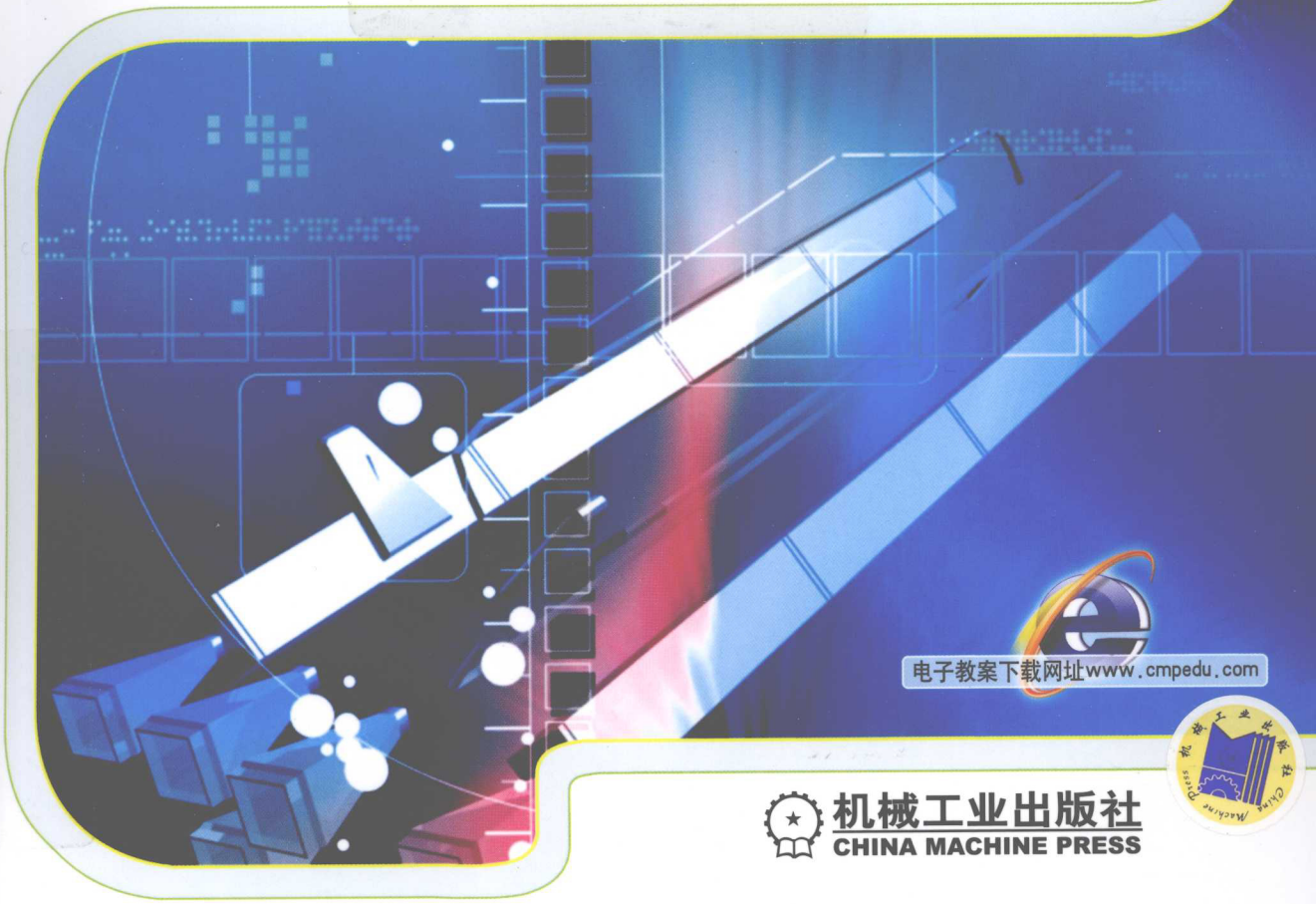
全国高等职业教育规划教材

C# 2008程序设计基础案例教程

江　南　主编



电子教案下载网址www.cmpedu.com

机械工业出版社
CHINA MACHINE PRESS

全国高职高专教育规划教材

C# 2008程序设计基础案例教程

主编

化学工业出版社

全国高等职业教育规划教材

C# 2008 程序设计基础案例教程

主编　江　南

参编　余新华　王永刚　都　娟

机械工业出版社

本书主要介绍 C#.NET 平台、.NET 框架及集成开发环境下的程序设计基础知识，分为数据基础、编程基础和高级应用 3 部分，旨在帮助读者建立程序设计的基本思想，掌握程序设计的基本方法。本书用一个完整的项目贯穿全书，把项目分解成若干个模块，每个模块包含课程学习所需的基本知识点，在每一部分的知识点中先提出目标，再介绍解决方案，通过大量的实例和实验，并结合理论讲解来帮助学生进一步理解程序设计的方法和思路，掌握面向对象程序设计的基本方法与技术。全书共分为 12 章，其中第 12 章是综合开发实例。每一章后面均有习题，可供读者复习参考。

