



全国高等职业教育规划教材

Visual C# 程序设计 应用教程

郭力子 编著



电子教案下载网址 www.cmpedu.com



机械工业出版社
CHINA MACHINE PRESS



全国高等职业教育规划教材

Visual C#程序设计应用教程

郭力子 编著



机械工业出版社

本书从实际应用的角度出发,全面系统地介绍了 Visual C#程序设计的基本知识和应用技术。全书共分 10 章,内容包括 C#和 Visual Studio 2005 概述,变量与表达式,流程控制和函数,数组、集合与字符串,面向对象编程基础,Windows 应用程序设计,数据库访问,文件处理和绘图基础,Web 应用程序设计基础,程序应用项目实践等。本书内容立足于课堂教学和面向应用,书中配有丰富的案例,各章均配有实训练习题和习题,便于读者掌握重点及提高程序设计动手能力。全书内容循序渐进,示例面向应用,全面兼顾了知识介绍、编程能力培养和实践技能训练。

本书可作为高职高专相关专业的教材,也可作为 C#程序设计培训机构的教材,还可作为.NET 开发人员的参考书。

本书配套授课电子课件,需要的教师可登录 www.cmpedu.com 免费注册、审核通过后下载,或联系编辑索取(QQ: 81922385, 电话: 010-88379739)。

图书在版编目(CIP)数据

Visual C#程序设计应用教程 / 郭力子编著. —北京: 机械工业出版社, 2010.10

全国高等职业教育规划教材

ISBN 978-7-111-31994-8

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

中国版本图书馆 CIP 数据核字 (2010) 第 186634 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: 王 颖

责任印制: 杨 曦

保定市中国画美凯印刷有限公司印刷

2011 年 1 月第 1 版·第 1 次印刷

184mm×260mm·18.5 印张·452 千字

0001—3000 册

标准书号: ISBN 978-7-111-31994-8

定价: 31.00 元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

电话服务

网络服务

社服务中心: (010) 88361066

门户网: <http://www.cmpbook.com>

销售一部: (010) 68326294

教材网: <http://www.cmpedu.com>

销售二部: (010) 88379649

读者服务部: (010) 68993821

封面无防伪标均为盗版

全国高等职业教育规划教材计算机类专业

编委会成员名单

主 任 周智文

副 主 任 周岳山 林 东 王协瑞 张福强
陶书中 龚小勇 王 泰 李宏达
赵佩华 陈 晴

委 员 （按姓氏笔画排序）

马 伟	马林艺	卫振林	万雅静
王兴宝	王德年	尹敬齐	卢 英
史宝会	宁 蒙	刘本军	刘新强
刘瑞新	余先锋	张洪斌	张 超
杨 莉	陈 宁	汪赵强	赵国玲
赵增敏	贾永江	陶 洪	康桂花
曹 毅	眭碧霞	鲁 辉	裴有柱

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近 60 所高等职业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”进行了全面的修订和增补,并更名为“全国高等职业教育规划教材”。

本系列教材是由高职高专计算机专业、电子技术专业和机电专业教材编委会分别会同各高职高专院校的一线骨干教师,针对相关专业的课程设置,融合教学中的实践经验,同时吸收高等职业教育改革的成果而编写完成的,具有“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。在几年的教学实践中,本系列教材获得了较高的评价,并有多品种被评为普通高等教育“十一五”国家级规划教材。在修订和增补过程中,除了保持原有特色外,针对课程的不同性质采取了不同的优化措施。其中,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。同时,根据实际教学的需要对部分课程进行了整合。

归纳起来,本系列教材具有以下特点:

- 1) 围绕培养学生的职业技能这条主线来设计教材的结构、内容和形式。
- 2) 合理安排基础知识和实践知识的比例。基础知识以“必需、够用”为度,强调专业技术应用能力的训练,适当增加实训环节。
- 3) 符合高职学生的学习特点和认知规律。对基本理论和方法的论述要容易理解、清晰简洁,多用图表来表达信息;增加相关技术在生产中的应用实例,引导学生主动学习。
- 4) 教材内容紧随技术和经济的发展而更新,及时将新知识、新技术、新工艺和新案例等引入教材。同时注重吸收最新的教学理念,并积极支持新专业的教材建设。
- 5) 注重立体化教材建设。通过主教材、电子教案、配套素材光盘、实训指导和习题及解答等教学资源的有机结合,提高教学服务水平,为高素质技能型人才的培养创造良好的条件。

由于我国高等职业教育改革和发展的速度很快,加之我们的水平和经验有限,因此在教材的编写和出版过程中难免出现问题和错误。我们恳请使用这套教材的师生及时向我们反馈质量信息,以利于我们今后不断提高教材的出版质量,为广大师生提供更多、更适用的教材。

