HZ BOOKS
华章教育

计 算 机 基 础 课 程 系 列 教 材

C#程序设计教程

第2版

郑阿奇 梁敬东 主编
朱毅华 时跃华 赵青松 编著

机械工业出版社
China Machine Press

计 算 机 基 础 课 程 系 列 教 材

C#程序设计教程

第2版

郑阿奇 梁敬东 主编
朱毅华 时跃华 赵青松 编著

机械工业出版社
China Machine Press

本书第2版以Microsoft Visual Studio .NET 2008/2010作为平台，在继承第1版的基本内容和基本方法的基础上，对内容体系结构进行调整、修改和优化，特别是大大增强了实例的实用性。本书内容包括：.NET与C#基础、C#编程基础、面向对象编程基础、面向对象编程进阶、Windows应用程序开发、GDI+编程、文件操作、数据库应用、多线程编程。

本书第1～9章是教程部分，讲解C#程序设计的基础知识，并包含大量实例；然后是习题部分，主要突出基本编程内容和基本概念；最后是实验部分，锻炼读者的编程和应用能力。

本书可作为高等院校相关专业C#程序设计课程的教材，也可供广大C#开发用户参考。

图书在版编目（CIP）数据

C#程序设计教程　第2版 / 郑阿奇，梁敬东主编. —北京：机械工业出版社，2011.8
（计算机基础课程系列教材）

ISBN 978-7-111-34942-6

Ⅰ. C…　Ⅱ. ①郑…　②梁…　Ⅲ. C语言－程序设计－高等学校－教材　Ⅳ. TP312

中国版本图书馆CIP数据核字（2011）第105466号

机械工业出版社（北京市西城区百万庄大街22号　邮政编码　100037）
责任编辑：李　荣
北京牛山世兴印刷厂印刷
2011年8月第2版第1次印刷
185mm × 260mm · 18.5印张
标准书号：ISBN 978-7-111-34942-6
定价：35.00元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换
客服热线：（010）88378991；88361066
购书热线：（010）68326294；88379649；68995259
投稿热线：（010）88379604
读者信箱：hzjsj@hzbook.com

前　言

C#是微软.NET中最简单、方便和高效的程序设计语言，在继承C++和Java等语言优点的基础上，不仅具有封装、继承和多态等特性，而且增加了不少创新元素，广泛应用于开发桌面系统、Web应用、数据库应用、网络应用等，已成为目前主流的开发工具。

2007年，我们结合当时的教学和应用开发需要，编写了本书第1版，受到广大师生的欢迎。随着Visual Studio .NET平台的不断升级，C#的功能变得更强，应用开发变得更方便。为此，我们以当前最新的Visual Studio .NET 2010作为平台，结合近年来的教学和应用开发实践，编写了本书第2版。

本书第2版在继承第1版的基本内容和基本方法的基础上，对体系结构进行了调整、修改和优化，实例实用性大大增强。本书分为教程、习题和实验三部分。教程部分首先介绍C#的特点并大致介绍.NET开发平台Visual Studio .NET 2010，通过一个简单实例了解控制台方式的操作过程；然后介绍Visual Studio .NET 2010开发环境，同时通过一个简单实例了解界面方式的操作过程；在此基础上，系统地介绍C#的编程基础、面向对象编程基础、面向对象编程进阶；最后介绍Windows应用程序开发、GDI+编程、文件操作、数据库应用、多线程编程。习题部分主要突出基本编程内容和基本概念。实验部分主要锻炼读者的编程和应用能力，读者可先跟着实例模仿，然后自己独立练习。一般来说，通过教程的学习、习题的练习，特别是认真地上机操作，在较短的时间内，读者便基本能够在Visual Studio .NET 2010环境下用C#解决一些小型应用问题。

本书为教师免费提供教学课件和所有应用实例的源文件，需要者请登录华章网站www.hzbook.com下载。

本书由南京农业大学朱毅华、时跃华和赵青松编写，郑阿奇（南京师范大学）和梁敬东（南京农业大学）统编、定稿。参加本教程编写的还有顾韵华、丁有和、刘毅、王燕平、汤玫、彭作民、崔海源、徐卫军等。

由于作者水平有限，不当之处在所难免，恳请读者批评指正。请发邮件到：easybooks@163.com。

编　者

2011年5月

目　录

第1章 .NET与C#基础

微软公司在2000年推出了.NET战略，它是微软面向互联网时代构筑的新一代平台，是微软在21世纪初的一个重大战略步骤。为了实现.NET技术，微软公司开发了一整套工具组件，这些组件被集成到Visual Studio（简称VS，本书使用的版本为VS 2010）开发环境中，在VS环境中可以开发运行新的.NET平台上的应用程序。VS是基于.NET框架重新设计的集成开发环境，而C#就是它的一个组成部分。在VS中，除了包括Visual C# .NET开发工具之外，还包括VB.NET、Visual J# .NET、Visual C++ .NET、ASP.NET等开发工具。本章主要介绍C#语言特点以及.NET开发平台和Visual Studio NET 2010（以下简称VS 2010）开发环境等知识。

1.1 C#语言简介

C#是.NET平台中用于应用开发的一种现代编程语言，随着微软.NET战略进入开发人员的视野，C#很快成为Windows应用开发语言中的宠儿。

1.1.1 C#的由来

C#（读作C sharp）是微软公司发布的一种面向对象的、运行于.NET Framework之上的高级程序设计语言。C#看起来与Java有着惊人的相似之处，它包括了诸如单一继承界面、与Java几乎同样的语法，以及编译成中间代码再运行的过程。但是C#与Java也有着明显的不同，它借鉴了Delphi的一个特点，与COM（组件对象模型）是直接集成的，而且它是微软公司.NET Windows网络框架的主角。

