

21 世纪

计算机应用技术系列规划教材

C#网络应用 编程基础

◎ 马骏 主编 ◎



人民邮电出版社
POSTS & TELECOM PRESS

图书在版编目 (CIP) 数据

C#网络应用编程基础 / 马骏主编. —北京: 人民邮电出版社, 2006.10

(21 世纪计算机应用技术系列规划教材)

ISBN 7-115-14736-1

I. C... II. 马... III. C 语言—程序设计—教材 IV. TP312

中国版本图书馆 CIP 数据核字 (2006) 第 070096 号

内 容 提 要

本书主要介绍 C#网络应用编程的基础知识及相关的應用。内容包括 C#语言基本知识和面向对象编程基础、C# Windows 窗体客户端应用编程、C# Web 窗体应用编程以及在文件管理、Internet 应用、SQL Server 数据库、图形图像和水晶报表等方面的应用。全书语言简洁, 重点突出, 思路清晰, 实用性强。同时有与本书配套的实验指导、电子教案、例题与习题的源程序和全部习题参考解答。

与本书配套的教材是《C#网络应用高级编程》和《C#网络应用编程开发实例与实验指导》。配套教材在本书的基础上进一步介绍了 C#语言更高级的编程技术以及用实际开发项目讲解数据库应用、智能客户端和网站开发的详细设计、实现与发布技术, 同时还包含了与本书配套的上机实验指导。

本书可作为高等院校计算机及相关专业的教材, 也可作为初、中级程序员的参考资料。

21 世纪计算机应用技术系列规划教材

C#网络应用编程基础

◆ 主 编 马 骏

责任编辑 邹文波

◆ 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@ptpress.com.cn

网址 <http://www.ptpress.com.cn>

河北三河市海波印务有限公司印刷

新华书店总店北京发行所经销

◆ 开本: 787×1092 1/16

印张: 27.75

字数: 672 千字

印数: 1—3 000 册

2006 年 10 月第 1 版

2006 年 10 月河北第 1 次印刷

ISBN 7-115-14736-1/TP · 5381

定价: 36.00 元

读者服务热线: (010)67170985 印装质量热线: (010)67129223



在 Internet 和计算机技术高速发展的时代，处处都有网络的影子。在网络无处不在的环境下，任何一种开发都不可避免地涉及网络应用。网络办公、网络游戏、网络电视、电子商务、网络资源搜索、邮件处理、五彩缤纷的网站、杂志订阅、远程控制……以及其他各种网络相关的应用软件，都是网络编程的实际应用。早期的网络编程难度大、效率低，使初学者望而生畏，而 C#语言和.NET 平台大大地简化了这些技术，使过去困难的网络应用编程变得非常轻松。

目前，很多高校计算机及相关专业都开设了网络应用编程方面的课程。在就业压力日益增大以及学生对所学知识与社会实际需求紧密结合的迫切要求的环境下，本书涉及的内容和知识面显得尤为重要。

Microsoft Visual Studio 2005（简称 VS2005）是 Microsoft 公司最新推出的先进的可视化开发工具。2005 年底，Microsoft 公司发布了 VS2005 英文版，并于 2006 年 1 月发布了 VS2005 Professional 简体中文版。本书以 VS2005 Professional 简体中文版为开发环境，充分考虑了学以致用和社会实际需求，用大量的实例，引导读者快速掌握用 C#语言进行各类网络应用程序编程的方法和技巧。使学生学习本书内容后能立即编写出与企事业单位实际应用紧密结合的程序，避免说起来似懂非懂，做起来寸步难行的尴尬局面。

C#语言是在 C 和 C++语言基础上开发的在.NET 平台上运行的为适应 Internet、Intranet 和各类网络应用而设计的编程语言，该语言综合了 C、C++和 Java 以及其他高级语言的优点，是一种语法优雅、类型安全、完全面向对象的编程语言。由于 C#语言是专门为.NET 平台而设计的开发语言，并于 2001 年由 ECMA（European Computer Manufacturers Association，欧洲计算机制造商协会）规定为高级语言开发标准（ECMA-334），2003 年被 ISO（International Standards Organization，国际标准化组织）规定为国际标准（ISO/IEC 23270），因此它比任何在.NET 平台下提供的其他语言都有无可比拟的优越性。毫无疑问，随着.NET 技术的普及，C#语言必将成为开发 Internet 和企业级应用程序的首选程序设计语言。

