

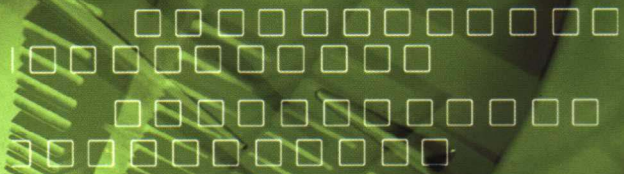
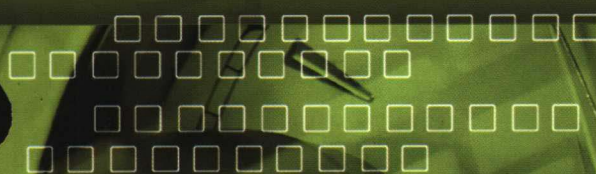
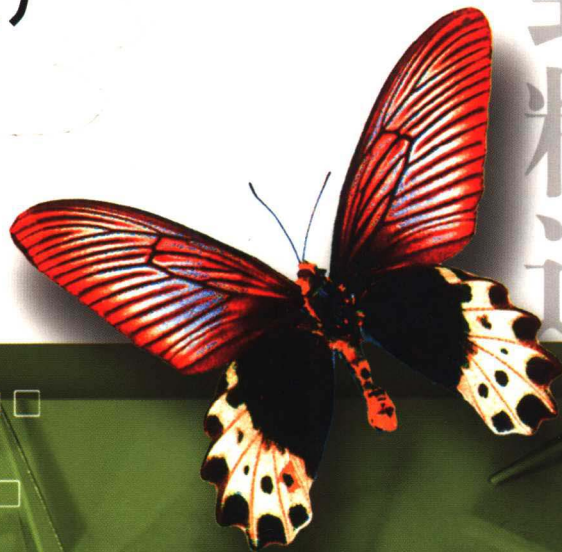
陈 铨 陈初铨 宫 正 编著  
邱仲潘 邱仲潘 审校

从入门到精通

# Visual C# 2005

## 从入门到精通 (普及版)

入门到精通



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>

TP312

2352

2007

# Visual C# 2005 从入门到精通

## (普及版)

陈 锵 陈初铨 宫 正 编著

邱仲潘 审校

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

## 内 容 简 介

本书全面介绍了 Visual C# 2005 的特点、使用方法及编程技巧,旨在提供 Visual C# 的“从入门到精通”式的综合性指南。具体内容包括: C#的基本语法、C#面向对象编程、C#高级语法、Visual C# 2005 的新特性、Visual C# 2005 的常用控件、Visual C# 2005 的基本应用、基于 ADO.NET 的数据库编程、C#网络编程和基于 ASP.NET 的 Web 编程等。

本书为“从入门到精通”类图书,适合阅读的读者范围涵盖初学者到高级技术人员。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。  
版权所有,侵权必究。

### 图书在版编目(CIP)数据

Visual C# 2005 从入门到精通: 普及版 / 陈锵等编著. —北京: 电子工业出版社, 2007.5

ISBN 978-7-121-03964-5

I. V… II. 陈… III. C 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2007)第 030072 号

责任编辑: 吴 源

印 刷: 北京天竺颖华印刷厂

装 订: 三河市金马印装有限公司

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

北京市海淀区翠微东里甲 2 号 邮编: 100036

开 本: 787×1092 1/16 印张: 25.5 字数: 650 千字

印 次: 2007 年 5 月第 1 次印刷

定 价: 38.00 元

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。联系电话: (010) 68279077; 邮购电话: (010) 88254888。

质量投诉请发邮件至 zits@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010) 88258888。

# 前 言

1995年，Sun公司正式推出了面向对象的开发语言Java。在提出了跨平台的概念后，Java就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。

微软也不甘落后，推出了基于Java语言的编译器Visual J++，Visual J++在最短的时间里由1.1版本升到了6.0版本。但是，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，因而对微软提出了诉讼，这使得微软处于极为被动的局面。就在人们认为微软的局面不可能再有改观的时候，微软却另辟蹊径，决定推出其进军因特网的庞大计划.NET计划，以及该计划中旗帜性的开发语言C#。于是Visual J++渐渐消失在人们的视线里，取而代之的就是C#语言。

C#的作者就是著名的安德尔斯，他在此之前开发了Turbo Pascal语言，并在其基础上开发了面向对象的应用开发工具Borland Delphi。C#语言从C和C++演变而来，它是为那些愿意牺牲C++一点底层功能，以获得更方便和更产品化的企业开发人员而创造的。C#的优点是现代、简单、面向对象和类型安全。虽然它借鉴了C和C++的许多东西，但是在一些特定的领域，C#为你提供了方便的功能，仅有的“代价”就是代码操作默认是类型安全的，并且不允许使用指针。大多数问题通过类型安全就可以解决了。但是，如果你需要使用指针，仍可以通过不安全代码使用它们。

随着Visual Studio 2005的发布，C#语言已更新为2.0版，即Visual C# 2005。它增加了下列新功能：泛型、迭代器、分部类、可空类型、匿名方法、命名空间限定符、静态类、新的编译器选项等。本书就是在Visual C# 2005环境下讲述C#语言从入门到精通的相关知识。

