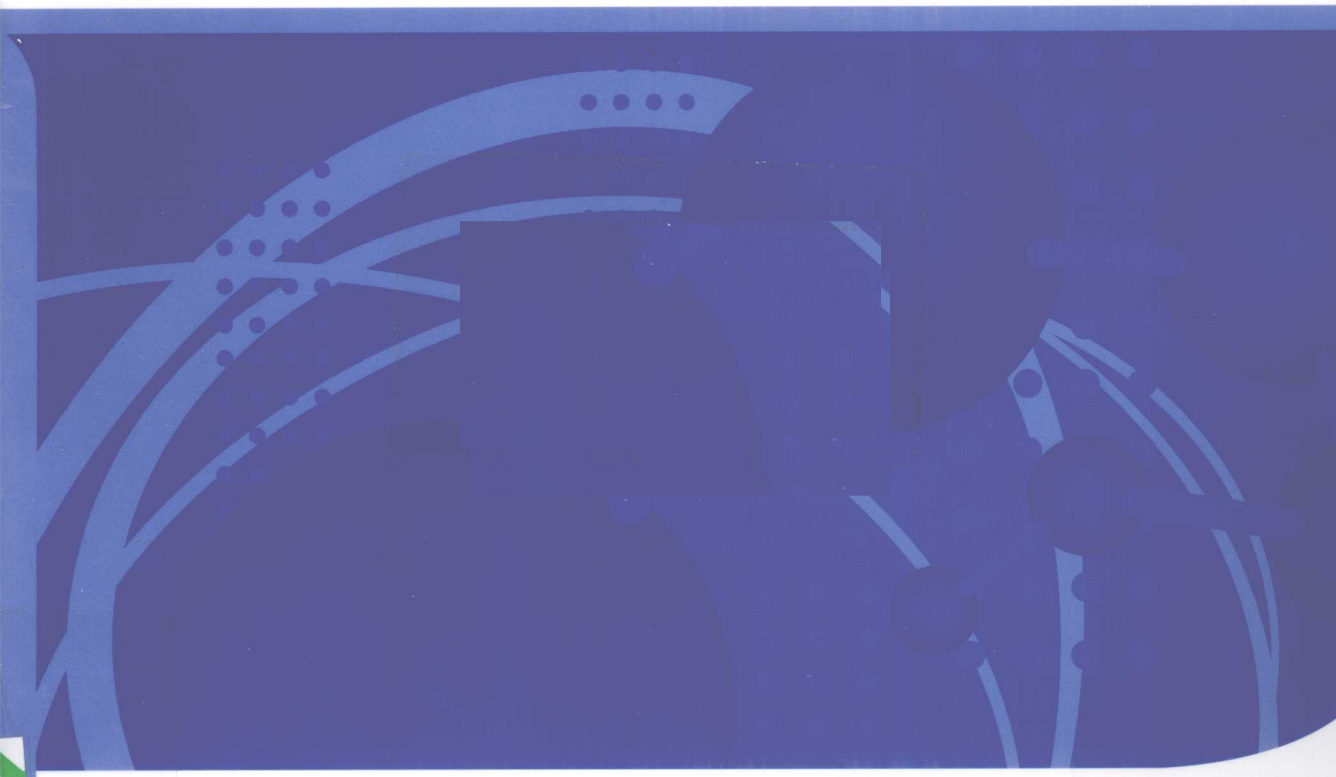
全国高职高专教育“十一五”规划教材

Visual C# 2008 应用开发教程

董淑娟



高等教育出版社
Higher Education Press

全国高职高专教育“十一五”规划教材

Visual C# 2008 应用开发教程

Visual C# 2008 Yingyong Kaifa Jiaocheng

董淑娟

高等教育出版社·北京
HIGHER EDUCATION PRESS BEIJING

内容提要

本教材按照基于工作过程的要求编写，通过一个完整的案例设计，全面细致地介绍了 Visual C#面向对象编程的概念和方法，包括 C#概述、模拟银行系统项目任务概述、C#基本语法、程序的基本结构、模拟银行系统的实体类设计、模拟银行系统操作类中方法的设计、ADO.NET 类在银行系统中的应用、银行系统主要操作类的实现、模拟银行系统的窗体设计、银行系统中水晶报表的应用、异常和调试以及继承和接口的实现。本教材以 Windows 窗体应用程序设计为主线，以银行系统的开发为主导，力求让读者通过本教材的学习，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。本教材每章均设计有与模拟银行系统配套的习题，以方便学生练习。

