中的类、方法和变量，构造方法，方法重载，内部类和分布类）、继承（继承概念、方法重写）及多态（多态概念和多态实现）。

第 8 章 C#面向对象高级语法（一）

第 8 章继续介绍面向对象的一些高级特性，包括静态（Static）变量和方法、密封（Sealed）变量和方法、访问规则（Access Control）、抽象类和方法、接口（Interface）及集合等。

第 9 章 C#面向对象高级语法 （二）

第 9 章继续介绍面向对象的一些高级特性，包括运算符重载、结构和枚举、字符串与正则表达式、委托与事件及泛型等。

第 10 章 异常处理

异常处理也是 C#的一个重要概念，它能够保证程序运行的健壮性。本章内容包括异常定义、异常分类、异常处理（try、catch、finally、throw 语句）及自定义异常。

第 11 章 使用 WinForm 建立用户图形界面

从第 10 章之后的内容属于 C#核心技术的应用部分，或曰高级部分。本章介绍如何使用 WinForm 建立用户图形界面，包括 WinForm 概述，WinForm 基础，常用组件、容器和事件处理等。

第 12 章 多线程高级编程

对多线程的支持，是 C#语言的一个重要优点，它可以实现代码的并行性，提供程序的性能。本章介绍多线程高级编程技术，包括线程简介，创建、初始化和启动线程，线程状态的转化，资源同时读取问题和线程的同步，线程的自动管理等。

第 13 章 使用输入输出类

输入输出是任何计算机语言都要涉及的，C#也不例外。在本章里将介绍如何使用输入输出类，包括 C#文件和目录的操作，流的概念和使用流来进行文件操作，以及 XML 技术等。

第 14 章 使用 TCP/IP 和 UDP/IP 开发网络程序

C#从一诞生起，就是面向网络的。在本章里介绍使用 TCP/IP 和 UDP/IP 开发网络程序，包括网络编程简介、理解 TCP/IP 及 UDP/IP 协议、开发 TCP/IP 网络程序、开发 UDP/IP

网络程序等。

第 15 章 ADO.NET 数据库开发

数据库的使用，是应用开发中必不可少的组成部分。在本章里介绍数据库的基础原理、SQL 语言、SQL Server 数据库的基本使用、使用 ADO.NET 开发数据库程序的步骤和 ADO.NET 的常用对象。

在本书的附录中，介绍 Visual Studio 工具、C#编程规范、C#与 Java 的比较、C#与 C++的比较以及单元测试工具介绍等内容。

这本书是否适合您

阅读此书，要求读者具备计算机编程基础知识。

本书结构清晰，注重实用，深入浅出，非常适合作为 C#核心技术的培训教材或自学教材，同时也适合作为 C#开发人员的参考书。

感谢

本书由梁立新主持编写，其他参与编写的人员有梁恒、林瀚、沈彬、于亚杰、孙夏、张瑞、张洪亮、武永琪、唐海余。

在本书的编写过程中得到了许多支持和帮助。北京亚思晟科技有限公司负责本书的审校和监制工作，部分工作人员利用宝贵时间为本书提供项目相关代码与文档，并测试了项目的功能和性能，朋友和家人，特别是妻子 Linda，对本书提供了有益的建议和帮助，在此表示衷心的感谢。最后，感谢电子工业出版社对本书出版的协助。

联系方式

北京亚思晟商务科技有限公司

地址：北京海淀上地东路 1 号院鹏寰国际大厦 501

网址：www.ascenttech.com.cn

电话：58859825/26/27/28/29

目 录

目 录

第 1 章　C#核心技术概述

目前，国内外信息化建设已经进入蓬勃发展阶段，Microsoft .NET 作为面向网络的平台，前景无限看好。C#作为.NET 平台的一个核心组件，起着至关重要的作用。在这里我们首先概要地介绍一下.NET 平台和 C#核心技术，主要包括以下几项：

- Microsoft .NET 概述
- C#的历史、现状和特点
- C#与面向对象
- C#核心技术体系结构

1.1　Microsoft .NET 介绍

1.1.1　Microsoft .NET 概述

.NET，相信你早已听过这个概念。微软希望.NET 能做到让任何人从任何地方、在任何时间、使用任何装置来获得互联网上的服务。Microsoft .NET 包括用于创建和操作新一代服务的.NET 基础结构和工具、用于建立新一代高度分布式的数以万计的.NET 组件服务，以及用于启用新一代智能互联网设备的.NET 设备软件。