在这本书中，作者站在一个初学者的角度，由浅入深，进而深入浅出地讲述了C#的方方面面。这个过程也许不像一个故事那样离奇曲折，引人入胜，但一次成功的求知旅程一样可以令人回味无穷。

本书以一个大多数程序员的启蒙程序——“Hello, World!”开始，共分10章，阐述了以下几个方面内容。

- C#的基本语法
- C#面向对象编程
- C#高级语法
- Visual C# 2005的新特性
- Visual C# 2005的常用控件
- Visual C# 2005的基本应用
- 基于ADO.NET的数据库编程
- C#网络编程

### ● 基于 ASP.NET 的 Web 编程

通过阅读本书，读者不仅可以精通 C# 的编程理论，还可以从不同的实例中汲取到宝贵的实践经验，这对读者今后的工作与学习都大有裨益。

本书适合的对象有：

- (1) 以前没有接触过 C# 或者对 C# 不甚了解，而想进入 C# 程序员队伍的读者朋友们；
- (2) C# 开发的初学者也应该好好品读一下本书；
- (3) 如果你在前一个版本的 C# 上已经有了相当深入的经验，那么你也可以购买本书作为一个参考，特别是书中有关 Visual C# 2005 新特性的内容，相信对你会有帮助。

本书由陈镭、陈初铨、宫正、孙赫雄老师等一起完成，并由邱仲潘审校，刘文红、刘文琼等同志也提供了大量帮助，在此一并表示感谢。

---

为方便读者阅读，若需要本书配套资料，请登录“华信教育资源网”(<http://www.hxedu.com.cn>)，在“资源下载”频道的“图书资源”栏目下载。

## 目 录

第 1 章 进入 C# 世界 .....	1	3.2 定义一个 C# 类 .....	58
1.1 欢迎来到 Visual Studio 2005 .....	1	3.2.1 类和接口 .....	58
1.1.1 经典示例程序 .....	1	3.2.2 构造函数和析构函数 .....	62
1.1.2 Visual Studio 2005 开发环境 .....	3	3.2.3 类成员的定义 .....	65
.....	3	3.2.4 this 关键字 .....	71
1.1.3 管理项目 .....	9	3.3 结构与类的区别 .....	74
1.2 初识 C# .....	14	3.3.1 实例在内存中的位置以及 .....	74
1.2.1 C# 语言概述 .....	14	内存回收 .....	74
1.2.2 .NET 框架 .....	16	3.3.2 结构不支持继承 .....	74
1.2.3 公共语言运行库 .....	18	3.3.3 结构字段的默认值 .....	75
第 2 章 C# 的特性 .....	21	3.3.4 关键字 this 的含义 .....	75
2.1 基本语法 .....	21	3.3.5 装箱和拆箱 .....	77
2.1.1 C# 语法概述 .....	21	3.3.6 结构的构造函数 .....	77
2.1.2 C# 中的注释 .....	21	3.4 运算符重载 .....	79
2.2 类型 .....	22	3.5 继承 .....	83
2.2.1 简单类型 .....	22	3.5.1 类的继承 .....	83
2.2.2 运算符与表达式 .....	28	3.5.2 继承下的构造函数和析 .....	84
2.3 枚举、结构体和数组 .....	31	构造函数 .....	84
2.3.1 枚举 .....	32	3.5.3 关键字 base 和 new .....	87
2.3.2 结构体 .....	34	3.6 多态 .....	89
2.3.3 数组 .....	36	3.7 虚方法与抽象方法、接口与 .....	95
2.4 变量和常量 .....	42	抽象类 .....	95
2.5 流程控制 .....	44	3.8 System.Object 类 .....	105
2.5.1 分支控制 .....	44	第 4 章 深入 C# .....	107
2.5.2 循环控制 .....	49	4.1 接口 (Interface) 和委托 .....	107
2.6 类型转换 .....	52	(Delegate) .....	107
2.6.1 隐式转换 .....	52	4.1.1 使用接口 .....	107
2.6.2 显式转换 .....	52	4.1.2 使用委托 .....	111
2.6.3 使用转换命令的显式转换 .....	53	4.1.3 如何选用接口和委托 .....	114
第 3 章 C# 面向对象编程 .....	56	4.2 事件 (Event) .....	114
3.1 面向对象的概念 .....	56	4.2.1 使用事件 .....	114
3.1.1 类和对象 .....	56	4.2.2 自定义事件 .....	118
3.1.2 类的特性 .....	56	4.2.3 在接口中声明事件 .....	121