机械工业出版社

前 言

Visual C#是微软公司的 Visual Studio .NET 开发平台上面向对象的编程语言。作为一种继承自 C/C++和 Java 的语言, Visual C#简单易学、面向对象而且类型安全, 已成为当今计算机程序设计的主流编程语言。Visual C#既具有 C、C++那样的强大功能, 又具有 Java 那样的面向对象机制, 还兼有类似 VB 开发的高效率和方便性。使用这种面向对象的、可视化的编程语言, 结合事件驱动的设计, 将使程序设计变得轻松快捷。

本书共分 10 章, 内容包括 C#和 Visual Studio 2005, 变量与表达式, 流程控制和函数, 数组、集合与字符串, 面向对象编程基础, Windows 应用程序设计, 数据库访问, 文件处理和绘图基础, Web 应用程序设计基础, 程序应用项目实践。

本书突出实际应用, 注重能力的培养。内容简明扼要、由浅入深、顺序渐进, 力求通俗易懂、简洁实用。全面兼顾了知识介绍、编程能力培养和实践技能训练。

本书案例简单明了、实用性强, 力求带给读者真实的编程体验。每章后都配有实训题, 其中很多都是书中案例的扩展, 有助于读者实践。另外, 还配有丰富的简答题和选择题供读者练习以巩固已学的知识内容。为了帮助读者快速理解, 书中案例前都有关于代码编写的简要分析, 代码部分尽量给予注释, 案例后针对重要或易错的内容还有提示说明。读者通过仔细阅读这些代码并且通过实训操作, 可以迅速培养和提高独立解决实际编程问题的能力。本书最后的项目实践, 可进一步帮助读者提高编写 C#应用程序的综合能力。

本书适用于高职高专相关专业, 也可作为 C#程序设计培训机构的教材或参考书。建议读者遵循书中介绍的方法和步骤实际建立案例程序, 然后对案例程序进行修改或扩展, 并通过对案例代码进行分析和讨论掌握案例背后所包含的概念、原理、知识点和方法等。这是学习程序设计最稳妥、有效和快捷的途径。

本书的全部示例代码、部分习题参考答案和采用 PowerPoint 制作的电子教案, 读者可直接从机械工业出版社教材服务网 <http://www.cmpedu.com> 下载。

