



高等学校电子信息类专业规划教材

C#程序设计

田 原 李素若 文 斌 编 著



清华大学出版社
<http://www.tup.tsinghua.edu.cn>



北京交通大学出版社
<http://press.bjtu.edu.cn>



21 世纪高等学校电子信息类专业规划教材

C # 程序设计

田 原 李素若 文 斌 编著

清华大学出版社
北京交通大学出版社
· 北京 ·

内 容 简 介

本书从最基本的词法、语法开始介绍 C# 语言,包括 C# 的数据类型、控制语句、名字空间、类、数组和枚举、集合、结构、接口、代表和事件以及错误和异常处理。另外,本书还介绍了 C# 的一些高级概念,包括文件和流、Windows 桌面应用程序开发、ASP.NET 中的 Web Form 和 Web Services、分布式应用程序创建、网络程序开发、数据库应用程序开发、多线程以及与 COM 对象的交互等。

本书结构清晰,内容丰富,可作为本科、大专层次院校的教材,也适用于使用 C# 进行软件开发的广大技术人员。

版权所有,翻印必究。举报电话:010-62782989 13501256678 13801310933

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

本书防伪标签采用特殊防伪技术,用户可通过在图案表面涂抹清水,图案消失,水干后图案复现;或将表面膜揭下,放在白纸上用彩笔涂抹,图案在白纸上再现的方法识别真伪。

图书在版编目(CIP)数据

C#程序设计/田原,李素若,文斌编著. —北京:清华大学出版社;北京交通大学出版社, 2005.6

(21 世纪高等学校电子信息类专业规划教材)

ISBN 7-81082-563-1

I. C… II. ①田… ②李… ③文… 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(2005)第 069344 号

责任编辑:周益丹 特邀编辑:吴炳林

出版者:清华大学出版社 邮编:100084 电话:010-62776969

北京交通大学出版社 邮编:100044 电话:010-51686414

印刷者:北京鑫海金澳胶印有限公司

发行者:新华书店总店北京发行所

开 本:185×260 印张:19.75 字数:473 千字

版 次:2005 年 7 月第 1 版 2005 年 7 月第 1 次印刷

书 号:ISBN 7-81082-563-1/TP·208

印 数:1~4 000 册 定价:28.00 元

本书如有质量问题,请向北京交通大学出版社质监组反映。对您的意见和批评,我们表示欢迎和感谢。

投诉电话:010-51686043,51686008;传真:010-62225406;E-mail:press@center.bjtu.edu.cn。

前 言

本书是关于 C#(读作 C Sharp)语言程序设计的一本教材。C#语言是一种崭新的面向对象的编程语言,它强调以组件为基础的软件开发方法。

基于组件的编程方法已出现多年,但 C#所提供的功能将使我们达到软件开发的新水平。这是向着 XML 的 Web 服务转变的新模式——将软件视为一种服务,这种服务是非连接型的、无状态的,并且符合国际的开放式标准。

把软件作为服务是下一代计算机系统的趋势。例如,C#非常适合于建立 Web 服务,适合于开发在遵循开放式标准的 Internet 上可重用的组件。软件开发不再限于过去几年中发展起来的单机结构。Web 服务使应用程序能够在 Web 上使用分布式服务,这就大大简化了软件开发,并提升了软件重用的规模。C#是在 Web 服务领域的主角,它有力地促进了软件作为服务这一大潮流。

本书不仅介绍 C#本身,而且展示了如何用 C#开发作为服务的软件。多少年来,程序都是作为目的单一的单机应用程序来编写的。通过研究和探索,我们已经认识到模块化的好处,这就逐渐发展出面向对象的方法,它为我们提供了大规模重用的可能性和可维护性。当协作和通信变成了商业的需求时,就自然地发展出了客户-服务器和连网技术。Internet 和 Web 为我们提供了分布式的、无状态的和安全的软件技术,包括 Applet 等。下一步将是 C#的用武之地——使 Internet 自动化。

本书中的全部内容都是关于 C#语言程序设计的问题。许多高级主题内容本身就足够写一本书了。但是,本书的重点并不是讲解这些高级主题的细节,尽管本书对其中的一些领域也有一定深度的介绍。每一章的目的都是展示怎样用 C#来完成给定的任务。即使是最理论化的章节,其重点仍然是如何使用 C#来编写代码,实现每章主题介绍的应用。

