21世纪

高等院校计算机系列教材

C#程序设计实用教程

唐耀 主编

中国水利水电出版社
www.waterpub.com.cn

21世纪高等院校计算机系列教材

C#程序设计实用教程

唐　耀　主编

中国水利水电出版社

内 容 提 要

本书是一本讲解微软的 C#的专题书籍。主要内容包括 C#基本特征、基础语法、面向对象技术、结构化异常处理、可视化程序设计、GDI+图形编程、基于流的 IO 操作、多线程编程和数据库应用等。由于 C#语言是专门为.NET 框架设计的新语言，所以，本书通篇紧紧结合.NET 平台进行讲解。为突出面向对象编程技术和.NET 框架类库应用这两大主线，全书精心编排了大量相关的实用例程，供读者学习参考，同时，每一章还有针对性地提供了思考练习题。

本书内容翔实、结构清晰、实用性强。初学者可以很容易地入门并逐渐掌握 C#程序开发的要旨，中级读者也可以快速地从书中获得不少有价值的参考信息。本书适合作为高等学校讲述 C#语言的教材以及作为初中级人员的自学参考工具书。

图书在版编目（CIP）数据

C#程序设计实用教程 / 唐耀主编. —北京：中国水利水电出版社，2005
（21 世纪高等院校计算机系列教材）
ISBN 7-5084-2426-3

Ⅰ. C… Ⅱ. 唐… 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（2004）第 108296 号

书　　名	C#程序设计实用教程
作　　者	唐　耀　主编
出版　发行	中国水利水电出版社（北京市三里河路 6 号　100044） 网址：www.waterpub.com.cn E-mail：mchannel@263.net（万水） sales@waterpub.com.cn 电话：（010）63202266（总机）　68331835（营销中心）　82562819（万水）
经　　售	全国各地新华书店和相关出版物销售网点
排　　版	北京万水电子信息有限公司
印　　刷	北京市天竺颖华印刷厂
规　　格	787mm×1092mm　16 开本　22.25 印张　495 千字
版　　次	2005 年 1 月第 1 版　2005 年 1 月第 1 次印刷
印　　数	0001—5000 册
定　　价	32.00 元

前　言

随着计算机应用的不断深入和扩展，计算机技术也在急速发展。为了适应开发各种不同用途软件的需要，在过去的几十年中，人们已经构建了1000多种程序设计语言，程序设计思想、软件开发工具都发生了巨大的变革，相关的参考图书也令我们目不暇接，当然，也包括了“疲惫”。今天，有什么能使我们迅速提高自身，从容应付未来的编程挑战？如果你有此想法，那么，建议使用微软的Visual C#.NET。

C#是微软公司在2000年7月发布的一种全新的简单、安全、面向对象的程序设计语言，它充分吸收了过去几十年计算机科学发展的经验教训，从C#的身上可以看到很多C++、Visual Basic、Delphi、Java等语言的优点，其目标是将Visual Basic的高效率同C++原有的强大功能充分地结合在一起，为未来几十年的程序设计提供一个优良的利器。可以预见，C#在对语言作出了革命性的更新后，依托.NET框架的支撑，必将成为未来几十年中应用程序开发的首选工具。为了对广大C#学习者提供一本有价值的、实用的书籍，我们编写了本书。

本书是一本关于C#的专题书籍，为了充分地展现C#的新特点和强大功能，以满足不同层次读者的学习需要，我们在内容设计方面力争做到强化基础、突出重点、注重应用；在文字编排方面力求语言精练、循序渐进，以保证书籍达到较高的质量。

全书分为三个部分：基础篇、提高篇和应用篇。

基础篇：第1~3章。主要讲述了C#的基本编程技术和基础语言规范。

提高篇：第4~7章。本篇的重点是面向对象技术，是初学者真正进入C#殿堂的必经之路。因此，对于类的封装、继承、多态进行了详细的讲解，并对接口、委托、事件等重要概念进行了专门阐述。最后，介绍了C#编程中的异常处理技巧。

应用篇：第8~12章。在熟悉了面向对象编程的基础上，紧密结合.NET框架，采取每章一个主题方向介绍了Windows桌面开发、GDI+编程、流IO、多线程和数据库开发的实用编程知识，并通过大量的例程引导读者进入编程的较高境界。

本书由唐耀、刘汉明、高国兴、李岐旭、侯玉芳、李成龙、杜斌、范士云编写，丁宁统稿。由于作者水平有限，加之时间仓促，疏忽与遗漏之处在所难免，敬请广大读者谅解和批评指正。