4.3 泛型 (Generic) .....	122	5.7.2 “:” 又回来了 .....	191
4.3.1 使用泛型 .....	123	5.8 外部程序集别名 .....	193
4.3.2 自定义泛型 .....	126	5.9 属性访问器的可访问性 .....	194
4.3.3 泛型与模板的区别 .....	130	5.9.1 访问修饰符的应用 .....	194
4.3.4 泛型和数组 .....	130	5.9.2 访问器可访问性域 .....	196
4.4 索引器 (Indexer) .....	132	5.10 委托中的协变和逆变 .....	198
4.4.1 什么是索引器 .....	132	5.10.1 协变 .....	198
4.4.2 实现接口索引器 .....	135	5.10.2 逆变 .....	199
4.4.3 索引器与属性的区别 .....	136	5.11 实例化委托的新方法 .....	200
4.5 迭代器 (Iterator) .....	136	5.12 固定大小的缓冲区 .....	200
4.5.1 使用迭代器 .....	136	5.12.1 关于创建固定大小的数据 结构 .....	200
4.5.2 深入迭代器 .....	139	5.12.2 不安全缓冲区与常规数组 的差异 .....	201
4.5.3 为整数列表和泛型列表实现 迭代器 .....	140	5.13 友元程序集 .....	202
4.6 异常 (Exception) .....	146	5.13.1 不同程序集之间的访问 .....	202
4.6.1 使用和处理异常 .....	146	5.13.2 强名称的不同程序集之间 的访问 .....	203
4.6.2 创建和引发异常 .....	151	5.14 内联警告控制 .....	205
4.6.3 异常分类和常见的异常 .....	154	5.15 volatile .....	205
4.7 处理器指令 .....	155	5.16 编译器增加和更改的选项 .....	206
4.8 使用垃圾回收和资源管理 .....	161	5.16.1 /errorreport 选项 .....	206
4.9 可空类型 .....	163	5.16.2 /keycontainer 和/keyfile 选项 .....	207
4.10 不安全代码和指针 .....	168	5.16.3 /linkresource 选项 .....	207
4.11 XML 文档注释 .....	173	5.16.4 /moduleassemblyname 选项 .....	207
<b>第 5 章 Visual C# 2005 新特性</b> .....	<b>179</b>	5.16.5 /platform 选项 .....	208
5.1 泛型 .....	179	5.17 C# 3.0 新特性简介 .....	208
5.2 迭代器 .....	179	5.17.1 隐式类型本地变量 .....	209
5.3 分部类 .....	179	5.17.2 匿名变量 .....	209
5.3.1 类学会了分身术 .....	179	5.17.3 扩展方法 .....	210
5.3.2 为什么要增加分部类的功能 .....	181	5.17.4 对象和 collection 初始 化符 .....	210
5.3.3 使用分部类的注意事项 .....	182	5.17.5 Lambda 表达式 .....	211
5.4 静态类 .....	184	5.17.6 查询表达式 .....	212
5.4.1 什么是静态 .....	184	5.17.7 表达式树 .....	213
5.4.2 静态类的作用 .....	185	<b>第 6 章 Visual C# 2005 常用控件</b> .....	<b>214</b>
5.5 可空类型 .....	188	6.1 公共控件 .....	214
5.6 匿名方法 .....	188	6.1.1 按钮 (Button) .....	214
5.6.1 命名方法和匿名方法 .....	188		
5.6.2 使用匿名方法的注意事项 .....	189		
5.7 命名空间别名限定符 .....	190		
5.7.1 命名空间被隐藏的问题 .....	190		

6.1.2	单选按钮 (RadioButton) 和 复选框 (CheckBox) .....	221	7.3.2	绘制简单的线条 .....	272
6.1.3	组合框 (ComboBox) 和列表框 (ListBox) .....	225	7.3.3	绘制几何图形和呈现图像 .....	275
6.1.4	进度条 (ProgressBar) .....	228	7.3.4	绘制文本 .....	278
6.1.5	日历控件 (MonthCalendar) 和日期时间提取控件 (DateTimePicker) .....	230	7.4	动态链接库 (DLL) .....	280
6.1.6	Tab 属性页 (TabControl) .....	233	7.4.1	什么是 DLL .....	280
6.1.7	列表视图 (ListView) .....	234	7.4.2	创建和使用 C# 动态链接库 .....	281
6.1.8	树形视图 (TreeView) .....	237	7.4.3	调试动态链接库 .....	285
6.2	对话框 .....	239	7.5	多任务编程 .....	286
6.2.1	文件夹浏览器 (FolderBrowserDialog) .....	239	7.5.1	多任务、进程与线程的 基础知识 .....	286
6.2.2	打开文件 (OpenFileDialog) .....	240	7.5.2	进程与线程控制 .....	287
6.2.3	保存文件 (SaveFileDialog) .....	242	7.5.3	线程同步 .....	294
6.2.4	字体对话框 (FontDialog) .....	244	<b>第 8 章</b>	<b>数据库编程</b> .....	300
6.2.5	颜色对话框 (ColorDialog) .....	245	8.1	ADO.NET 简介 .....	300
6.3	菜单和工具栏 .....	246	8.1.1	数据库编程与 ADO.NET .....	300
6.3.1	菜单 (MenuStrip) .....	246	8.1.2	关于 ADO.NET 的类 .....	303
6.3.2	快捷菜单 (ContextMenu Strip) .....	249	8.1.3	Windows 应用程序与 ADO.NET .....	310
6.3.3	工具栏 (ToolStrip, ToolStripContainer) .....	251	8.2	ADO.NET 应用 .....	313
6.3.4	状态栏 (StatusStrip) .....	255	8.2.1	使用 DataReader 从数据库 中读取数据 .....	313
6.4	组件 .....	258	8.2.2	使用 DataSet 从数据库中 读取数据 .....	315
6.4.1	定时器 (Timer) .....	258	8.2.3	更新数据库的内容 .....	317
6.4.2	图像列表 (ImageList) .....	258	8.2.4	访问数据集中的多个表 .....	320
6.4.3	错误提示器 (ErrorProvider) .....	259	8.2.5	深入理解 ADO.NET 中的 SQL 语句 .....	322
<b>第 7 章</b>	<b>基本应用</b> .....	261	8.2.6	数据绑定 .....	326
7.1	典型的应用程序 .....	261	<b>第 9 章</b>	<b>网络编程</b> .....	331
7.1.1	Windows 应用程序 .....	261	9.1	建立连接 .....	331
7.1.2	控制台应用程序 .....	265	9.1.1	TCP/IP 协议 .....	331
7.2	文档和视图 .....	267	9.1.2	使用 TcpListener 和 TcpClient 收发信息 .....	333
7.2.1	文档和视图的概念 .....	267	9.1.3	使用 Socket 类代替 TcpListener 和 TcpClient .....	338
7.2.2	实现一个简单的多文档 编辑器 .....	268	9.2	典型的网络应用 .....	341
7.3	绘图 .....	270	9.2.1	下载网页 .....	341
7.3.1	C# 绘图机制 .....	271	9.2.2	下载和上传文件 .....	343
			9.2.3	接收电子邮件信息 .....	347



