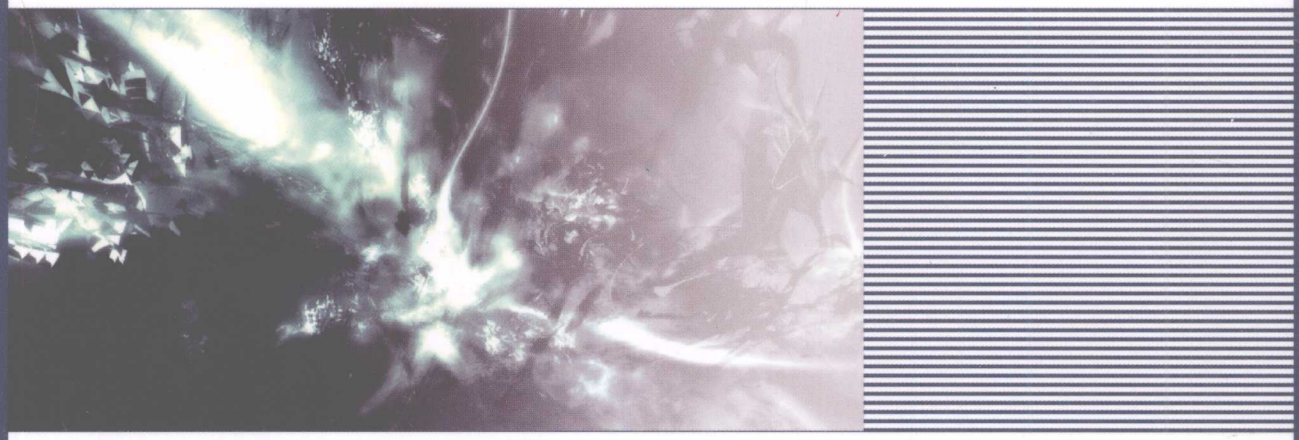
21世纪高等院校网络工程规划教材

21st Century University Planned Textbooks of Network Engineering



C#网络编程技术教程

C# Network Programming

金华 华进 编著

- 侧重编程基本思路和方法
- 强调C#网络开发技术
- 循序渐进引用典型案例

人民邮电出版社
POSTS & TELECOM PRESS

21世纪高等院校网络工程规划教材

21st Century University Planned Textbooks of Network Engineering

C#网络编程技术教程

C# Network Programming

金华 华进 编著

人民邮电出版社

北京

图书在版编目（CIP）数据

C#网络编程技术教程 / 金华，华进编著. —北京：人民邮电出版社，2009.2
21世纪高等院校网络工程规划教材
ISBN 978-7-115-18941-7

Ⅰ. C… Ⅱ. ①金…②华… Ⅲ. C语言－程序设计－高等学校－教材 Ⅳ. TP312

中国版本图书馆CIP数据核字（2008）第155125号

内 容 提 要

本书是一本系统介绍C#网络编程技术的教材，全书共11章，由两部分组成。第1部分（第1～5章）为基础知识，内容包括Visual C#.NET集成开发环境、C#语言的基础知识、面向对象的程序设计、C#的Windows编程方法、C#的常用数据类、网络编程的基本概念、网络套接字编程以及多线程编程的方法概述等。第2部分（第6～11章）介绍网络编程的几个常用领域，内容包括TCP/UDP编程、FTP编程、电子邮件协议编程、HTTP编程、Web Service编程以及密码术网络编程等。

本书可作为高等院校计算机、网络工程、通信工程、信息安全等专业的教材，也可作为相关工程技术人员的参考用书。

21世纪高等院校网络工程规划教材

C#网络编程技术教程

◆ 编　　著　金　华　华　进
责任编辑　滑　玉
执行编辑　张　鑫

◆ 人民邮电出版社出版发行　　北京市崇文区夕照寺街14号
邮编　100061　　电子函件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
北京隆昌伟业印刷有限公司印刷

◆ 开本：787×1092　1/16
印张：17.75
字数：443千字　　2009年2第1版
印数：1－3 000册　　2009年2北京第1次印刷

ISBN 978-7-115-18941-7/TP

定价：29.00元

读者服务热线：(010)67170985　印装质量热线：(010)67129223
反盗版热线：(010)67171154

前　言

随着 Internet 的快速发展，面向网络的开发技术已经成为 IT 发展的重要分支之一。为了支持下一代 Internet 的编程结构，微软公司推出了.NET 战略平台，而 C#正是.NET 技术的核心开发语言。C#采用面向对象的编程技术，提供了开发 Windows 应用程序、Web 应用程序最便捷、最行之有效的方法。用户不需掌握太多的专业编程知识就能够设计出高难度的图形化应用程序，使得用户真正从烦琐复杂的工作中解脱出来。因此，掌握 C#网络编程技术是网络开发人员开发网络应用程序的基本要求之一。

