高等学校计算机教材

C#语言基础教程

张　威　编著

仇　爽　审校

人民邮电出版社

图书在版编目（CIP）数据

C#语言基础教程/张威编著.—北京：人民邮电出版社，2001.7

高等学校计算机教材

ISBN 7-115-09441-1

I.C… II.张… 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（2001）第 039251 号

内 容 简 介

本书重点讲述了 C#语言关键字、语法和程序结构。书中不仅介绍了 C#语言的各种简单数据类型、运算符和运算表达式、常量、变量、数组、程序顺序结构、选择结构以及循环结构等传统的程序基本元素，还讲述了类和面向对象的基本概念、C#语言类成员的使用、继承、接口、代理、编译预处理以及程序调试、代码属性等高级知识。

本书内容丰富、全面、系统，并列举了大量的程序实例和课后习题，不仅适合 C#的初中级读者，还可帮助 C 和 C++的用户顺利过渡到 C#。本书可作为本专科学生学习计算机编程语言的教科书，也可以作为广大编程爱好者学习和提高的参考书。

高等学校计算机教材

C#语言基础教程

◆ 编　　著　张　威
　审　　校　仇　爽
　责任编辑　李振广

◆ 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@ pptph.com.cn
　网址　http://www.pptph.com.cn
　读者热线:010-67129212　010-67129211(传真)
　北京汉魂图文设计有限公司制作
　北京朝阳隆昌印刷厂印刷
　新华书店总店北京发行所经销

◆ 开本:787×1092　1/16
　印张:20
　字数:476 千字　　2001 年 7 月第 1 版
　印数:1－4 000 册　　2001 年 7 月北京第 1 次印刷

ISBN 7-115-09441-1/TP·2325

定价:26.00 元

前　言

C#语言是 Microsoft 公司发布的一种全新的编程语言，作为 Microsoft 公司.NET 战略的一枚重要的棋子，它一经发布就引起了广大编程爱好者和软件工程技术人员的密切关注。

C#语言是一个全新的、面向对象的、现代的编程语言。作为一种编程设计语言，它具有高级的语法结构、优秀的编程开发环境和高效率编译工具。用 C#语言编写的应用程序可以充分利用.NET 框架体系带来的各种优点，完成各种各样高级的功能，例如用来编写基于通用网络协议的 Internet 服务软件，也可以编写 Windows 图形用户界面程序，还可以编写各种数据库、网络服务应用程序。

C#语言不仅可以用来编写各种高级功能的软件，也可以作为一个编程入门语言。C#语言继承了 C/C++语言的全部优点，语法灵活，能够充分利用操作系统的底层功能。同时 C#语言又具备了完全的面向对象特性，可以让编程初学者直接了解面向对象编程的各种概念，做到一步到位地学习现代的编程思想和手段。

本书在编写过程中收集了大量国外最新权威资料，经过数月的研究、揣摩，结合作者深厚的 C 语言技术根底和开发经验，精心组织编写。在无数次的尝试和比较中，我们发现，只有深刻理解和掌握 C#语言的设计理念，及其对于面向对象、代码重用、开发机制的革新，才能把它用好、写好。希望进一步深入学习的读者也可以从这几个方面去努力。

本书由张威编写，王胜虎、赵红东也参加了部分内容的编写和整理。由于时间仓促，书中难免存在一些不妥之处，诚望广大读者原谅，并请提出宝贵意见，以便我们在再版时改正。

编著者

目　录

第 1 章　C#语言基础

本章教学目的:

- 熟悉 C#语言的开发环境
- 掌握 C#源程序的编辑、编译过程
- 掌握 C#应用程序的基本结构

在本章，将简要介绍 C#（发音为 C Sharp）语言，它是一种最适合作为.NET 平台开发的语言。

1.1　C#的由来

C 和 C++语言是被广泛使用的编程语言。尽管这两种语言为程序员们提供了丰富的控制软件功能的方法和灵活的程序结构，但是利用 C/C++语言开发 Windows 应用程序显然复杂了很多，特别是相对于 Microsoft 推出的 Visual Basic 语言来说，利用 C/C++语言开发具有 Windows 图形界面的软件所消耗的时间要长得多。

近年来，随着因特网的发展，越来越多的计算机应用技术是基于网络的，而以前的 C/C++应用程序在网络方面的功能不够强大，毕竟网络的飞速发展仅仅是近十年的事情，而 C/C++语言至少有二十年以上的历史了。