作者

2004年10月

目　　录

提高篇

应用篇

基础篇

本篇介绍了 C#的新特点、集成开发环境、项目管理、一般程序结构与开发步骤和C#基础语法等，为初学者提供了一个快速进入C#编程的入口。主要包括以下知识点：

- C#新特点
- 集成开发环境
- 项目组织与管理
- C#程序开发与调试初步
- 基本数据类型
- 枚举与数组
- 变量与表达式
- 流程控制

第 1 章　Visual C#.NET 简介

知识点：

- .NET 框架
- 程序结构
- 开发环境
- 程序调试

本章导读：

本章主要介绍了微软 Visual C#.NET 的渊源及特点、.NET 框架的结构、Visual Studio .NET 集成开发环境、简单的程序开发、编译和调试等内容。通过本章的学习，读者可以对 Visual C#.NET 程序设计获得初步认识，为后续的学习打开方便之门。

1.1　C#简述

1.1.1　什么是 Visual C#.NET

过去的几十年计算机语言一直在加速发展。流行的说法是，20 世纪 80 年代看 C 语言，90 年代则是属于 C++的黄金时代。那么，21 世纪的理想语言是什么？微软为推行其.NET 战略，已经在各种场合和媒体上大声地宣扬，C#为本世纪前 20 年的程序开发作好了准备。就笔者的观点，C#语言的确是目前各类应用开发的优良利器。

C#是微软公司在 2000 年 7 月发布的一种全新的简单、安全、面向对象的程序设计语言，它充分吸收了过去几十年中计算机科学发展的经验教训，体现了当前最新的程序设计技术的功能和精华，从 C#的身上可以看到很多 C++、Visual Basic、Delphi、Java 等语言的优点。而且，C#于 2001 年 12 月获得了国际标准组织 ECMA 批准的 C#标准 ECMA-334，并于 2003 年 4 月被国际标准化组织 ISO 采纳为 ISO/IEC 23270 标准，这一切都为 C#的发展推广奠定了坚实的基础。

通常，我们对于 C#和 Visual C#.NET 可以不加区分，但严格地说，两者是有区别的。C#只是一门语言或者说是一个标准，它是专门为微软的.NET 平台设计的。作为 Visual Studio .NET 套件中的语言之一（该套件还包括 Visual Basic .NET、Visual C++ .NET 和 J#.NET 语言），充当了微软推行.NET 战略的拳头产品。但是，难保今后不会出现其他使用 C#语言的开发工具（就像有 Visual C++和 C++ Builder 一样）。Visual C#.NET 则是指“C#语言+.NET 框架”。由于目前 C#必须和.NET 平台一同使用，因此，C#语言的学习与以往的语言学习有显著的不同：对以往的编程工具可以先学习语言语法，再学习编程环境（例如，先学 C++再使用 Visual C++）；而对 C#的学习一开始就要进入到 Visual Studio .NET 中

进行编程学习。

1.1.2　Visual C#.NET 的特点

Visual C#.NET 以 Visual Studio .NET 为开发工具，包括交互式开发环境、可视化设计器（用于生成 Windows 和 Web 应用程序）、编译器和调试器。Visual C#.NET 的特征主要体现在两个方面。

1. 语言的变化

C#是在 C、C++的基础上改进而来，作为一种全新的语言，它继承了 C、C++的强大功能，同时，吸收了 Visual Basic 语言的简单易用特点。虽然从整体上来说，它基本继承了 C 语言的语法风格，但还是有明显的区别和改进，具体的语言变化细节将在书中的各处体现。

2. .NET 框架支持

Visual C#.NET 完全集成了.NET 框架。.NET 框架封装了传统的 Windows API，为用户提供了全新的编程接口，并吸收了微软 20 世纪 90 年代中后期发展的各种新技术（COM+组件、ASP 技术、XML 支持等），为程序提供了对语言互操作性、垃圾回收、增强的安全性和改进的支持。

其中，.NET 框架的支持是 C#运行和功能实现的基础，离开了.NET 框架，C#就是无源之水、无本之木。因此，对于 C#的学习，大部分精力要放在对.NET 框架的熟练掌握和应用方面。