为了能够编写一本真正适合于课程教学的 C#网络编程教材，我们结合多年来的网络编程教学体会和经验做了一些有益的探讨，在内容安排上从基础理论知识出发，先介绍原理与编程思想，再通过具体实例加以说明。本书具有如下特点。

- 内容的设置注意循序渐进与合理搭配，力求通过简单实例介绍 C#基础知识，重点突出实际应用中的常用类和方法，便于读者快速掌握 C#基本编程方法。
- 基础理论与实用开发相结合，所选实例都具有较强的概括性和实际应用价值。
- 应用内容力求全面，涵盖了网络编程的常用领域。
- 突出应用编程思想与开发方法的介绍，即对各种协议编程先介绍其编程思想，再辅以实例说明。

本书共 11 章，先介绍了在 Visual Studio.NET 2005 集成开发环境下的 C#编程基础知识，内容包括面向对象编程基础、C#常用数据类、C#的 Windows 编程等；在此基础上，引入 C#网络编程，以简明直观的方式介绍了网络编程涉及的常用技术，内容包括 TCP/IP、套接字编程和多线程编程；最后分别介绍了 TCP/UDP 编程、FTP 编程、电子邮件协议编程、HTTP 编程、Web Service 编程、密码术与网络编程等，内容涵盖了基本的网络应用技术。本书各章都配有习题，便于读者理解掌握所学内容。本书所有实例程序都在 Microsoft Visual Studio 2005 集成开发环境下调试通过。

本书由金华、华进编著，李峰、张志祥、韩小祥参与了部分内容的编写。此外，袁晓云、曾宇、刘芳等参加了整理与编排工作。全书由金华统稿。

由于作者水平有限，加之编写时间仓促，书中难免存在疏漏和不妥之处，敬请广大读者批评指正。

编　者

2008 年 9 月

目　录

第 1 章　C#.NET 概述

本章学习目标

（1）了解.NET 框架及其特点。

（2）了解 C#开发背景，以及它与.NET 框架之间的关系。

（3）掌握 Visual C#.NET 集成开发环境。

（4）掌握创建 C#程序的一般方法和 C#程序结构。

1.1　.NET 平台介绍

Microsoft .NET 是微软公司于 2000 年 6 月 22 日发布的下一代计算计划，该计划的主要目的是让网络由呈现式的平台转为完全的平台，使微软已有的软件在 Web 时代适用于传统的 PC。具体来讲，就是通过.NET 计划，可以让我们在任何时间、任何地点，使用任何设备获取信息并且得到服务。

1.1.1　什么是.NET

.NET 是微软公司推出的一个全新概念的技术，它代表了一个集合、一个环境和一个可以作为平台支持下一代 Internet 的可编程结构。同时.NET 也是一种新的计算平台，它简化了在高度分布式 Internet 环境中开发应用程序的过程，为用户提供更加丰富和完善的解决方案。

Microsoft .NET 方案由以下四个关键部分组成。

（1）.NET 构件块服务，即对某些特定服务程序的访问，如用于文件存储的服务、日历管理或 Passport.NET（一种身份鉴别服务）。

（2）.NET 设备软件，是运行于新型 Internet 设备上的软件。

（3）用户体验，包括自然界面、信息代表和智能标签等功能，这些技术可以自动建立超链接，这些超链接指向与用户创建的文档中的单词或短语相关的信息。

（4）基础结构，由.NET 框架、Microsoft Visual Studio .NET、.NET 企业服务器和 Microsoft Windows .NET 组成。

Visual Studio.NET 是微软公司为实现其.NET 技术而开发的一整套工具组件。它简化了开发功能强大、性能可靠的企业网络解决方案。通过提供端到端的网络开发能力以及可伸缩、可复用的服务器端组件，Visual Studio.NET 大大提高了生产率，并促使商务活动更加有效地关注快速多变而又充满竞争的市场需求。

Visual Studio.NET 的主要作用如下。

（1）提供加速开发过程的高效工具。Visual Studio.NET 提供了一个统一的、紧密集成的可视化编程环境，以帮助用户简化开发网络应用程序的过程，缩短学习使用方法的时间。它提供了一种新的语言——C#。通过共享的 HTML、XML 和样式单编辑器，用户可以轻松地借助包括 C#在内的任何一种 Visual Studio 语言来开发网络应用程序。

