

新
版

21世纪

高职高专系列教材

Visual C# 2005 程序设计教程

◎崔 淼 陈明非 主编

◆ 提供电子教案增值服务



机械工业出版社
CHINA MACHINE PRESS



21 世纪高职高专系列教材

Visual C# 2005 程序设计教程

崔 淼 陈明非 主编

图 中 图 书 在 册 编 号 (CIP) 登 录

Visual C# 2005 程序设计教程 / 崔淼, 陈明非主编. — 北京: 机械工业出版社, 2007.1

(21 世纪高职高专系列教材)

ISBN 978-7-111-20710-8

I. V... Ⅱ. 崔... Ⅲ. C 语言—教材—高职高专—教材 Ⅳ. TP312.1

中国版本图书馆 CIP 数据核字 (2007) 第 001399 号

机械工业出版社 (北京市百万庄大街 22 号 邮政编码 100037)

责任编辑: 郝永升

责任印制: 郝永升

三河市国英印务有限公司

2007 年 1 月第 1 版·第 1 次印刷

184 mm × 260 mm · 16.75 印张 · 410 千字

0001—2000 册

定价: 24.00 元



机械工业出版社

本书以 Visual Studio 2005 为程序设计环境,采用案例方式,全面细致地介绍了 C#可视化面向对象编程的概念和方法。本书以学习者没有任何程序设计知识背景为立足点,以 Windows 应用程序设计为主线,以学以致用为主导,力求学习者通过本教材的学习,能较快地具备开发应用程序的基本能力,为进一步深入学习编程打下良好的基础。本书每章均配有一定数量的习题,以方便学生练习。

为配合教学,本书的配套教材《Visual C# 2005 程序设计教程实训与习题解答》对本书中的习题做了详细解答,同时还为每章增加了实训实验。

本书为高职高专学校教材,也可作为初学编程人员的自学教程。

图书在版编目(CIP)数据

Visual C # 2005程序设计教程/崔森,陈明非主编. —北京:机械工业出版社, 2007.1

(21 世纪高职高专系列教材)

ISBN 978-7-111-20710-8

I. V... II. ①崔... ②陈... III. C 语言—程序设计—高等学校:技术学校—教材 IV. TP312

中国版本图书馆CIP数据核字(2007)第001966号

机械工业出版社(北京市百万庄大街 22 号 邮政编码 100037)

责任编辑:赵丽欣

责任印制:洪汉军

三河市国英印务有限公司

2007年1月第1版·第1次印刷

184 mm × 260 mm · 16.75 印张 · 410 千字

0001—5000 册

定价:24.00 元

凡购本书,如有缺页,倒页,脱页,由本社发行部调换
销售服务热线电话:(010) 68326294

购书热线电话:(010) 88379639 88379641 88379643

编辑热线电话:(010) 88379739

封面无防伪标均为盗版

21 世纪高职高专计算机专业系列教材

编委会成员名单

主 任 周智文

副主任 周岳山 林 东 王协瑞 赵佩华

程时兴 吕何新 陈付贵 朱连庆 陶书中

委 员 (按姓氏笔画排序)

马 伟 马林艺 卫振林 于恩普

王养森 王 泰 王德年 刘瑞新

余先锋 陈丽敏 汪赵强 姜国忠

赵国玲 赵增敏 顾可民 贾永江

顾 伟 陶 洪 龚小勇 眭碧霞

曹 毅 鲁 辉 翟社平

秘书长 胡毓坚

出版说明

根据《教育部关于以就业为导向深化高等职业教育改革的若干意见》中提出的高等职业院校必须把培养学生动手能力、实践能力和可持续发展能力放在突出的地位,促进学生技能的培养,以及教材内容要紧密结合生产实际,并注意及时跟踪先进技术的发展等指导精神,机械工业出版社组织全国 40 余所院校的骨干教师对在 2001 年出版的“面向 21 世纪高职高专系列教材”进行了修订,现丛书名改为“21 世纪高职高专系列教材”。

在几年的教学实践中,本系列教材获得了较高的评价。因此,在修订过程中,各编委会保持了第 1 版教材“定位准确、注重能力、内容创新、结构合理和叙述通俗”的编写特色。同时,针对教育部提出的高等职业教育的学制将由三年逐步过渡为两年,以及强调以能力培养为主的精神,制定出了本次教材修订的原则:跟上我国信息产业飞速发展的节拍,适应信息行业相关岗位群对第一线技术应用型操作人员能力的要求,针对两年制兼顾三年制,理论以“必须、够用”为原则,增加实训的比重,并且制作了内容丰富而且实用的电子教案,实现了教材的立体化。

针对课程的不同性质,修订过程中采取了不同的处理办法。核心基础课的教材在保持扎实的理论基础的同时,增加实训和习题;实践性较强的课程强调理论与实训紧密结合;涉及实用技术的课程则在教材中引入了最新的知识、技术、工艺和方法。此外,在修订过程中,还进行了将几门课程整合在一起的尝试。所有这些都充分地体现了修订版教材求真务实、循序渐进和勇于创新的精神。在修订现有教材的同时,为了顺应高职高专教学改革的不深入,以及新技术新工艺的不断涌现和发展,机械工业出版社及教材编委会在对高职高专院校的专业设置和课程设置进行了深入的研究后,还准备出版一批适应社会发展的急需教材。