本书可作为高职高专院校计算机专业的教材或教学参考书，也可作为程序开发人员的入门培训教材或参考书。

图书在版编目（CIP）数据

C#2008 程序设计基础案例教程 / 江南主编. —北京：机械工业出版社，2010.4
（全国高等职业教育规划教材）
ISBN 978-7-111-30155-4

Ⅰ. ①C… Ⅱ. ①江… 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 048528 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 常建丽
责任印制：乔 宇

北京机工印刷厂印刷（三河市南杨庄国丰装订厂装订）

2010 年 4 月第 1 版·第 1 次印刷
184mm × 260mm·16 印张·395 千字
0001—4000 册
标准书号：ISBN 978-7-111-30155-4
定价：26.00 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
社服务中心：（010）88361066
销 售 一 部：（010）68326294
销 售 二 部：（010）88379649
读者服务部：（010）68993821

网络服务
门户网：http://www.cmpbook.com
教材网：http://www.cmpedu.com

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专业
编委会成员名单

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，并有多个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

C#.NET 是程序开发所应用的编程语言的一个主流方向，广泛应用于各类软件项目的开发。根据新浪网的调查，2007 年 IT 技术人员主要使用的基于.NET 平台的开发语言是 C#，占到 56.3%；C#吸取了 Delphi，JAVE，C++，VB 这些语言的很多优点。从调查数据分析，目前使用.NET Framework 2.0 的用户多达 67.4%。显而易见，C#已经占尽商机，甚至已成为参与企业应聘时的第一项基本能力。掌握 C#语言将成为计算机专业学生在专业学习上的一种基本能力。一本能够理论联系实际、使学生能够掌握基本编程经验的教材能使学生受益匪浅。

本书以应用广泛的 C#2008 平台为语言基础，用一个完整的项目贯穿全书，把项目分解成若干个模块，每个模块包含课程学习所需的基本知识点，在每一部分的知识点中先提出学习目标，再介绍解决方案，并对解决方案中涉及到的知识点作适当的展开，既有实践操作的步骤，又能够联系实际讲解理论知识，使学生不仅能够学习到操作要点，而且可以结合理论知识进一步提高对其的掌握，为学生学习其他知识打下良好的基础。

本书以高职高专院校计算机专业教学为主，内容通俗易懂，在内容的安排上力求循序渐进，由浅入深，先通过案例介绍让学生理解程序设计过程，使学生对程序有一个直观的了解，再详细阐述设计过程中用到的理论知识，并做到举一反三，同时配套了相应的实验和实训来加深对相应知识点的掌握。

本书第 1、12 章由江南编写，第 4～6、8 章由余新华编写，第 9～11 章由王永刚编写，第 2、3、7 章由都娟编写。

本书在编写过程中得到了机械工业出版社和重庆电子工程职业学院郎登何老师的大力支持，在此表示衷心的感谢。由于时间仓促与编者水平有限，书中难免存在疏漏或错误之处，恳请广大读者不吝赐教。

为了配合教学，本书提供了电子教案，读者可在机械工业出版社网站 www.cmpedu.com 下载。

编　者

目　录

第1章 概 述

本章要点

- ✧ .NET 的基本概念
- ✧ VS.NET 开发环境
- ✧ 使用 C#开发应用程序的基本步骤与方法
- ✧ C#基本语法
- ✧ 使用 C# 编程的相关概念

本章首先介绍了.NET Framework 的基本概念和 Visual Studio 2008 开发环境，之后通过一些简单示例演示了使用 C# 2008 开发项目的基本步骤，并以此了解 C#语言的基本语法。