本书共分 18 章。从最基本的词法、语法开始讲解 C#语言,包括 C#的数据类型、控制语句、名字空间、类、数组和枚举、集合、结构、接口、代表和事件以及错误和异常处理。另外,本书还讲解了 C#的一些高级概念,包括 Windows 桌面应用程序开发、ASP.NET 中的 Web Form 和 Web Services、分布式应用程序创建、网络程序开发、数据库应用程序开发、多线程以及 COM 对象的交互等。

本书结构清晰,内容丰富,可作为本科、大专层次院校的教材,也适用于使用 C#进行软件开发的广大技术人员。

本书第 1 章由文斌编写,第 2~12 章由田原编写,第 13~第 18 章由李素若编写。田原负责审定全书内容。

由于作者水平有限,书中不妥或错误之处在所难免,敬请读者指正。

作 者

2005 年 7 月

目 录

第 1 章 .NET 和 C#简介.....	1
1.1 .NET 和 C#	1
1.2 .NET SDK	3
1.2.1 .NET SDK 简介	3
1.2.2 命令行编译	3
1.3 Visual Studio .NET 简介	4
1.3.1 Visual Studio .NET 系统要求	4
1.3.2 Visual Studio .NET 组件	4
1.3.3 Visual Studio .NET 集成环境 IDE	5
第 2 章 C#语言入门	13
2.1 “Hello,world”程序.....	13
2.2 类型.....	14
2.3 变量和参数.....	16
2.4 内置变量.....	17
2.5 数组类型.....	18
2.6 系统统一类型.....	20
2.7 自动内存管理.....	21
2.8 表达式.....	23
2.9 局部常量和局部变量.....	24
2.10 语句	24
2.10.1 if 语句	25
2.10.2 switch 语句	25
2.10.3 while 语句	26
2.10.4 do 语句	26
2.10.5 for 语句	27
2.10.6 foreach 语句	27
2.10.7 break 语句和 continue 语句	28
2.10.8 return 语句	28
2.10.9 throw 语句	28
2.10.10 try 语句.....	28
2.10.11 checked 和 unchecked 语句	28
2.10.12 lock 语句	28
2.10.13 语句标记和 goto 语句.....	28

第3章 表达式与控制语句	30
3.1 表达式	30
3.1.1 表达式分类	30
3.1.2 简单表达式	30
3.2 函数	31
3.2.1 函数分类	31
3.2.2 参数列表	32
3.3 操作符	33
3.3.1 操作符的优先级	33
3.3.2 基本操作符	34
3.3.3 单目操作符	39
3.3.4 算术操作符	40
3.3.5 移位操作符	41
3.3.6 关系操作符	41
3.3.7 位逻辑操作符	41
3.3.8 条件逻辑操作符	41
3.3.9 操作符的重载	42
3.3.10 条件语句	45
3.3.11 赋值语句	45
3.4 语句	47
3.4.1 语句块	47
3.4.2 空语句	47
3.4.3 标签语句	48
3.4.4 声明语句	48
3.4.5 表达式语句	48
3.4.6 选择语句	48
3.4.7 循环语句	50
3.4.8 跳转语句	51
第4章 名字空间	53
4.1 名字空间的声明和调用	53
4.1.1 名字空间的声明	54
4.1.2 名字空间的使用	56
4.1.3 using 指令	57
4.1.4 名字空间的别名	58
4.2 名字空间和基类	59
第5章 类	64
5.1 描述类的概念	64
5.1.1 抽象类	64
5.1.2 密封类	65

5.1.3 基类	65
5.2 类成员	66
5.2.1 类成员声明	66
5.2.2 继承	66
5.2.3 new 修饰符	67
5.2.4 权限修饰符	67
5.2.5 静态和实例成员	67
5.3 常量	68
5.4 字段	70
5.4.1 静态和实例字段	70
5.4.2 只读字段	72
5.4.3 常量和静态只读字段的区别	72
5.5 方法	73
5.5.1 方法参数	74
5.5.2 虚拟方法	79
5.5.3 重载基类中的方法	83
5.5.4 抽象方法	84
5.5.5 外部方法	85
5.6 属性	86
5.6.1 存取标记	86
5.6.2 虚拟、隐藏和抽象存取标记	89
5.7 事件	90
5.8 索引	95
5.9 操作符	101
5.10 实例构造函数	101
5.10.1 默认构造函数	102
5.10.2 私有构造函数	102
5.11 静态构造函数	103
5.12 析构函数	105
第6章 数组和枚举	106
6.1 数组	106
6.1.1 数组的基本概念	106
6.1.2 数组声明	106
6.1.3 数组的初始化	108
6.1.4 数组的应用	109
6.2 枚举	113
6.2.1 枚举的声明	114
6.2.2 枚举成员	114