（2）提供对各种网络应用程序的快速设计能力。借助 Web Form，用户可以用他们在开发基于窗体的桌面应用程序时所使用的技巧来创建跨平台、跨浏览器的网络应用程序。

（3）利用 XML 和 Web Service 来简化分布式计算。Web Service 借助标准的 Internet 协议在网络上调用商务逻辑。HTTP 被作为 Web Service 传输的基础协议，该协议使得对功能的请求能够穿越各种团体所使用的防火墙。XML 被用来对上述功能请求的参数进行格式统一，从而使这些请求能够使用于所有的软件和硬件。这样使得对 Web Service 的访问可以通过任何一种语言、使用任何一种组件模型在任何一种操作系统上实现。

（4）快速构建中间层商务组件。Visual Studio 的一个核心目标就是要为基于服务器的应用程序提供应用程序快速部署工具。利用 Visual Studio.NET 创建的组件将为您的商务运作提供足够的功能和伸缩性。

（5）构建可靠的可伸缩解决方案。利用 Visual Studio.NET，用户可以非常轻松地创建具有自动伸缩能力的可靠的应用程序和组件。

简而言之，.NET 是一种面向网络、支持各种用户终端的开发平台。.NET 的核心内容之一就是要搭建第三代 Internet 网络平台，这个网络平台将解决网站之间的协同合作问题，从而可以最大限度地获取信息。在.NET 平台上，不同网站之间通过相关的协定联系在一起，网站之间自动交流、协同工作，从而提供最全面的服务。

1.1.2 .NET 的核心组件

.NET 包括以下核心组件。

（1）一组用于创建互联网操作系统的构建块。包括用于用户认证的 Passport.NET 以及用于文件存储的服务，用户首选项日历管理和众多的其他任务。

（2）构建和管理新一代服务的基本结构和工具。包括 Visual Studio.NET 企业服务器、.NET Framework 和 Windows.NET。

（3）能够启用新型智能互联网设备的.NET 设备软件。

（4）.NET 用户体验。

1.1.3 .NET 的新特性

.NET 是一种全新的技术，其平台由公共语言运行时、基础类库和公共语言规范组成。基础类库展现了公共语言运行时的功能，类似于 Windows API。但基础类库提供了比 API 更为高层的功能来方便代码的重用。作为一个新的平台，它包括了很多新特性：一致的编程模式，简化的编程模式，平台与处理器独立，支持多语言的开发，自动内存管理，一致的出错处理方式，完美的安全机制，XML 和 SOAP 的引入等。具体说明如下。

（1）一致的编程模式。在.NET 环境中，所有的应用程序都采用通用的面向对象的编程模式，而 Windows 环境中既有 DLL 函数也有 COM 对象。

（2）简化的编程模式。这是最令开发人员欢迎鼓舞的消息，在.NET 环境下，由于 CLR 的作用，程序开发人员不再需要深入了解和 Windows 或 COM 架构相关的 GUID、Iunknown( )、AddRef( )、Release( )和 HRESULTS 等知识。.NET 平台不但隐藏了实现细节，而且在新的平台上，这些概念已经消失了。

（3）平台与处理器独立。微软中间语言（MSIL）独立于 CPU，是一种比传统机器语言层次更高的语言。对于任何操作平台，只要支持.NET 运行就可以运行.NET 应用程序。现在所有的 Windows 平台均可以实现这一点，将来甚至在非 Windows 操作系统上也可以实现这一点，直接在 C++这样的语言中使用。

（4）支持多语言的开发。按照 COM 的原理，代码重用是建立在二进制代码的级别上的。在.NET 环境下，代码重用可以建立在源码级别上，也就是说，用 C#语言编写的某个类可以直接在 C++这样的语言中使用。.NET 有这样的巨大威力在于它为所有支持.NET 编程的语言提供了一整套通用类型系统。

（5）自动内存管理。对于所有开发人员而言，最难解决的就是内存泄漏的问题。在.NET 环境下这个问题得到彻底解决，自动内存管理功能已经纳入 CLR 之中。该功能会定时检查被丢弃的内存，并进行相应的回收，从而使程序员可把复杂的内存管理交给平台自己来处理。

（6）一致的出错处理方式。相信所有的 Windows SDK 程序员都对 Windows 环境下混乱的错误处理方式感到厌烦，如 Win32 错误代码、异常情况处理和 HRESULT 等。在.NET 环境下，所有的程序都采用统一的错误处理方式（产生异常）。