信息技术以前所未有的速度飞快地向前发展,信息技术已经成为经济发展的关键手段,作为与之相关的教材要抓住发展的机遇,找准自身的定位,形成鲜明的特色,夯实人才培养的基础。为此,担任本系列教材修订任务的教师,将努力把最新的教学实践经验融于教材的编写之中,并以可贵的探索精神推进本系列教材的更新。由于高职高专教育正在不断的发展中,加之我们的水平和经验有限,在教材的编审中难免出现问题和错误,恳请使用这套教材的师生提出宝贵的意见和建议,以利我们今后不断改进,为我国的高职高专教育事业作出积极的贡献。

机械工业出版社

前 言

Visual C#是微软公司推出的 Visual Studio 2005 开发平台中面向对象的编程语言。利用这种面向对象的、可视化的编程技术,结合事件驱动的设计,将使程序设计变得轻松快捷,使程序员不必再像早期面向过程的程序设计那样把大量的精力投入到程序设计的技术细节上,而可以将主要精力集中在解决实际问题与设计友好界面上。因此,它在各个领域都得到了广泛的应用。

本书结合大量具体实例,面向毫无编程基础的读者,从零开始逐步讲解 Visual C#程序设计的整个过程。读者在学习 Visual C#前不需要先学习 C、C++、VC、VB 等编程语言。这样,一方面节省了教学课时,另一方面可以培养学生从一开始就正确地建立起面向对象和可视化的编程概念。本书在叙述上尽量避免使用难懂的术语,以深入浅出的语言结合直观的图示、示例进行讲解,使读者能够较轻松地理解面向对象编程的基本概念与思想,于不知不觉之中掌握 Visual C#编程的方法和技巧。

本书采用“任务驱动”方式讲解例题,即先给出设计目标,然后介绍为实现该目标而采取的设计方法。为初学者考虑,程序设计中的操作以详尽的表述结合图例来说明,以求读者对每一步操作清清楚楚;代码设计尽量随时给予注释,以求读者对每一行代码的意义及其前后联系明明白白。针对初学者的特点,本书在编排上注意做到简明扼要、由浅入深和循序渐进,力求通俗易懂、简捷实用。本书概念清晰、逻辑性强、层次分明、例题丰富,符合教师教学和学生的学习习惯。本书在编写的主导思想上突出一个“用”字,紧紧抓住培养学生基本编程技能这个纲,以求达到学以致用目的。

作者讲授程序设计语言多年,有丰富的程序设计教学经验,主要是站在程序设计的层面上,而不是某种语言上。本书以可视化程序设计为主线,以相关的 C#控件为辅助,这样通过对本书的学习,学生不但可以学会程序设计的基本知识、设计思想和方法,还可学会可视化程序设计的通用方法与步骤,并可以较容易地过渡到其他语言(如 VB.NET、Delphi、Java 等)的学习与使用。

为方便学生上机练习和教师组织编程训练,我们编写了本书的配套教材《Visual C# 2005 程序设计教程实训与习题解答》,对本书中的习题做了详细解答,同时还为每章增加了实训实验。

完成本书教学的课时数约为 70 学时,其中授课为 36 学时,上机练习为 34 学时。

为了便于教师教学,本书的教学课件和例题源程序代码,可从机械工业出版社的网站 <http://www.cmpbook.com> 下载。

本书由崔淼、陈明非主编,参加编写的还有秦国防、刘瑞新、宋海军、宋鑫、吕振雷、刘笑迎、彭静静、郭晓燕、李艳静、靳小翠、郭金良、胡晴、臧顺娟、彭守军、李智、刘大明。

由于作者水平有限,书中疏漏和不足之处难免,敬请广大读者指正。

本书编写组 编 者

目 录

出版说明	2.4.2 运算符的优先级与结合性	33
前言	2.5 C#常用方法与属性	34
第1章 Visual C#概述	2.5.1 日期时间类常用方法与属性	34
1.1 C#集成开发环境	2.5.2 常用数学方法与字段	35
1.1.1 起始页	2.5.3 字符串的方法与属性	36
1.1.2 设计器视图与代码视图	2.5.4 类型转换方法	36
1.1.3 解决方案资源管理器	2.5.5 随机方法	37
1.1.4 属性	2.6 习题	37
1.1.5 工具箱	第3章 顺序结构程序设计	39
1.1.6 子窗口的操作	3.1 赋值语句	39
1.2 使用 Visual Studio 的帮助	3.1.1 单赋值语句	39
1.2.1 动态帮助	3.1.2 复合赋值语句与连续赋值 语句	40
1.2.2 智能感知	3.2 输入与输出	40
1.2.3 MSDN 帮助	3.2.1 焦点与 Tab 键序	41
1.3 对象的属性、事件和方法	3.2.2 使用标签与文本框实现输入与 输出	41
1.4 窗体对象与控件对象	3.2.3 使用链接标签	43
1.4.1 窗体对象	3.3 使用图片框与图片列表控件	45
1.4.2 几种常用控件	3.3.1 使用图片框控件	46
1.5 应用程序开发的一般步骤	3.3.2 使用图片列表控件	49
1.6 习题	3.4 习题	52
第2章 C#语法基础	第4章 选择结构程序设计	54
2.1 C#程序的代码结构	4.1 单分支选择结构	54
2.1.1 C#程序代码的组成要素	4.1.1 if...else 语句	54
2.1.2 C#程序的格式	4.1.2 if...else if 语句	58
2.2 基本数据类型	4.2 多分支选择结构	60
2.2.1 数值类型	4.3 使用单选按钮与复选框控件	63
2.2.2 字符类型	4.3.1 单选按钮	63
2.2.3 布尔类型和对象类型	4.3.2 复选框	64
2.3 变量与常量	4.4 使用分组框控件	66
2.3.1 变量	4.5 习题	66
2.3.2 常量	第5章 循环程序结构设计	69
2.3.3 类型转换	5.1 循环语句	69
2.4 运算符与表达式		
2.4.1 运算符与表达式类型		