1.1 .NET 的基本概念

在.NET 框架公布于众之前，在 Windows 平台上所进行的面向构件的软件开发大都采取 COM 构件的形式。组件对象模型（Component Object Model，COM）也称为构件对象模型，是用于创建可重用二进制构件的一个编程标准。按照这个标准，开发者可以编写较少的代码来解决较小的问题域中的问题。通过将问题分解为构件，从而简化解决方案，用于解决某一问题的构件或解决其他类似问题。

COM 在具有诸多优点的同时，也存在不少缺点。COM 标准中最常见的一个问题，被业界比喻为“DLL 地狱”。当 COM 接口在注册表中被登记和索引之后，如果又发布了 COM 构件的新版本，“DLL 地狱”问题便会产生。由于 COM 标准与二进制构件的紧耦合性质，版本控制经常让软件开发者感到头疼。不仅如此，如果 DLL 在文件系统中的存储位置发生变化，而 DLL 在注册表中的信息没有被相应地修改，则 DLL 中的 COM 接口将成为不可访问的接口。例如，当我们安装了一个新的应用程序后，如果这个应用程序使用了与其他应用程序共享的某个构件的新版本，便可能导致新安装的应用程序不能正常工作，还会破坏使用这个共享构件的所有的其他应用程序。

COM 构件在软件开发中还存在其他一些问题，包括内存管理困难、拖延开发时间、所提供的典型 GUI 控件不能完全满足许多开发任务的需要，以及缺乏语言之间的互操作性，如 C++和 Visual Basic 之间的互操作性等。

为了解决软件开发中的上述问题和其他问题，Microsoft 开始研究后来被称为 COM+ 2.0 的解决方案。这一解决方案旨在设计实现一个托管环境。在该环境中，执行的代码提供了增强的类型安全性、代码安全性以及丰富的几乎令人难以置信的实用类库和函数库，使得开发者更容易完成开发工作。此解决方案最终演化成现在的.NET 框架。.NET 框架的 1.0 版、1.1 版和 2.0 版此前已相继发布，现在 3.0 版也已问世。

1.2 .NET 框架的工作原理

在介绍 C#语言的基础知识之前，需要了解一些 C#语言和所有其他.NET 语言的工作机理、这些语言在.NET 框架中的位置，以及它们与.NET 框架中其他组成部分之间的关系。

.NET 框架的一个重要组成部分是公共语言运行时（Common Language Runtime，CLR）。C#与以往的 C++语言及其他类似语言不同，C#语言运行于一个托管（Managed）的环境。用 C# 语言编写的代码，在公共语言运行时的上下文中执行。公共语言运行时负责管理内存和安全性，并将 C#代码与其他非托管代码隔离，以使应用程序的正常操作不受恶意的或设计不良的代码所影响。

图 1-1 显示了.NET 框架的层次结构。

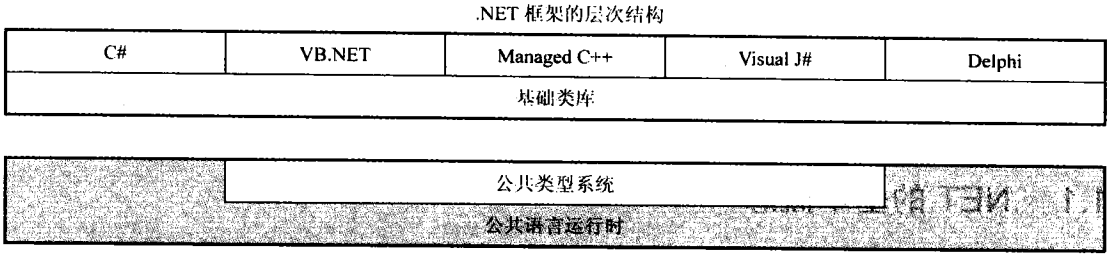


图 1-1 .NET 框架的层次结构

.NET 框架的最上层是各种.NET 语言（图 1-1 所示中列出的只是 CLR 上能够运行的语言中的几种）。在这些语言之下的是基础类库（Base Class Library，BCL），它是在.NET 框架中事先编写的各种类和实用程序的集合，它提供了创建应用程序所需的基本服务，如用于加密、数据访问、文件 I/O、创建 Web 应用和创建 Windows 应用的程序代码等。CLR 是一种驱动大家在 BCL 中编写的各种代码和创建的各种应用程序的托管的执行引擎。