9.2.4 实现 Ping 命令.....	351	10.2.4 状态管理.....	378
<b>第 10 章 基于 ASP.NET 的 Web 编程</b> ...	<b>362</b>	<b>10.3 Web 服务</b> .....	<b>386</b>
10.1 ASP.NET 技术概述.....	362	10.3.1 Web 服务的概念.....	386
10.1.1 ASP.NET 技术.....	362	10.3.2 Web 服务的优点.....	387
10.1.2 ASP.NET 2.0 新特性.....	363	10.3.3 Web 服务的体系结构.....	388
10.2 ASP.NET 编程基础.....	365	10.3.4 Web 服务平台.....	388
10.2.1 创建一个简单的 Web 页面... 365		10.3.5 Web 服务的例子.....	392
10.2.2 事件处理.....	369	<b>10.4 部署 Web 应用程序</b> .....	<b>394</b>
10.2.3 输入验证.....	371		

# 第 1 章 进入 C# 世界

## 1.1 欢迎来到 Visual Studio 2005

### 1.1.1 经典示例程序

相信曾经接触过编程的读者对“Hello, World!”都不陌生,它几乎可以算是计算机界最有名的示例程序了。就像初生的婴儿要先学会啼哭一样,一个编程初学者也必须先学会“Hello, World!”。当然,从来没有接触过编程的读者也不用担心,因为“Hello, World!”就是为初学者而设计的,决不会包含任何晦涩难懂的内容。

现在,就让我们开始学习这新生儿的第一声啼哭吧!

也许有的读者初涉编程,想要马上看到自己的程序顺利地运行起来,但是,别急,让我们先了解一下自己要做什么,而不是怎么做。这是因为即使将来读者精通了 C#, 并成为了一个编程高手,也不得不在开始任何项目之前构思一下。所谓“谋定而后动”,因此我们最好从编程的第一课就养成这个良好的习惯。

那么,先来看看我们的目的吧。毫无疑问,读者是为了学好 C#才购买了这本书,所以在看到 Visual C# 2005 之后,最好先跟它打个招呼,说明“我要开始学你了”。如此这般,在彼此熟识了以后,有助于进一步的了解。这也就是我们编写“Hello, World!”程序的全部目的了。

要打招呼有很多种方法,可以招招手、点点头或是说一句简短的问候语。显而易见,“Hello, World!”就是一句简短的问候。唯一的疑问是,我们要怎么把它说出口呢?这就要交给 Visual C# 2005 的开发环境来完成了,我们要做的就是给开发环境一点点指示。

首先,在 Visual Studio 2005 的初始界面,单击“文件”菜单中的“新建”命令,选择“项目”;另外,读者同样可以在此界面的“最近的项目”页面中单击“创建项目”。然后,可以看到类似于如图 1.1 所示的新建项目界面。

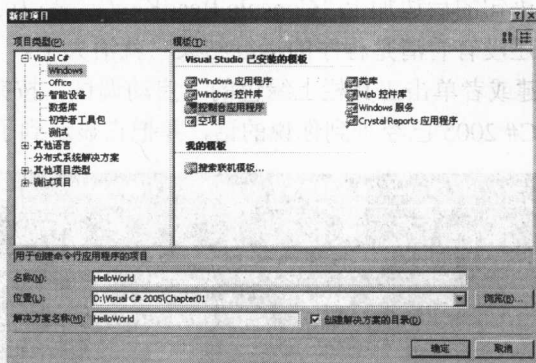


图 1.1 新建项目

在如图 1.1 所示的左边的树形结构中,选择“Visual C#”→“Windows”,再在“Visual Studio 已安装的模板”中选择“控制台应用程序”,接着将新建项目的“名称”和“解决方案名称”都命名为“HelloWorld”,“位置”则设定为读者想要保存项目代码的路径,然后单击“确定”按钮。