在本书的编写过程中, 作者参考了大量的相关资料, 吸取了许多同仁的经验, 在此谨表谢意。

鉴于作者水平有限, 书中错误和不足之处难免, 恳请广大读者批评指正。

作 者

目 录

出版说明

前言

第 1 章 C#和 Visual Studio 2005

概述 1

1.1 C#语言简介 1

1.2 Visual Studio 2005 集成开发

环境 2

1.2.1 菜单栏与工具栏 3

1.2.2 设计器与编辑器窗口 4

1.2.3 解决方案资源管理器窗口 5

1.2.4 工具箱窗口 5

1.2.5 属性窗口 5

1.2.6 类视图窗口 6

1.2.7 服务器资源管理器窗口 6

1.3 创建 C#应用程序 6

1.3.1 创建控制台应用程序 6

1.3.2 创建 Windows 应用程序 8

1.4 C#应用程序的一般结构 11

1.5 C#中标识符命名风格 11

1.6 实训 12

1.7 习题 12

第 2 章 变量与表达式 13

2.1 输入与输出 13

2.2 变量与常量 14

2.2.1 变量的声明 14

2.2.2 变量的种类 15

2.2.3 常量的声明与使用 16

2.3 数据类型 16

2.3.1 值类型 17

2.3.2 引用类型 20

2.3.3 数组类型 20

2.3.4 类型转换 21

2.3.5 装箱与拆箱 22

2.4 运算符与表达式 25

2.4.1 算术运算符 26

2.4.2 赋值运算符 28

2.4.3 关系和类型测试运算符 29

2.4.4 逻辑运算符 31

2.4.5 其他特殊运算符 31

2.4.6 运算符的优先级 32

2.5 应用示例——常用变量
类型的运算 33

2.6 实训 34

2.7 习题 35

第 3 章 流程控制和函数 37

3.1 选择语句 37

3.1.1 if 语句 37

3.1.2 switch 语句 39

3.2 循环语句 41

3.2.1 while 语句 41

3.2.2 do...while 语句 42

3.2.3 for 语句 44

3.2.4 foreach 语句 46

3.3 跳转语句 47

3.3.1 break 语句和 continue
语句 47

3.3.2 goto 语句 48

3.3.3 return 语句 49

3.4 函数 49

3.4.1 函数的定义和使用 49

3.4.2 函数参数的传递方式 51

3.5 异常处理 55

3.5.1 try...catch 语句 56

3.5.2 try...catch...finally 语句 57

3.6 应用示例——四则运算器 58

3.7 实训 60

3.8	习题	60	5.6	委托	104
第4章	数组、集合与字符串	63	5.7	应用示例——类的继承的 综合应用	105
4.1	数组	63	5.8	实训	107
4.1.1	一维数组	63	5.9	习题	108
4.1.2	多维数组	65	第6章	Windows 应用程序设计	110
4.2	集合	66	6.1	在 Visual Studio 中创建窗体 (Form)	110
4.2.1	ArrayList 类	66	6.1.1	添加控件	111
4.2.2	Hashtable 类	68	6.1.2	设定布局	111
4.3	字符串	70	6.1.3	控件的属性、方法和 事件	111
4.3.1	字符串的形式与声明	70	6.2	按钮控件	113
4.3.2	字符串的处理	71	6.3	文本类控件	115
4.4	应用示例——使用二维 数组	74	6.3.1	标签控件	115
4.5	实训	76	6.3.2	文本框控件	115
4.6	习题	76	6.3.3	格式文本控件	120
第5章	面向对象编程基础	79	6.4	菜单、工具栏和状态栏 控件	120
5.1	面向对象程序设计概述	79	6.4.1	菜单控件	120
5.2	类	80	6.4.2	工具栏控件	123
5.2.1	类的声明	80	6.4.3	状态栏控件	125
5.2.2	对象和类的实例化	80	6.5	选择类控件	127
5.2.3	类的成员	82	6.5.1	单选按钮控件和复选框 控件	127
5.2.4	类成员的访问	82	6.5.2	列表框控件	130
5.2.5	构造函数和析构函数	84	6.5.3	下拉组合框控件	133
5.2.6	类的继承	86	6.5.4	列表视图控件	135
5.2.7	类的多态	88	6.5.5	树视图控件	137
5.3	字段、属性和索引器	89	6.6	容器类控件	138
5.3.1	字段	89	6.6.1	面板控件	139
5.3.2	属性	89	6.6.2	分组框控件	139
5.3.3	索引器	91	6.6.3	选项卡控件	141
5.4	方法	93	6.7	通用对话框	144
5.4.1	方法的声明	93	6.7.1	“打开文件”和“保存 文件”对话框	145
5.4.2	方法的调用	94	6.7.2	其他对话框	147
5.4.3	静态方法	95	6.7.3	“模式”和“非模式” 对话框	148
5.4.4	方法的重载	96			
5.4.5	方法的覆盖	97			
5.4.6	使用 base 访问	99			
5.5	抽象类与接口	100			
5.5.1	抽象类	100			
5.5.2	接口	102			

6.8 其他控件	148	8.1.2 文本文件处理	193
6.8.1 日期控件	148	8.1.3 二进制文件处理	196
6.8.2 定时器控件	150	8.1.4 文件夹处理类	197
6.9 多窗体的概念与创建	150	8.1.5 文件夹基本操作	198
6.10 应用示例——文本编辑器	153	8.1.6 XML 文件读/写	202
6.11 实训	156	8.2 GDI+绘图基础	205
6.12 习题	156	8.2.1 创建绘图对象	205
第 7 章 数据库访问	158	8.2.2 基本图形绘制	205
7.1 关系数据库和 SQL 基本 概念	158	8.2.3 图像处理基础	209
7.1.1 关系数据库基本概念	158	8.2.4 使用图像文件	210
7.1.2 SQL 基本概念	159	8.3 应用示例——图片处理 观看器	211
7.2 ADO.NET 简介	162	8.4 实训	215
7.2.1 ADO.NET 的基本构成	162	8.5 习题	215
7.2.2 .NET 数据提供程序	163	第 9 章 Web 应用程序设计基础	217
7.3 应用 ADO.NET 访问 数据库	164	9.1 ASP.NET 基础	217
7.3.1 数据库连接	164	9.1.1 ASP.NET 的特点	217
7.3.2 显示数据记录	166	9.1.2 IIS 安装与虚拟目录	217
7.3.3 更新、插入和删除数据 记录	169	9.1.3 创建 ASP.NET Web 应用 程序	218
7.3.4 使用参数化 SQL 语句 访问数据库	172	9.1.4 用文本编辑器创建 ASP.NET 应用程序	221
7.4 使用 DataSet	175	9.1.5 ASP.NET 的基本语法	222
7.4.1 DataSet 和 DataAdapter	175	9.2 ASP.NET 常用控件	223
7.4.2 使用 DataSet 显示和更新 记录	176	9.2.1 HTML 控件	223
7.5 控件的数据绑定	178	9.2.2 Web 标准控件	226
7.5.1 数据绑定	178	9.2.3 Validation 验证控件	230
7.5.2 使用 TextBox 控件	179	9.2.4 Data 数据控件	232
7.5.3 使用 DataGridView 控件	181	9.2.5 母版页	237
7.6 使用存储过程	183	9.3 ASP.NET 内置对象	240
7.7 应用示例——学生信息管理 系统	184	9.3.1 Page 对象	241
7.8 实训	187	9.3.2 Request 对象	242
7.9 习题	187	9.3.3 Response 对象	244
第 8 章 文件处理和绘图基础	190	9.3.4 Session 对象	247
8.1 文件和目录处理	190	9.3.5 Application 对象	248
8.1.1 文件处理类	190	9.3.6 Server 对象	250
		9.4 应用示例——网上 聊天室	251
		9.5 实训	258