1.1.3　C#与其他语言的关系

1. 与 C、C++的关系

就像名字上的相似性一样，C#是从 C、C++语言演变改进而来的，存在着血缘关系。C#基本上继承了 C 语言的语法风格，同时，又从 C++那里继承了面向对象特性。但是，不能够简单地将 C#看成 C++在.NET 框架上的翻版。毕竟，它们之间的不同点也是很明显的。主要表现有：第一，C#的对象模型已经面向 Internet 进行了重新的设计，使用的是.NET 框架的类库，与 C++的对象模型结构完全不一样。因此，在编程中没有太大的相识感。第二，C#语言不再提供对指针类型的支持，使得程序不能随便访问内存地址空间，从而更加健壮。用惯了指针的程序员需要一定的适应过程。第三，在面向对象技术方面，C#不再支持多重继承，避免了以往类层次结构中由于多重继承带来的可怕后果。相应的功能可以通过对接口的多重继承来实现。

由于.NET 框架的支持，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。同时，公共语言运行时（Common Language Runtime）为 C#程序提供了一个托管的运行时环境，使程序比以往更加稳定、安全。C#是被设计用来满足新的在线环境功能的开发需求，解决新环境中的程序设计问题，而不是用来替代 C++，二者在未来的一段时期将共处。

2. 与 Java 相比较

从整体上来看，C#与 Java 极其相似，甚至超过了 C#与 C、C++的相似程度，不过，两者还是有区别的。例如，Java 通过 Java 虚拟机来实现平台的可移植性，而 C#则是首先被编译成一种中间语言（类似 Java 的字节码），然后，在执行时由公共语言运行时中的即时编译器编译成本机代码交由 CPU 处理。而且，Java 虚拟机只能执行 Java 程序，而即时编译器能够编译任何.NET 框架支持的语言（如 C#、Visual Basic、J#）编写的程序。

鉴于 C#与 Java 的相似性，学习过 Java 的人员对 C#掌握起来不会感到太难，反过来也一样。由于二者支持平台（或者说运行环境）的不同，相互之间不会取代，而会长期共存下去。

1.2 .NET 框架

前面已经多次提到“.NET 框架”一词，本节就先解释其具体含义，以满足更好地理解和学习 C#的需要。

.NET 框架是一种全新的用来开发应用程序的计算平台，它简化了在高度分布式互联网环境中的应用程序开发，为开发人员提供了面向对象的编程环境以及安全、可靠、高效的代码执行环境。它主要包含两个重要组件：公共语言运行时 CLR（Common Language Runtime）和.NET 框架类库。

1. 公共语言运行时

“运行时”并非什么新东西，例如，Visual Basic 有某种形式的运行时，Visual C++也有一个叫 MCVCRT.DLL 的运行时，其他语言（Perl、SmallTalk 等）也都使用运行时的概念。.NET 框架的公共语言运行时是 .NET 框架的基础，主要负责组件管理、内存分配、启动和停止线程与进程，以及强制执行安全策略等，可以简单看作一个在执行时管理代码的代理。它与那些旧运行时的区别在于，CLR 被设计用来支持和服务于多种语言。以运行时为目标的代码称为托管代码，而不以运行时为目标的代码称为非托管代码。

2. .NET 框架类库

.NET 框架类库是一个综合性的面向对象的可重用类型集合，是对 Windows API 封装的全新设计，为开发人员提供了一个统一的、面向对象的、分层的和可扩展的庞大类库。

以前，C++开发人员使用 MFC 基础类库，而 Java 开发人员使用 Windows 基础类库，现在，.NET 框架一统这些完全不同的模型库，使得从 Visual Basic、JScript、C++到 C#的所有编程语言具有对框架的相似访问，开发人员只需选择使用自己熟悉的语言。并且，通过创建跨所有编程语言的公共 API 集，公共语言运行库使得跨语言继承、错误处理和调试成为可能。图 1-1 显示了.NET 框架的构造模型。

从图 1-1 可以看出，公共语言运行时 CLR 处于.NET 框架的最底层，这是.NET 框架最重要的软件层，实现了对操作系统的封装，提供了程序的执行环境。在其上是.NET 框架基础类库，类库中的类支持基本的输入输出、字符串操作、安全管理、网络通信、线程管理及文本操作等一些通用的任务。再往上的数据与 XML、Web 服务、Windows 应用、Web

应用类库构成了有史以来最大的类库，并且还将继续发展。这些类库目前总共提供了大约5000 多个类，总称为 FCL（Frame Class Library）。和 Microsoft 的早期框架相比，.NET 是彻底面向对象的，提供了更丰富更广泛的可用组件（都以类的形式提供）。程序员不仅可以在应用里使用这些类，而且可以通过继承、重载和根据自己的需要扩展框架类。