5.1.1 for 循环语句	69	7.1.2 定义派生类	110
5.1.2 while 循环语句	70	7.1.3 密封类	114
5.1.3 do...while 循环	72	7.2 多态性	115
5.2 循环的嵌套与转向语句	73	7.2.1 概述	115
5.2.1 循环的嵌套	73	7.2.2 虚方法	116
5.2.2 转向语句	75	7.2.3 抽象类与抽象方法	118
5.3 列表框和组合框	76	7.3 习题	123
5.3.1 列表框	76	第 8 章 数组与自定义类型	125
5.3.2 组合框	79	8.1 数组概述	125
5.4 使用定时器控件	81	8.1.1 数组与数组元素	125
5.4.1 定时器控件的属性、方法和事件	81	8.1.2 数组的类型	126
5.4.2 定时器控件应用实例	81	8.2 声明与访问数组	126
5.5 习题	82	8.2.1 声明与访问一维数组	126
第 6 章 面向对象编程技术	86	8.2.2 声明多维数组	131
6.1 类和对象概述	86	8.2.3 声明数组列表	134
6.1.1 类和对象的概念	86	8.2.4 声明控件数组	135
6.1.2 类与对象的本质	86	8.3 数组参数	137
6.1.3 类成员	87	8.3.1 整个数组为参数	137
6.2 类的字段与属性	87	8.3.2 params 关键字	139
6.2.1 类定义	88	8.4 数组与数组列表的属性与方法	139
6.2.2 声明与使用对象	89	8.5 自定义数据类型	143
6.2.3 值类型与引用类型	91	8.5.1 结构类型	143
6.3 类的方法	92	8.5.2 枚举类型	144
6.3.1 声明与调用方法	92	8.6 习题	147
6.3.2 参数传递	95	第 9 章 菜单、工具栏、状态栏与对话框	150
6.3.3 重载方法	98	9.1 菜单	150
6.4 类的构造函数	100	9.1.1 主菜单的设计	150
6.4.1 声明构造函数	100	9.1.2 快捷菜单	154
6.4.2 重载构造函数	102	9.2 工具栏	157
6.5 静态成员	104	9.2.1 工具栏控件	157
6.5.1 静态数据成员	104	9.2.2 工具栏控件应用实例	159
6.5.2 静态方法	104	9.3 状态栏	160
6.5.3 C# 中类型转换的常用静态方法	106	9.3.1 状态栏控件	160
6.6 习题	107	9.3.2 状态栏控件应用实例	161
第 7 章 继承	109	9.4 对话框	162
7.1 类的继承性	109	9.4.1 模式对话框和非模式对话框	162
7.1.1 类的继承	109	9.4.2 消息框	163

9.4.3 字体对话框	164	12.4.2 添加、删除、修改数据库 记录	207
9.4.4 打开文件对话框	165	12.4.3 多表查询应用实例	209
9.4.5 自定义对话框	167	12.5 习题	213
9.5 习题	168	第 13 章 创建 ASP.NET 网站	215
第 10 章 键盘和鼠标事件	170	13.1 ASP.NET 概述	215
10.1 键盘事件	170	13.2 创建一个简单的 ASP.NET 网站	216
10.1.1 KeyPress 事件	170	13.2.1 创建站点	216
10.1.2 KeyDown 和 KeyUp 事件	172	13.2.2 设计 Web 页面	217
10.2 鼠标事件	176	13.2.3 设置对象属性	217
10.2.1 鼠标事件发生的顺序	176	13.2.4 编写程序代码	217
10.2.2 MouseDown 和 MouseUp 事件	177	13.2.5 运行 ASP.NET 应用程序	218
10.2.3 MouseMove 事件	178	13.2.6 发布 ASP.NET 网站	219
10.3 习题	180	13.3 ASP.NET 的常用控件和 对象	223
第 11 章 ActiveX 控件	181	13.3.1 ASP.NET 的常用控件	223
11.1 概述	181	13.3.2 Page 对象	228
11.2 创建 ActiveX 控件	182	13.3.3 Response 对象	229
11.2.1 创建 ActiveX 控件的步骤	182	13.3.4 Request 对象	229
11.2.2 测试 ActiveX 控件	185	13.3.5 Application 对象和 Session 对象	230
11.2.3 将创建的 ActiveX 控件添加至 工具箱	186	13.3.6 Server 对象	231
11.3 使用 Flash 动画播放控件	187	13.4 访问数据库	234
11.3.1 Flash 动画播放控件的注册	187	13.4.1 使用 GridView 数据控件	234
11.3.2 Flash 动画播放控件的使用	188	13.4.2 使用 FormView 数据控件	237
11.4 习题	192	13.5 创建和使用 WebService	239
第 12 章 创建数据库应用程序	193	13.5.1 创建 WebService	239
12.1 数据库基础知识	193	13.5.2 在 ASP.NET 应用程序中调用 WebService	241
12.1.1 有关数据库的概念	193	13.5.3 在 Windows 应用程序中调用 WebService	242
12.1.2 关系型数据库	193	13.6 习题	245
12.2 使用数据库系统	195	第 14 章 异常处理与部署应用 程序	247
12.2.1 Microsoft Access	195	14.1 异常处理	247
12.2.2 Microsoft SQL Server	196	14.1.1 用 try...catch 语句捕获异常	247
12.3 SQL 查询基础	200	14.1.2 用 try...finally 语句清除 异常	249
12.3.1 查询语句 Select	200		
12.3.2 插入记录语句 Insert	202		
12.3.3 修改记录语句 Update	202		
12.3.4 删除记录语句 Delete	202		
12.4 访问数据库	203		
12.4.1 浏览数据库	203		