这时我们就可以看到代码了。在系统自动生成的 Program.cs 文件中,包含了几行简单的代码,如图 1.2 所示。

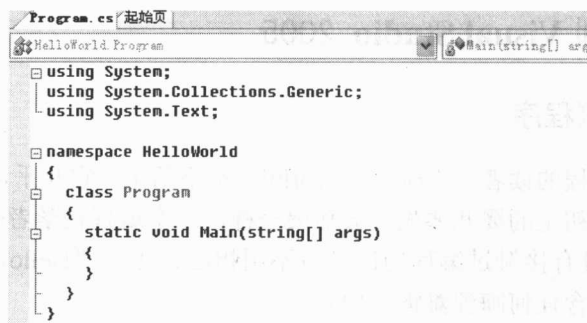


图 1.2 代码视图

先不去管这些代码的意义何在,等了这么久,我们要先跟 Visual C# 2005 打声招呼了。即包含英文单词 Main 的这行代码及其下边的一对大括号,就是整个程序的入口,我们只需要在这两个大括号中添加适当的代码,就可以实现“Hello, World!”的功能。由于我们建立的是控制台程序,所以自然要去引用一些控制台本身特有的方法。现在,让我们在那对大括号中输入控制台对应的英文单词 Console,然后按下“.”,调用控制台的 WriteLine 方法,说出那句我们想了好久的话吧!添加代码后的主程序如图 1.3 所示。

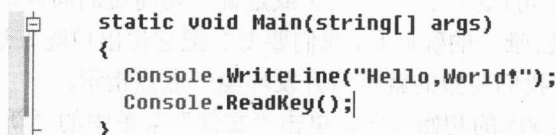


图 1.3 添加代码

在该主程序的第一条语句的后边加上“Console.ReadKey();”是为了产生一个“按任意键继续”的效果,不然你可能还没有看清楚程序的运行结果它就消失了。

这时你可以按下 F5 键或者单击工具栏上绿色的“启动调试”按钮来运行你的程序了。接下来,你就会发现 Visual C# 2005 已经听到你说的话,并把它显示到了屏幕上,如图 1.4 所示。

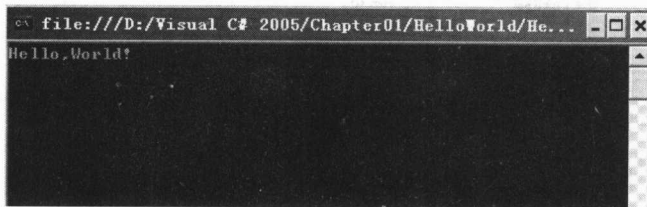


图 1.4 运行结果

这下子，你可以好好欣赏自己的成果了。它就是“Hello, World!”，虽然这只是一个简单的黑底白字界面程序，但其不失直观与简洁，并且令人印象深刻。不管是如今的 Windows 操作系统，还是你现在正使用的 Visual Studio 2005 开发工具，都离不开伟大的“Hello, World!”的启蒙，千万不要小看它。

当然，如果你不想再看这个屏幕了，或者想要结束程序的调试，按下任意键就可以让它消失。毕竟，我们不能把时间都浪费在欣赏成就上。

至此，我们这个“Hello, World!”项目就算构建完成了，读者也看到了它的运行效果。接下来，让我们大致理解一下这部分代码。虽然这几行代码很简单，但却可以体现不少 C#语言的特点。

开头有 3 条 using 语句，顾名思义，它们是用来引用本程序需要的一些现成的东西，如 Console.WriteLine 方法的实现代码，C#将这些东西放在命名空间（Name Space）的集合中。代码中的 System、System.Collections.Generic 和 System.Text 都是命名空间的名称。

在接下来的代码中，读者可以看到，我们的“Hello, World!”也使用关键字 namespace 声明成了命名空间。这是 C#语言的规则，便于以后其他的代码模块引用这块代码。

在“Hello, World!”命名空间中，包含了名为 Program 的类。在以后的学习中，读者也可以发现，C#语言中充满了形形色色的类，它们体现出 C#很强的面向对象（Object Oriented）特点。

下面的这个静态（Static）方法是 C#程序中必须包含的 Main 方法。不但程序控制在该方法中开始和结束，而且用户还可以在其中创建对象和执行其他方法。

在 Main 方法中调用的 WriteLine 方法是控制台类 Console 的输出方法之一，而 Console 类是在 .NET 框架的运行时库中实现的，包含在命名空间 System 中，此命名空间正是第一行代码所引用的。

关于 C#的“Hello, World!”程序我们就简单介绍到这里，也许有些读者还没有完全理解这个程序的各个细节，但这并没有关系，因为本小节只是带领读者初识 Visual C# 2005 的编程环境，并让读者对 C#编程有某种程度上的理解，说这么多就足够了。关于 C#语言更深入的知识以及它在各方面的应用，我们将在以后的章节中逐步阐述。

## 1.1.2 Visual Studio 2005 开发环境

Microsoft Visual Studio 2005 中的许多开发环境的创新和增强功能有助于开发人员更快、更好地编写代码。

在编写应用程序之前，我们先简要地了解一下 Visual Studio 2005 IDE 的功能。

### 1. 配置设置

每个开发者都有自己的开发风格，Visual Studio IDE 使你能够把各个 IDE 元素摆列成最适合自己的开发风格。

从菜单栏的“工具”→“选项”，打开如图 1.5 所示的窗口。“选项”对话框提供了开发环境大量的设置类别。

在 Visual Studio 2005 之前的版本中，每次使用新的 IDE 都必须重新设置首选项，这让开

发人员不胜其烦。而 Visual Studio 2005 提供的“导入和导出设置”可以快速轻松地还原首选项设置。如图 1.6 所示,选中选项对话框的“环境”设置中的“导入和导出设置”选项,在出现的“将我的设置自动保存到此文件”中输入保存路径。该文本框中的配置文件在每次关闭 IDE 时都会进行更新。配置文件可以是本地文件也可以是网络文件。如果是网络文件,那么在其中的一台计算机上修改设置,则其他使用这一配置文件的计算机也会使用新的设置。