所以，无论是经验丰富的程序员，还是初涉编程语言的自学者都在寻找一种编程语言，希望这种编程语言简单、易用、易学，同时具有强大而丰富的功能。

对于 C/C++程序员，理想的解决方法是，将 C/C++语言能够利用开发平台底层的功能同 Visual Basic 语言能够快速开发应用程序的特性结合起来。这个应用程序开发环境能够将原有的应用程序较好地继承发展，并且可以同步地生成基于因特网标准的应用程序。

对于初学者来说，理想的编程入门语言要像 Basic 语言一样具有较少的关键字，又要像 C 语言一样具有松散简单的程序结构和灵活的编程语法，同时这门编程语言还要具有所有现代编程语言的特性，是面向对象的编程语言。换句话说，他们希望入门语言像 Basic 语言一样简单，又像 C/C++语言一样具有不可限量的能力。这样，他们只需要学习这一种语言就可以了。

鉴于此，IT 行业巨头们纷纷推出了自己的解决方案，例如 Sun 公司推出了 Java 语言，Microsoft 公司推出了 C#—— 一种现代的面向对象的编程语言。

C#语言是由 Microsoft 公司于 2000 年 6 月 26 日对外正式发布的。C#语言从 C/C++语言

继承发展而来，它不仅对原有的 C/C++程序有较好的继承，还是一种比 C/C++语言更加高级的编程语言。C#语言是 Microsoft .NET 战略中的一枚重要的棋子，在.NET 开发环境中，C#语言可以使广大程序员更加容易地建立基于 Microsoft .NET 平台以 XML（扩展标识语言）为基础的因特网服务应用程序。

Microsoft 公司推出的 C#集成开发环境——Visual Studio.NET 提供了类似于 Visual Basic 的开发环境——Visual C#，使得 C/C++语言的优点同 Visual Basic 的优点结合了起来，可以成为初学 Windows 图形界面软件编程者的首选语言。使用 C#语言编程，还可以将其开发的组件转换为以因特网为基础的服务，这样，应用程序就可以作为一种服务在因特网上发布了。

1.2 了解.NET

*C#语言源于 C/C++，作为 Microsoft .NET 的开发工具存在，与 Microsoft.NET 有着不可分割的联系。*所以，学习 C#语言之前应该对 Microsoft .NET 有一定的了解。

.NET 是什么？.NET 是一个用于建立应用程序的平台，它在内部封装了大量的功能强大的应用程序接口函数（API），利用这些函数可以开发各类 Windows 应用软件；.NET 还是一个开发平台，它向广大的程序员提供了功能强大的集成开发环境（IDE）——Visual Studio.NET；在未来，.NET 还是一个运行、发布应用程序的平台，Windows 的下一个版本——Whistler 将集中体现这一特性，它可以将应用程序作为一种服务，通过因特网提供给分布在世界各个角落的网络用户。总之，.NET 是一个用来建立、开发、运行和发布基于因特网的服务和应用程序的平台，.NET 是一门新的技术，同时也是一门令人兴奋的属于因特网的技术。

.NET 的核心是 Microsoft .NET Framework，叫做微软.NET 框架体系。在这个体系中，Microsoft 的软件工程师们将各种开发 Windows 应用程序的应用程序接口（API）封装在了各种“类”中。使用.NET 类库来开发应用程序，就不再需要原来 Visual C++的微软基础类（MFC）了，也不再需要各种复杂的 Windows 32 位 API 函数，我们所要做的就是使用.NET 提供的各种“类库函数”。并且，.NET 还封装了可以直接应用在因特网应用程序开发上的各种类库函数。对于程序开发人员来讲，.NET 框架体系结构就是由若干封装了涵盖 Windows 各个方面应用的类库组成的。

开发.NET Framework 有两方面的目的：一方面是改善 Windows 应用程序的开发过程，特别是组件对象模型（COM）的开发过程。另一方面是为了创建一个将软件作为“服务”来发布的开发平台。在 Microsoft 的最终产品中将集中体现这两个目的。那么，软件开发人员的生产效率将得到巨大的提高，开发应用程序将更加容易，开发的应用程序将更加可靠。特别是，这个最终的产品将向众人展示一个新的计算机概念：因特网服务。所谓因特网服务就是指通过使用标准的 Internet 协议，例如 XML 或者 SOAP，将为不同平台开发的不同的软件和组件有效地结合在一起。