14.1.3	try...catch...finally 语句结构	250	14.2.1	生成安装包	253
14.1.4	抛出异常和常用异常类	251	14.2.2	编译安装项目	255
14.1.5	用户自定义异常	252	14.3	习题	256
14.2	部署 Windows 应用程序	253			

第1章 Visual C#概述

Visual C#（简称 C#）是美国微软公司推出的 Windows 操作系统下的应用程序开发平台 Visual Studio.NET（简称 VS.NET）家族中的一员，C#是对 C++进行了很大改进的一种完全面向对象的可视化程序设计语言。由于 C#采用了类似于 Visual Basic 的较易使用的程序设计界面，从而成为了一种更加简单易学、功能强大的应用程序开发工具。

1.1 C#集成开发环境

Visual Studio 可用于多种类型的程序开发，包括基于 Web 的应用程序、XML Web Service、基于 Windows 的应用程序和移动应用程序等。Visual Studio 提供了一系列核心程序设计语言：Visual Basic、Visual C++、Visual C#和 Visual J#。Visual Studio 允许这些编程语言共享工具，甚至可以创建由不同编程语言混合在一起的应用程序，从而结束了各种不同的编程语言只能各自使用独立的程序设计界面的局面。这些编程语言利用 .NET 框架（.NET Framework）的功能（.NET 框架提供了对简化应用程序开发的关键技术的访问）。

初次运行 Visual Studio 2005 将打开“选择默认环境设置”对话框，如图 1-1 所示。在“选择默认环境设置”列表框中选择“Visual C#开发设置”选项，单击【启动 Visual Sduio】按钮，经过配置后打开 Visual Studio 2005 界面。

如果在使用的过程中想改变默认环境设置，则可以执行“工具”菜单中的“导入和导出设置”命令，即打开“导入和导出设置向导”对话框，进行必要的设置，从而使集成开发环境更适合 C#程序的设计。

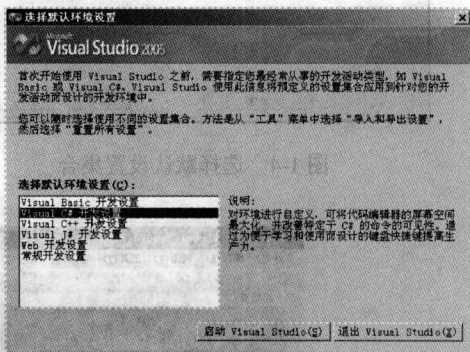


图 1-1 选择默认环境设置

在如图 1-2 所示的“欢迎使用‘导入和导出设置向导’”对话框中，选择“重置所有设置”单选项。单击【下一步】按钮，打开向导的“保存当前设置”对话框，选择“否，仅重置设置，从而改写我的当前设置”单选项，如图 1-3 所示。单击【下一步】按钮。

在打开的“选择一个默认设置集合”对话框中，在“要重置为哪个设置集合”列表框中，选择“Visual C#开发设置”选项，如图 1-4 所示。单击【完成】按钮。系统重置后，打开向导的“重置完成”对话框，如图 1-5 所示。单击【关闭】按钮，完成设置。

将默认配置设置为“Visual C#开发设置”，Visual Studio 2005 的主界面如图 1-6 所示。Visual Studio 环境有许多子窗口作为其组成部分。

