SHEN RU QIAN CHU

许春杰 等 编著

深入浅出 C#

人民邮电出版社
www.pptph.com.cn

附光盘
CD-ROM

深入浅出 C#

许春杰 等 编著

人民邮电出版社

图书在版编目（CIP）数据

深入浅出 C#／许春杰编著．－北京：人民邮电出版社，2001，9
ISBN 7-115-09632-5

Ⅰ．深… Ⅱ．许… Ⅲ．C 语言－程序设计 Ⅳ．TP312

中国版本图书馆 CIP 数据核字（2001）第 058444 号

内容提要

本书从最基本的词法、语法开始讲解 C#语言，包括 C#程序的编译和调试、编译预处理、C#的数据类型、控制语句、名字空间、数组和枚举、类、接口、属性、代表和异常处理。另外，本书还涉及了一些 C#的高级概念，包括编写数据库程序、编写和调用 DLL 以及和 COM 的交互等。本书的最后讲述了如何用 C#编写 Windows 程序。

本书是 C#的入门书，适用于有一定 C++基础的程序员阅读。

深入浅出 C#

◆ 编　　著　许春杰　等
责任编辑　张立科

◆ 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
北京鸿佳印刷厂印刷
新华书店总店北京发行所经销

◆ 开本:787×1092　1/16
印张:19
字数:456 千字　　2001 年 9 月第 1 版
印数:1－6000 册　　2001 年 9 月北京第 1 次印刷

ISBN 7-115-09632-5/TP·2451

定价:34.00 元(附光盘)

前　言

在过去的 20 年里，C/C++语言已经广泛应用在商业应用软件的开发中。但是 C/C++的灵活性是牺牲了其开发效率的。如果和其他的开发语言相比（比如说 VB），相同功能的 C/C++软件通常会需要更长的开发周期。正是由于 C/C++开发的复杂性和需要较长的开发周期，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。

目前有一些开发语言通过牺牲语言的灵活性（一些必要的灵活性）来换取开发效率，而有些语言则对开发人员有过多的限制（比如说限制使用底层控制代码）。这些语言不能够轻易地与现存的系统相结合，并且也不能够与当前的 Web 开发相结合。

一种合理的 C/C++替代语言应该是能够为在现在和将来的平台上的高效开发提供有力的支持，并使 Web 开发可以非常方便地与现存的应用开发相结合，同时，C/C++开发人员都倾向于在必要时使用底层代码。

在这个问题上微软公司的解决方案是推出一种命名为 C#（读作 C Sharp）的开发语言。C#是一种先进的、面向对象的语言，通过 C#可以让开发人员快速地建立大范围的基于微软.NET 平台的应用，并且提供大量的开发工具和服务帮助开发人员开发基于计算和通信的各种应用。

从开发语言的角度来讲，C#可以更好地帮助开发人员避免错误，提高工作效率，而且同时具有 C/C++的强大功能。

由于 C#是一种面向对象的开发语言，因此 C#可以大范围地适用于高层商业应用和底层系统的开发。

和 C/C++相比，C#会帮助开发者通过更少的代码完成相同的功能，并且能够更好地避免错误发生；C#的垃圾收集机制将减轻开发人员对内存的管理负担；C#中的变量将会自动根据环境进行初始化，变量类型是安全的，并且通过提供内置的版本支持来减少开发费用。

C#从 C/C++中派生而来，与 C/C++有着非常密切的联系，所以 C/C++程序人员能够非常容易地掌握它。它的目标是结合 VB 的高效和 C++的强大功能。C#作为 MSDEV 7.0 的一部分提供，除了 C#，MSDEV 7.0，还支持 VB、VC、VBScript 和 JScript，所有这些语言一起提供下一代 Windows 服务。

本书的定位是作为 C#的入门读物，主要讲述 C#的基本概念，所以书中不涉及大量的关于微软的.NET 框架方面的内容，而把注意力集中于 C#语言本身上，把它作为一种全新的语言，从最基本的词法、语法开始讲解，前 12 章主要讲述了 C#的基本概念，第 13 章讲述了 C#的一些高级概念，包括编写和调用 DLL 以及和 COM 的交互等，第 14 章讲述了如何用 C#编写 Windows 程序。