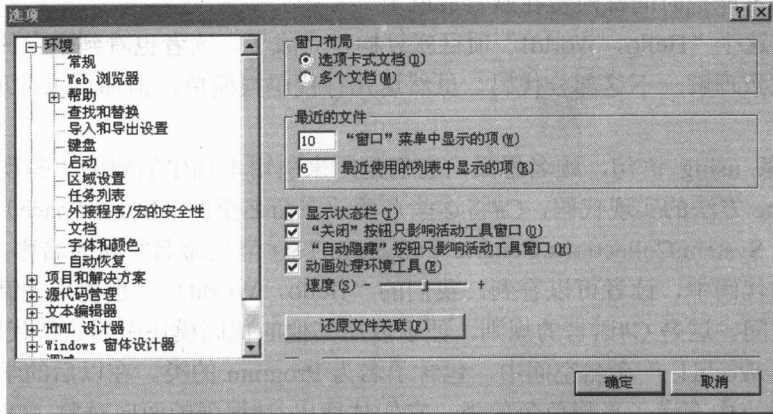


图 1.5 “选项”对话框

在“使用团队设置文件”复选框中,可以指定一组开发人员中共享的设置。只要把设置文件保存到网络共享的配置文件中,开发团队的人员在更新他们的 IDE 配置时就会使用团队的配置文件。即只需修改该网络共享的配置文件,其他的成员在下次启动 IDE 的时候就会自动更新该设置。

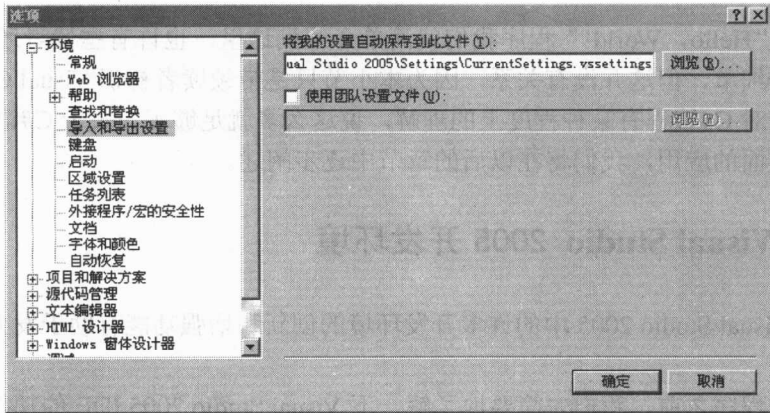


图 1.6 “导入和导出设置”对话框

## 2. 导入和导出设置

Visual Studio 2005 的菜单项“工具”中包含的“导入和导出设置”用于导入或者导出特定的设置类别,或者将环境重设为一个默认设置集合。如图 1.7 所示。