本教材可作为培养应用型，技能型人才的计算机相关专业的教材，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# 2008 应用开发教程/董淑娟编. —北京：高等教育出版社，2010.3

ISBN 978-7-04-028846-9

Ⅰ. ①V… Ⅱ. ①董… 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 011675 号

策划编辑 杜 冰 **责任编辑** 许兴瑜 **封面设计** 张志奇 **责任绘图** 尹文军
版式设计 张 岚 **责任校对** 王效珍 **责任印制** 韩 刚

出版发行	高等教育出版社	**购书热线**	010-58581118
社 址	北京市西城区德外大街 4 号	**咨询电话**	400-810-0598
邮政编码	100120	**网 址**	http:// www.hep.edu.cn
总 机	010-58581000		http:// www.hep.com.cn
		网上订购	http:// www.landraco.com
经 销	蓝色畅想图书发行有限公司		http:// www.landraco.com.cn
印 刷	北京鑫丰华彩印有限公司	**畅想教育**	http:// www.widedu.com
开 本	787×1092 1/16	**版 次**	2010 年 3 月第 1 版
印 张	19.5	**印 次**	2010 年 3 月第 1 次印刷
字 数	470 000	**定 价**	25.10 元

物料号 28846-00

前 言

Visual C# 2008 是微软公司推出的 Visual Studio 2008 环境的重要组成部分，是面向对象的编程语言。Visual C# 2008 在保持 C 语言优美表示形式的同时，实现了应用程序的快速开发。

本教材结合模拟银行系统展开，逐步讲授 Visual C# 2008 程序设计的整个过程。本教材采用简明的语言描述解决问题的方法，使读者能够轻松地理解面向对象编程的基本概念与思想，感受在学习中获取知识的乐趣。

本教材在内容编排上采用“基于工作过程”的方式，先为任务的开展给出相关的知识储备，并配备有相关的实例，然后依据知识储备和模拟银行系统项目完成相关的任务。程序的实现以详尽的表述结合图例来说明，代码设计给予算法描述，以便读者对代码的编写有理可循。本教材操作性强，以一个完整的案例项目贯穿全书，并将分散的知识点合理地融入其中。项目采用三层结构实现：数据处理层、业务逻辑层、业务表现层，能有效地帮助读者建立项目开发的理念，符合“基于工作过程”的教学要求，是学习完整项目开发的好选择。

编者多年讲授程序设计语言，具有丰富的程序设计教学经验，以读者接受新知识的特点和培养读者编写代码的方法为出发点设计编排教材内容。旨在通过本教材的学习，让读者在掌握 C#程序设计基本知识的同时，还掌握 Windows 项目开发的方法与步骤。

本教材可作为培养应用型、技能型人才的计算机相关专业的教材，完成本教学的时数为 120 学时（其中授课为 60 学时，上机练习为 60 学时）左右。

本教材由董淑娟编写，由于编者水平有限，书中疏漏和不足之处在所难免，敬请广大读者批评指正。

编 者

2009 年 11 月

目 录

1 第1章 Visual C#概述

学习目标

① 了解.NET 平台与.NET Framework。
② 了解 Visual Studio 2008 集成开发环境。
③ 了解软件的开发过程。
④ 了解 C#程序的结构。
⑤ 掌握安装开发环境并配置开发环境的过程。
⑥ 掌握不同类型应用程序的创建方法。
⑦ 熟练掌握帮助文档的使用。

Visual Studio 是微软公司推出的集成开发环境，是目前最流行的 Windows 平台应用程序开发环境。Visual Studio 基于微软公司的智能客户端应用程序构想而构建，它使开发人员能够快速创建可提供最高质量和丰富用户体验的应用程序。目前已经发展到 Visual Studio 2008 版本，Visual Studio 2008 使开发人员能够快速创建高质量、用户体验丰富而又紧密联系的应用程序，充分展示了微软公司开发智能客户端应用程序的构想。借助 Visual Studio 2008，采集和分析信息将变得更为简单、便捷，业务决策也会因此变得更为有效。任何规模的组织都可以使用 Visual Studio 2008 快速创建能够利用 Windows Vista 和 Office 2007 的更安全、更易于管理并且更可靠的应用程序。