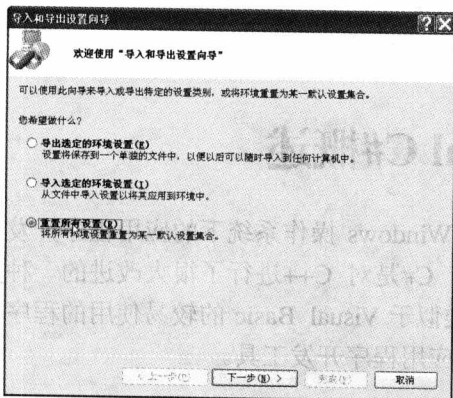


图 1-2 重置所有设置

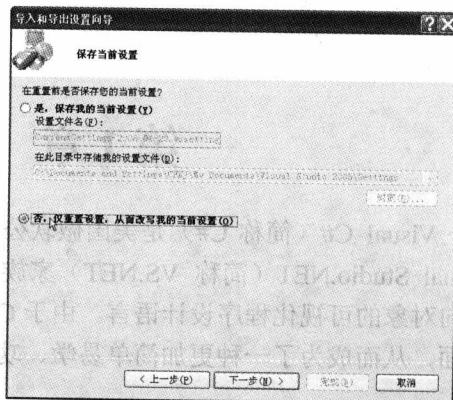


图 1-3 改变当前设置

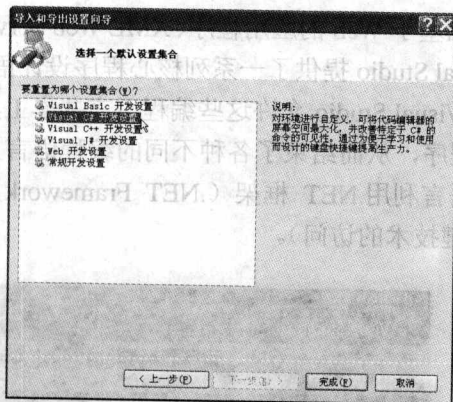


图 1-4 选择默认设置集合

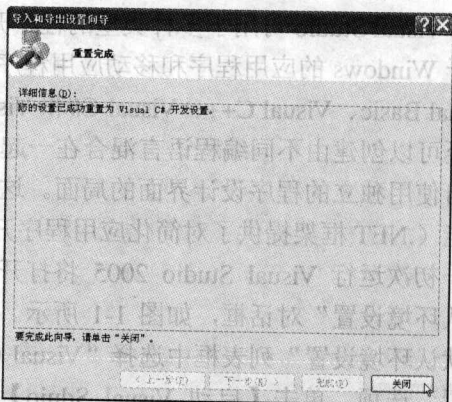


图 1-5 完成设置

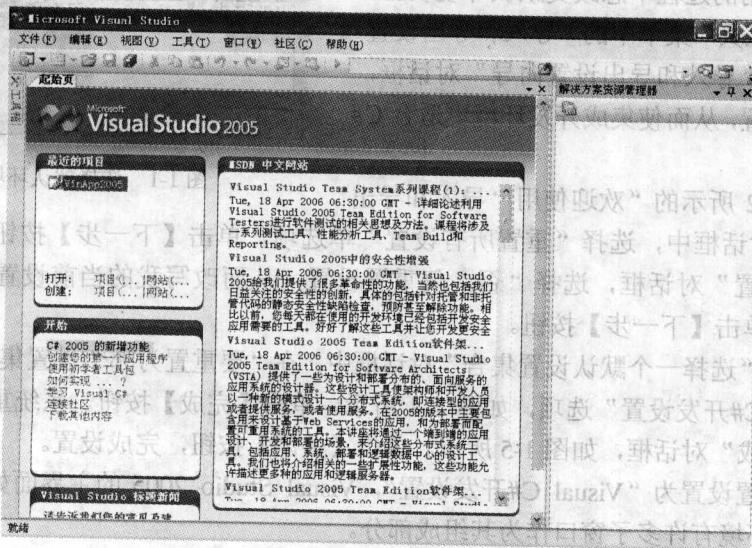


图 1-6 Visual Studio 2005 主窗口

1.1.1 起始页

在 Visual Studio 的主区域, 显示有起始页面窗口, 如果该窗口未显示, 可以选择菜单“视图”→“其他窗口”→“起始页”命令, 打开该窗口。该窗口有 4 个区域。

1. 最近的项目

在“最近的项目”区域列出了最近创建或打开过的项目列表。列表默认显示最近创建或打开的 6 个项目, 可以使用“工具”菜单中的“选项”命令打开“选项”对话框来设置显示的项目数, 最多可以显示 10 个最近创建或打开的项目。

在该区域下方有“打开”与“创建”两个选项, 其后均有“项目”与“网站”两个超链接, “项目”超链接用于打开已有项目(包括网站)或创建新的项目, “网站”超链接则专门用于打开或创建 Web 应用程序。

2. 联机资源

起始页的其他三个区域为联机资源超链接区域, 主要用于查找 Visual Studio 资源, 如果安装了 MSDN (Microsoft Developer Network) 并且已连接因特网, 则可以通过这三个区域查找本机和因特网上的 Visual Studio 资源。

例如, 单击“开始”区域的“C# 2005 的新增功能”超链接, 将打开 Visual Studio 的帮助窗口, 显示本机 MSDN 中的 Visual C# 2005 中的新增功能, 如图 1-7 所示。单击“开始”区域中的“连接社区”超链接, 将在帮助窗口打开因特网上 MSDN 的 Visual Studio 资源, 如图 1-8 所示。