9.6 习题	259	10.4 详细设计	264
第 10 章 程序应用项目实践	261	10.4.1 数据库层	264
10.1 多层应用程序	261	10.4.2 数据库访问层	268
10.2 系统功能概述	262	10.4.3 业务逻辑层	274
10.3 概要设计	263	10.5 实训	282
10.3.1 应用程序模块设计	263	10.6 习题	282
10.3.2 应用程序结构	264	参考文献	283

第1章 C#和 Visual Studio 2005 概述

1.1 C#语言简介

C#（读作“C Sharp”）语言是由微软公司开发的基于.NET 框架（.NET Framework）的一种新型程序设计语言。

C#语言是一种面向对象和类型安全的程序设计语言，它继承了 C 语言的语法、关键词和运算符，同时与 Java 有很多相似的地方，是在 C++定义的对象模型基础上加以改进而开发的。

C#语言是一种可以单独学习的计算机程序设计语言，但是微软公司最初设计 C#语言的初衷是为了编写.NET 框架，同时 C#语言中使用的函数库都来自于.NET 框架定义的函数库。因此，有必要先了解一下.NET 框架。

C#语言也常被直接简称为 C#。

1. .NET 框架

.NET 框架是微软公司的一个多语言组件开发和运行环境，其目的是在统一框架下，可以应用多种语言创建、配置和运行 Windows 应用程序和 Web 应用程序。.NET 框架主要由公共语言运行库（Common Language Runtime, CLR），.NET 框架类库（.NET Framework Class Library），Microsoft ADO.NET，ASP.NET，VB.NET、C#.NET、C++.NET、J#.NET 等语言运行库几部分组成，如图 1-1 所示。

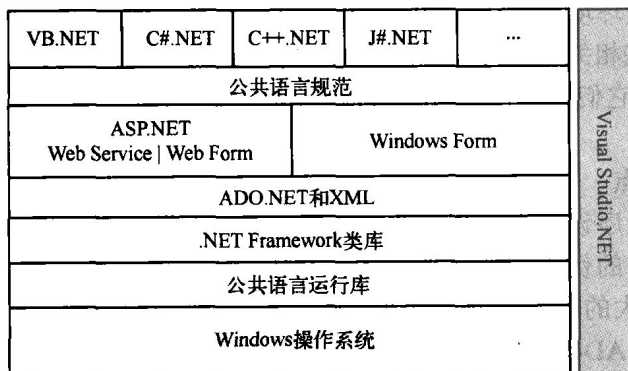


图 1-1 .NET 框架的组成

公共语言运行库是.NET 框架的运行环境，它负责运行和维护用户编写的任何程序代码，包括内存管理、线程执行、代码执行、代码安全验证、编译及其他系统服务等。符合公共语言规范（Common Language Specification, CLS）的任何语言都可以在公共语言运行库上运行。.NET 框架支持的主要语言有 Visual Basic.NET（简称 VB.NET）、C#.NET、

C++.NET 和 J#.NET。其中, C#是.NET 框架开发平台首选的开发语言。

.NET 框架类库是.NET 框架面向所有的.NET 程序语言提供的一个公共的基础类库。这个类库构成了.NET 框架的核心部分, 而各个类又包含了多个方法、属性和其他成员。利用这些类和类所包含的方法、属性和其他成员, 就可以快速搭建各种应用程序。常见的几组类库如下。