### (1) 导出选定的环境设置

选择“导出选定的环境设置”,并单击“下一步”按钮,出现如图 1.8 所示的对话框。选择

要导出的设置，并单击“下一步”按钮，出现如图 1.9 所示的对话框。重命名文件，并单击“保存”按钮，导出操作完成。

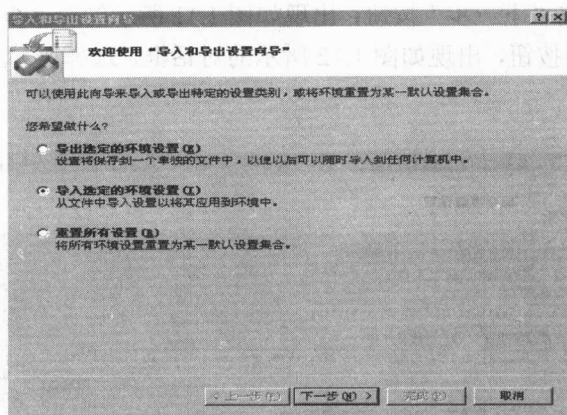


图 1.7 导入和导出设置向导

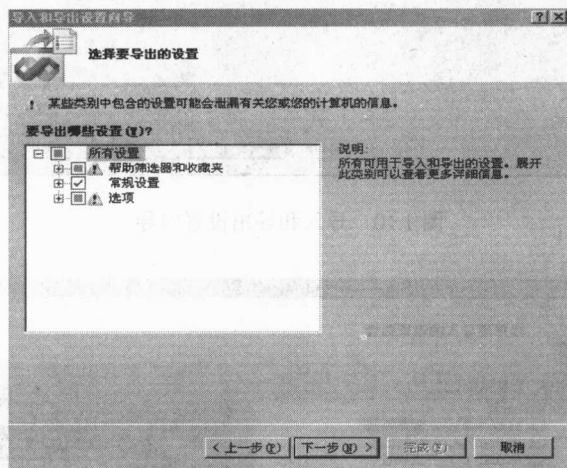


图 1.8 导入和导出设置向导

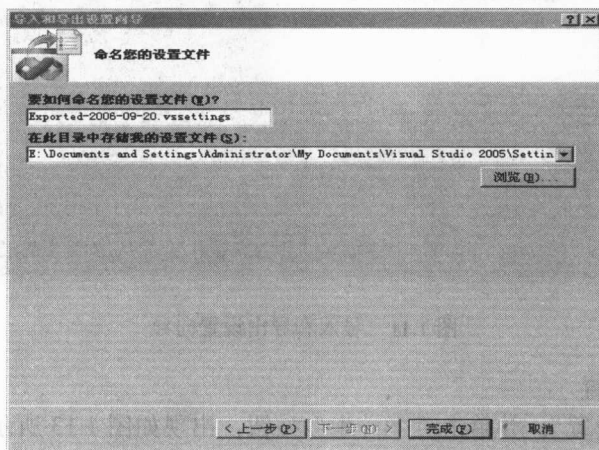


图 1.9 导入和导出设置向导

### (2) 导入选定的环境设置

选择“导入选定的环境设置”，并单击“下一步”按钮，出现如图 1.10 所示的对话框。选择要导入的设置，并单击“下一步”按钮，出现如图 1.11 所示的对话框。选择要导入的设置集合，然后单击“下一步”按钮，出现如图 1.12 所示的对话框，选择要导入的设置，单击“完成”按钮，导入操作完成。

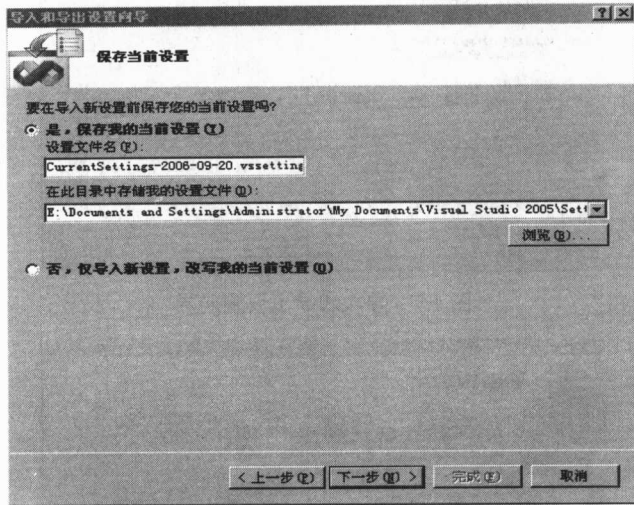


图 1.10 导入和导出设置向导

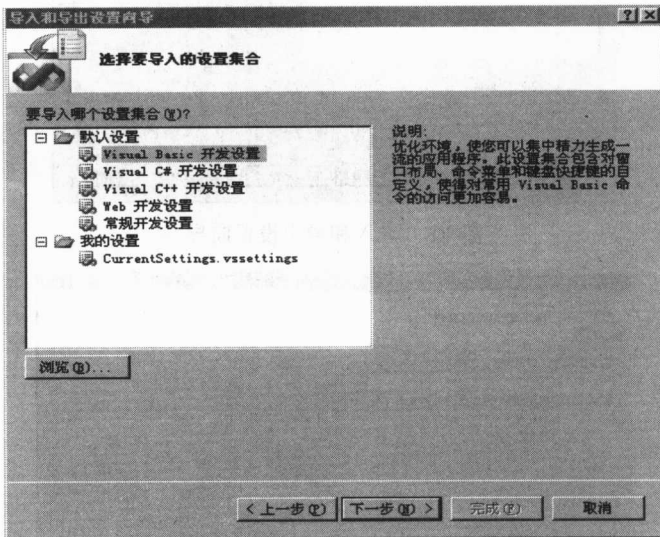


图 1.11 导入和导出设置向导

### (3) 重置所有设置

选择“重置所有设置”，并单击“下一步”按钮，出现如图 1.13 所示的对话框。选择“在重置前是否保存您的当前设置”的单选按钮，并做相应的输入。单击“下一步”按钮，出现如