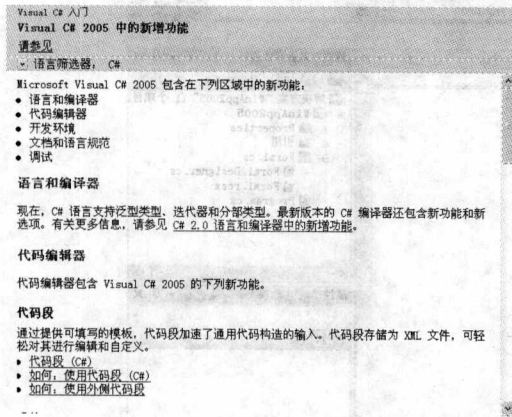


图 1-7 本机的 MSDN 资源

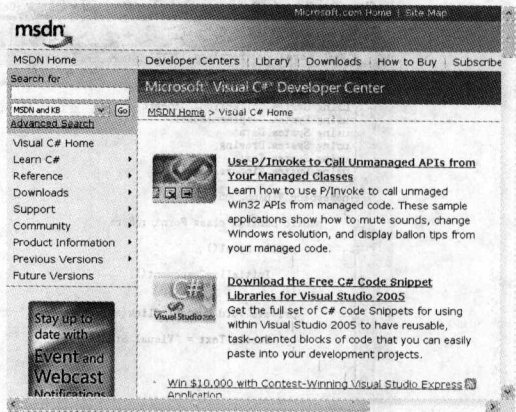


图 1-8 因特网的 MSDN 资源

1.1.2 设计器视图与代码视图

设计器视图与代码视图是程序设计最常用的两个子窗口。设计器视图用来设计 Windows 窗体或 Web 窗体, 代码视图用于设计程序源代码。

创建 Windows 或 Web 应用程序时, 系统将默认打开设计器视图, 如图 1-9 所示。在设计器视图中可以为 Windows 或 Web 界面添加控件。

双击设计器视图中的窗体, 将打开代码视图, 代码视图与设计器视图共享一个屏幕区域, 通过窗口上部的标签进行切换, 如图 1-10 所示。代码视图实质上是一个纯文本编辑器, 在