（7）完善的安全机制。.NET 的出现是为了迎合下一代因特网环境下的企业级计算，一般的访问控制已经不能满足要求，所以在安全方面，.NET 相对于 Windows 等其他系统而言，有了更深入的改进，如从装载一个类开始就进行确认性检查；在访问代码和相应资源时，实施代码访问安全措施。.NET 还提供了一整套机制来判断角色和确认身份信息，并且能做到跨进程和跨机器，从而确保所需的代码在远端不会受到破坏。.NET 的安全性也深深嵌入到 CLR 结构中，以确保应用程序本身安全。这些安全机制是对现有操作系统安全机制的一种本质上的扩展，从而使.NET 在安全性上进一步加强。

（8）XML 和 SOAP 的引入。回忆一下过去的分布式应用程序的设计，通常设计两层应用程序，在此基础上出现了如 CORBA、IIOP、RMI 和 DCOM 这样的协议。人们已经熟悉了这样的分布式系统。但是这种系统的弊端就是灵活性差，因为这种设计方式使得应用程序固定在服务器端。而因特网是整个松散连接和分布非常广的世界。原有的 Client/Server 结构已经过时，因此就提出了全新的编程模式，而 XML 和 SOAP 能使这种模式很好地工作。在.NET 中，XML 和 SOAP 已经深深地融入其中并成为非常重要的组成部分。

1.1.4　.NET 框架

.NET 框架是.NET 平台的基本架构，其目的是为了更容易建立网络应用程序和网络服务。此外，Microsoft.NET 框架还规定了代码访问安全和基于角色的安全。通过代码访问安全机制，为应用程序指定完成工作所必需的权限，从而保障按照开发人员的意图全面、细致地设计安全可靠的应用程序。.NET 平台的框架结构如图 1.1 所示。整个平台由以下 7 部分组成。

（1）公共语言运行时（Common Language Runtime，CLR）。它是整个开发框架的基础。从微软公司的设计意图来看，是利用它来实现跨平台操作。新平台的程序编译过程类似 Java，

程序先被编译成微软中间语言。但不会被编译成执行程序，而是采用即时编译（Just In Time，JIT）的方式。具体来说，是由当前操作系统平台上运行的公共语言运行时即时把微软中间语言编译成当前操作系统的机器码，从而实现.NET 的跨平台特性。

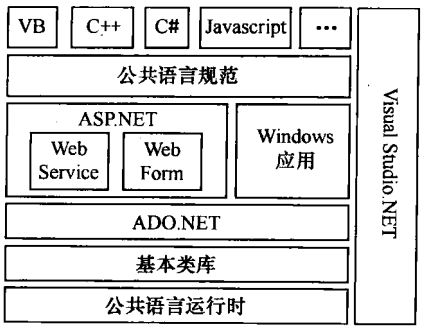


图 1.1 .NET 框架结构

（2）由 CLR 所提供的一组基本类库。在编程接口与公共语言运行时之间，是一个功能强大的基本类库。利用它，程序员可以方便、安全、可靠地访问.NET 平台提供的功能。

（3）数据库编程接口（数据和 XML 类）。.NET 提供了全新的数据库访问技术——ADO.NET，为数据库的管理和访问提供了一系列组件，使用它们可以大大简化数据库的编程工作。

（4）应用程序编程接口。应用程序编程接口包括 ASP.NET、Web Service、Web Form 和控制台程序等模块。利用这些编程接口，开发人员可以快速开发出高质量的.NET 平台的应用程序。

（5）支持多种开发语言的公共语言规范。最上层的多语言支持是通过公共语言规范来实现的。具体来说，微软提出了一种称为微软中间语言（MSIL）的概念，任何其他语言都要先编译成微软中间语言。这样任何开发语言只要能支持公共语言规范，就能集成到微软的.NET 平台上。

（6）多种开发语言。如 VB、C++、C#、Javascript 等。

（7）全面支持.NET 开发的集成开发环境 Visual Studio.NET。

1.2 C#语言简介

1.2.1 C#的开发背景

C 和 C++这两种语言为程序员提供了丰富的功能、高度的灵活性和强大的底层控制能力。而这一切都不得不在效率上做出不同程度的牺牲。和其他的开发语言相比（如 Visual Basic），C/C++通常需要更长的开发周期。特别是对于 VC++来说，大部分的程序结构都被封装在 MFC 中。因此对于初学者来说程序结构显得十分混乱，学习变得十分艰苦。此外，在 VC++版本不断更新的过程中，既要适应新的技术（如 COM、ALT 等），又要与前一个版本兼容，VC++在此之后的每一次升级都使程序结构变得越来越复杂，而且出现了越来越多的变量类型，从而带来更多的问题。因此许多 C/C++开发人员都在寻找一种可以在功能和开发效率间提供更多平衡的开发语言。