1) System 类: 该类提供支持其他类所需要的基本服务, 如底层服务、输入输出、文本处理等任务。

2) Data 类: 该类提供处理数据库的操作。

3) XML 类: 该类提供处理 XML 文件的操作。

4) Web 类: 该类用来完成用户界面和 Web 服务的任务, 这是 ASP.NET 开发人员最常用的类。

5) Windows Forms: 该类提供传统的桌面应用程序服务。

6) Drawing 类: 该类提供创建图形、复杂字体等操作。

ADO.NET (Microsoft ActiveX Data Object) 是微软公司的新一代数据库访问技术。ADO.NET 为非连接的编程模型提供了改进的支持。它还提供了丰富的 XML 支持。

ASP.NET 是一个建立在公共语言运行库基础上的编程模型, 可以使用 ASP.NET 在服务器上构建功能强大的 Web 应用程序。ASP.NET Web 窗体提供了简单易用而且功能强大的方法来构建动态的 Web 用户界面 UI (User Interface)。

.NET 框架支持以下 3 种类型的用户界面。

1) Windows 窗体, 它运行在 WIN32/64 客户端上。

2) Windows 控制台应用程序。

3) Web 窗体, 它通过 ASP.NET 工作。

2. 开发环境

.NET 为使用与开发人员提供了功能强大、种类丰富的管理与开发工具。这些工具包括: .NET 的集成开发环境 Visual Studio 200x (从 2000 年以来, Visual Studio 的发展经历了许多版本), 以及一些相关的工具, 如编译、连接、反汇编、证书管理、注册工具、XML 工具等。正是由于有了它们的支持, .NET 日趋强大, 现已成为计算机软件和系统开发的主要首选平台之一。

3. C#语言的特点

C#语言代表着程序设计语言演变的一个新阶段, 它继承了 C++和 Java 这两种世界上最重要的计算机语言的优点, 并且还增加了委托、索引等创新性特点; 同时, C#语言利用了.NET 作为其强大的平台, 使得它在 Windows 图形用户界面、ASP.NET Web 应用、XML Web Service 及 ADO.NET 数据库等方面有广泛的应用。总体而言, C#语言具有简单易学, 面向对象, 与 Web 的紧密结合, 安全稳定, 支持多线程, 丰富的类库, 灵活性与兼容性等主要特点。

1.2 Visual Studio 2005 集成开发环境

Microsoft Visual Studio 2005 是基于.NET 框架的应用程序集成开发环境, 也是目前使用

最广泛的 Windows 平台上的应用程序开发工具。它使开发人员能够快速创建 Windows 操作平台下的 Windows 应用程序和网络应用程序，或者创建网络服务、智能设备（手机、掌上电脑、智能终端等）应用程序和 Office 插件。在 Visual Studio 2005 中可以实现以下功能。

- 1) 可视化的窗口和网页设计器：支持以拖动方式生成控件，同时提供源码查看视图。
- 2) 智能化的代码编辑器：支持语句自动完成、语法检查等功能。
- 3) 集成的编译、调试功能。
- 4) 项目管理功能：生成并管理应用程序文件，并可将应用程序安装到远程服务器上。

Visual Studio 2005 集成开发环境包括了菜单栏、工具栏、工具箱、属性窗口、解决方案资源管理等各功能窗口。选择“视图”菜单下的相关命令显示或关闭各个功能窗口。各个窗口可以打开后重叠在同一位置上。通过切换各窗口顶部或底部的选项标签就可以在不同的窗口之间进行切换。单击窗口右上角的按钮可以把窗口固定在所在的位置，这时该按钮变成, 再次单击该按钮，可使窗口重新浮动。主窗口如图 1-2 所示。