由于作者水平有限，书中如有不妥之处，恳请广大读者批评指正。

编者

2001 年 9 月

目　录

第 1 章　.NET 和 C#简介

本章首先简单介绍了.NET 平台，然后讲述如何建立一个用 C#语言开发.NET 程序的开发环境。

1.1　.NET 和 C#

C#包含在.NET 平台中，学习 C#前应该先了解一下.NET，实际上很难用一句话把.NET 说清楚。微软公司是这样来描述.NET 的：“.NET 是一个革命性的新平台，它建立在开放的 Internet 协议和标准之上，采用许多新的工具和服务用于计算和通信。”这就是说，.NET 是一个全新的软件平台。

简单地说，.NET 就是一个开发和运行软件的新环境。只不过这个环境提供了许多基于 Web 的服务，更加易于使用，使得多种语言之间，以及网络上计算机之间的基于组件的交互访问更加方便。.NET 的核心是一大套构件库，这套库既像 VCL（Delphi 和 C++Builder 的控件库），也像 Java Class 和 COM（组件对象模型）。

.NET 的最终目的就是让用户在任何地方、任何时间以及利用任何设备都能访问他们所需要的信息、文件和程序。而用户不需要知道这些东西保存在什么地方，甚至连如何获得等具体细节都不需要知道。他们只需发出请求，然后只管接收就是了，所有后台的复杂性操作都被完全屏蔽了。

.NET 作为第三代 Internet 思想，它与 Java 有着极大的不同，用户数据生存于网络，而不只是生存于不同平台，.NET 允许不同应用程序之间能互相传递信息，而 Java Beans 是不能和其他语言建立的构件共享数据的。.NET 是一个语言无关平台，它是基于 XML（Extensible Markup Language）和 SOAP（Simple Object Access Protocol，简单对象访问协议，用于 Internet 上客户端对服务器端 COM 对象的访问），允许开发者利用各自不同的语言来使用它所有的功能。可以肯定，在网络飞速发展的今天，.NET 将很快成为主流。

C#是一种以.NET 为基础的语言。.NET 平台提供了一个运行 C#语言的环境，叫做 Common Language Runtime 或者称为 CLR，CLR 和 Java 的虚拟机十分相似。CLR 管理着代码的执行，并且提供了跨语言集成、跨语言异常处理和良好的安全性等服务。

为了使 CLR 有这些功能，编译器就要把源程序编译成一个中间的语言代码，简称 IL（Intermediate Language）。

由 C#或其他能产生受管代码的编译器所生成的受管代码就是 IL 码。

.NET 环境提供了许多核心的运行时服务，比如异常处理和安全策略。为了使运行代码能够使用这些服务，代码必须要给运行时的环境提供一些信息，这种代码就是受管代码。所有的 C#、VB.NET、JScript.NET 默认都是受管的。

✍ 注意：Visual C++ 7.0 不是受管的，但是编译器能够使用命令行选项（/com+）

> 注意：Visual C++ 7.0 不是受管的，但是编译器能够使用命令行选项（/com+）产生受管代码。

由 C#编译器生成的受管代码（Managed Code）并不是原始的可执行代码，它是中间语言（IL）代码。IL 代码明显的优势在于它的 CPU 无关性，目标计算机上要有一个即时编译器才能把 IL 代码转换成原始的可执行代码。

尽管 IL 代码由编译器产生，但它并不是编译器提供给运行时仅有的东西。编译器同时产生有关代码的元数据（metadata），它包含了比代码更多的东西，例如，各种类型的定义、各种类型成员及其他数据。元数据基本上是关于类型库、注册表内容和其他用于 COM 的信息。尽管如此，元数据还是直接和执行代码合并在一起，并不处在隔离的位置。

IL 和元数据存放于扩展了 PE（Portable Executable）格式的文件（.exe 或者.dll 文件）中。当这样的一个 PE 文件被装载时，运行时从文件中定位和分离出元数据和 IL。

虽然 IL 代码被包装在一个有效的 PE 文件中，但是这些代码并不能被执行，除非它被转换成为可执行代码，那就需要一个即时编译器（JIT）。

C#代码的完整编译过程如图 1-1 所示。从图 1-1 可以看出，C#的源文件通过 C#编译器生成了 PE 文件（.exe 或者.dll 文件），但这个是扩展了的 PE 文件，里面还包含了 IL 数据和元数据（metadata）。运行的时候通过即时编译器（Just In Time Complier）编译成可执行的代码。