在这一过程中，人们改进、开发出了许多语言以提高软件生产率，但都或多或少以牺牲 C 和 C++程序员所需要的灵活性为代价。这样的解决方案在程序员身上套上了太多的枷锁，限制了他们能力的发挥，也不能很好地与原有的系统兼容，而且它们与当前的 Web 应用结合得也不好。

合理的 C/C++替代语言应该可以对现存和潜在的平台上的高效开发提供有效的支持，并可以使 Web 开发非常方便地与现存的应用相结合。此外应该提供一些 C/C++开发人员在必要的时候使用底层编程的功能。

C#是从 C 和 C++演变而来的，并且结合了 C/C++的强大功能、Java 的面向对象特性和 Visual Basic 的易用性，从而成为一种简单的类型安全、面向对象的编程语言，也是.NET 公

共语言运行环境的内置语言。C#代码被编译为托管代码，这意味着它可以从公共语言运行库的服务中获益。这些服务包括语言互操作性、垃圾回收、增强的安全性和改进的版本支持。C#同时还具有 Delphi 的一些优点。

1.2.2　C#语言的特点

如前所述，C#是专门为.NET 应用而开发的语言，这从根本上保证了 C#与.NET 框架的完美结合。.NET 框架为 C#提供了一个强大的、易用的、逻辑结构一致的程序设计环境。在.NET 运行库的支持下，.NET 框架的各种优点在 C#中表现得淋漓尽致。具体地讲，C#具有如下的特点。

1. 语法简洁

在默认情况下，C#的代码在.NET 框架提供的可操作环境下运行，不允许直接进行内存操作。它所带来的最大特色是取消了指针，与此相关的那些在 C++中被疯狂使用的、各式各样的操作符，如“::”或“->”已经不再出现。C#只支持一个“.”，对于使用者来说只需要明白具体的名字嵌套就可以了。

语法中的冗余是 C++中的常见问题，如“const”和“#define”以及各种各样的字符类型等。C#对此进行了简化，使用统一的类型系统，摒弃了 C++中多变的类型系统，只保留了常见的形式，别的冗余形式都从它的语法结构中清除出去了。因此不必再去记忆基于不同处理器架构的隐含的类型，甚至各种整型的变化范围。

2. 面向对象设计

C#具有所有关键的面向对象概念和面向对象语言所应有的一切特性，如：封装、继承和多态性。通过精心的面向对象设计，C#成为创建各种组件（从高级商业对象到系统级应用）的最好选择。

C#语言在限制至关重要的核心的语法上，其面向对象技术相对来说已经成熟，而且有很高的效率，同时，要鼓励用户建立好的类结构，这样对继承性就肯定会有一定的限制。例如，一个类可以从无数个类中继承接口，但它只能从一个基类中继承其实现的方式。又如多态函数的模糊性问题在 C#里用一种更清晰的新语法来解决，即声明虚拟和“纯虚拟”的函数。特别是一个类可以先行提供方法执行方式，在该方法的前面通过使用 abstract 关键字来进行声明，使得其子类也这么做。在 C#的类型系统中，每种类型都可以看作一个对象。C#提供了装箱（boxing）与拆箱（unboxing）的机制来完成这种操作，而不给用户带来麻烦。

C#的属性方法是 C#类机制最好的特性之一。这种方法把属性的读写集中到一个地方，以便更容易对它进行控制和使用。其整体效果是使得从对象的角度去看它也是对象，因为它们的属性在语法上更像特性，而不是类似伪函数的调用。使用索引符来对属性的特殊形式进行说明，它通过一种非常直观的语法来显示类中的数组。

C#只允许单继承，即一个类不会有多个基类，从而避免了类型定义的混乱。C#中没有全局函数，没有全局变量，也没有全局常数，都必须封装在一个类中。代码将具有更好的可读性，并且减少了发生命名冲突的可能。

整个 C#的类模型建立在.NET 虚拟对象系统（Virtual Object System，VOS）的基础之上，其对象模型是.NET 基础架构的一部分，而不是其本身的组成成分。这样可以满足比较好的兼容性。