第7章 集合	117
7.1 预定义的集合	117
7.1.1 ArrayList 集合	117
7.1.2 BitArray 集合	118
7.1.3 HashTable 集合	119
7.1.4 Queue 集合	120
7.1.5 SortedList 集合	121
7.1.6 Stack 集合	122
7.2 集合的接口	122
7.3 创建集合	123
7.3.1 列表的集合	123
7.3.2 使用 SiteList 集合	133
7.4 小结	134
第8章 结构	135
8.1 结构声明	135
8.2 结构的应用	136
第9章 接口	140
9.1 抽象类与接口	140
9.2 接口成员	140
9.2.1 方法	141
9.2.2 特性	141
9.2.3 索引	141
9.2.4 事件	141
9.3 隐式实现	142
9.3.1 单个类接口的实现	142
9.3.2 模拟多态性行为	145
9.4 显式实现	149
9.5 映射	154
9.6 继承性	156
第10章 代表和事件	159
10.1 代表	159
10.1.1 定义代表	159
10.1.2 创建代表方法处理程序	159
10.1.3 接通代表和处理程序	160
10.1.4 通过代表调用方法	160
10.1.5 多重代表	161
10.1.6 代表的相等性	163
10.2 事件	163
10.2.1 定义事件处理程序	163

10.2.2	注册事件	164
10.2.3	实现事件	166
10.2.4	触发事件	168
10.2.5	修改事件添加/删除方法	169
10.3	小结	173
第 11 章	处理异常和错误	175
11.1	try/catch 块	175
11.2	finally 块	177
11.3	预定义异常的类	178
11.4	处理异常	178
11.4.1	处理多个异常	178
11.4.2	处理和传递异常	179
11.4.3	从异常中恢复	182
11.5	设计自己的异常	184
11.6	checked 和 unchecked 语句	186
11.7	小结	188
第 12 章	Windows 桌面应用程序开发	190
12.1	桌面应用程序开发一般步骤	190
12.1.1	第一步:新建项目	190
12.1.2	第二步:界面设计	191
12.1.3	第三步:组件属性设置	192
12.1.4	第四步:程序设计	192
12.2	表单	193
12.2.1	属性	193
12.2.2	表单的基本方法	197
12.2.3	表单响应的主要事件	199
12.2.4	表单应用的例子	200
第 13 章	ASP.NET 中的 Web Form	204
13.1	ASP.NET 简介	204
13.1.1	ASP.NET 的功能	206
13.1.2	ASP.NET 与 ASP 比较	208
13.1.3	ASP.NET 中文件类型	209
13.2	基于 Web 表单的开发	209
13.2.1	Web Form 介绍	209
13.2.2	第一个 Web Form	210
13.2.3	页面处理过程	213
13.2.4	View State	214
13.3	使用 ASP.NET 开发应用程序	215
13.3.1	Global.asax 文件	215

13.3.2 Web.config	216
13.4 服务器控件.....	219
第 14 章 ASP.NET 中的 Web Services	224
14.1 Web Services 简介	224
14.2 VS.NET 中的 XML Web services	227
14.2.1 XML Web services 简介	228
14.2.2 使用 Visual Studio 创建 XML Web services	228
14.2.3 使用 Visual Studio 部署 XML Web services	231
14.2.4 Webservice 目录	236
第 15 章 创建分布式应用程序	241
15.1 将创建的分布式应用程序的结构.....	241
15.2 分布式应用程序的创建过程.....	242
15.2.1 创建中间层业务对象.....	243
15.2.2 创建用户界面	246
15.2.3 部署解决方案	251
第 16 章 利用 C#开发网络程序	252
16.1 Socket 类开发	252
16.1.1 说明.....	252
16.1.2 开发实例	256
16.2 NET 类开发	259
16.2.1 程序设计的主要思路及实现方法	260
16.2.2 创建过程	261
第 17 章 C#数据库应用程序开发	263
17.1 建立连接.....	263
17.2 查看数据.....	264
17.3 操作数据.....	268
17.4 调用存储过程.....	272
17.5 检索数据集.....	276
17.6 小结.....	279
第 18 章 C#高级应用	280
18.1 多线程.....	280
18.1.1 创建新的线程	280
18.1.2 同步化	281
18.1.3 小结.....	284
18.2 与 COM 对象的交互	284
18.2.1 创建一个非管理 COM 类包容器.....	284
18.2.2 在 C#代码中声明一个非管理的 COM 类	285
18.2.3 在 C#代码中创建一个非管理 COM 类的实例	286
18.2.4 C#中的 QueryInterface	286