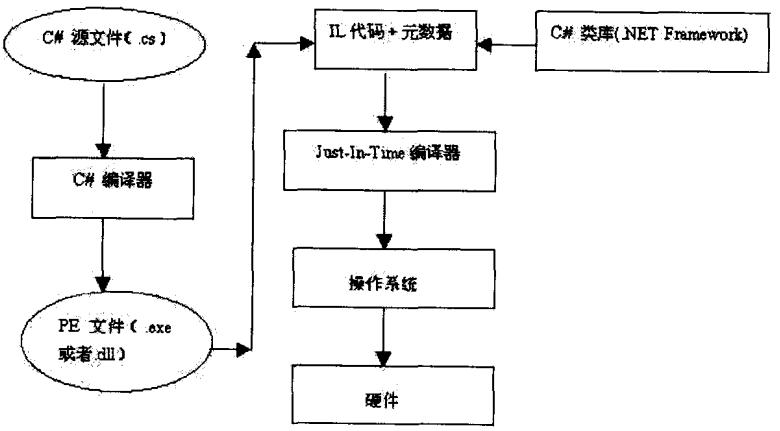

图 1-1 C#源程序的编译过程

C#编译器为什么不把整个 IL（中间语言）的 PE 文件编译成可执行的代码呢？答案是 IL 代码编译成 CPU 规格的代码需要大量的时间。即时编译的效率更高，因为一些程序段从来就没有被执行过。C#编译器把源程序编译成 IL（中间语言）代码后，在运行的时候 JIT 会把它编译成执行代码。

C#是一种简单易学面向对象的编程语言，它是由 C/C++派生而来的。C#（读作 C－Sharp）作为 Microsoft 的下一代面向对象的 C 语言产品能让开发人员在.NET 平台上快速建立大量的应用程序，C#为 C/C++程序员提供了开发飞速发展的 Web 应用程序所需的强大而灵活的功能。C#和 Java 的核心与 C++相比有着相同的优势和局限，但 Microsoft 声称 C#并非为了和 Java 竞争，而是想让它成为 C++的发展，比起 C++，C#将更容易被理解，将来大量.NET 平台的应用将由 C#开发。

开发.NET 程序需要有.NET SDK（就是所谓的 NGWS Runtime），这个开发包里有命令行

编译器 csc，用来编译 C#源程序；或者可以使用 Microsoft Visual Studio .NET 7.0（以下简称 Visual Studio 7.0），它提供了对.NET 应用程序开发的全面支持。

C#是 Visual Studio 7.0 中的一个组成部分。除了 C#，Visual Studio 7.0 中还包括 Visual Basic、Visual C++、VBScript 等。所有这些语言都提供了面向下一代窗口服务平台（NWGS）的编程方法，它们还提供了一个普通的执行引擎和一个丰富的类库。

C#是从 C/C++发展起来的，所以熟悉 C/C++语言的人会很快地学好这种语言。C#与 C++语言相比，语法基本上是一样的，只不过 C#中没有指针，并且编程人员可以直接编写 OOP（面向对象编程）的代码，而在 C++中通常写的都是 C 的代码。C#语言结合了 VB 的灵活性和 C 语言的强大的功能，是新一代的编程语言。

像编写 C 程序一样，可以使用任何文本编辑器编写 C#程序，但是要调试和编译 C#程序，需要有 Microsoft 的.NET SDK 的支持。如果要单纯地学习 C#语言，那么用.NET SDK 所提供的命令行编译程序和一个 SDK 调试程序就足够了。但是如果要开发 Windows 程序，那么就需要使用 Visual Studio 7.0 这个强大的集成开发环境来调试和编译程序。

1.2 .NET SDK 的安装

如果不想安装 Visual Studio 7.0，那么可以使用.NET SDK 来调试 C#程序，.NET SDK 带一个命令行编译器和两个调试工具：The SDK debugger 和 The IL Disassembler。

微软公司提供了.NET SDK 的下载和免费安装，感兴趣的读者可以到微软公司的网站 http://download.microsoft.com/download/VisualStudioNET/Install/2204/NT5/EN-US/setup.exe 上下载安装软件，软件大小为 108MB，可以运行在 Windows 2000、Windows NT 4.0、Windows 98 和 Windows Me 操作系统上。

安装.NET SDK 前需预先安装 IE 5.5 或以上版本的浏览器和 Data Access Object 2.6，这两个软件可以在微软公司的网站 http://www.microsoft.com/windows/ie/download/ie55.htm 和 http://www.microsoft.com/data/download.htm 上下载。