总之，C#具备了良好的开发环境。提供了最好的方式来体现面向对象的概念，使编程过程中更好地体现了面向对象的思想。

3. 与 Web 紧密结合

.NET 中新的应用程序开发模型要求越来越多的解决方案必须与 Web 标准相统一，例如 HTML（超文本标记语言）和 XML。由于历史的原因，现存的一些开发工具不能与 Web 紧密地结合。而在 C#中可以通过 SOAP 实现上述所要求的统一，使 C#能与 HTML 和 XML 建立联系，进而实现大规模、深层次的分布式开发。

由于有了 Web 服务框架的帮助，对程序员来说网络服务看起来就像是 C#的本地对象。程序员们能够利用他们已有的面向对象的知识与技巧开发 Web 服务。仅需要使用简单的 C#语言结构，C#组件将能够方便地为 Web 服务，并允许它们通过 Internet 被运行在任何操作系统上的任何编程语言所调用，从而扩充 Web 的功能。举个例子，XML 已经成为网络中数据结构传送的标准。为了提高效率，C#允许直接将 XML 数据映射成为结构，这样就可以有效地处理各种数据。

4. 完整的安全性和错误处理

语言的安全性与错误处理能力是衡量一种语言是否优秀的重要依据。C#语言就具有完善的安全性和错误处理能力。任何人都会犯错误，即使是最熟练的程序员也不例外。忘记变量的初始化、对不属于自己管理范围的内存空间进行修改这些错误常产生难以预见的后果。一旦这样的软件被投入使用，寻找与改正这些简单错误的代价将会让人无法承受。C#的先进设计思想可以消除软件开发中的许多常见错误，并提供了包括类型安全在内的完整的安全性能。为了减少开发中的错误，C#会帮助开发人员通过更少的代码完成相同的功能。这不但减轻了编程人员的工作量，同时更有效地避免了错误的发生。

C#非常擅长于隐藏基本内存管理，因为它使用了无用存储单元收集器和引用。但在很多时候，开发人员需要对内存进行直接访问。众所周知，指针的使用会带来很大的风险，但指针在编写高效代码时的强大性和灵活性也是极其突出的，为了平衡风险和效率，在 C#语言里，只允许在特别标识的代码块里使用指针。也就是说 C#为了提高代码的安全性，虽然允许使用人们普遍认为不安全的 C++习惯用语代码，但要求用户在前面加上标识不安全的关键字，让开发人员知道该段代码是危险的，并作为服务于 C#编译器的一个标识符，降低对不安全代码的限制。

.NET 运行库提供了代码访问安全特性，它允许管理员和用户根据代码的 ID 来配置安全等级。在默认情况下，从 Internet 和 Intranet 下载的代码都不允许访问任何本地文件和资源。例如，一个在网络上的共享目录中运行的程序如果访问本地的一些资源，那么异常将被触发；若复制到本地硬盘上运行则一切正常。内存管理中的垃圾收集机制减轻了开发人员对内存管理的负担。.NET 平台提供的垃圾收集器（Garbage Collection，GC）将负责资源的释放与对象撤销时的内存清理工作。

变量是类型安全的，C#中不能使用未初始化的变量。对象的成员变量由编译器负责将其置为零，当局部变量未经初始化而被使用时，编译器将作出提醒。C#不支持不安全的指向，不能将整数指向引用类型，例如对象。当进行下行指向时，C#将自动验证指向的有效性。C#中提供了边界检查与溢出检查功能。

5. 版本控制

以前系统中的组件和动态链接库如要升级，由于这些组件或动态链接库都要在注册表中

注册，因此可能带来一系列的问题。例如，安装新程序时自动安装新组件替换旧组件，有可能某些必须使用旧组件才可以运行的程序，使用新的组件却运行不了，甚至可能导致程序的崩溃。为了帮助开发人员处理这些问题，C#在语言中内置了版本控制功能，一方面可以减少开发费用，另一方面又可以使开发人员更加轻易地开发和维护各种商业应用。在.NET 中那些新安装的组件和动态链接库不必在注册表中注册，每个程序都可以使用自带的组件或动态链接库。由于不需要在注册表中注册，软件的安装也变得容易了，一般将运行程序及库文件复制到指定文件夹中就可以了。

6. 兼容性

C#允许与 C 风格的需要传递指针型参数的 API 进行交互操作，DLL 的任何入口点都可以在程序中进行访问。C#遵守.NET 公共语言规范（CLS），从而保证了 C#组件与其他语言组件之间的互操作性。元数据概念的引入既保证了兼容性，又实现了类型安全。