本书的所有程序均在 Microsoft VS2005 Professional 简体中文版开发环境下调试通过，是一本非常实用的学习 C#网络应用编程的教材。全书共分 14 章。其中第 1~4 章介绍 C#语言基本知识和面向对象编程基础，第 5~10 章介绍窗体控件、文件管理、Internet 应用、数据库、图形图像以及组件设计等 Windows 窗体界面下的客户端应用编程，第 11~14 章介绍网页设

计、Web 服务器控件、网站开发应用设计和报表处理。

读者通过阅读、上机练习和调试运行，能快速理解用 C#语言进行实际应用编程的方法和技巧，为提高实际动手开发能力打下良好的基础。另外，由于本书的重点在于实际应用，对语言本身的语法介绍不是太多，因此需要读者有一定的 C++或 Java 等高级语言编程基础。

由于 C#语言以及 VS2005 开发工具包含的知识太多，一本教材不可能将其精髓全部展现出来，因此本书只是尽可能将最常用、最实用的各种基本应用编程方法介绍给读者，更复杂的高级编程技术和综合应用则在《C#网络应用高级编程》和《C#网络应用开发实例与实验指导》中介绍。

本书由马骏主编，并对全书内容进行了统稿、修改、整理和定稿。具体编写人员有马骏（第 1、7、8、9 章）、赵新强（第 4、10 章及 11.1、11.6 节）、毋琳（第 2、5、12 章及 11.3、11.5 节）、王玉璟（第 3、13 章及 11.2、11.4 节）和郭祖华（第 6、14 章）。参与编写和代码调试等工作的还有陈明、王芳、杨韶华、张瑞青、郑珂、金梁、李国雁、王鸿铭、臧国轻和朱二兵等。

为了配合教学需要，本书还提供配套的 PowerPoint 多媒体教学课件、全书所有例题和习题的源程序代码以及所有习题参考解答。读者可以到人民邮电出版社网站（<http://www.ptpress.com.cn>）下载，也可以与马骏联系（mj@henu.edu.cn）。

由于编者水平有限，书中难免存在错误之处，敬请读者批评指正。

编 者
2006 年

目 录



第 1 章 概述	1
1.1 VS2005 开发环境的功能和特点	2
1.2 Microsoft.NET Framework	3
1.2.1 公共语言运行库	4
1.2.2 .NET Framework 类库	4
1.3 C#应用程序的结构	5
1.3.1 命名空间	5
1.3.2 控制台应用程序	6
1.3.3 Windows 应用程序	7
1.3.4 ASP.NET 网站	8
1.3.5 C#程序一般格式举例	9
1.3.6 程序代码注释	11
1.4 断点设置与程序调试	12
1.5 本书使用的命名约定	12
习题	13
第 2 章 数据类型与流程控制	14
2.1 C#语言概述	14
2.1.1 C#语言的特点	14
2.1.2 C#语言与其他语言的比较	15
2.2 C#数据类型	16
2.2.1 值类型	19
2.2.2 引用类型	24
2.2.3 值类型之间的数据转换	28
2.2.4 装箱和拆箱	30
2.2.5 运算符与优先级	30

2.3 C#流程控制语句	31
2.3.1 条件分支语句	32
2.3.2 循环语句	36
2.3.3 跳转语句	41
2.3.4 循环与跳转综合举例	42
2.3.5 异常处理	44
习题	46
第3章 面向对象的编程基础	48
3.1 面向对象的程序设计	48
3.1.1 类的组织	48
3.1.2 构造函数	52
3.1.3 方法	53
3.1.4 属性	58
3.1.5 事件	60
3.2 常用类操作和数据处理	60
3.2.1 Convert 类	60
3.2.2 String 类	61
3.2.3 StringBuilder 类	67
3.2.4 DateTime 类和 TimeSpan 类	68
3.2.5 Math 类	70
习题	71
第4章 面向对象的高级编程	73
4.1 封装、继承与多态	73
4.1.1 封装性	73
4.1.2 继承	74
4.1.3 多态性	78
4.2 版本控制	83
4.3 接口	87
4.3.1 接口的声明与实现	88
4.3.2 显式方式实现接口	90
4.3.3 通过接口实现多继承	91
4.4 委托	93
4.5 序列化与反序列化	95
4.5.1 二进制序列化与反序列化	96
4.5.2 XML 和 SOAP 序列化与反序列化	99
4.6 泛型处理	102
4.6.1 泛型的定义	103