下面介绍.NET Framework 的一些重要特性。

1．公共类型系统

用一种语言编写的程序缺乏与用另一种语言编写的程序进行数据交换的能力，这是许多编程语言存在的一个共同问题。例如，由于不同语言中参数的传递顺序存在差别，在一个 Pascal 程序中调用 C 语言编写的方法时，需要格外小心。在 C 程序中调用 Pascal 方法时，也是如此。另外，C 语言中的字符串是一个以 ASCII 码字符 NULL（在程序中通常用'\0'表示）为结尾的字符数组。而在 Pascal 语言中，字符串实际上是一个字符数组，数组中的第一个字节实际上包含了字符串的长度。

上述问题的一个解决方案是，允许所有的语言共享对数据类型的某种公共表示。公共类型系统（Common Type System）恰恰提供了一个公共的数据类型集合。例如，如果你引用了一个 VB.NET、C#、J#、Delphi (.NET)、托管 C++或其他任何一种.NET 语言的字符串，公共类型系统能够确保你所引用的字符串对这些不同的语言来说是完全相同的实体。这是因为 string 类型是在.NET 框架本身中定义的数据类型，而不是在语言中定义的数据类型。让数据类型的定义与编程语言分离，便能够创建一个允许开发者采用 VB.NET 和 C#语言混合编程，且不存在通信问题的编程环境。

2．清除垃圾：在垃圾收集环境中编码

此前提到，CLR 能够为开发者分担许多内存管理工作。如果你使用过 C# 之前的版本，那么对垃圾收集这一概念便不会陌生。

CLR 中的一个构件是垃圾收集器（Garbage Collector，GC）。当用 C#语言声明了一些新的变量（有关 C#中的变量声明，将在下一章介绍），这些变量就被纳入垃圾收集器的管理之下。当一个回收周期到来时，垃圾收集器将对变量执行检查，如果某些变量不再被使用，垃圾收集器将销毁这些变量（称为回收）。有关垃圾收集器的更多内容，读者可以参阅 MSDN。现在，你只需知道在你和你声明的变量背后隐藏了一个帮你管理内存和清除废物的垃圾收集器便足够了。

3．Microsoft 中间语言（MSIL）

和传统的编程语言不同，.NET 编译器不会生成能直接进入 CPU 执行的本机代码。相反，它生成的是所谓的 Microsoft 中间语言（MSIL 或 IL）代码，这是一种用于某种虚拟处理器的机器语言，这种虚拟处理器不是当前市场上可得到的任何一种商用 CPU。尽管 IL 代码比大多数现代编程语言低级，但它比纯 Intel 汇编语言要高级一些。IL 是一种面向堆栈的语言，它不仅能直接对 CPU 寄存器寻址，并且能够察觉高级概念，如异常和对象创建等。

IL 代码不能直接被 CPU 识别执行，它运行在 CLR 平台之上，因此要运行由.NET 编写的应用程序，应先安装.NET Framework。

4．即时编译（JIT）

既然 CPU 不能直接执行 IL 代码，那 CLR 又是如何将 IL 代码转换成本机代码并执行呢？通过了解.NET 应用程序开始执行时所发生的过程，就可以找出这个问题的答案。

1）所有的.NET 可执行文件在启动后都首先加载 mscoree.dll，这个文件实际上是.NET 运行库，然后调用该 DLL 文件所提供的_CorMainExe 函数。

2）_CorMainExe 中的代码查看存储在可执行文件内的元数据，以获取应用程序入口点（通常是 Sub Main 函数）的地址。

3）入口点中的 IL 代码被传递给实时（Just-In-Time，JIT）编译器（该编译器位于.NET 运行库中），转换为本机代码，然后执行该代码。

4）当主函数调用其他方法时，.NET 运行库使用 JIT 编译器将这些方法中的 IL 代码转换成本机代码，然后执行该代码。在应用程序生存期内，每一种方法只能进行一次这种实时编译，因为当再次调用某一方法时，本机代码会被保存在内存中并被调用。

5．异常处理

.NET 提供了一种基础结构，让面向.NET 的编译器支持异常处理。特别是，它提供了一组.NET 类来表示异常，语言的互操作性则允许错误处理代码处理被抛出的异常对象，无论错误处理代码使用什么语言编写，都是如此。错误处理对象包括了给用户提供的相应信息和在代码的某个地方检测到错误的确切信息。本书将在第 5 章详细讨论异常。