.NET 框架体系主要有两大部分组成，一部分是最基本的通用语言运行时库（Common Language Runtime），另一部分是一些提供了具体功能的类库，例如网络应用的 ASP.NET、数据库应用的 ADO.NET、Windows 窗口（Forms）类等等。它们之间的关系以及它们同 Windows

操作系统之间的关系可以参见图 1-1 所示。

通过图 1-1 可以比较直观地了解.NET 框架体系中 Windows 操作系统、框架体系的各种类库和开发语言之间的关系。首先，所有的.NET 应用是建立在 Windows 操作系统提供的种种强大的功能之上的，这是.NET 框架体系应用程序运行的基础。接下来，是各种.NET 的运行时库和各种类库，其中运行时库是基本系统应用的基础，而基本系统应用又是网络应用、Windows 图形界面、数据库应用、XML 应用的基础，而后面的四种类库互相是独立的。最后，提供给程序开发人员使用的最直接的工具就是各种.NET 开发语言，包括了 C#、C++、Visual Basic、JScript 等。

那么，.NET Framework 给软件开发人员带来的好处是什么呢？简单地说，高效、易开发、易管理、性能更佳。具体而言，包括以下几个主要的方面。

■ 允许使用多种开发语言。在.NET 框架体系中，为各种语言提供了应用程序接口（API），所以无论是在本书中讲述的 C#，还是传统的 C++，以及 Microsoft 自己的 Visual Basic 还是 JScript，都可以作为.NET 的开发语言。这样，熟悉传统语言的开发人员就不必花费很多时间来学习新的语言。

图 1-1　.NET 框架体系

■ 充分利用优秀的开发工具——Visual Studio .NET。Visual Studio 6 是业内认为最优秀的开发环境之一，其最新的版本 Visual Studio .NET 将 Visual C++、Visual C#、Visual Basic 等语言的集成开发环境（IDE）统一在同一个用户界面中了，减少了开发程序的步骤。

■ 书写更少的代码。每一个程序员的梦想就是书写更少的代码来实现更加强大的功能，那么，依靠.NET Framework 的内建机制和 Visual Studio.NET 的强大功能，程序员在编写复杂的应用程序时，就能够将精力集中在自己的程序逻辑上。

■ 充分利用 Windows NT/2000 优秀的应用程序服务功能。Windows NT/2000 具有强大的应用程序服务功能：先进快速的事件处理和事件消息队列机制，快速的数据访问系统，同时也是一个优秀的 Web 服务器。. NET Framework 能够充分利用 Windows 操作系统的优势，例如 COM+的应用。

■ 以 XML/SOAP 作为将软件发布为服务的核心。. NET 最主要的目的是将软件作为因特网上的服务来发布，所以它采用了 XML 和 SOAP 等因特网标准作为网络服务的核心标准。

■ 同以前版本的应用程序相比，具有良好的继承，可以将以前的应用程序统一在.NET 平台中。例如，在.NET 之前开发的 COM 应用可以不做任何修改就能在.NET 环境中使用。

■ 使用 ADO.NET 访问数据将更加容易简便。通过.NET 提供的内部机制，ADO.NET 使用 XML 作为数据的本地化格式，这为用户提供了灵活显示和处理数据的手段。

此外，.NET 框架体系结构在应用程序安全性、可靠性等方面都有很大的改进，这里就不再一一讲述了。

另外，.NET 框架体系结构是一个复杂庞大的体系，在本书中，只讲述与基本编程有关的语法和一部分基本系统应用的类库，而对于网络应用的 ASP.NET、数据库应用的 ADO.NET、XML 应用的 XML.NET、图形用户界面编程的 Windows Forms 等内容请参阅其他书籍，或者 MSDN 文档。

1.3 熟悉开发环境

学习一种编程语言，首先要熟悉这种语言开发出来的应用程序的运行环境，也就是操作系统。使用 C#语言开发的应用程序都是运行在 Microsoft Windows 操作系统中，在本小节，只展示一些与开发 C#程序有关的 Windows 环境特点。如果大家对 Windows 操作系统很熟悉，则可以跳过本小节。

另外，本书中所讲述的内容和实例程序的编辑、编译、运行，都是在 Windows 2000 Professional 版环境中进行的。

1.3.1 长文件名

随着 Windows 各种版本在世界各地的流行，人们已经逐渐淡忘了 DOS 6.22 时代或者 Windows 3.1 时代的文件命名原则——8.3 文件命名。也就是说，当时的文件只能够使用 8 个字符来命名，而文件的扩展名只能够使用 3 个字符。