4.6.2 泛型的引用	103
4.6.3 常用的泛型集合	104
习题	106
第 5 章 Windows 窗体界面设计	107
5.1 鼠标事件与键盘事件	107
5.2 Windows 窗体基本控件	110
5.2.1 控件中常用的属性和方法	110
5.2.2 TextBox 控件	110
5.2.3 MaskedTextBox 控件	112
5.2.4 RichTextBox 控件	116
5.2.5 CheckBox 控件	120
5.2.6 CheckedListBox 控件	123
5.2.7 RadioButton 控件	125
5.2.8 ComboBox 控件	126
5.2.9 ListBox 控件	129
5.2.10 Timer 组件	130
5.2.11 PictureBox 控件	131
5.3 菜单与工具栏	134
5.3.1 MenuStrip 控件	134
5.3.2 ContextMenuStrip 控件	138
5.3.3 ToolStrip 控件	141
5.4 其他 Windows 窗体控件	143
5.4.1 ToolTip 组件	143
5.4.2 ProgressBar 控件	145
习题	146
第 6 章 目录与文件管理	148
6.1 目录管理	148
6.1.1 目录的创建、删除与移动	149
6.1.2 FolderBrowserDialog 控件	154
6.2 文件管理	157
6.2.1 文件复制、删除与移动	157
6.2.2 文件属性与设置	160
6.2.3 OpenFileDialog 控件	161
6.2.4 SaveFileDialog 控件	163
6.3 文件的读写	163
6.3.1 文件和流	163
6.3.2 文件的打开	164

6.3.3 文本文件的读写	164
6.3.4 二进制文件的读写	166
习题	167
第7章 正则表达式与 Internet 操作	168
7.1 正则表达式	168
7.1.1 正则表达式简介	168
7.1.2 Regex 类	171
7.1.3 Match 类	173
7.1.4 MatchCollection 类	174
7.1.5 Group 类	176
7.2 Windows 应用程序中嵌入 Web 浏览器	177
7.3 通过 Internet 搜索网络资源	180
习题	184
第8章 ADO.NET 与 SQL Server	186
8.1 创建数据库与数据库表	188
8.2 ADO.NET 数据访问对象	189
8.2.1 SqlConnection 对象	189
8.2.2 SqlCommand 对象	191
8.2.3 SqlDataAdapter 对象	195
8.2.4 DataTable 对象	197
8.2.5 DataSet 对象	199
8.3 数据绑定技术	201
8.3.1 DataSet 设计器与数据源	201
8.3.2 单个数据库表的绑定	203
8.3.3 关联数据库表的绑定	208
8.3.4 直接编写代码完成数据绑定	209
8.4 数据处理控件	213
8.4.1 BindingSource 组件	213
8.4.2 BindingNavigator 控件	223
8.4.3 DataGridView 控件	226
8.5 存储过程	239
习题	244
第9章 二维图形图像处理	245
9.1 GDI+概述	245
9.1.1 GDI+使用的坐标系	246