7. 灵活性

C#在简化语法的同时，并没有失去灵活性。尽管它不是一种无限制的语言，比如它不能用来开发硬件驱动程序，在默认的状态下没有指针等。但是这些都没有影响它的灵活性，在学习过程中你将发现，它仍然是那样的灵巧。

尽管 C#代码的默认状态是类型安全的。但是如果需要，C#仍然允许你将某些类或者某些方法声明为非安全的。进行这样的声明后，你将可以使用指针结构和静态数组了，并且调用这些非安全的代码不会带来任何其他的问题，也不需要更多的条件。

C#的灵活性还表现在多个方面。例如，C#可以通过委托（delegate）机制来模拟指针的功能；C#不支持类的多继承但是通过对接口的继承你将获得这一功能等。

1.2.3　C#与其他语言的比较

1. C#和 C++的比较

从语言上讲，C#是 C++的后续版本，是对 C++的发展，沿用了 C++的思想，面向对象编程等，但又去除和限制了一些相对来说不好的技术，比如指针技术，从另一方面讲 C++和 C 都是国际化的标准，有标准组织维护，并不属于任何一个公司，但是 C#却是微软公司自己在 C++基础上扩充、发展出来的，C#并不是一个国际标准，它只是微软公司的一家之言。但从纯技术角度讲，C#在 C++基础上扩充出来（或限制出来）以后，更注重实际应用，思想上完全面向对象，限制掉了 C++的一些东西，又增加了一些比如内存自动回收等功能。但这样一些功能的扩充，其实是限制了的 C++版本，在 C#上编程，完全没有 C++上那么自由，但这样的改进却在软件开发效率思想上有了很大的提高，所以 C#是为了提升开发效率和软件工业化的产物。相比之下，C#能做到的，C++都能做到，并且更完全，只不过 C++更复杂。C#作为.NET 的核心编程语言是专为.NET 服务的，C#是专门编写.NET 软件的语言。具体地讲，C#与 C++的主要区别有如下几点：

（1）编译目标。C++代码直接编译为本地可执行代码，而 C#默认编译为中间语言（IL）代码，执行时再通过 Just-In-Time 将需要的模块临时编译成本地代码。

（2）内存管理。C++需要显式地删除动态分配给堆的内存，而 C#不需要这么做，C#采用垃圾回收机制自动在合适的时机回收不再使用的内存。

（3）指针。C++中大量地使用指针，而 C#使用对类实例的引用，如果确实想在 C#中使用指针，必须声明该内容是非安全的。不过，一般情况下 C#没有必要使用指针。

（4）字符串处理。在 C#中，字符串是作为一种基本数据类型来对待的，因此比 C++中对字符串的处理要简单得多。

（5）库。C++依赖于以继承和模板为基础的标准库，C#则依赖于.NET 基库。

（6）类继承。C++允许类的多继承，而 C#只允许类的单继承，通过接口才能实现多继承。

2. C#和 Java 的比较

（1）C#与 Java 的相同之处。

① 两者都编译成跨平台的、跨语言的代码，并且代码只能在一个受控制的环境中运行。

② 自动收集垃圾内存，并且取消了指针。在 C#中可以使用指针，不过必须注明 unsafe 关键字。

③ 都不需要头文件，所有的代码都被“包（package）”限制在某个范围内，并且因为没有头文件，所以消除了类定义的循环依赖。

① 所有的类都是从对象派生出来的，并且必须使用 new 关键字分配内存。

② 用对象加锁的方式来支持多线程。

③ 都具有接口的概念。

（2）C#与 Java 的区别。

① C#面向对象的程度比 Java 高。

C#中的基本类型都是面向对象的。例如，当定义一个 int 类型的变量以后，就可以通过这个变量来访问 int 类型的成员。实际上，C#每一个基本类型都内置了相应的类，如 int->Int32。Int32 是系统提供的一个类，基本类型 int 就映射为类 Int32。

在 Java 中找不到这样的内置关系。Java 只是提供了操作这些基本类型的工具类，如 Interger 对应于 int。但是这只是一个简单的操作基本类型的工具，int 在 Java 中并不是面向对象的。

② C#具有比 Java 更强大的功能。

C#拥有 VB 开发的快捷和 C++强大的特点。任何 VB 和 C/C++程序所能做到的都可以简单地用 C#语言实现。也就说在 Windows 平台下，C#足以取代 VB 和 C/C++，而且 C#较 Java 在面向对象的开发上还简单。C#提高了语言面向对象的技术和思想，结合了面向模块和面向对象的技术，使用户使用起来更方便、更快捷。C#还提供了非常强大的兼容性，可以用 C#调用已有的 VB、C/C++、COM 和 VBX 等，几乎是所有微软产品的总接口。