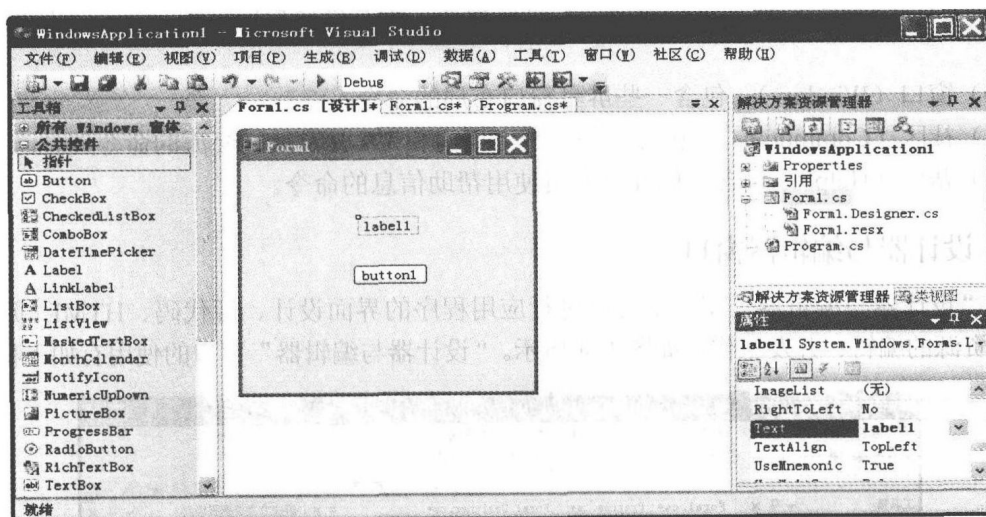


图 1-2 Visual Studio 2005 集成开发环境的主窗口

1.2.1 菜单栏与工具栏

Visual Studio 2005 的菜单栏包括多个菜单项，如图 1-3 所示。这些菜单项会根据不同的使用状况而显示，使用这些菜单项以及它们的子菜单提供的命令可以完成几乎所有 Visual Studio 的功能。对应这些菜单项，工具栏提供了更为直接方便的选项。使用“视图”→“工具栏”窗口命令，就可以确定需要在窗口中显示不同的工具栏。

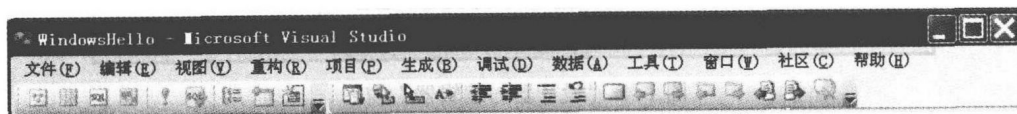


图 1-3 Visual Studio 2005 的菜单栏

菜单栏中的每个菜单都有标题，每个标题都有一个下拉式菜单。

- 1) 文件 (File): 主要包括新建 (New)、打开 (Open)、保存 (Save) 以及打开和关闭

解决方案等命令。

2) 编辑 (Edit): 主要包含一些符合 Windows 操作风格的进行文件编辑的各项命令, 如撤销 (Undo)、复制 (Copy)、粘贴 (Paste)、删除 (Delete)、查找 (Find) 和替换 (Replace) 等命令。

3) 视图 (View): 包含显示与隐藏工具栏、工具箱 (Toolbox) 和各种独立的工具窗口的所有命令。

4) 重构 (Refactor): 包括用来实现一些常见代码重构的命令。

5) 项目 (Project): 包括向当前项目添加、改变和删除组件、引用 Windows 对象和添加部件等命令。

6) 生成 (Build): 包含代码生成的有关命令。

7) 调试 (Debug): 包含调试程序的命令, 启动和终止当前应用程序运行的命令。

8) 数据 (Data): 包含数据的生成和预览命令。

9) 工具 (Tools): 包括进程调试、数据库连接、宏和外接程序管理、设置工具箱和选项等命令。

10) 窗口 (Window): 包含一些屏幕窗口布局的命令。

11) 社区 (Community): 包含一些在网上提问和 VS 开发社区交流的命令。

12) 帮助 (Help): 包含方便开发人员使用帮助信息的命令。

1.2.2 设计器与编辑器窗口

在“设计器与编辑器”窗口中可以进行应用程序的界面设计、源代码、HTML 页、CSS 表单及资源的编辑、开发工作, 如图 1-4 所示。“设计器与编辑器”窗口的使用有如下特点:

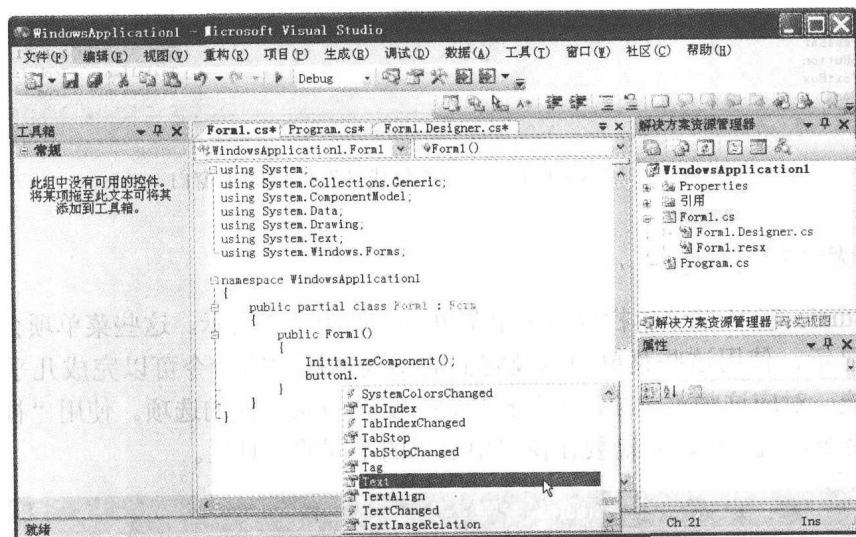


图 1-4 “设计器与编辑器”窗口

1) 采用“智能感知”技术, 在编码时不必离开文本编辑器以执行语言元素搜索, 可以保持上下文, 查找所需的信息, 直接向代码中插入语言元素。

2) 语法着色。默认情况下, 常规代码为黑色, 注释为绿色, 而 C#语言的关键字为蓝色。

1.2.3 解决方案资源管理器窗口

“解决方案资源管理器”用于管理和监控方案中的项目，解决方案资源管理器窗口通常位于 Visual Studio 界面的右上方，如图 1-2 所示。在菜单栏中选择“视图”→“解决方案资源管理器”命令，可以打开“解决方案资源管理器”。

一个解决方案中可以包含多个项目，它使用户能够方便地组织需要开发和设计的项目和文件以及配置应用程序或组件。解决方案资源管理器中采用树状视图显示方案及其中方案中项目的层次结构。方案中包含项目及项目中的条目。

1.2.4 工具箱窗口

“工具箱”窗口通常位于 Visual Studio 界面的左方，如图 1-2 所示。选择“视图”→“工具箱”菜单命令可以打开“工具箱”窗口。

“工具箱”窗口中包含若干标签，展开这些标签，可以显示不同种类的控件。这些控件是 Visual Studio 提供的封装好的.NET 框架的各种类的对象。使用时可用鼠标将其拖放到设计器窗口的表单上。

1.2.5 属性窗口

“属性”窗口通常位于 Visual Studio 界面的右下方，如图 1-2 所示。选择“视图”→“属性窗口”菜单命令可以打开当前对象的“属性”窗口。

控件、文档和窗体具有一些指定它们将如何与用户进行交互以及在运行时操作过程中需要的信息的属性。可以使用属性窗口编辑和查看文件、项目和方案的属性。此外，属性窗口还能在设计时查看和修改设计器中被选对象的属性和事件。

“属性”窗口上方有 5 个图标按钮（见图 1-5），从左到右依次介绍如下。

- 按分类顺序：按类别列出选定对象的所有属性及属性值，可以折叠类别以减少可见属性数。展开或折叠类别时，可以在类别名左边看到加号（+）或减号（-）。类别按字母顺序列出。
- 按字母顺序：按字母顺序对选定对象的所有设计时的属性和事件排序。若要编辑可用的属性，在它右边的单元格中单击并输入更改内容。
- 属性：显示对象的属性。很多对象也有可以使用“属性”窗口查看的事件。
- 事件：显示对象的事件。此“属性”窗口工具栏控件仅当窗体或控件设计器在一个 Visual C#项目的上下文中处于活动状态时才可用。
- 属性页：显示选定项的“属性页”对话框。“属性页”显示“属性”窗口中的可用属性的子集、同集或超集。使用该按钮可以查看和编辑与项目的活动配置相关的属性。

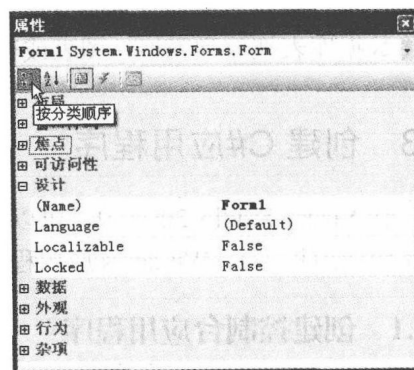


图 1-5 “属性”窗口

1.2.6 类视图窗口

“类视图”窗口与“解决方案资源管理器”窗口共享显示区域，单击“解决方案资源管理器”窗口下方的“类视图”选项卡就可以切换到“类视图”窗口（见图 1-6）。或者选择“视图”→“类视图”菜单命令，也显示“类视图”窗口。

“类视图”窗口中以树结构显示了当前工程中的所有类，并在每个类中列出了成员变量和成员函数，每一个类首先列出带有紫色图标的成员函数，然后是带有绿蓝色图标的成员变量，每个成员的图标左边都有一个标志，以表示成员类型和存取类别的信息，保护型成员图标旁边的标志为一把钥匙，私有成员的标志是一把挂锁，而公有成员图标旁边没有标志。