9.1.2 Graphics 类	248
9.1.3 颜色	249
9.1.4 画笔和画刷	249
9.2 绘制基本图形	256
9.2.1 直线	256
9.2.2 矩形	258
9.2.3 多边形	261
9.2.4 曲线	262
9.2.5 椭圆	265
9.2.6 扇形	266
9.3 图像处理	271
9.3.1 图像的显示与保存	271
9.3.2 图像的平移、旋转和缩放	273
9.3.3 图像的拉伸与反转变换	275
习题	278
第 10 章 基于组件的开发技术	279
10.1 组件和控件	279
10.2 组件的设计	280
10.2.1 组件设计的一般方法	281
10.2.2 在 Windows 应用程序中调用设计的组件	282
10.3 用户控件	283
10.4 自定义控件	286
10.4.1 设计 Windows 自定义控件	286
10.4.2 调用 Windows 自定义控件	289
10.5 扩展控件	290
10.5.1 设计扩展控件	290
10.5.2 调用扩展控件	293
习题	294
第 11 章 网站建设与网页制作	295
11.1 网站的创建复制与发布	295
11.1.1 网站的创建	295
11.1.2 复制网站	297
11.1.3 发布网站	298
11.2 HTML 和 XHTML	299
11.2.1 HTML 和 XHTML 的格式与标记	299
11.2.2 常用 HTML 和 XHTML 标记	304
11.3 页面定位与布局方法	305

11.3.1 表格.....	306
11.3.2 利用表格进行页面布局	310
11.3.3 页内框架.....	311
11.3.4 层的使用与布局方法	313
11.4 母版页.....	316
11.5 CSS 样式设计.....	319
11.5.1 样式与样式表	319
11.5.2 样式规则.....	323
11.5.3 样式生成器	328
11.5.4 CSS 单位	330
11.6 使用 JavaScript 创建网页特效.....	331
11.6.1 脚本语言.....	331
11.6.2 常用的特效	332
习题.....	341
第 12 章 ASP.NET Web 服务器控件.....	342
12.1 标准控件.....	342
12.1.1 简单控件	342
12.1.2 Table 控件.....	346
12.1.3 HiddenField 控件	349
12.2 数据操作控件	350
12.2.1 SqlDataSource 组件	350
12.2.2 GridView 控件.....	354
12.2.3 DataList 控件.....	357
12.2.4 DetailsView 控件.....	361
12.2.5 FormView 控件	362
12.3 验证控件.....	364
12.3.1 ValidationSummary 控件	364
12.3.2 RequiredFieldValidator 控件.....	364
12.3.3 RangeValidator 控件	366
12.3.4 CompareValidator 控件	367
12.3.5 RegularExpressionValidator 控件	370
12.3.6 CustomValidator 控件	371
12.4 导航控件.....	372
12.4.1 SiteMapPath 控件.....	373
12.4.2 Menu 控件	375
12.4.3 TreeView 控件.....	378
习题.....	383

第 13 章 ASP.NET Web 应用程序开发	384
13.1 Web 应用程序中的常用对象	384
13.1.1 Response 与 Request	384
13.1.2 Application 与 Session 对象	386
13.2 Global.asax 文件	386
13.2.1 Global.asax 中的常用事件	386
13.2.2 统计网站在线人数	387
13.3 网页的切换与数据传递	388
13.3.1 网页的切换	388
13.3.2 网页间的数据传递	389
13.4 在 Web.Config 中保存配置	392
13.4.1 设置 sessionState	392
13.4.2 保存数据库连接信息	393
13.4.3 用户身份验证与授权	394
13.5 利用 SQL Server 数据库实现用户身份验证	394
13.6 通过更新数据库保存新用户注册信息	398
13.7 统计网站登录次数	402
13.8 文章的浏览和搜索	403
13.9 文件的上传与下载	406
13.10 在线播放音频视频	407
13.11 发布即时消息	410
习题	411
第 14 章 水晶报表	413
14.1 水晶报表基础知识	413
14.1.1 水晶报表的分类	413
14.1.2 嵌入式水晶报表设计器	414
14.1.3 报表节	415
14.1.4 水晶报表对象模型	415
14.1.5 将 ADO.NET 数据集作为报表数据源	417
14.2 在 Windows 应用程序中使用水晶报表	417
14.3 在 ASP.NET 网站中使用水晶报表	427
习题	431

第 1 章

概述



Microsoft Visual Studio.NET 是 Microsoft 公司为适应 Internet 高速发展的需要，而推出的新的开发平台。2003 年，Microsoft 公司发布了 VS.NET 2003，提供了在 Windows 操作系统下开发各类基于 .NET 框架 1.1 的全新的应用程序开发平台；2005 年底，Microsoft 公司又发布了基于 .NET 框架 2.0 的 VS2005 开发平台，植入了适用于大型团队开发的各种优秀的复杂功能，并于 2006 年 1 月发布了 VS2005 Professional 简体中文版。