而在 Windows 9x/NT/2000 以及未来的版本中，这个限制就不再存在了，用户可以使用

任意多的字符构成文件名（最大限制是 255 个字符），例如，一个文件可以命名为 WelcometoBeijing.txt。

长文件名的好处是显而易见的，从文件名上可以非常直观的反映文件的基本信息，比如上面提到的名为 WelcometoBeijing.txt 的文件，通过文件名就可以了解文件的主要内容。在编辑使用文件时，强烈推荐使用长文件名，以便于以后的维护和管理。

不过，有些应用软件是在长文件名出现之前就已经问世了，所以，Windows 9x 操作系统为每个长文件名的文件准备了一个别名。在这些别名中有一个“~”字符。例如，上面的 WelcometoBeijing.txt 文件，在 Windows 9x 操作系统的 MS-DOS 方式窗口中或者叫做控制台方式中，dir 命令则会显示这个文件的别名——Welcom~1.txt。如果还有另外一个文件叫做 WelcometoShenyang.txt，则在上面的操作中显示 Welcom~1.txt 表示第一个文件，而 Welcom~2.txt 表示第二个文件。而它们的长文件名则显示在了窗口的右边，如图 1-2 所示。

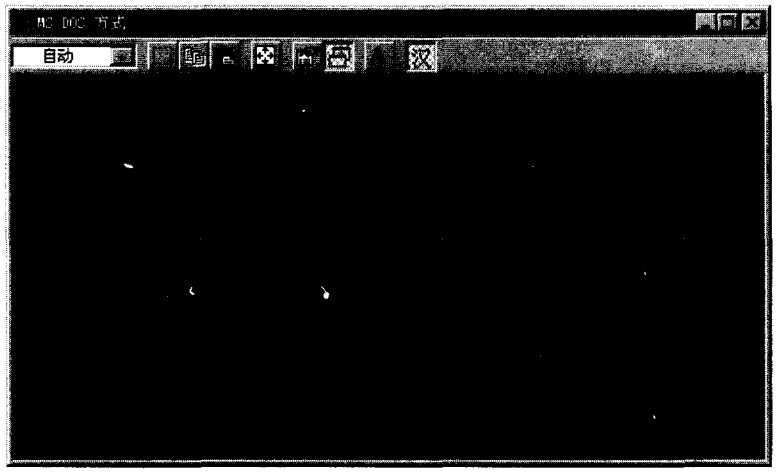

图 1-2　dir 命令显示文件

1.3.2　控制台方式

控制台方式这个名词应该是来自于 Unix/Linux 操作系统。为了用户在 Unix/Linux 的 X Window 环境中还可以使用命令行方式，手动地键入各种 Unix/Linux 命令，例如建立一个子目录或者拷贝一个文件等，X Window 提供了一个类似于字符界面的窗口，叫做控制台。而在 Windows 9x 操作系统中，为了照顾大多数从 MS-DOS 操作系统过渡到 Windows 操作系统的用户，Windows 提供了一个可以运行在 Windows 图形界面中的字符界面——MS-DOS 窗口，叫做 MS-DOS 方式。

然而，无论是在 Visual C++ 或者 Visual C#中，工程项目类型都包含一种叫做 Windows Console Project（Windows 控制台方式工程）的项目，使用这个项目开发的应用程序就是在所谓的 MS-DOS 方式下运行的。特别是随着 Windows 2000，一个真正的 32 位操作系统的发布，MS-DOS 方式就被称为了“命令提示符”，我们认为，把 Windows 2000 中的命令提示符叫做控制台方式显得更加准确。至少，可以将 Visual C++或者 Visual C#开发的应用程序有一个比较统一的称谓。所以在本书中，将会把所有 Windows 版本中的 MS-DOS 方式或

者命令提示符都统一叫做控制台方式。

Windows 2000 中的控制台比前面版本的 MS-DOS 方式先进了很多，主要有以下两个方面的改进。

（1）支持了滚动条，这样可以查看启动控制台之后，所执行的所有命令的运行状况。

（2）增加对于命令行的记忆功能，这样不需要执行 Doskey 之类的应用程序就可以通过上下光标键选择执行过的命令。

一些 Windows 的应用程序也可以从控制台方式中利用命令行的方式运行，比如准备编辑一个文件，可以在控制台方式中直接键入：

```
notepad readme.txt
```