1.1 Visual Studio 的发展

1998 年，微软公司发布了 Visual Studio 6.0，所有开发语言的开发环境版本均升至 6.0。这也是 Visual Basic 最后一次发布，从下一个版本（7.0）开始，Visual Basic 进化成了一种新的面向对象的语言：Visual Basic.NET。

2002 年，随着.NET 口号的提出与 Windows XP / Office XP 的发布，微软发布了 Visual Studio .NET。在这个版本的 Visual Studio 中，微软剥离出 Visual FoxPro 作为一个单独的开发环境以 Visual FoxPro 7.0 单独销售，同时取消了 Visual InterDev。与此同时，微软引入了建立在 .NET 框架上的托管代码机制以及一门新的语言 C#（读作 C Sharp，意为 C++++）。C# 是一门建立在 C++ 和 Java 基础上的现代语言，是写 .NET Framework 的语言。

2003 年，微软对 Visual Studio 2002 进行了部分更新，并以 Visual Studio 2003 的名义发

布。Visio 作为使用统一建模语言（UML）架构应用程序框架的程序被引入，同时被引入的还包括移动设备支持和企业模版。.NET Framework 也升级到了 1.1 版。

2005 年，微软发布了 Visual Studio 2005。.NET 字眼从各种语言的名字中被抹去，但是这个版本的 Visual Studio 仍然还是面向.NET Framework（2.0 版本）。它同时也能开发跨平台的应用程序，如开发使用微软操作系统的手机程序等。总体来说，它是一个非常庞大的软件，甚至包含代码测试功能。

随着 Windows Vista 和 Office 2007 的发布，Visual Studio 9 也渐渐浮出水面。Visual Studio 9 支持建立于 DHTML 基础上的 AJAX 技术，这种微软在 Visual InterDev 时代提出的基于异步的客户端动态网页技术，在当年并没有像微软预期中的那么流行起来，反而随着 GMail 等的应用而东山再起，渐渐成为主流网络应用之一。同时，Visual Studio 9 会强化对于数据库的支持以及微软新的基于工作流（Workflow）的编程模型。为了保持与 Office 系列的统一，Visual Studio 9 的名称改为 Visual Studio 2008。

2008 年，Visual Studio 2008 发布，它使开发人员能够快速创建高质量、用户体验丰富而又紧密联系的应用程序，充分展示了微软公司开发智能客户端应用程序的构想。借助 Visual Studio 2008，采集和分析信息将变得更为简单便捷，业务决策也会因此变得更为有效。

1.2 Visual Studio 2008

Visual Studio 2008 提供了高级开发工具、调试功能、数据库功能和创新功能，帮助在各种平台上快速创建当前最先进的应用程序。

Visual Studio 2008 包括各种增强功能，例如可视化设计器（使用 .NET Framework 3.5 加速开发）、对 Web 开发工具的大量改进，以及能够加速开发和处理所有类型数据的语言增强功能。Visual Studio 2008 为开发人员提供了所有相关的工具和框架支持，帮助创建引人注目、令人印象深刻并支持 AJAX 的 Web 应用程序。

开发人员能够利用这些丰富的客户端和服务器端框架轻松构建以客户为中心的 Web 应用程序，这些应用程序可以集成任何后端数据提供程序、在任何当前浏览器内运行并完全访问 ASP.NET 应用程序服务和 Microsoft 平台。Visual Studio 2008 在三个方面为开发人员提供了关键改进：

① 快速的应用程序开发。

为了帮助开发人员迅速创建先进的软件，Visual Studio 2008 提供了改进的语言和数据功能，例如语言级集成查询（LINQ），各个编程人员可以利用这些功能更轻松地构建解决方案以分析和处理信息。

② 突破性的用户体验。

Visual Studio 2008 为开发人员提供了在最新平台上加速创建紧密联系的应用程序的新工具，这些平台包括 Web、Windows Vista、Office 2007、SQL Server 2008 和 Windows Server 2008。对于 Web，ASP.NET AJAX 及其他新技术使开发人员能够迅速创建更高效、更强交互性和更个性化的新一代 Web 体验。