目前流行的各类网络应用编程技术中，开发人员使用的编程模式有：C/S (Client/Server) 模式、B/S (Browser/Server) 模式、Web 服务 (Web Service)，以及其他传统 C/S 结构的变形，例如，P2P (Peer to Peer)、智能客户端等。

20 世纪 80 年代和 90 年代，C/S 模式得到了广泛应用。近几年来，由于网络技术的迅速发展，很多应用系统的体系结构从传统的 C/S 模式向更加灵活的多极分布式结构演变，使得软件系统的网络体系结构再次跨入一个新阶段。

与快速发展的 Internet 相对应，B/S 模式的编程应用也更加广泛，功能也越来越强。当然 B/S 模式实际上只是 C/S 模式的其中一种表现形式，只不过它的客户端是我们比较熟悉的几种固定的浏览器软件。为了区别于传统的 C/S 模式，才有意将其称为 B/S 模式。

从开发角度看，C/S 模式和 B/S 模式各有优缺点，适用的环境也不相同。

首先，在系统的性能方面，只要拥有可上网的浏览器，就可以使用 B/S 模式的系统。不过，B/S 模式的客户端只能完成浏览、查询和数据输入等简单功能，绝大部分工作要由服务器承担，这就对服务器提出了很高的要求，因此无形中增加了用户在这一方面的投入。

其次，在系统升级方面，C/S 模式中若有某一模块发生改变，可能要关联到其他模块的变动，使系统升级的成本增大；而 B/S 模式在开发、维护阶段几乎所有的工作都集中在服务器端，只需更新服务器端的软件就可以了。如果系统升级比较频繁，那么 B/S 模式的产品就具有维护工作量少的优势。

但是，在安全性上，B/S 模式则略显不足，毕竟通过网络传输的数据安全系数并不高，例如，只要拥有合法的用户名和密码，任何人都可以进入到用户的系统中；而 C/S 模式由于需要特定的客户端软件，甚至可以限定只有某一台计算机才可以使用这个客户端软件，因而对安全性有更多的保障。

为了解决传统 C/S 模式在系统升级方面和模块开发方面的不足, 随后又发展了一种新的开发模式, 即 Web 服务, Web 服务的思想实际上就是将各种功能做成组件或控件的形式, 然后直接将这些组件或控件安装在服务器上。这样, 不论是 C/S 模式还是 B/S 模式的应用程序, 都可以通过 Internet 直接调用提供的服务。采用这种形式, 一方面解决了模块不完善引起的功能不足; 另一方面也解决了升级困难的问题。

在网络应用编程技术中, 除了将数据库、功能实现与功能界面分离, 并有专用的数据库服务器、应用程序服务器和 Web 服务器等专用服务外, 还有一种用于不同计算机用户之间, 不经过中继设备直接交换数据或服务的技术, 即 P2P (Peer-to-Peer), 称之为对等互连或点对点技术。利用 P2P 技术可以让 Internet 用户更充分地利用网络资源, 每个人都可以通过 Internet 直接连接到其他用户的计算机, 并进行文件或数据的交换, 而不需要经过专用服务器。由于这种技术消除了中间环节, 因此使网络数据的传输变得更快捷、更直接。

可见, 网络应用开发的模式多种多样, 编程方法也不尽相同。但是, 从另一个角度来看, 不论使用哪种模式, 从外观上来看主要可以分为以下几类。

(1) Windows 应用程序

Windows 应用程序实现 C/S 模式、P2P、智能客户端等客户端和服务端的功能与用户操作界面, 用于交互性操作较多的场合, 例如, 大型网络游戏、复杂的网络办公软件、大量网络信息传递以及其他高端的网络开发设计等。

(2) Web 应用程序

Web 应用程序也叫网站, 实现 B/S 模式的功能和简单客户端用户操作页面, 用于通过 Internet 传递页面、交互性操作不多的场合, 例如, 目前流行的各类网站以及功能比较简单的基于 B/S 模式的网络办公管理等。