.NET Framework 的最新版本是 3.0，这个版本引入了一些全新的功能。在这个版本中，可以用 Windows Presentation Foundation（WPF）建立新的应用程序类型，基于 Windows Communication Foundation（WCF）、Windows Workflow Foundation（WWF）和 Windows CardSpace 的应用程序和库。

1.3 浏览开发环境

Visual Studio 2008 包含了一系列高效的、智能的开发工具，方便开发人员进行开发。在正式进入 C#编程语言学习之前，需要先掌握集成开发环境的使用。

启动 Visual Studio 2008 后，首先看到的就是中间的起始页。使用起始页可以轻松地访问或创建项目、阅读最新的开发文章。若要访问起始页，请在“视图”菜单中选择“其他窗口”，然后单击“起始页”。图 1-2 所示就是 Visual Studio 2008 开发环境的起始页。

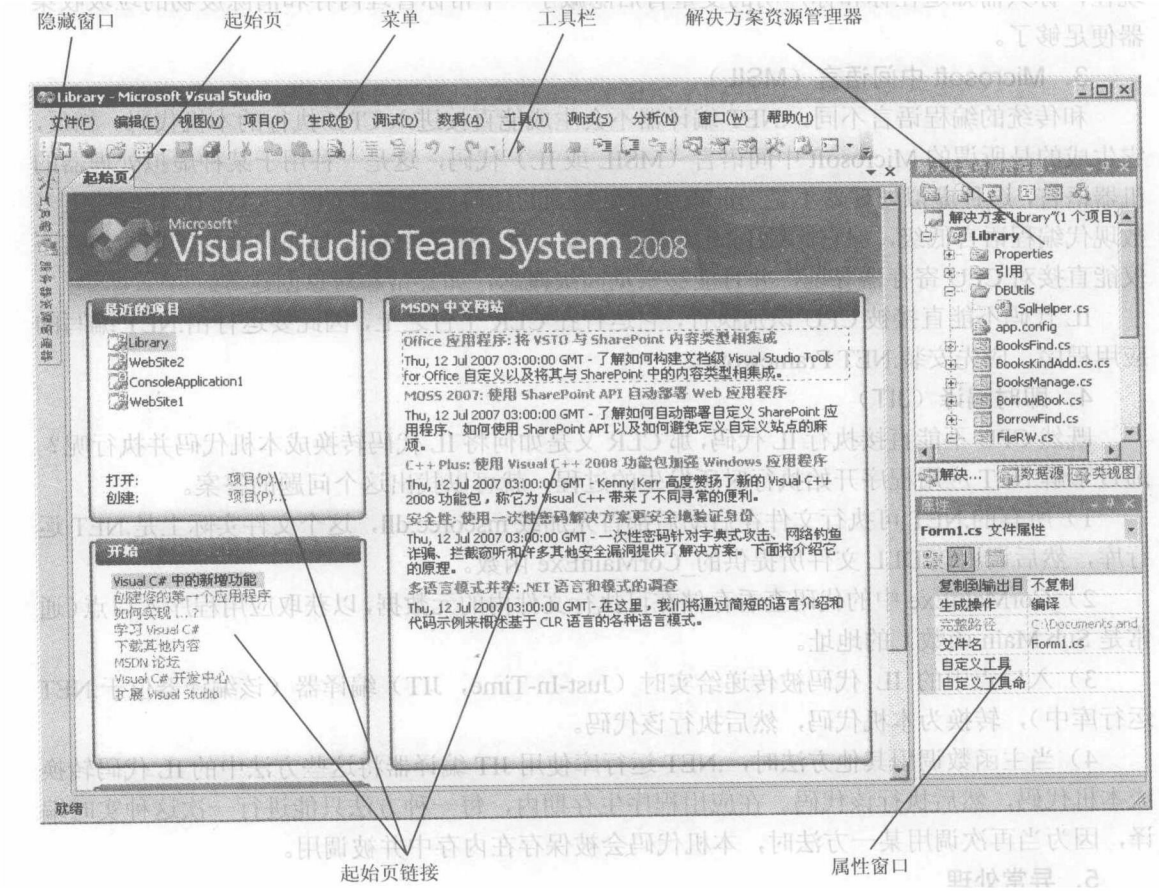


图 1-2 Visual Studio 2008 开发环境的起始页

可以使用“工具”菜单的“选项”子菜单打开如图 1-3 所示的“选项”对话框，在对话框中可以根据自己的需要配置集成开发环境。例如，可以建立项目的默认保存位置，改变窗口的默认外观和行为，还可以创建常用命令的快捷方式。