③ 高效的团队协作。

Visual Studio 2008 提供了帮助开发团队改进协作的扩展和改进方面的服务项目，包括帮助将数据库专业人员和图形设计人员加入到开发流程的工具。

1.3 Microsoft.NET Framework

.NET Framework 具有两个主要组件：公共语言运行库和 .NET Framework 类库。公共语言运行库是 .NET Framework 的基础。可以将公共语言运行库看做是一个在执行时管理代码的代理，它提供内存管理、线程管理和远程处理等核心服务，并且还强制实施严格的类型安全以及可提高安全性和可靠性的其他形式的代码准确性。事实上，代码管理的概念是公共语言运行库的基本原则。以公共语言运行库为目标的代码称为托管代码，而不以公共语言运行库为目标的代码称为非托管代码。.NET Framework 的另一个主要组件是类库。

1.3.1 公共语言运行库

.NET Framework 的核心是其运行库的执行环境，称为公共语言运行库（CLR）或.NET 运行库。通常将在 CLR 的控制下运行的代码称为托管代码（managed code）。

但是，在 CLR 执行编写好的源代码之前，需要编译它们（在 C#中或其他语言中）。在.NET 中，编译分为两个阶段：

① 把源代码编译为 Microsoft 中间语言（IL）。

② CLR 把 IL 编译为平台专用的代码。

1.3.2 .NET Framework 类库

.NET Framework 类库是一个由 Microsoft .NET Framework SDK 中包含的类、接口和值类型组成的库。该库提供对系统功能的访问，是建立 .NET Framework 应用程序、组件和控件的基础。

.NET Framework 提供了用于解决常见编程任务的构件（预制的软件），从而能够快速构造具有出色的最终用户体验的紧密联系的应用程序。在 .NET Framework 模型业务流程上有效构建的紧密联系的应用程序有利于在异类环境中实现系统集成。

Visual Studio 和 .NET Framework 的结合使用减少了对公用管道代码的需要，从而缩短了开发时间并使开发人员能够集中精力解决业务问题。

.NET Framework 3.5 在 .NET Framework 3.0 的基础上进行了增强。得到增强的功能领域包括基类库、Windows Workflow Foundation、Windows Communication Foundation、Windows Presentation Foundation 和 Windows CardSpace。

1.4 C#应用程序

在 C#中，每一个解决方案可以由一个或多个项目组成，每一个项目可以由一个或者多个类组成，所有的代码都必须封装在某个类中。一个类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#源程序文件的扩展名为.cs，比如WindowsApplication1.cs。常用的C#应用程序分为以下三种：控制台应用程序、Windows窗体应用程序和ASP.NET网站。

1.4.1 控制台应用程序

控制台应用程序适合于对界面的运行速度要求不高的项目，通过命令行的方式实现程序的运行情况，如例1-1。

【例1-1】 使用控制台应用程序创建一个项目，运行显示“hello visual studio!”。

实现的步骤如下：

（1）新建项目

启动Visual Studio 2008，选择菜单栏中的“文件”菜单，在打开的文件菜单中用鼠标指向“新建”，在下一级菜单中选择“项目”，将打开“新建项目”对话框，如图1-1所示。

也可以单击工具栏中的新建项目按钮，或者在起始页的“项目”选项卡中单击“新建项目”按钮，以打开“新建项目”对话框。在“项目类型”列表框中选择“Visual C#”，这时在“Visual Studio已安装的模板”列表框中列出了Visual C#可以创建的各种项目，选择“控制台应用程序”。在对话框的右上方的下拉列表框中可以选择.NET Framework的不同版本，这里选择.NET Framework 3.5，也就是最新版本的.NET Framework。然后在对话框的下方“名称”文本框中填写项目的名称，在“位置”文本框中输入项目的存储路径，也可以通过“浏览”按钮选择项目的存放位置，在“解决方案名称”文本框中可编辑该项目默认的解决方案名称。

创建新项目时，如果在Visual Studio 2008中有已打开的项目，在“解决方案”下拉列表框中将显示“创建新解决方案”和“添入解决方案”选项，选择“添入解决方案”选项，将把创建的项目包含在当前的解决方案中；如果选择“创建新解决方案”选项，则关闭当前解决方案，将创建的项目放在新的解决方案中。