图 1.14 所示的对话框，选择所要重置的设置集合，并单击“完成”按钮，重置所有设置操作完成。

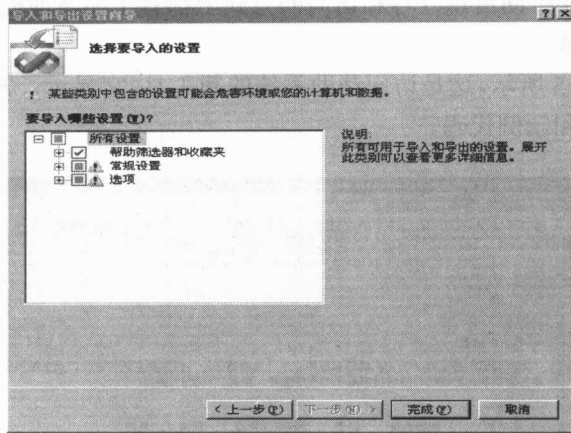


图 1.12 导入和导出设置向导

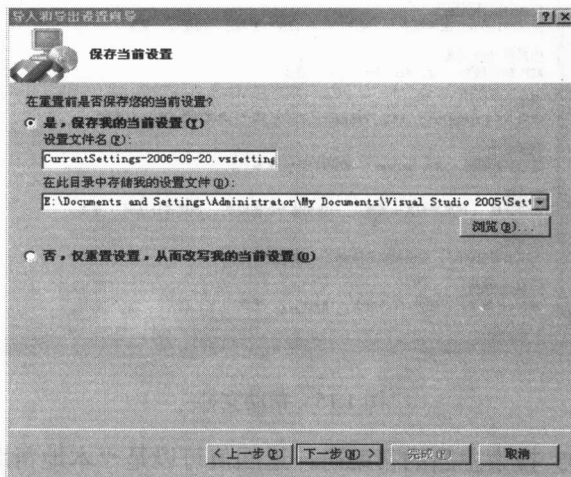


图 1.13 导入和导出设置向导

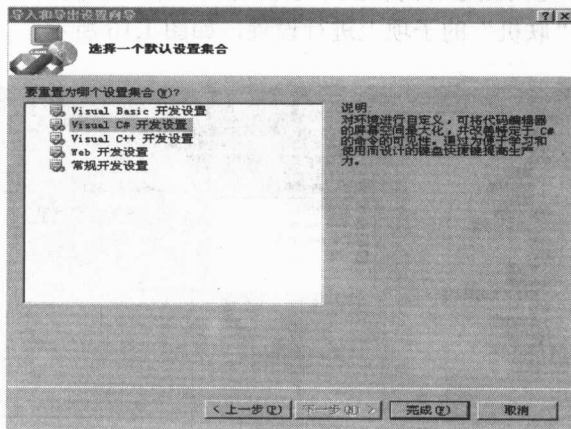


图 1.14 导入和导出设置向导



### 3. 社区和帮助

Visual Studio 2005 在帮助信息内容和访问的方式上都有了显著地提高。这些提高包括基于社区的资源更全面的集成。

如何实现? 如图 1.15 所示, 这是访问帮助系统的新工具, 它显示了常见的开发人员的任务, 帮助中包含了指导内容和示例代码。

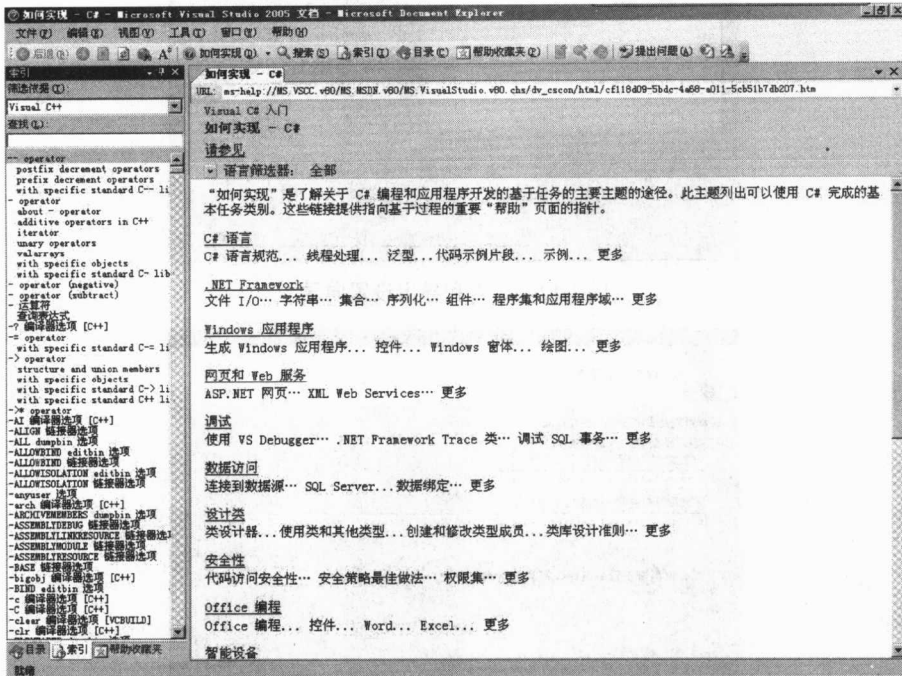


图 1.15 帮助文档

在 Visual Studio 2005 搜索主题的帮助时, 返回值可以是“本地帮助”、“MSDN Online”、“CodeZone”、“问题”中的相关页面链接的列表。该功能除了能够从 MSDN 上获取全面的内容之外, 还能够更方便、更快捷地访问社区中最新的代码示例和技术文章。我们可以在“选项”对话框中的“帮助”的“联机”的子项上进行设置, 如图 1.16 所示。

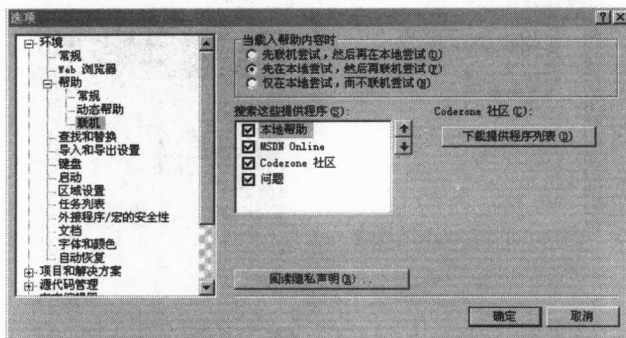


图 1.16 联机帮助