下面介绍开发环境中的几个主要窗口。

1. 解决方案资源管理器

Visual Studio 2008 是以解决方案（Solution）和项目来组织资源的。解决方案就是要创建的应用程序，应用程序下的各个模块可以被建立成一个个的项目。解决方案和项目还可以包

含一些项，这些项表示创建应用程序所需的引用、数据连接、文件夹和文件。一个解决方案可包含多个项目，而一个项目通常包含多个项。项目和项目以及项目和解决方案之间的连接可以通过解决方案资源管理器管理。图 1-4 所示就是解决方案资源管理器窗口。

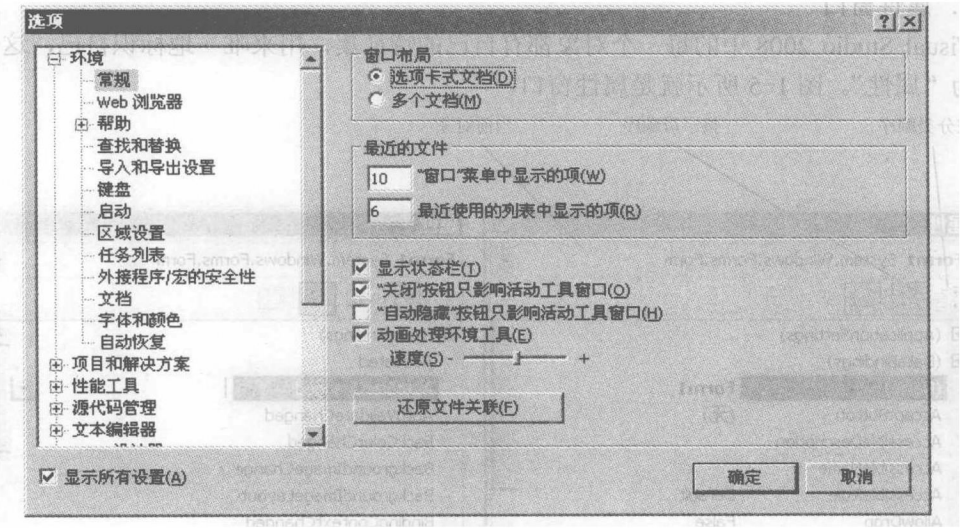


图 1-3 “选项”对话框

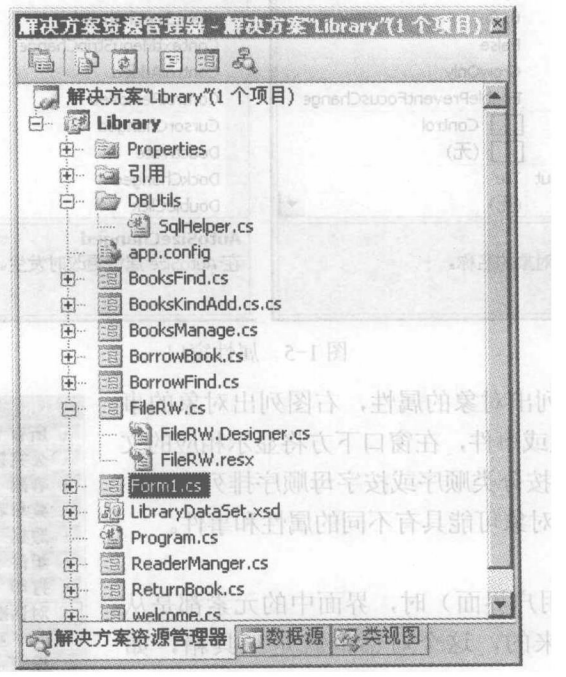


图 1-4 解决方案资源管理器窗口

解决方案资源管理器提供项目及其文件的有组织的视图，并且提供对项目和文件相关命令的便捷访问。与此窗口关联的工具栏提供适用于列表中突出显示的项的常用命令。若要访

问“解决方案资源管理器”，可在视图菜单上选择“解决方案资源管理器”。