18.2.5 综合实例	286
18.3 创建受管理 DLL 和调用外部 DLL	291
18.3.1 编译和调用一个动态链接库	291
18.3.2 调用外部 DLL 库	294
18.4 代码安全	294
18.4.1 安全性	295
18.4.2 安全策略	295
18.4.3 安全和性能	298
参考文献	301

第 1 章 .NET 和 C#简介

本章首先简单介绍.NET平台C#(读作C Sharp),然后介绍.NET SDK和命令行编译,最后介绍Visual Studio .NET安装的系统要求、包含的组件及集成开发环境IDE。

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CLR(Common Language Runtime),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+)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 JIT(Just In Time Compiler)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的 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 代码。

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,除非它被转换成可执行代码,为此需要一个即时编译器。

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#源文件通过 C#编译器生成 PE 文件(.exe 或.dll 文件),但这个扩展了的 PE 文件,其中应包含 IL 数据和元数据(metadata)。运行的时候通过即时编译器编译成可执行的代码。

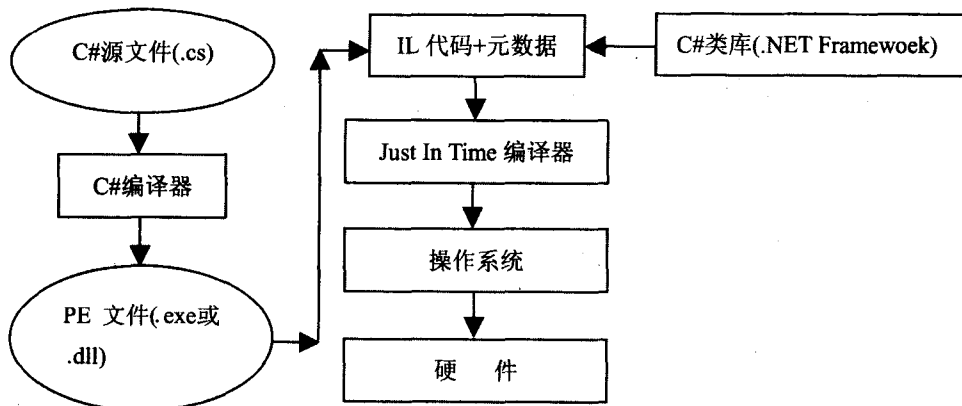


图 1-1 C#源程序的编译过程

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,在运行时 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#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#程序;或者可以使用 Microsoft Visual Studio .NET 7.0(以下简称 Visual Studio .NET),它提供了对 .NET 应用程序开发的全面支持。

C#是 Visual Studio .NET 中的一个组成部分。除了 C#, Visual Studio .NET 中还包括 Visual Basic、Visual C++、VB Script 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—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。但是如果耍开发 Windows 程序,那么就需要使用 Visual Studio .NET 这个强大的集成开发环境来调试和编译程序。

1.2 .NET SDK

1.2.1 .NET SDK 简介

如果不想安装 Visual Studio .NET,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。

微软公司允许下载和免费安装 .NET SDK,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 108 MB,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。

1.2.2 命令行编译

如果只是安装了 .NET SDK,那么就只能用命令行来进行编译,这里介绍常用的编译命令,其他的编译命令请查看 .NET 文档。

(1) 编译 File.cs 生成 File.exe:

```
csc File.cs
```

(2) 编译 File.cs 生成 File.dll:

```
csc /target:library File.cs
```

(3) 只编译 File.cs,但是不生成可执行文件(用来检查语法):

```
csc /nooutput File.cs
```

(4) 编译 File.cs,生成 My.exe:

```
csc/out:My.exe File.cs
```

(5) 通过引用 some.dll 编译 File.cs(适用于 File.cs 用到的名字空间在 some.dll 里面定义):

ese /r:some.dll File.cs

1.3 Visual Studio .NET 简介

与 .NET SDK 自带的命令行调试工具相比, Visual Studio .NET 的集成开发环境的功能要强大得多, 本节将先讲述安装 Visual Studio .NET 的系统要求和其包含的有关组件, 最后介绍其集成开发环境 IDE。