1.1.2　Microsoft .NET 平台的意义

我们来看一下 Microsoft .NET 对开发人员、IT 专业人员以及企业应用的巨大意义。

1. 对于开发人员

Microsoft .NET 的策略是将互联网本身作为构建新一代操作系统的基础，对互联网和操作系统的设计思想进行合理扩展。.NET 对开发人员来说十分重要，它不但会改变开发人员开发应用程序的方式，而且使得开发人员能创建出全新的各种应用程序。新型开发范例的核心是 Web 服务这个概念的引入。Web 服务是一种应用程序，它可以通过编程并使用标准的 Internet 协议，像超文本传输协议（HTTP）和 XML，将功能展示在互联网和企业内部

网上。我们可将 Web 服务视做 Web 上的组件编程，从理论上讲，开发人员可通过调用 Web 应用编程接口（API），将 Web 服务集成到应用程序中。其调用方法与调用本地服务类似，不同的是 Web API 调用可通过互联网发送给位于远程系统中的某一服务。

.NET 正是根据这种 Web 服务原则而创建的，微软目前正着手提供这个基本结构，以便通过.NET 平台的每一部分来实现这种新型的 Web 服务。而 Visual Studio .NET、.NET 框架、Windows .NET 和.NET 企业服务器，正是为进行基于 Web 服务模型的应用程序开发而量身定做的新一代开发工具和基本结构。

2. 对于 IT 专业人员

目前，IT 专业人员能够利用与构建.NET 平台所使用的相同的技术。.NET 企业服务器和 Windows 操作系统，为创建具有高度可管理性的、能迅速投入市场的应用程序提供了坚实基础。它们利用的是可扩展标记语言（XML），因此随着 Web 体系结构的革新，在此平台上创建的程序依然很有价值。

开发应用程序的.NET Web 服务模型将为企业应用程序的创建开辟一条新路。通过企业内外多种服务的联合，很容易把企业内部数据和客户及合作伙伴的相关数据结合在一起，这大大简化了应用程序的创建过程。这就为最终用户发掘了巨大的功能涵盖性。

3. 对于企业

Microsoft .NET 平台将从根本上改善计算机和用户之间进行交互的方式，最大限度地发挥电子商务中计算技术的重要作用。.NET 能实现确保用户从任何地点、任何设备都可访问其个人数据和应用程序。除此之外，.NET 技术还可实现多个应用程序在逻辑上的松散或紧密的耦合链接。根据设计，.NET 使得用户无须在如何与计算机进行交互上费力，从而全身心地投入到使计算机自动执行任务、实现最终目标的工作中。通过使用 XML 行业标准，可将用户数据进行跨站点和应用程序的链接，从而轻松实现当前很难实现的操作。

.NET 把雇员、客户和商务应用程序整合成一个协调的、能进行智能交互的整体，而各公司无疑将是这场效率和生产力革命的最大受益者。简言之，.NET 将致力于为人类创造一个消除任何鸿沟的商务世界。

1.1.3 Microsoft .NET 的基本模块

.NET 包括 5 个主要组成部分，即 Windows .NET、.NET 框架（.NET Framework）、.NET 企业服务器、模块构建服务（Building Block Services）和 Visual Studio .NET。Window .NET 是融入了.NET 技术的 Windows，它紧密地整合了.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 Microsoft 公司的下一代 Windows 桌面平台。.NET 框架的目的是便于开发商更容易地建立网络应用程序和 Web 服务。它的关键特色是提供了一个多语言组件开发和执行的环境。.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，它为企业未来开展电子商务提供了高可靠性、高性能、可伸缩性以及可管理性。模块构建服务是.NET 平台中的核心网络服务集合。Visual Studio .NET 是基于 XML 的编程工具和环境。它便于快速开发符合.NET 体系的软件服务，使其在独立设备、企业数据中心和互联网之间的传送更加容易。

其中最核心的部分是.NET 框架，.NET 框架的设计目标包括以下几项：

- 提供一个一致的面向对象的编程环境，而无论对象代码是在本地存储和执行的，还是在本地执行但在互联网上分布的，抑或是在远程执行的。
- 提供一个将软件部署和版本控制冲突最小化的代码执行环境。
- 提供一个可提高代码（包括由未知的或不完全受信任的第三方创建的代码）执行安全性的代码执行环境。
- 提供一个可消除脚本环境或解释环境的性能问题的代码执行环境。
- 使开发人员的经验在面对类型大不相同的应用程序（如基于 Windows 的应用程序和基于 Web 的应用程序）时保持一致。
- 按照工业标准生成所有通信，以确保基于.NET 框架的代码可与任何其他代码集成。

.NET 框架包括两个主要组件：公共语言运行时（Commnon Language Runtime，CLR）和.NET 框架类库（Framework Class Library，FCL）。