③ C#的速度比 Java 快。

为了跨平台 C#也采用了 JIT 编译器，但不是简单地移植和套用原有的 JIT 技术，而是 JIT 的发展与提升，所以 C#比 Java 的执行速度快。

1.3 Visual C#.NET 集成开发环境

.NET 应用程序的创建通常有两种方法：一种是编辑器＋开发包的方法，另一种是利用.NET 可视化开发工具进行开发。

前者可以使用文本编辑器来编写代码，然后借助从微软站点免费下载的.NET 软件开发工

具包（Software Development Kit，SDK），通过 C#命令行编译器（csc.exe）来构建.NET 程序。但这种方法会带来许多麻烦，因为它在 SDK 中不提供代码生成使用工具（向导）、图形调试器和 IntelliSense 功能，从而导致效率低下。

为了减轻在命令行构建软件的负担，提高开发效率，大多数开发人员都使用后一种方法进行开发。Visual Studio 2005 正是微软公司推出的，现阶段用于创建 C#.NET 应用程序的最简单、快捷的开发工具，使用它可以开发控制台应用程序、Windows 应用程序和 Web 应用程序。下面介绍 Visual Studio 2005 的集成开发环境。

1.3.1　启动界面

每次启动 Microsoft Visual Studio 2005 时，首先进入的是如图 1.2 所示的启动界面。

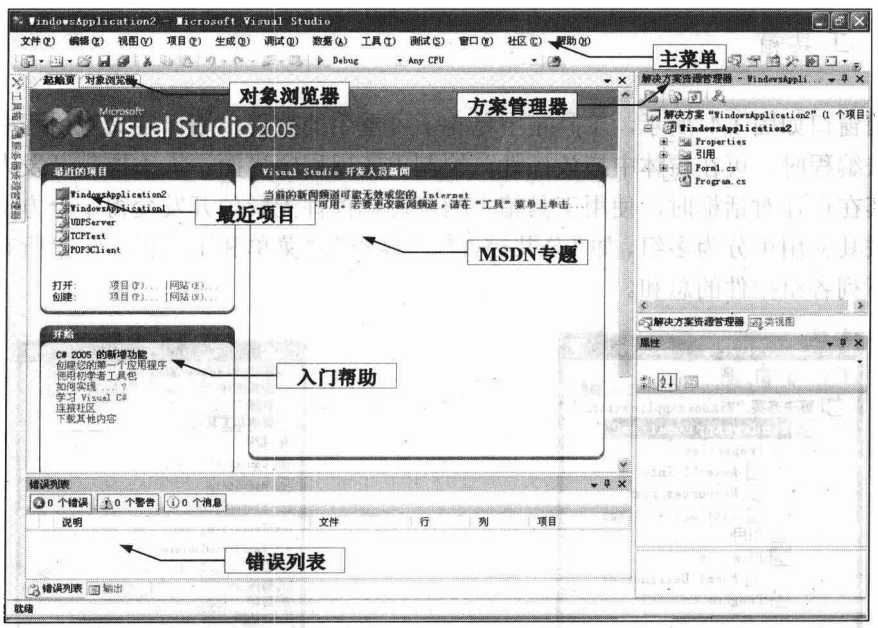


图 1.2　Microsoft Visual Studio 2005 启动界面

按照从上到下，从左到右的顺序来介绍该界面。与大多数应用程序窗口一样，应用程序名称的下方是程序主菜单，接着是常用功能的快捷按钮工具栏。最左边是工具箱窗口和服务器资源管理器窗口。在主窗口中有两个页面，一个是“对象浏览器”，采用明细列表的方式显示了 C#中的类、方法及其介绍；另一个是“起始页”，其中有 3 个小窗口：

（1）最近的项目。显示最近创建并保存的项目，通过双击其中的项目，可以快速打开该项目。在其下方还有两个快捷方式链接，分别对应程序的打开项目和创建项目功能。

（2）开始。可以以在线的方式帮助初学者快速入门。单击其中某项内容条目后，便可打开微软文档浏览器，显示详细的帮助内容。

（3）Visual Studio 开发人员新闻。该窗口中也是提供以在线的方式显示 MSDN 的重要信息。

右侧是 Visual Studio 2005 的解决方案资源管理器、类视图、属性窗口等，界面的下方是错误列表窗口，用于在代码编辑、程序调试中为程序开发人员提供错误信息。在启动界面中的这些子