(3) 控制台应用程序

控制台应用程序使用统一的重定向文本输出形式, 用于交互性操作非常少、主要偏重于内部功能实现的场合。

(4) 组件与 Web 服务

组件与 Web 服务实现模块级的功能。Web 服务主要用于在服务器端通过 Internet 提供给 Windows 应用程序和 Web 应用程序调用的功能模块。这些模块既可以提供给服务器端应用程序调用, 也可以提供给客户端应用程序调用。

在实际应用中, 可以根据具体的客户要求和环境选择合适的应用程序类型, 也可能在一个解决方案中同时包含不同应用程序类型的项目, 即将几种模式综合使用。

1.1 VS2005 开发环境的功能和特点

Microsoft 公司推出的 Visual Studio 2005 (简称 VS2005) 提供了一个全新的开发环境, 在此环境下, 可以用多种语言开发 Windows 窗体应用程序、ASP.NET 网站、移动设备应用程序、组件与控件、数据报表和 XML Web 服务等多种复杂的系统。

VS2005 分为 Team System 版本、Professional 版本、Standard 版本和 Express 版本, 分别适用于大型软件开发团体、中小型软件开发公司以及业余开发者, 是一种全面的、先进的、

完全适应当前网络发展的高级软件开发平台。

C#语言是在 C、C++和 Java 语言基础上重新构造的、语法与 C++和 Java 语言都比较相似的、基于.NET 框架的，一种完全面向对象的、类型安全的编程语言，也是.NET 的首选编程语言。从开发效率来讲，C#语言为应用程序开发人员提供了快速的开发手段，但又不牺牲 C++语言的特点和优点。从继承性来讲，C#语言在更高层次上重新实现了 C 和 C++语言。从语法形式和易用性来讲，C#语言几乎综合了目前流行的所有高级语言的优点，提供了一种语法优雅、功能完善而又容易使用的外在表现形式。

表 1-1 描述了基于.NET 框架的一些编程工具和编程特点。

表 1-1 基于.NET 框架的编程工具及特点

功 能	描 述
Windows 窗体设计器	提供图形化设计界面，通过对窗体中的控件进行拖放，可以快速创建应用程序的用户界面
Windows 窗体工具	提供了 Windows 窗体设计器、Windows 应用程序模板、基本的项目引用和初始代码，以帮助用户创建标准 Windows 窗体应用程序
Web 窗体工具	提供了 Web 窗体设计器、ASP .NET Web 应用程序模板、基本的项目引用和初始代码，以帮助用户创建以浏览器作为主界面的 Web 窗体应用程序
XML Web Services 工具	提供了一个 ASP .NET Web Service 模板，可以用来构成 Web 应用程序的基本结构。此时，Web 应用程序的基本架构将构建在 Web 服务器和本地解决方案文件之上
多语言支持	开发环境集成了所有的.NET 平台编程语言，包括 Visual C#语言
数据访问	包括用于创建数据库应用程序的组件、可视化数据库工具以及一个可靠的 ADO.NET 类集，使用户可以方便地操作所有类型的数据
错误处理	包括支持跨语言调试的调试工具以及结构化异常类
向导	可以用来快速创建复杂的通用任务

1.2 Microsoft.NET Framework

Microsoft.NET Framework 是支持生成与运行下一代应用程序和 XML Web services 的内部 Windows 组件，其主要目标如下。

- 提供一个一致的面向对象的编程环境，而无论对象代码是在本地存储和执行，还是在本地执行但在 Internet 上分布，或者是在远程执行的。
- 提供一个将软件部署和版本控制冲突最小化的代码执行环境。
- 提供一个可提高代码（包括由未知的或不完全受信任的第三方创建的代码）执行安全性的代码执行环境。
- 提供一个可消除脚本环境或解释环境的性能问题的代码执行环境。
- 使开发人员的经验在面对类型大不相同的应用程序（如基于 Windows 的应用程序和基于 Web 的应用程序）时保持一致。
- 按照工业标准生成所有通信，以确保基于.NET Framework 的代码可与任何其他代码集成。