1.3.1 Visual Studio .NET 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 3GB。

显示器: 800 × 600, 256 色(推荐: 16 位彩色)。

1.3.2 Visual Studio .NET 组件

Visual Studio .NET 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 .NET 企业级开发工具包括:

- (1) 报表工具 Crystal Report;
- (2) 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;
- (3) Visual Studio SDK, 包括用来为程序开发在线帮助的 HTML Helper 1.3 SDK;
- (4) 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++ 程序的时候自动生成的一个数据库文件;
- (5) DIA SDK, 用来查询编译器产生的 PDB 文件中的信息;
- (6) Visual Studio Analyzer SDK, Visual Studio Analyzer 的开发包;
- (7) Windows Management Instrumentation SDK Tools, 是对 Visual Studio 的 ATL 向导的扩展;
- (8) 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, 并且具有良好的可缩放性;
- (9) Enterprise Samples, 企业版例子, 包含了两个电子商务程序;
- (10) Visual Studio Modler, 用来创建图表和模型;
- (11) Application Center Test for Visual Studio .NET, 用来对 Web 程序进行测试;
- (12) 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 Server。

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。

1.3.3 Visual Studio .NET 集成环境 IDE

Visual Studio .NET 是微软的集成开发环境(IDE),在该集成开发环境中,用户可以创建、归档、运行和调试使用各种 .NET 编程语言编写的程序。Visual Studio .NET 同时提供了操纵各种类型文件的编辑工具。Visual Studio .NET 是建立商业相关和任务相关应用程序的强大而复杂的工具。

1. Visual Studio .NET 集成开发环境 IDE 概览

当 Visual Studio .NET 第一次执行时,会显示起始页(如图 1-2 所示)。在这一页的左侧包含了一些有用的链接,用户可以单击其中一项的名字(例如“开始”)来浏览它的内容。把用鼠标左键点击一次称为选中或单击,把用鼠标左键快速点击两次称为双击(注意,由于使用的版本不同,Visual Studio .NET 所显示的方式可能会有一些细微的差别)。

图 1-2 显示了“开始”链接被选中时,起始页显示的内容。“开始”项包含了最近打开过的项目的链接。最近打开过的项目出现在列表中(如图 1-2 中的 WindowsApplication2),还显示了项目的修改日期。用户也可以选择【文件】菜单下的【最近的项目】来查看这些内容。当 Visual Studio .NET 第一次装载的时候,“开始”项将是空的。注意在页面上出现的两个按钮:“打开项目”和“新建项目”。按钮是突起的矩形区域,如果单击它,就可以执行一个操作。

下面简要描述一下图 1-2 中起始页的其他链接。“新增功能”项显示了 Visual Studio .NET 的新特性和更新的内容,其中包括代码示例和新的编程工具的下载。“网上社区”项包含了使用新闻组、Web 页和其他的在线资源同其他软件开发人员联系的方法。“标题新闻”项提供了一种浏览新闻、文章和使用向导的方法。使用“联机搜索”项可以浏览 MSDN (Microsoft Developer News,微软开发人员信息)在线库。MSDN 站点包括关于各种技术的大量文章、下载内容和教程。“下载”项允许用户得到更新内容以及代码示例。“XML Web services”项向程序员提供了关于网络服务的信息,网络服务是 Internet 上的可重用软件。“Web 宿主”项为想把自己的软件(如 Web 服务)在线共享的开发者提供了相关信息。“我的配置文件”项允许用户自定义 Visual Studio .NET,如设置键盘和窗口布局选项。用户也可以通过从【工具】菜单选择【自定义...】或【选项...】来配置 Visual Studio .NET。Visual Studio .NET 甚至可以浏览 Web, Internet Explorer 浏览器是 IDE 的一部分。要访问 Web 页,在地址栏中键入地址并按回车键。除起始页外,在 IDE 中还有其他的几个窗口,在下面的几个部分里将讨论这些窗口。