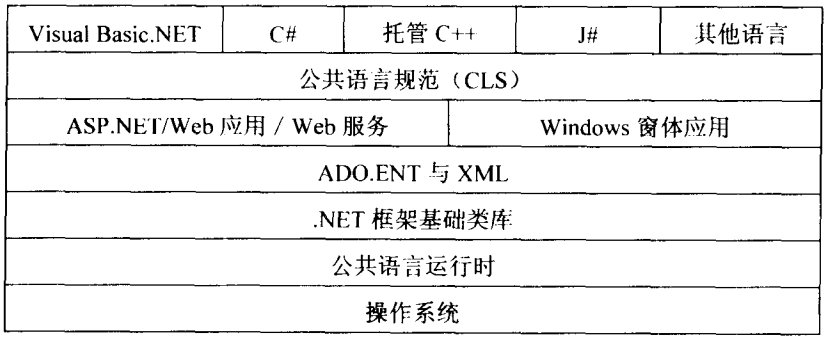

图 1-1　.NET 框架结构

.NET 框架是与语言无关的。这通过所有的语言都要遵循公共语言规范（Common Language Specification，CLS）来实现。CLS 提供了集成语言必需的一些规则，符合 CLS 的编译器都能够生成彼此相互操作的对象，目前 Visual Studio.NET 提供了四种正式的语言：Visual Basic.NET、Visual C#.NET、托管 C++.NET 和 J#.NET。

按照微软的设计目标，框架将可以提供操作系统独立性。虽然目前还只是运行在 Windows 系统上（如桌面 Windows 上的.NET 框架和嵌入式 Windows CE 上的.NET 框架精简版），将来会推出其他操作系统（如 Linux，Macintosh 和像 Palm Pilots 这样的便携式设备）上的.NET 版本。这样，在理论上将有可能使同一个程序运行在支持.NET 的所有设备上，实现超级跨平台能力。

1.3　MIL 中间语言

在以往的程序开发工具中，对于项目源代码进行编译后，最终得到的.exe 文件都是为硬件所识别的机器代码，这限制了程序的可移植性。因为对于不同的计算机，或者说不同类型的 CPU，其指令系统是不同的。这种硬件系统的差异导致不同的计算机上使用的计算机语言不同。理论上说，任何类型的 CPU 都应该可以编译 C++程序，这需要对不同类型的 CPU 都开发不同版本的 C++编译器。然而，编译器的开发不但非常昂贵，而且非常复杂和耗时。因此，在 Windows 平台上编译的程序要想在 Linux 系统上运行是不行的，除非使用 Linux 系统上的 C 编译器重新编译。至于 X86 架构程序要移植到 MIPS 系统就更困难了。

现在，.NET 框架提供了一个可移植的交叉语言平台，使不同语言开发的程序可以在不同环境下的不同 CPU 上执行。.NET 框架的这种跨平台程序移植能力是如何实现的呢？

Java 将程序编译为字节码，并使用一个名叫 Java 虚拟机的运行时来解释执行字节码程序。因此，在不同的平台上只要运行 Java 虚拟机就可以运行 Java 程序。.NET 框架与此类

似，用.NET 框架支持的语言编写的源程序被编译后，生成的.exe 文件不能够被计算机直接识别执行，这样的.exe 文件称为 Microsoft 中间语言文件（Microsoft Intermediate Language，MIL），在程序执行时，MIL 文件会用 CLR 中带有的即时编译器（Just In Time，JIT）再次编译，生成机器代码，由计算机的处理器执行。如图 1-2 所示。

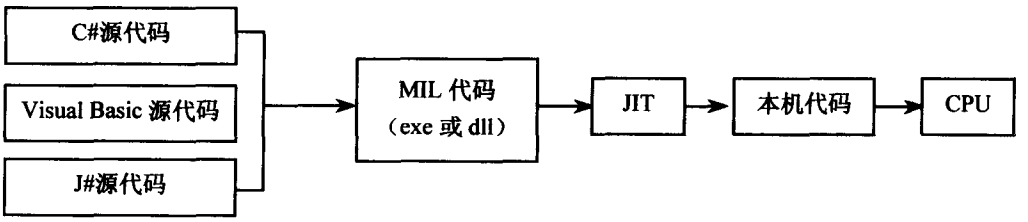

图 1-2 编译执行过程

C#生成的中间语言文件 MIL 与其他.NET 语言生成的 MIL 文件完全相同，.NET 平台不会区分语言，CLR 是公用的，在运行期间既支持 C#开发，又支持其他语言（例如 C#.NET）的开发。这样说来，只要是使用 Visual Studio .NET（不管使用哪种语言）开发的程序，都可以在安装有.NET 框架的平台上运行。