公共语言运行时是.NET 框架的基础，可以将它看做一个在执行时管理代码的代理，它提供内存管理、线程管理和远程处理等核心服务，并且强制实施严格的类型安全，以及可提高安全性和可靠性的其他形式的代码准确性。代码管理的概念是公共语言运行时的基本原则。以公共语言运行时为目标的代码称为托管代码。.NET 框架用统一的命令集来支持任何的编程语言，支持混合语言编程，确保程序的可移植性。托管代码只是意味着在内部可执行代码与运行自身间存在已定义好的合作契约。对于像生成对象、调用方法等这样的任务，被委托给了运行语言，这使得运行语言能为可执行代码提供额外的服务。

.NET 框架类库是一个综合性的面向对象的可重用类的集合，可以使用它来开发多种应用程序，这些应用程序包括传统的命令或图形用户界面应用程序，也包括基于 ASP.NET 所提供的最新创新的应用程序（如 Web 窗体和 XML Web 服务）。.NET 框架类库确保用户程序能够访问公共语言运行时环境。

语言互操作是一种代码与使用其他编程语言编写的另一种代码进行交互的能力。语言互操作有助于最大程度地提高代码的重用率，从而提高开发过程的效率。公共语言运行时提供内置的语言互操作支持。.NET 框架通过如下 3 个基础来保障语言互操作：

- 中间语言（Microsoft Intermediate Language，简称为 MSIL 或 IL），所有的.NET 语言都被编译为中间语言。这是一组可以有效地转换为本机代码且独立于 CPU 的指令。IL 包括用于加载、存储、初始化对象及调用对象的方法的指令，还包括用于

算术和逻辑运算、控制流、直接内存访问、异常处理和其他操作的指令。要使代码可运行，必须先将 IL 编译为特定 CPU 的机器码，这通常是通过即时（JIT）编译器来完成的。在编译过程中，必须通过验证过程来查看是否可以将代码确定为类型安全。代码只需即时编译一次。当再次运行编译过的代码时，将运行已经通过 JIT 编译得到的机器码。这种进行 JIT 编译然后执行代码的过程一直重复到执行完成时为止。

- 通用类型系统（Common Type System，CTS）：通用类型系统提供了定义、管理和使用类型的规范。它提供了所有支持语言互操作的语言都必须遵守的规则集，以确保不同语言所创建的类型能够进行交互操作。
- 公共语言规范（Common Language Specification，CLS）：公共语言规范定义了编译器和库管理器必须遵守的规则，以确保它们所生成的语言和代码能够与其他.NET 语言进行互操作。公共语言规范是通用类型系统的一个子集。

C#是一种完全符合公共语言规范的语言，能够与其他.NET 语言进行互操作。C#与 Microsoft .NET 的关系体现在两个方面：其一，C#的设计目标就是用来开发在.NET 框架中运行的代码，因此.NET 框架是 C#程序的运行环境；其二，C#的编程库是.NET 框架类库，即 C#的数据类型和操作类都来自.NET 类库。

1.2 C#的历史、现状和特点

1.2.1 C#产生的历史

1999 年，微软公司开始研发一种名为“Cool”的新开发语言，而具体内幕一直是个谜。直到 2000 年 6 月 26 日微软公司在奥兰多举行的“职业开发人员技术大会”（PDC 2000）上，这个谜底才揭晓，这种新的、先进的、面向对象的开发语言就是 C#（发音为“C Sharp”）。那么微软公司为什么要开发 C#，C#究竟能给开发者带来什么好处呢？

1995 年，Sun 公司正式推出了面向对象的开发语言 Java，并提出了跨平台、跨语言（Write Once and Run Anywhere）的概念，此后 Java 逐渐成为了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大有“山雨欲来风满楼”之势。

很快，在众多研发人员的努力下，微软公司也推出了自己基于 Java 语言的编译器 Visual J++。Visual J++在最短的时间里由 1.1 版本升级到了 6.0 版本。这不仅仅是数字上的变化，集成在 Visual Studio 6.0 中的 Visual J++ 6.0 的确有了质的变化：不但虚拟机（JVM）的运行速度大大加快，而且增加了许多新特性，同时支持调用 Windows API，这些特性使得 Visual J++成为强有力的 Windows 应用开发平台，并成为业界公认的优秀 Java 编译器。

不可否认，Visual J++具有强大的开发功能，但其主要运用在 Windows 平台的系统开发

中。Sun 公司认为 Visual J++违反了 Java 的许可协议，即违反了 Java 开发平台的中立性，因而，对微软公司提出了诉讼，这使得微软公司处于极为被动的局面。就在人们认为微软公司的局面不可能再有改观的时候，微软公司却另辟蹊径，决定推出其进军互联网的的庞大计划——.NET 计划和该计划中旗帜性的开发语言——C#。