要建立新的 C#程序,单击起始页中的“新建项目”按钮,然后就会出现如图 1-3 所示的

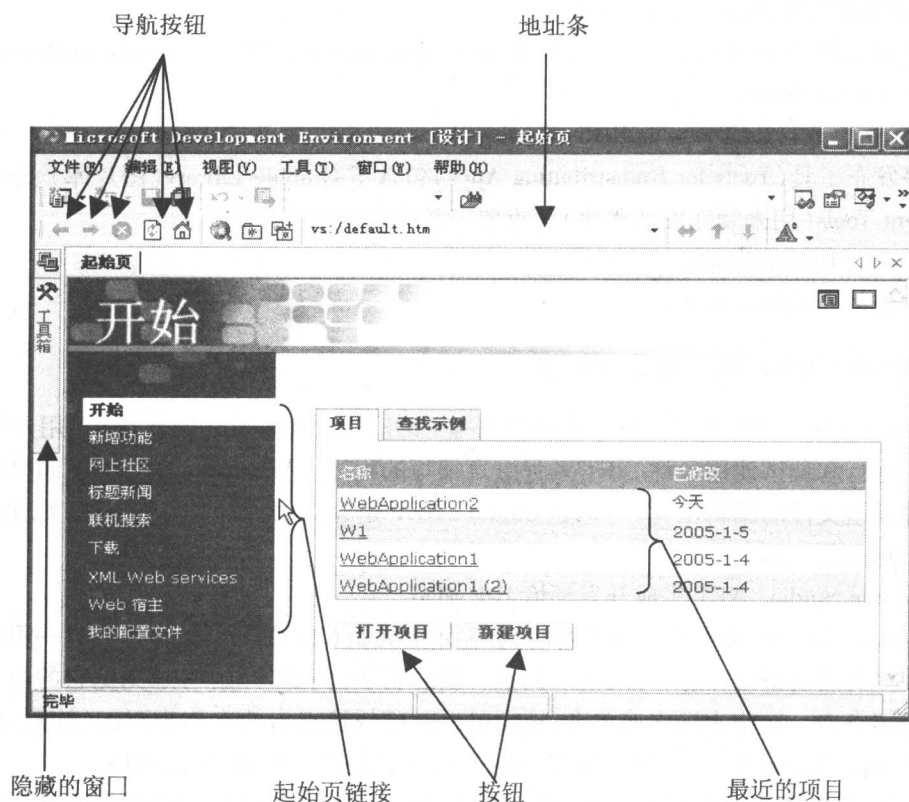


图 1-2 Visual Studio .NET 中的起始页

对话框。Visual Studio .NET 将这些程序组织到项目和解决方案中。项目是一组相关的文件,如 C#代码、图像和文档;解决方案是一组代表完整的应用程序的项目,或者说是一组相关的应用程序。每个解决方案中的每个项目可以执行不同的任务。

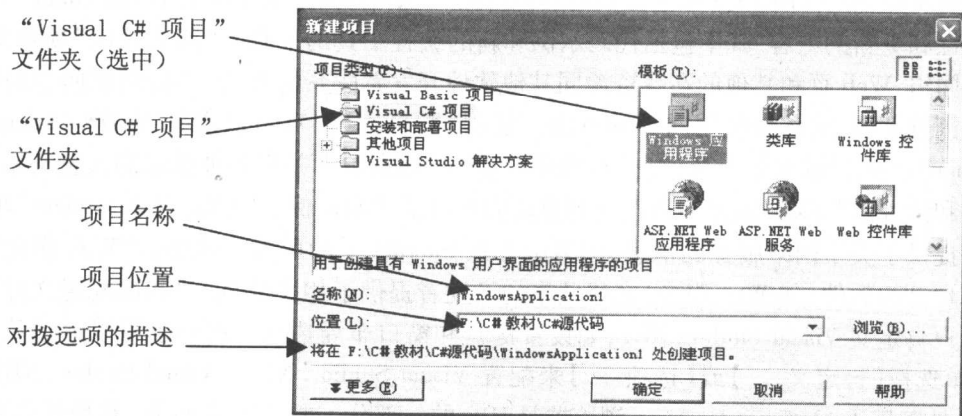


图 1-3 “新建项目”对话框