Visual Studio.NET 开发环境中各部分的组成和相互关系如图 1-1 所示。

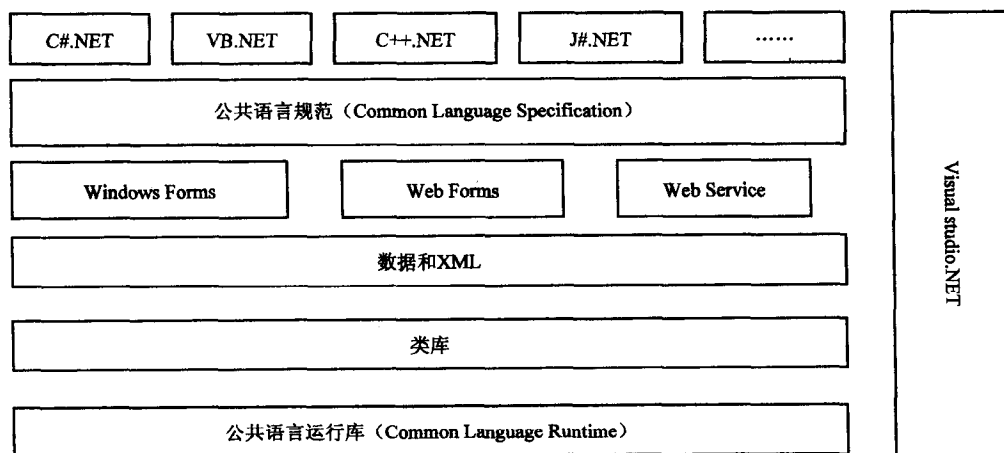


图 1-1 Visual Studio.NET 开发环境

❖ 注意：使用 .NET Framework 提供的类库开发的应用程序，必须在安装了 .NET Framework 的计算机上才能运行。这种架构与 Java 语言必须由 Java 虚拟机支持相似。

1.2.1 公共语言运行库

公共语言运行库 (Common Language Runtime, CLR) 是 .NET 框架的基础，可被看做是一个在执行时管理代码的代理，提供诸如内存管理、线程管理和远程处理等核心服务，而且还强制实施严格的类型安全以及确保安全性和可靠性的其他形式的代码的准确性。事实上，代码管理是实施运行库的基本原则。以运行库为目标的代码称为托管代码；不以运行库为目标的代码称为非托管代码。

凡是使用符合公共语言规范的程序语言开发的程序，均可以在任何安装有 CLR 的操作系统中执行。CLR 可以大幅度简化应用程序的开发，同时由于代码在托管模式下运行，有效地避免了内存泄漏等情况。使用 .NET 提供的编译器可以直接将源程序编译为 .EXE 或者 .DLL 文件，但是需要注意的是，此时编译出来的程序代码并不是 CPU 能直接执行的机器代码，而是一种中间语言 (Intermediate Language, IL) 代码，在代码被调用执行时，CLR 的 Class Loader 会将需要的 IL 代码装入内存，然后再通过即时 (Just-In-Time) 编译方式将其临时编译成所用平台的 CPU 可直接执行的机器代码。

1.2.2 .NET Framework 类库

.NET Framework 类库是一个与 CLR 紧密集成的可重用的类型集合。该库提供对系统功能的访问，是建立 .NET 框架应用程序以及组件和控件的基础。

类库提供了 Internet 和企业级开发所需要的各种功能，完全支持 Web 标准及其应用，而且使用简单，扩充方便。

所有类库均分配到不同的命名空间下。

1.3 C#应用程序的结构

在 C#语言中, 每一个应用程序可以由一个或者多个类组成, 所有的程序都必须封装在某个类中。一个应用程序可以由一个文件组成, 也可以由多个文件组成, 文件名可以和类名相同, 也可以不同。C#源文件的扩展名为.cs, 如 `WindowsApplication1.cs`, 一个源文件中可以有一个类, 也可以有多个类; 如果加上 `partial` 修饰符, 也可以把一个类分散保存到多个文件中, 然后由编译器在编译时自动将其组合到一起。

下面是一个最简单的 C#语言应用程序的例子。

```
using System;
class HelloWorld
{
    public static int Main()
    {
        Console.WriteLine("Hello World");
    }
}
```