该视图可以进行通常的文本编辑操作，如选定、复制、移动、撤销、恢复等。

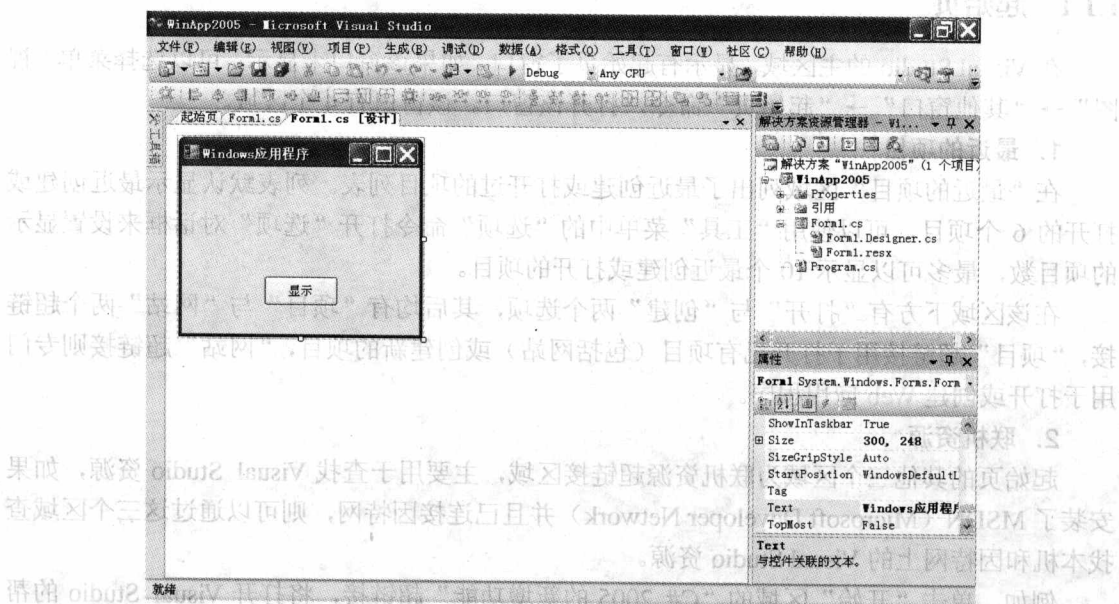


图 1-9 设计器视图

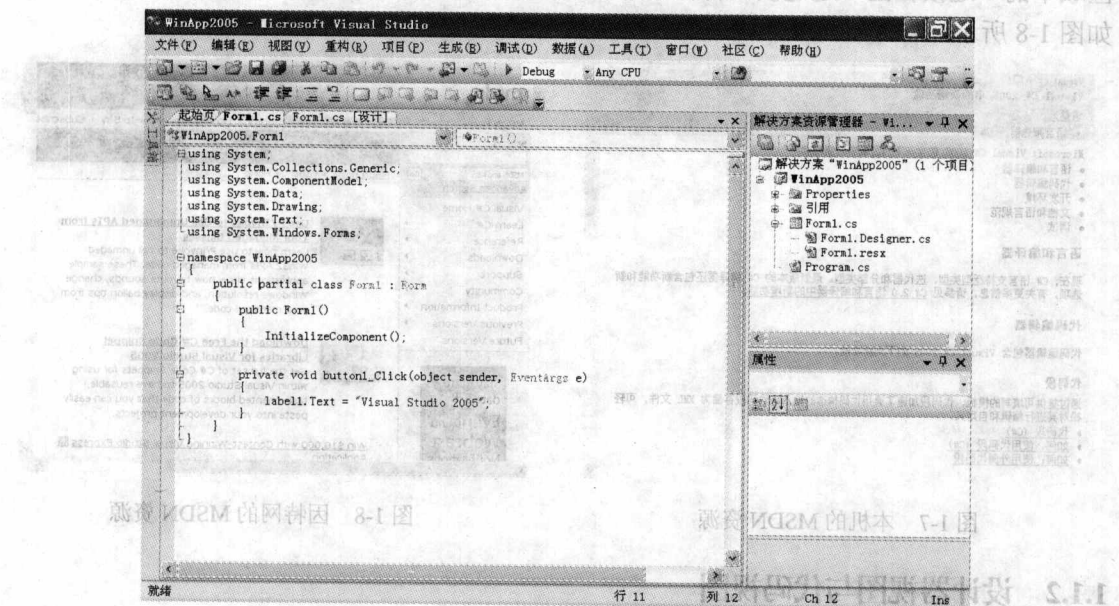


图 1-10 代码视图

打开已有的 Windows 或 Web 项目时，其显示方式由上一次关闭解决方案时的状态确定，如果关闭时是设计器视图，则打开的仍是设计器视图；如果关闭时是代码视图，则打开的仍是代码视图；如果都没有打开，则两种视图均不显示。

1.1.3 解决方案资源管理器

使用 Visual Studio 开发的应用程序叫解决方案，每一个解决方案可以包含一个或多个项目。一个项目通常是一个完整的程序模块，一个项目可以有多个项。“解决方案资源管理器”子窗口显示 Visual Studio 解决方案的树型结构，如图 1-11 所示。在“解决方案资源管理器”中可以浏览组成解决方案的所有项目和每个项目中的文件，可以对解决方案的各元素进行组织和编辑。

双击某个项目中的文件将打开相应的视图，即可以对该文件进行编辑。例如，双击窗体文件将打开设计器视图，双击代码文件将打开代码视图。另外，如果选择了某一文件，在“解决方案资源管理器”的左上方将动态地出现相应的打开该文件的按钮，单击该按钮也可以打开选择的文件。如选择窗体文件后，将出现设计器视图按钮与代码视图按钮, 单击按钮将打开相应的视图。如果选择的是代码文件，将只显示代码视图按钮。

在“解决方案资源管理器”中，可以使用鼠标的右键操作。例如，右键单击“解决方案资源管理器”中的“引用”项，可以选择“添加引用”或“添加 Web 引用”的操作，如图 1-12 所示。

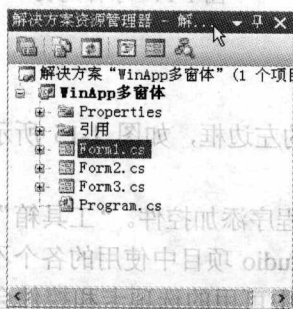


图 1-11 解决方案资源管理器



图 1-12 右键操作

当一个解决方案包含多个项目时，其中必须且只能有一个项目作为默认的启动项目，该项目是程序运行的入口。启动项目的名称以粗体显示。

1.1.4 属性

“属性”子窗口用于设置解决方案中各个子项的属性，当选择设计器视图、解决方案、类视图中的某一子项时，“属性”子窗口将以两列表格的形式显示该子项的所有属性，如图 1-13 所示。

在窗口的上部有一下拉列表框，显示当前选定的对象名称及所属类型。可以单击该列表框的下拉按钮，从打开的列表中选择其他对象。例如，选择的如果是窗体中的控件对象，在设计器窗口中，被选择的对象会自动处于选定状态，原来选定的对象将被取消。

在表格中左列是属性名称列表，右列对应各属性名称的属性值。选择某一名称，可以在右列修改该属性值。

属性默认按字母顺序排列，单击窗口中的字母排序按钮与分类排序按钮, 可以在两种排序方式之间切换。

选择设计器视图中的窗体或窗体中的控件，在“属性”子窗口中将显示事件按钮，单击该按钮，窗口将显示被选择窗体或控件的事件列表，如图 1-14 所示。双击某一事件名称，将打开代码视图，并添加该事件方法的声明。

属性与事件窗口的下部有一个属性或事件功能说明区域，当选择某一属性或事件时，说明区域显示文字说明属性或事件的作用，这对初学者很有用。如果该区域没有显示，可将鼠标指向窗口列表框下部边框，当鼠标变为双向箭头时，向上拖动鼠标，该区域即可显示，如图 1-13 所示。