1.3 Visual Studio 7.0 的安装

与.NET SDK 自带的命令行调试工具相比，Visual Studio 7.0 的集成开发环境的功能要强大得多，本节将讲述如何安装 Visual Studio 7.0。

1.3.1 Visual Studio 7.0 系统要求

安装 Visual Studio 7.0 系统的计算机配置要求如下。

处理器：至少 Pentium II 450 MHz（推荐：Pentium III 733 MHz 或以上）。

内存：至少 128 MB（推荐: 256 MB）。

可用磁盘空间：至少 3 GB 。

显示器： 800 ×600，256 色（推荐：16 位彩色）。

操作系统：微软公司的 Windows 2000、 Windows NT 4.0、 Windows Me 或 Windows 98。

1.3.2 Visual Studio 7.0 安装步骤

首先安装 Visual Studio 7.0 支持环境，安装方法如下。

（1）先确定系统已经安装了中文版 Windows 2000 SP1 或中文版 Windows 98 Second Edition SP1。

（2）安装 IE 5.5。

（3）插入安装盘，安装/IEQFE 目录下的 Internet Explorer QFE，文件名为 q11258.exe。

（4）执行/jet/jetsetup.exe，安装 Jet Engine。

（5）执行/MDAC/mdac_type.exe，安装 MDAC2.6。

（6）执行/mso9.msi，安装 MSO9。

（7）执行/msxml3.msi，安装 MS XML 3.0。

（8）重新启动计算机。

执行/FrameworkSDK/setup.exe 文件，安装.NET Framework SDK；执行/bootstrap.msi 文件，安装 Visual Studio.NET bootstrap。安装完支持环境后，就可以开始安装 Visual Studio .NET 了。

（1）先放入 Disk1，自动启动安装程序后，将其关闭。

（2）执行/Setup/setup.exe 文件，就正式开始安装 Visual Studio 7.0 了，安装首页如图 1-2 所示。由于在本书的写作过程中，Visual Studio 7.0 版本还是测试版，所有 product key（产品序列号）都是自动输入的，只要选择“I accept the agreement”选项，即可进入下一步安装过程。出现了选择 Visual Studio 7.0 组件的画面，如图 1-3 所示。

图 1-2 安装首页

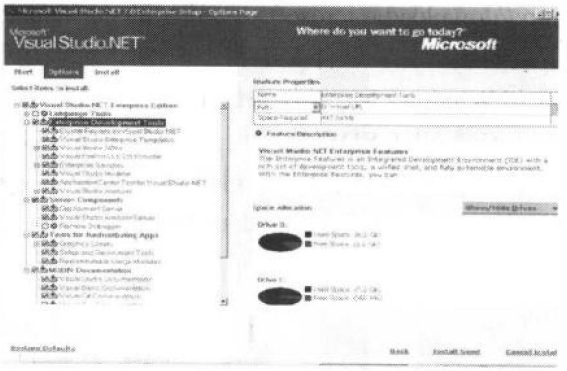

图 1-3 选择组件

Visual Studio 7.0 具有如下组件：

（1）语言工具（Language Tools）

Visual Studio 7.0 语言工具包括 Visual C++ 7.0、Visual Basic 7.0、Visual FoxPro 7.0 等。在这里并没有列出 C#，因为所有的语言都使用同一个集成开发环境，所以只要安装了其中任何一个语言工具，就可以使用 C#开发 Windows 程序。

（2）企业级开发工具（Enterprise Development Tools）

Visual Studio 7.0 企业级开发工具包括：

报表工具 Crystal Report；

- Enterprise Template，使用该模板可以定义重复使用的初始结构，从而可以加快开发

的速度；

- Visual Studio SDK，包括用来为程序开发在线帮助的 HTML Helper 1.3 SDK；
- VC Browser Toolkit SDK，用来查询 BSC 数据库中的信息，BSC 数据库是在编译 Visual C++程序的时候会自动生成一个数据库文件；
- DIA SDK，用来查询编译器产生的 PDB 文件中的信息；
- Visual Studio Analyzer SDK，Visual Studio Analyzer 的开发包；
- Windows Management Instrumentation SDK Tools，是对 Visual Studio 的 ATL 向导的扩展；
- Visual Foxpro OLE DB Provider，可以用来访问 Visual FoxPro 的数据库，和原来的 ODBC 不同的是，Visual Foxpro OLE DB Provider 支持存储过程，以及良好的可缩放性；
- Enterprise Samples，企业版例子，包含了两个电子商务程序；
- Visual Studio Modler，用来创建图表和模型；
- Application Center Test fro Visual Studio .NET，用来对 Web 程序进行测试；
- Visual Studio Analyzer，用来分析程序的性能，隔离错误等。