每一个应用程序都应该有一个入口点, 表明该程序从哪里开始执行。为了让系统能找到入口点, 入口方法名规定为 `Main`, 注意: `Main` 的首字母大写, 而且后面的小括号不能省略。`Main` 方法声明为 `public static`, 除非有特殊理由, 一般不要更改 `Main` 方法的声明。

类中的每一个方法都要有一个返回值, 对于没有返回值的方法, 必须声明返回值为 `void`。但是要注意, `Main` 方法的返回值只能是 `void` 或者 `int`。

C#语言和 C++、Java 语言一样, 也区分大小写。

1.3.1 命名空间

Microsoft.NET Framework 提供了一千多个类, 用于完成各种各样的功能。根据类的功能不同, 又把这些类划分到不同的命名空间中。命名空间包含可在程序中使用的类、结构、枚举、委托和接口, 一个命名空间又可以包含其他的命名空间。这种划分方法有些类似于文件的保存方式: 把文件分类放到不同的文件夹下, 而且不同文件夹下的文件名可以相同也可以不同。同样道理, 不同命名空间下的类名可以相同也可以不同。注意: 命名空间只是一种逻辑上的划分, 而不是物理上的存储分类。

如果要调用命名空间下某个类提供的方法, 可以使用下面的语法:

命名空间. 命名空间……命名空间. 类名称. 静态方法名(参数, ……);

或者

命名空间. 命名空间……命名空间. 实例名称. 方法名(参数, ……);

例如:

```
System.Console.WriteLine ("Hello World");
```

这条语句使用的命名空间为 `System`。在 `System` 命名空间下, 有一个 `Console` 类, 该类提供了静态的 `WriteLine` 方法, 此语句调用此方法输出字符串 “Hello World”。

显然，每句都加上命名空间太繁琐了。为了快速引用需要的功能，一般在程序的开头引用命名空间来简化代码书写形式。例如上面的语句，前缀“System.”表示 Console 类在 System 命名空间下，如果在程序的开头写上：

```
using System;
```

那么，语句中就不需要加上命名空间前缀了，而可以直接写为

```
Console.WriteLine("Hello World");
```

表 1-2 列出了 .NET 框架中常用的命名空间。

表 1-2 .NET 框架类中常用的命名空间

命名空间	类的描述
System	定义通常使用的数据类型和数据转换的基本 .NET 类
System.Collections	定义列表、队列和位数组组合字符串表
System.Data	定义 ADO.NET 数据库结构
System.Drawing	提供对基本图形功能的访问
System.IO	允许读写数据流和文件
System.Net	提供对 Windows 网络功能的访问
System.Net.Sockets	提供对 Windows 套接字的访问
System.Runtime.Remoting	提供对 Windows 分布式计算平台的访问
System.Security	提供对 CLR 安全许可系统的访问
System.Text	ASCII、Unicode、UTF-7 和 UTF-8 字符编码处理
System.Threading	多线程编程
System.Timers	在指定的时间间隔引发一个事件
System.Web	浏览器和 Web 服务器功能
System.Windows.Forms	创建使用标准 Windows 图形接口的基于 Windows 的应用程序
System.XML	提供对处理 XML 文档的支持

1.3.2 控制台应用程序

控制台应用程序也叫 Console 应用程序，用于在命令行方式下运行，适用于对界面设计要求不高、偏重于快速实现程序逻辑功能的场合。

在 VS2005 开发环境下，也可以直接按<F5>键编译并运行 Console 应用程序，但是运行时，读者会发现屏幕一闪就过去了，无法看清输出的内容。为了能观察到输出结果，可以在 Main 方法的最后加上 Console.ReadLine(); 语句，意思是读取键盘输入的字符串直到遇到回车键为止。观察输出结果后，按一下回车键就又返回到开发环境下了。

本书所举的所有控制台应用程序的例子，都可以在程序的最后加上 Console.ReadLine(); 语句，并在 VS2005 开发环境下直接按<F5>键编译并运行。

【例 1-1】编写一个控制台应用程序，显示“Hello World”。

(1) 进入 VS2005 开发环境，选择【文件】→【新建】命令，在弹出的窗体中，选择【控制台应用程序】模板，输入项目名 HelloWorld，如图 1-2 所示。