设置完成后单击“确定”按钮，即创建了一个基于控制台的应用程序，并打开代码编辑窗口，如图1-2所示。

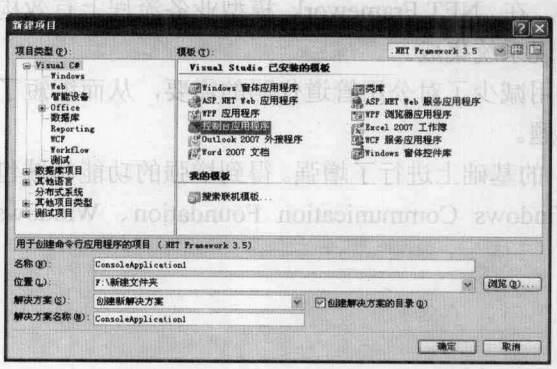


图1-1 “新建项目”对话框

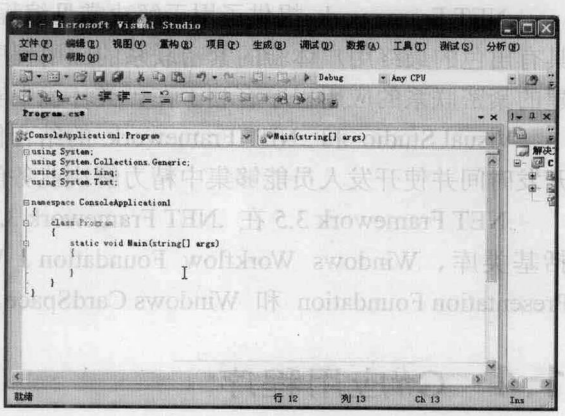


图1-2 代码编辑窗口

（2）编写代码

在代码编辑窗口中带有 using 的代码为系统的命名空间，用于添加 .NET Framework 类库的引用，或者其他自定义类的引用。在新建项目时，系统自动添加一些必需的类库名称。如下面的代码所示：

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
```

“namespace ConsoleApplication1”是该项目的命名空间，namespace 为保留字，ConsoleApplication1 为该项目的名称，必要时可以更改该名称，可以与项目的名称不一致。

在该项目的命名空间内可以包括一个或多个类，多个类可以在一个文件中，也可以是多个文件，但是一个项目有且只有一个 Main()方法，程序从该方法开始运行。该例的代码如下：