与 Java 虚拟机的运行时解释执行不同的是，.NET 框架的即时编译器 JIT 是按照需要执行的。在程序执行时，JIT 会分析代码并产生高效的机器代码，运行速度极快；而且，JIT 还能够智能地识别代码是否已经编译过。因此，程序只在需要的时候进行编译，保证程序运行的高速度。另外，Java 虚拟机局限于 Java 语言程序，而.NET 框架中的 JIT 是语言交叉的。

1.4 Visual Studio .NET 开发环境

开发 C#程序有两种方式：一种是使用任何的文本编辑器编写程序代码，然后以.cs 保存源文件，并使用命令行编译器（csc.exe）进行编译；另一种方式就是使用 Visual Studio .NET 提供的集成开发环境 IDE 进行开发。Visual Studio .NET 的 IDE 集成开发环境为开发人员提供了各类功能强大的工具、调试器和帮助系统，包括自动缩进、智能单词完成、代码着色等便利功能，所以，使用 IDE 进行项目开发当然是程序人员的首选。

1.4.1 默认开发环境

当在系统中安装了 Visual Studio .NET（安装中包括了 C#语言），就可以在操作系统的“开始”→“程序”菜单中选择启动 Visual Studio .NET。初次打开 Visual Studio 时，屏幕以默认样式开始，可以看到，整个窗口主要包含 9 个区域：标题栏、菜单栏、工具栏、服务器资源管理器、工具箱、解决方案资源管理器、帮助窗口、状态栏和中间的开始页。如图 1-3 所示。

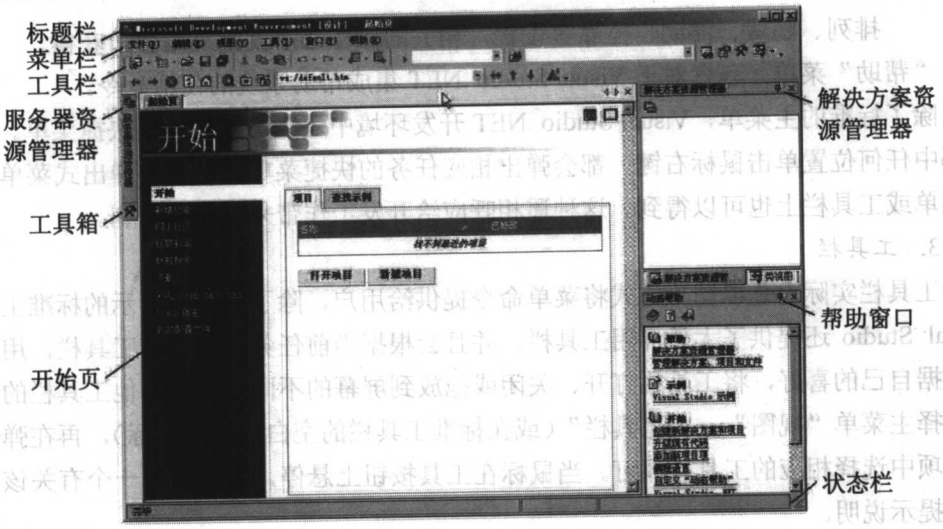

图 1-3　初始窗体

下面，分别介绍其中的各个区域，以帮助读者快速地熟悉 Visual Studio .NET 中各种工具的使用。

1. 标题栏

标题栏中主要提供了两个信息：

一是在[ ]符号中显示了 Visual Studio .NET 的运行状态。有三种状态：设计时、运行时和中断时。

- 设计时：进行界面设计和代码编写时。
- 运行时：运行程序时，不能够进行界面设计和代码编辑。
- 中断时：暂时中断程序运行，不能够进行界面和代码修改，按 F5 或 F11 键继续程序运行。

二是显示了当前方案的名称，并在“-”符号后面显示了当前编辑的源文件的名称。

2. 菜单栏

刚启动时，Visual Studio .NET 含有 6 个主菜单项：文件、编辑、视图、工具、窗口、帮助。随着当前任务的变化，主菜单项以及包含的子菜单项也会自动变化。

- “文件”菜单：主要完成方案项目和源文件的管理。
- “编辑”菜单：包含一些标准的 Windows 编辑操作命令。
- “视图”菜单：Visual Studio .NET 中包含了大量的任务窗口，如果同时显示在屏幕上，就无法有效地工作了。本菜单项下面主要包含对于需要的窗口和工具的开关。
- “工具”菜单：主要完成对于集成环境设置的自定义，还可以连接数据库和外接程序。
- “窗口”菜单：为了方便程序员的使用和节省有效屏幕空间，各种窗口在屏幕上的布置非常灵活，除了能根据当前任务自动调整外，还能够任由用户进行层叠、