（3）服务器组件（Server Components）

服务器组件（Server Components）包括 Development Server、Visual Studio Analyzer Server 和 Remote Debug Sever。

（4）程序发布工具（Tools for Redistributing Apps）

程序发布工具（Tools for Redistributing Apps）包含了 Graphic Library（图片库）、Setup and Deployment Tools（用来制作安装程序）等重要工具。

（5）MSDN Documentation

微软公司提供的参考文档。

选择好要安装的组件后，在右上角的 Path 栏中选择要安装的路径，如图 1-4 所示。选择完毕后即可开始安装，如图 1-5 所示。

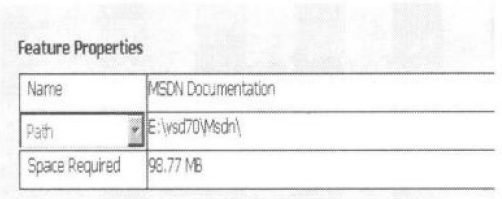

图 1-4　选择安装的路径

图 1-5　开始安装

安装程序会自动完成安装。如果在安装中出现问题，安装程序会给出错误信息。安装完毕，安装程序会给出提示信息。

1.3.3 Visual Studio 7.0 安装中常见问题

- 在执行安装程序过程中，提示插入 Disk1，但插入 Disk1 还是没有用，解决的方法是将 Disk1、Disk2 中的所有内容拷贝到硬盘，如果提问是否覆盖文件，请选择覆盖，然后再执行/Setup 目录下的 setup.exe。
- 安装完 Visual Studio.NET，并成功地启动了它，但在新建 C#项目的时候如果提示没有找到 MsCorlib.dll，这说明还没有正确地升级组件，重新按上面升级组件的顺序安装组件，Visual Studio .NET 不需要重新再安装。
- 如果升级完组件后还是不能安装 Visual Studio，提示找不到相应的语言版本，那么需要安装一个补丁程序，请遵照以下的步骤进行：

先装 Windows 的 sp1，再安装 Office 2000 的 FrontPage 2000 Server Extensions。一般在装 Office 2000 时会按照提示进行安装，如果要对其进行升级，在网址 http://www.microsoft.com/windows2000/downloads/recom mended/q274294/default.asp中，请选择简体中文版。

第 2 章　C#语言入门

本章将简单介绍 C#语言的基本概念，并结合基本概念介绍相应的 C#程序。C#的目的是把 Visual Basic 的易用和 C++的高性能结合起来。C#中的很多概念和 C/C++语言中的十分类似，但是也有自己独特的概念，本章对这些概念的说明简洁明了，目的是为以后章节的阅读打下基础。读完本章，读者可以对 C#语言有一个大概的了解。

2.1　“Hello,world”程序

用 C#编写的规范的“Hello,world”程序如下：

```
//源程序 hello.cs
using System;
class Hello
{
static void Main() {
Console.WriteLine("Hello, world");
}
}
```

C#程序的默认文件扩展名为.cs，如 hello.cs。

使用命令行语句：

```
csc hello.cs
```

编译，产生一个名为 hello.exe 的可执行文件。

执行 hello.exe，程序的输出是：

```
hello,world
```

这段程序的解释如下：

“using System;”行引用了一个由.NET 运行时间库提供的称为 System 的名字空间。System 名字空间包括了以 main 方式引用的控制类。名字空间提供分层的方式来组织类库，“hello,world”程序使用 Console.WriteLine 作为 System.Console.Writeline 的简写。System 是一个名字空间，Console 是一个在名字空间定义的类，WriteLine 是该类中定义的静态方法。

main 函数是类 hello 的静态成员。“hello,world”通过使用类库输出。C#自身不提供类库。C#使用.NET 的类库，其他语言如 Visual Basic 和 Visual C++也使用这个类库。

程序不使用#include 来包含程序文本。

下面是一个带参数的“Hello,world”程序：

```
//源程序：hello1.cs
using System;
```