```
namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("hello visual studio!");
            Console.Read();
        }
    }
}
```

值得一提的是，控制台应用程序和 Windows 窗体应用程序有且仅有一个 Main()方法，它是程序运行的入口点。

（3）编译程序

源代码编写完成后，单击工具栏中的“启动”按钮 ▶，或者按 F5 键，或者选择“调试”菜单中的“启动”命令，将编译该程序；如果编译没有错误，即运行程序，在控制台窗口显示运行结果。

在 Visual Studio 2008 开发环境下，运行时，运行结果屏幕一闪就过去了，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.Read ();”语句，意思是读取键盘输入的字符，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.Read ();”语句，并在 Visual Studio 2008 开发环境下直接按 F5 键编译并运行。

（4）保存程序

通常只要执行启动命令编译运行程序或者执行“生成”菜单中的“生成解决方案”命令，

程序即予以保存，而不需再专门进行保存。

如果程序被修改，并且没有被生成或运行，在关闭 Visual Studio 2008 环境或关闭解决方案时，会询问是否保存更改。

如果需要对程序进行保存而又不想生成、运行程序或关闭解决方案，这时可以单击工具栏中的“全部保存”按钮，或选择“文件”菜单中的“全部保存”命令，以保存项目。

1.4.2 Windows 窗体应用程序

Windows 窗体应用程序是在 Windows 操作系统中以图形界面运行的程序，可以使用鼠标进行相关的操作，设计时可以表现为窗体，并且可以利用开发环境提供的控件设计窗体，自动生成部分设计代码。

【例 1-2】 使用 Windows 窗体应用程序实现例 1-1。

实现的步骤如下：

（1）新建项目

启动 Visual Studio 2008，在菜单栏中选择“文件”→“新建”→“项目”命令，打开“新建项目”对话框，在“项目类型”列表框中选择“Visual C#”，在“Visual Studio 已安装的模板”列表框中选择“Windows 窗体应用程序”，并设置项目的名称和解决方案的名称以及项目的存放路径。

（2）窗体的设计

窗体是 Windows 应用程序中最常见的对象，也是程序设计的基础。各种控件对象必须建立在窗体上。同 Windows 环境下的应用程序窗口一样，窗体具有控制菜单、标题栏、最大化/还原按钮、最小化按钮、关闭按钮以及边框。通过窗体的属性设置可以达到我们想要的效果。

在创建 C#的 Windows 窗体应用程序项目或 Web 应用程序项目时，Visual Studio 2008 会自动提供一个窗体，但是一个应用程序往往是由多个窗体组成的，要为应用程序添加窗体，可以通过“解决方案资源管理器”的右键快捷菜单或“项目”菜单中的“添加 Windows 窗体”命令来实现。窗体常用的属性如表 1-1 所示。

表 1-1 窗体属性

属性名称	说明
Name	窗体的名称，该名称既是窗体对象的名称，也是保存在磁盘上的窗体文件的名称，窗体文件的扩展名为.cs。
BackColor	窗体颜色。
BackgroundImage	窗体背景图片设置。
Enabled	窗体是否启用。
Font	窗体中控件默认的字体、字号与字形。
ForeColor	窗体中控件文本的默认颜色。
Location	窗体相对于容器左上角的位置，通常程序主窗体应是相对于屏幕（桌面）左上角的位置。
Locked	窗体是否可以移动和改变大小。
MaximizeBox	窗体是否具有最大化/还原按钮。
MinimizeBox	窗体是否具有最小化按钮。

续表

属性名称	说明
Opacity	窗体是透明状态、半透明状态还是不透明状态。
StartPosition	窗体运行时在屏幕上显示的位置。
Text	窗体标题栏中显示的标题内容。
AcceptButton	使用键盘 Enter 键触发该按钮的单击事件。
CancelButton	使用键盘 Esc 键触发该按钮的单击事件。
MainMenuStrip	关联到该窗体上的菜单。

本例只用到一个窗体，所以该窗体为应用程序提供的窗体。

将窗体的 Name 属性和 Text 属性修改为“hello visual studio!”，StartPosition 属性设置为 CenterScreen，即运行时窗体处于屏幕中间的位置。

① 添加控件

在工具栏中单击“工具箱”按钮，或者在菜单栏上选择“视图”→“工具箱”命令，在窗体设计器的左边显示“工具箱”，在展开的窗格中单击“所有 Windows 窗体”选项卡中的 Label 控件，然后在窗体中单击，或者直接拖动控件到窗体中，这样就为窗体添加了一个标签。

② 修改控件的属性

在工具栏中单击“属性窗口”按钮，或者在菜单栏上选择“视图”→“属性窗口”命令，在窗体设计器的右边显示属性窗口。

接受默认 Label 控件的名称，设置 Label 控件的 Text 属性为“hello visual studio!”。

（3）编写代码

双击窗体，或按 F7 键，打开代码编辑窗口。本例不用写任何代码。

（4）运行调试程序

单击工具栏上的“启动”按钮，运行程序。如果显示的信息在窗口中没有居中，关闭运行的程序，用鼠标拖动控件“label1”进行调整，然后再运行程序以观察效果，反复调整和运行，直到满意为止。程序运行结果如图 1-3 所示。

图 1-3 程序运行结果

单击运行结果窗口右上角的“关闭”按钮，结束程序，回到编辑状态。

（5）保存程序

单击工具栏的“全部保存”按钮，保存修改结果。其实当在集成开发环境中单击“启动”按钮运行 C#程序后，该程序即被完全保存，如果之后未作任何修改，就不需要专门保存。只有在对 C#程序作了修改，又未运行，这时如要保存修改结果，才需要专门保存。

1.4.3 ASP.NET 网站

【例 1-3】 创建一个 ASP.NET 网站，运行时界面显示“hello visual studio!”。

（1）新建项目

进入 Visual Studio 2008 开发环境，选择“新建网站”→“ASP.NET Web 网站”，修改“位

置”为“文件系统”，并指明要创建的网站保存路径，比如 D:\My Documents\Visual Studio 2008\WebSites\WebSite1，如图 1-4 所示。单击“确定”按钮，就会出现如图 1-5 所示的画面。

（2）界面设计

切换到设计模式，从“工具箱”中向设计窗体拖放一个 Label 控件，设置其 Text 属性为“hello visual studio!”，如图 1-5 所示。

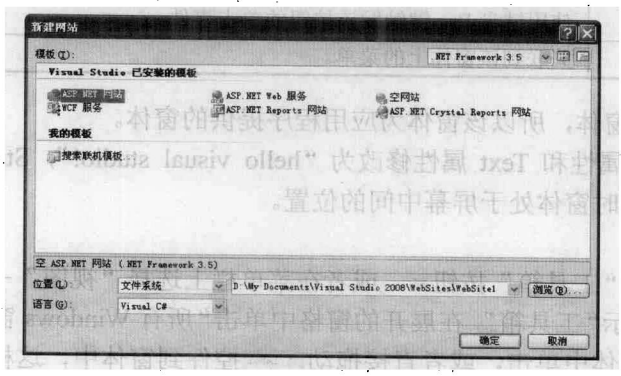


图 1-4 新建网站

（3）浏览页面

单击鼠标右键，在弹出的快捷菜单中选择“在浏览器中查看”命令，屏幕上就会弹出一个类似于 IE 浏览器的窗口，并在其上面显示有“hello visual studio!”。

图 1-5 网页设计界面

1.5 C#程序结构分析

以例 1-1 为例，分析 C#程序结构。

1.5.1 命名空间

例 1-1 程序的第一行代码 using System 表示导入 System 命名空间，该命名空间是基类命名空间之一，程序中的 Console 类在该名字空间中，即 System 命名空间中定义了 Console 类，该类提供了字符的界面输入输出功能。System.Collections.Generic 命名空间包含定义泛型集合的接口和类，可以使用泛型集合来创建强类型集合，这种集合能提供比非泛型强类型集合更好的类型安全性和性能。System.Linq 命名空间支持使用语言集成查询（LINQ）进行查询的类和接口。System.Text 命名空间包含表示 ASCII、Unicode、UTF-7 和 UTF-8 字符编码的类，用于将字符块转换为字节块和将字节块转换为字符块的抽象基类，以及操作和格式化 String 对象而不创建 String 的中间实例的 Helper 类。在后面的内容中将渐渐了解。

C#程序是以命名空间组织代码的，只有在导入命名空间后，才能使用该命名空间的类或对象，命名空间的声明格式是关键字“namespace”后跟命名空间和命名空间主体，然后跟一个分号，如例 1-1 程序命名空间 namespace ConsoleApplication1。

访问某一个命名空间的某一类时，需要在程序的开始使用 using 关键字导入命名空间，或者在访问类时，通过以下格式访问：

命名空间.类名

如果不想在程序开始导入命名空间，也可以在使用时导入，如 Console 类，可以通过以下方式访问：