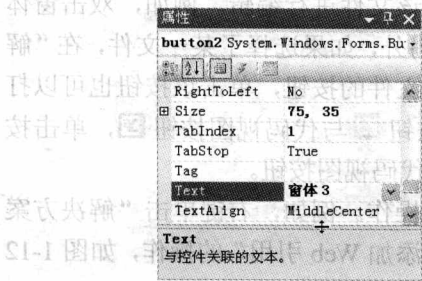


图 1-13 属性列表



图 1-14 事件列表

1.1.5 工具箱

默认状态下，“工具箱”处于隐藏状态，位于窗口的左边框，如图 1-15 所示。当鼠标指向“工具箱”时，将显示相应窗口，如图 1-16 所示。

“工具箱”用于向 Windows 应用程序或 Web 应用程序添加控件。“工具箱”使用选项卡分类管理其中的控件，打开“工具箱”将显示 Visual Studio 项目中使用的各个不同的控件列表。根据当前正在使用的设计器或编辑器，“工具箱”中可用的选项卡和控件会有所变化。图 1-16 是 Windows 应用程序窗体设计器视图状态下的工具箱。

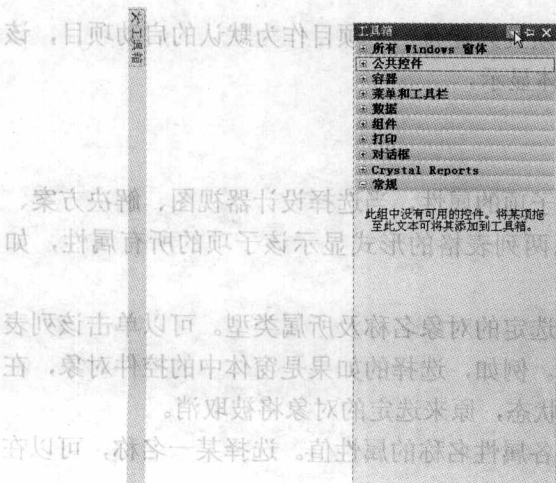


图 1-15 隐藏状态



图 1-16 工具箱

Visual Studio 2005 将控件放在不同的选项卡中，各选项卡以“+”号表示折叠状态，以

“-”号表示展开。其中，“所有 Windows 窗体”包含了其下各选项卡中的所有控件。控件以名称的字母顺序排列，目的是便于用户选择控件。

如果“工具箱”没有出现，可以选择“视图”菜单中的“工具箱”命令。

可以从“工具箱”中选择控件并将其拖动到窗体设计器中，还可以将某一代码片断拖回到“工具箱”中，以便将来重新使用。例如，可以将“按钮”(Button)控件从“公共控件”选项卡中拖放到 Windows 应用程序的设计器视图，即添加控件；或从一个编辑器中选择并拖出一个代码片段，然后将其存储在“工具箱”中，以便将来重复使用。

如果选择了“工具箱”中的某一项后，想撤销选择，可单击控件列表中的“指针”控件。

1.1.6 子窗口的操作

在 Visual Studio 集成环境中，有两类子窗口，一类是在主区域显示的窗口，如起始页、代码视图、设计器视图窗口等；一类是在主区域周围显示的子窗口，如工具箱、解决方案资源管理器、类视图、属性、动态帮助、输出、任务列表等。

在主区域显示的窗口，如果不止一个，则诸多窗口将以选项卡的形式叠放在一起，在最前端显示的为当前活动窗口。可以通过选项卡的标签切换各个窗口，在窗口的右上角有一个“关闭”按钮，用于关闭窗口。关闭按钮只对当前活动窗口有效。要关闭某一窗口，首先要使该窗口成为当前活动窗口，然后单击“关闭”按钮，则可关闭窗口。

在主区域周围的窗口也是由若干窗口共享某一屏幕区域，以选项卡的形式叠放在一起，通过标签进行切换。这些窗口的标题栏右部都有两个操作按钮，一个是关闭按钮，另一个是图钉按钮。关闭按钮用于关闭窗口，而图钉按钮用于决定窗口的隐藏与显示状态，在显示状态时又可以有停靠显示与浮动显示两种方式。

当图钉按钮为横向时，窗口为隐藏状态，这时窗口以标签形式显示在 Visual Studio 的左、右、下边框上，这时用鼠标指向标签，则窗口显示，鼠标移出窗口，则窗口又重新隐藏。隐藏状态的主要作用是为主区域的窗口开辟尽可能大的屏幕空间。

当图钉按钮为纵向时，窗口为显示状态，默认为停靠方式，即窗口附着在 Visual Studio 的左、右、下边框上。这时鼠标指向窗口的标题栏，拖动鼠标，使窗口离开边框，窗口即为浮动显示方式，这时标题栏上的图钉按钮将消失。如要使浮动方式变为停靠方式，只需拖动窗口至 Visual Studio 的边框上即可。

1.2 使用 Visual Studio 的帮助

在 Visual Studio 的集成环境中，可以使用动态帮助、智能感知和 MSDN 资源等多种帮助形式。

1.2.1 动态帮助

动态帮助与解决方案资源管理器共享一个屏幕区域。“动态帮助”对用户当前操作提供相关的帮助主题列表。当用户在 Visual Studio 环境中进行某一项操作时，“动态帮助”将搜索 MSDN 库，查找与该操作相关的帮助主题，以超链接的形式显示在动态帮助窗口，并把它认为可能最有用的主题列在第一位。单击一个主题链接后，此主题将会显示在 Visual Studio