图 1-6 “类视图”窗口

1.2.7 服务器资源管理器窗口

“服务器资源管理器”窗口与“工具箱”窗口共享显示区域，单击“工具箱”窗口下方的“服务器资源管理器”选项卡就可以切换到“服务器资源管理器”窗口。或者选择“视图”→“服务器资源管理器”菜单命令，也可以打开“服务器资源管理器”窗口，如图 1-7 所示。

“服务器资源管理器”是 Visual Studio 的服务器控制台。通过它可以查看和操作服务器及其中包含的数据库。例如，可以通过服务器管理器连接数据库，创建数据库表或将数据库表拖放到表格、图表或查询中。



图 1-7 “服务器资源管理器”窗口

1.3 创建 C# 应用程序

在 Visual Studio 2005 中，可以便捷地使用 C# 语言开发 Windows 控制台应用程序（简称控制台应用程序）、Windows 应用程序以及 Web 应用程序。

1.3.1 创建控制台应用程序

【例 1-1】 创建一个控制台应用程序。

在 Visual Studio 2005 中，创建控制台应用程序的过程如下：

- 1) 启动 Visual Studio 2005。
- 2) 依次选择“文件”→“新建”→“项目”菜单命令，打开“新建项目”对话框。
- 3) 在“项目类型”窗口中选择“Visual C#”节点，选择“Windows”→“控制台应用程序”菜单命令。输入应用程序名称 ConsoleHello，选择合适的路径保存，如图 1-8 所示。
- 4) 单击“确定”按钮。Visual Studio 将创建项目“ConsoleHello”，并在“解决方案资源管理器”中列出为 ConsoleHello 项目自动生成的文件，同时在编辑器窗口中显示出

Program.cs 文件生成的程序框架，如图 1-9 所示。

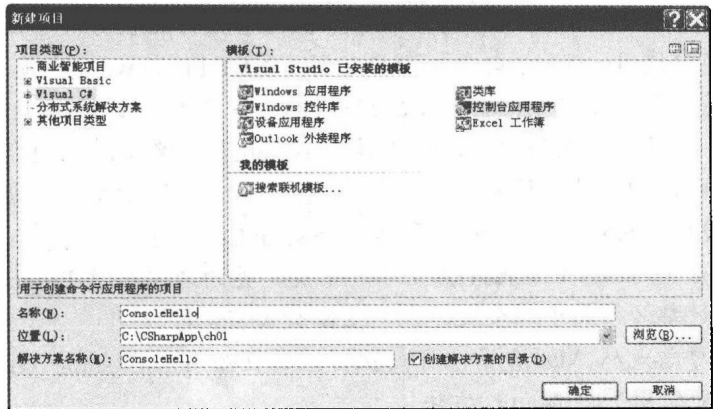


图 1-8 新建控制台应用程序

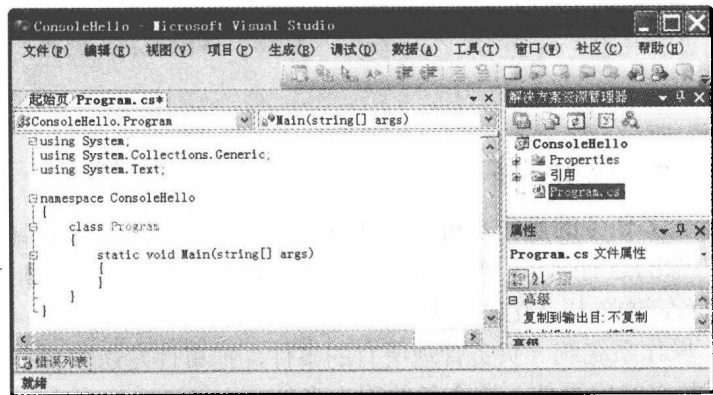


图 1-9 Program.cs 文件生成的程序框架

5) 在编辑器窗口的空的 Program.cs 程序框架中添加如下代码。

```
using System; //引入命名空间
using System.Collections.Generic;
using System.Text;
namespace ConsoleHello //定义名字空间
{
    class Program { //定义类
        static void Main(string[] args) {
            Console.WriteLine("Hello C# World! "); // 输出一行文字
            Console.ReadLine(); // 等待输入，防止窗口一闪而过
        }
    }
}
```

6) 选择“调试”→“启动调试”菜单命令运行程序。运行结果弹出 Windows 的命令窗口，在其中显示“Hello C# World!”字样，如图 1-10 所示。

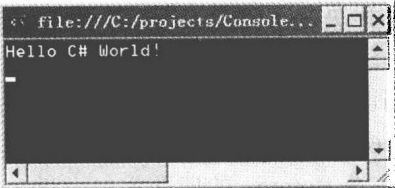


图 1-10 程序运行结果