```
System.Console.WriteLine("hello visual studio!");
```

使用同样的方法可以访问到用户定义的命名空间。

1.5.2 类

C#语言是完全面向对象的，所以要求程序中每个元素都要属于一个类，例 1-1 中系统默认声明类 Program。与很多语言一样，C#程序中每一对“{}”构成一个程序块，大括号可以嵌套，即程序块内可以出现子程序块，可以嵌套任意层，但是“{”与“}”必须成对出现。

1.5.3 Main()方法

项目的命名空间内可以包括一个或多个类，多个类可以在一个文件中，也可以是多个文件，但是一个项目有且只有一个 Main()方法，程序从该方法开始运行。C#规定，Main()方法是程序的入口，当程序执行时，直接调用该方法。该方法必须包含在一个类中。

一个 C#程序中有且只有一个 Main()方法，否则，出现错误。

1.5.4 注释

任何语言都支持注释，因为程序的编制过程中比较重要或比较难以理解的地方往往需要加以说明，而这种说明不能参与程序的运行，这就需要由注释来完成。注释是写在程序中的解释性文字，合理的注释会提高代码的阅读性和可理解性。C#提供了三种注释风格。

① 在程序的一行中以“//”开始注释，其后可以编写任何内容，只要这些内容在一行上即可。如下面的注释语句是正确的：