并按回车，Notepad 就会运行，如果在当前的目录中存在 readme.txt 这个文件，则 Notepad 会直接打开这个文件，否则会为用户直接创建一个新的名为 Readme.txt 的文件。

甚至如果在当前的目录中确实存在 Readme.txt 这个文件，则在控制台中直接键入：

```
Readme.txt
```

并按回车，则 Notepad 就会直接运行，并打开这个文件。

进入控制台方式并不一定要通过在开始“菜单”的“附件”子菜单中寻找“命令提示符”命令。我们可以在“开始”菜单中执行“运行”命令，然后，在弹出的对话框中键入 cmd，接着单击“确定”按钮，就可以进入 Windows 2000 的控制台方式了，如图 1-3 所示。

退出 Windows 2000 的控制台方式，可以像关闭 Windows 应用程序一样直接单击窗口右上角的“关闭”按钮，也可以在控制台方式中键入：

```
exit
```

然后回车，就可以退出控制台方式了。

图 1-3 Windows 2000 下的控制台方式

1.4 安装 C#编译器

编程语言按照执行方式可以被分为两类，一类是解释型语言，例如 Basic 语言，利用这

类语言编写的源程序在执行时，由语言的解释器解释源代码，解释一条代码就执行一条代码，直到将所有的源程序代码解释完毕或者中途出现错误为止。

另一类语言是编译语言，利用这类语言编写的源程序必须利用编译器将源代码经过编译转换为目标文件（扩展名为 obj 的文件），然后再通过连接（Link）将目标文件转换成为二进制文件也就是可执行文件（扩展名为.exe 或者.com 等的文件），才可以运行。

编译语言同解释型语言相比将语法查错等工作放在了编译过程中，一旦编译连接结束，则形成可以执行的文件。所以，用编译语言编写的程序执行速度较快。本书中介绍的 C#语言就属于编译语言。

在编译和运行 C#应用程序之前，读者应该在自己的计算机中已经成功地安装了一个 C#编译器。这个编译器可以是由 Microsoft .NET Framework SDK 提供，也可以使用 Visual Studio .NET 提供的编译器。

Framework SDK 可以通过微软的网站获取：http://msdn.microsoft.com/net/

安装 Framework SDK 或者 Visual Studio .NET 是一个相对较为简单的过程，同安装一个普通的 Windows 应用程序相比没有什么特殊之处，只要按照安装向导的指示一步一步照做即可。不过，在安装 Visual Studio.NET 之前，对用户的操作系统有一些要求。

目前，可以在 Windows ME 版本中成功安装 SDK 或者 Visual Studio.NET，如果在 Windows 98 或者 Windows NT 4.0 则分别要安装 Windows 98 SP1 和 Windows NT 4.0 SP6，而在 Windows 2000 操作系统中，则必须安装 Windows 2000 SP1。这些操作系统升级软件包都可以在微软的网站上直接下载。另外，所有的操作系统都必须安装了 Internet Explorer 5.5 或者以上版本。

为了可以正确的编译 C#源程序，安装程序会对 Windows 操作系统的环境变量进行修改，加入 Framework SDK 的相关路径。

1.5　选择一个 C#编辑器

C#源程序和 C/C++源程序一样，是没有任何特殊格式的纯文本文件，所以编辑 C#应用程序十分容易，可以使用 Windows 环境中的任何一个可以编辑纯文本文件的工具，一行一行地敲入程序。只要读者可以在 Windows 环境中编写文档，那么在 Windows 环境中编辑 C#源程序就没有丝毫的困难。而至于纯文本编辑器，可以使用 Windows 自带的记事本，也可以使用 Visual C++ 6 或者 Visual Studio.NET 的集成开发环境（IDE），或者使用一个支持首行缩进等功能的文本编辑器，例如 Ultra Editor 等等。

需要牢记，C#应用程序源程序文件的扩展名是“cs”。文件名称可以根据个人的喜好确定，不过最好是给自己的源文件起一个直观的名称，这样便于以后的维护管理。

无论对于初学者还是 C/C++的老用户，或是 Visual Basic 程序员，都推荐使用 Visual Studio.NET 提供的集成开发环境（IDE）来编辑 C#源程序。主要原因有以下几点。

（1）Visual Studio .NET 的 IDE 支持首行缩进，而且 C#源程序的关键字等可以用不同的颜色直观地表示出来。

（2）利用 Visual Studio 的 IntelliSense 功能，能够动态地查找、提示编辑过程中的错误。