图 1-4 所示是标准的“解决方案资源管理器”视图，它以树状形式列出了当前解决方案的所有项。在该窗口上可以为当前解决方案添加或删除项，也可以执行其他管理任务。

2．属性窗口

Visual Studio 2008 中的每一个对象都有自己的特征集，用来唯一地标识自己，这个特征集称为“属性”。图 1-5 所示就是属性窗口。

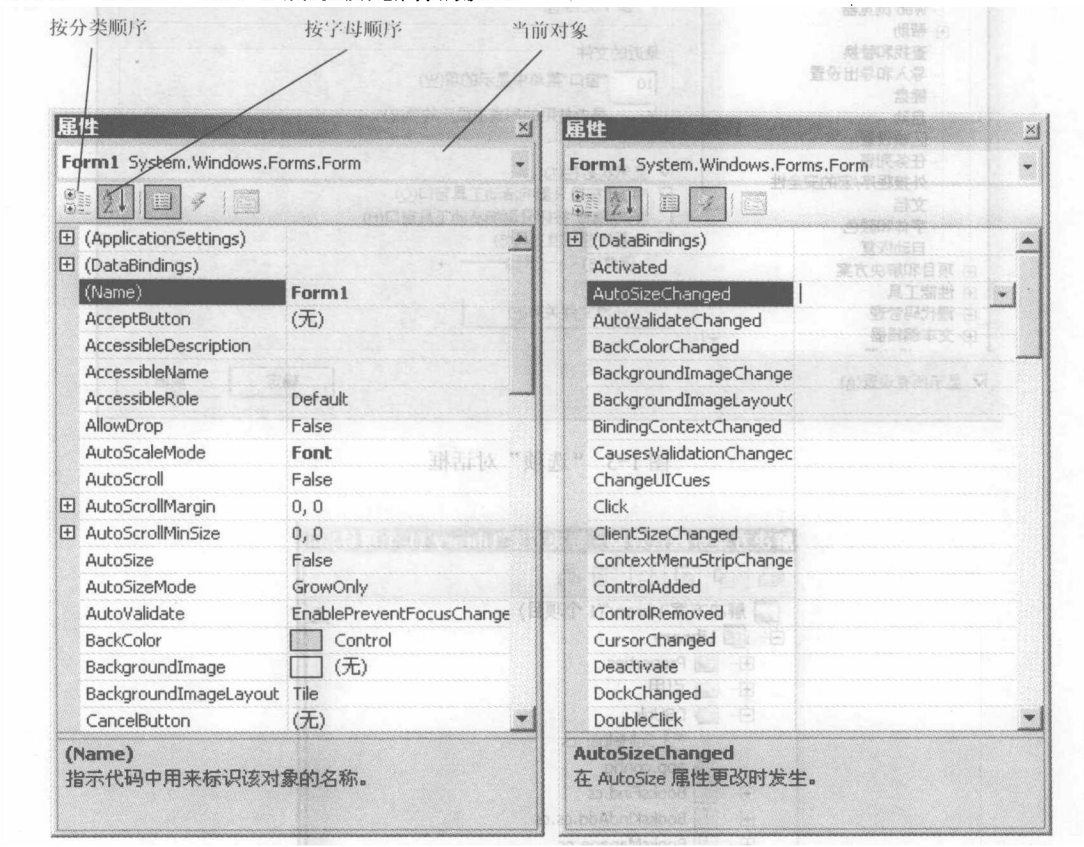


图 1-5　属性窗口

图 1-5 中的左图列出对象的属性，右图列出对象的事件。选择其中一个属性或事件，在窗口下方将显示相应的文字提示。用户可以选择按分类顺序或按字母顺序排列窗口中的属性或事件。不同的对象可能具有不同的属性和事件。

3．工具箱

绘制 GUI（图形用户界面）时，界面中的元素都是从其他对话框中拖曳过来的，这个对话框就是工具箱，如图 1-6 所示。

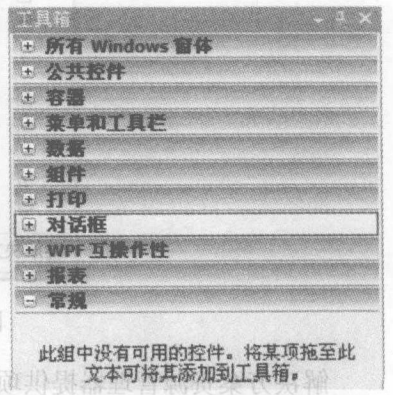


图 1-6　工具箱

可将每个工具箱中的图标拖放到设计视图的界面上，这将会创建相应控件的实例。工具箱中有多个选项卡，对其中的控件进行分类，单击选项卡旁边的“+”，可以展开