微软公司的.NET 是一项非常庞大的计划，也是其此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。Visual Studio .NET 则是微软.NET 的技术开发平台，其重要性可见一斑，而 C#就集成在 Visual Studio .NET 中。

为了支持.NET 平台，Visual Studio .NET 在原来 Visual Studio 6.0 的基础上进行了极大的修改和变更。在 Visual Studio .NET β 版中你会发现 Visual J++消失了，取而代之的就是 C#语言。

微软公司对 C#的定义是：“C#是一种类型安全的、现代的、简单的，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（简称 VB）的高生产率和 C++的行动力。”这个定义是恰如其分的，因为在 Visual Studio .NET 中，我们已经可以利用 C＃极其轻松地开发出强大的企业级分布式应用系统了。

1.2.2 C#的优势

1. 快速应用开发（RAD）功能

支持快速应用开发（Rapid Application Development，RAD）可以说是目前开发语言最为重要的一大功能，这也正是 C/C++的致命伤。网络时代应用系统的开发必须按照网络时代的速度来进行。支持快速开发可以使得开发人员的开发效率倍增，从而使得他们可以从繁重的重复性劳动中解放出来。C#的 RAD 功能之一是垃圾收集机制，它将减轻开发人员对内存的管理负担。利用 C#的这些功能，可以使开发者通过较少的代码来实现更强大的应用程序，并且能够更好地避免错误发生，从而缩短应用系统的开发周期。

2. 语言的自由性

用 C#编写的程序能最大程度地和任何支持.NET 的语言互相交换信息。能够继承和使用任何语言所编写的程序可以称得上是知识的继承，这样做的好处是节省了大量的工作，而不必把 COBOL 等语言强行改成另一种语言，.NET 让各种语言真正地互相交流了。C#和其他.NET 语言有着最好的协作，这点对开发人员非常重要。全球从事编写软件的人当中，大约有 50%以 Visual Basic 作为基本的编程工具。在跨入.NET 编程时代的时候，这些人轻松地使用了 VB.NET 开发 Web 上的应用程序。想想以后你居然可以和你的那些只懂 VB 的同事真正在一起共同开发.NET 的应用程序，这是不是一件让人轻松和高兴的事情呢？

3. 强大的 Web 服务器端组件

在 C#的 Web 编程中，最让人兴奋的是 Web 服务器端的组件，它们不仅包括传统组件，还包括那些能够自动连接五花八门服务的可编程组件。你可以用 C#编写自己的服务器端组件。服务器端组件和标准的 HTML 提供的服务自然是不一样的，使用它们可以更自由、更容易地进行数据绑定。更多服务器组件很快会出现，有了这些强大的组件，我们可以设计出功能更加强大的企业级分布式应用系统。

4. 支持跨平台

随着互联网应用程序的应用越来越广泛，人们逐渐意识到由于网络系统错综复杂，使用的硬件设备和软件系统各不相同，因此开发人员所设计的应用程序必须具有强大的跨平台性。C#编写的应用程序就具有强大的跨平台性，这种跨平台性也包括了 C#程序的客户端可以运行在不同类型的客户端上，比如 PDA、手机等非 PC 装置中。

5. 与 XML 的融合

由于 XML 技术真正融入到了.NET 和 C#之中，C#的编程变成了真正意义上的网络编程，甚至可以说.NET 和 C#是专为 XML 而设计的。使用 C#的程序员可以轻松地用 C#内含的类来使用 XML 技术。就这方面而言，C#给程序员提供了更多的自由和更好的性能来使用 XML。

6. 对 C++的继承

C#继承并保留了 C++的强大功能，例如， C#保留了类型安全的检测和重载功能，C#还提供了一些新功能取代了一些原来的 ANSI C 预处理程序的功能，提高了语言的类型安全等安全性。

1.3 C#与面向对象

C#是面向对象的编程语言。要想学好 C#, 首先就要建立起面向对象的思想。对于 C#来讲，一切皆对象，所以我们首先介绍一下面向对象的基本概念，之后所有的内容将围绕着面向对象展开，例如类（Class）、对象（Object）、方法（Method）、实例（Instance）等，这对深入学习、理解 C#语言大有益处。

面向对象作为一种思想及编程语言，为软件开发的整个过程，包括从分析设计到维护，提供了一个完整解决方案。面向对象堪称是软件发展取得的里程碑式的伟大成就。

从 20 世纪 80 年代后期开始进行的面向对象分析（OOA）、面向对象设计（OOD）和面向对象程序设计（OOP）等新的系统开发方式模型的研究，在有些文献中被统称为 OO 范型。