C#语言定义主要是从C和C++继承而来的，而且语言中的许多元素也反映了这一点。C#在从C++继承的可选选项方面比Java要广泛一些（比如structs），它还增加了自己新的特点（比如源代码版本定义）。

1.1.2 C#的特点

作为编程语言，C#是简单的、完全面向对象的，而且是类型安全的。重要的是，C#是一种现代编程语言。在类、名字空间、方法重载和异常处理等方面，C#去掉了C++中的许多复杂性，借鉴和修改了Java的许多特性，使其更加易于使用、不易出错。下面列举了一些C#在设计上的优点。

（1）简单性

C#在设计上的首要目标就是简单性。

没有指针是C#的一个显著特性。在默认情况下，用户使用一种可操控的（managed）代码进行工作时，一些不安全的操作，如直接的内存存取，将是不允许的。

在C#中不再需要记住那些源于不同处理器结构的数据类型，如可变长的整数类型。C#在CLR层面统一了数据类型，使得.NET上的不同语言具有相同的类型系统。可以将每种类型看做一个对象，不管它是初始数据类型还是完全的类。

整型和布尔数据类型是完全不同的类型。这意味着if判别式的结果只能是布尔数据类型，如果是别的类型则编译器会报错。那种混淆了比较和赋值运算的错误不会再发生。

（2）现代性

许多在传统语言中必须由用户自己来实现的或者干脆没有的特征，都成为基础C#实现的一个部分。金融类型对于企业级编程语言来说是一个很受欢迎的附加类型。用户可以使用新的decimal数据类型进行货币计算。

安全性是现代应用的头等要求，C#通过代码访问安全机制来保证安全性，根据代码的身份来源，可以分为不同的安全级别，不同级别的代码在调用时会受到不同的限制。

（3）面向对象

C#支持面向对象的所有关键概念：封装、继承和多态性。整个C#的类模型是建立在.NET虚拟对象系统（Virtual Object System, VOS）之上的，这个对象模型是基础架构的一部分，而不再是编程语言的一部分——它们是跨语言的。

C#中没有全局函数、变量或常数。每样东西必须封装在一个类中，或者作为一个实例成员（通过类的一个实例对象来访问），或者作为一个静态成员（通过类型来访问），这会使用户的C#代码具有更好的可读性，并且减少了发生命名冲突的可能性。

多重继承的优劣一直是面向对象领域争论的话题之一，然而在实际的开发中很少用到，在多数情况下，从多个基类派生所带来的问题比这种做法所能解决的问题更多，因此C#的继承机制只允许一个基类。如果需要多重继承，用户可以使用接口。

（4）类型安全性

当用户在C/C++中定义了一个指针后，就可以自由地把它指向任意一个类型，包括做一些并不完全的操作，如将一个整型指针指向双精度型数据。只要内存支持这一操作，它就能工作，这当然不是用户所设想的企业级编程语言类型的安全性。与此相比，C#实施了严格的类型安全机制来保护它自身及其垃圾收集器。因此，程序员必须遵守关于变量的一些规定，如不能使用未初始化的变量。对于对象的成员变量，编译器负责将它们置零。局部变量用户应自己负责。如果使用了未经初始化的变量，编译器会提醒用户。这样做的好处是：用户可以摆脱因使用未初始化变量得到一个可笑结果的错误。

边界检查。当数组实际上只有$n-1$个元素时，不可能访问到它“额外”的数组元素n，因此，不可能重写未经分配的内存。

算术运算溢出检查。C#允许在应用级或语句级检查这类操作中的溢出，当溢出发生时会出现一个异常。

C#中传递的引用参数是类型安全的。

（5）版本处理技术

在过去的几年中，几乎所有的程序员都和所谓的“DLL地狱”打过交道，产生这个问题是因为许多计算机上安装了同一DLL的不同版本。DLL是Dynamic Link Library的缩写，是一种编译为二进制机器代码的函数库。DLL在调用程序运行时才被调入内存执行，而不是在编译时链接到可执行程序内部，这样可以使程序代码在二进制级别实现共享，而不必在每个应用程序中编译一个副本。如果DLL中的代码更新了，只需要替换DLL文件即可更新所有使用该DLL的程序。然而这同时带来了DLL文件版本的问题，不同版本的DLL可能与不同调用程序不兼容，同一版本的DLL也不能同时为不同的调用程序服务，结果造成应用程序出现无法预料的错误，或者在用户计算机中不得不更换文件名来保存同一DLL的多个版本。这种混乱的状态称为“DLL地狱”。C#则尽力支持这种版本处理功能，虽然C#自己并不能保证提供正确的版本处理结果，但它为程序员提供了这种版本处理的可能性。有了这个支持，开发者可以确保当他开发的类库升级时，会与已有的客户应用保持二进制级别上的兼容性。

1.2 .NET开发平台

1.2.1 .NET框架概述

.NET框架（.NET Framework）是.NET战略的核心。这个框架执行应用程序和Web服务，包括类库（称为.NET框架类库或FCL），提供安全性并提供许多其他编程功能，可以建立.NET应用程序。使用.NET开发的程序需要在.NET框架下才能运行。

.NET框架的体系结构包括5部分，分别为：

- 程序设计语言及公共语言规范（CLS）。
- 应用程序平台（ASP.NET及Windows应用程序等）。
- ADO.NET及类库。
- 公共语言运行时（CLR）。

• 程序开发环境（Visual Studio）。

.NET框架的结构如图1-1所示。构建在Windows操作系统之上的是公共语言运行时，其作用是负责执行程序，提供内存管理、线程管理、安全管理、异常处理、通用类型系统与生命周期监控等核心服务。在CLR之上的是.NET框架类库，它提供许多类与接口，包括ADO.NET、XML、IO、网络、调试、安全和多线程等。

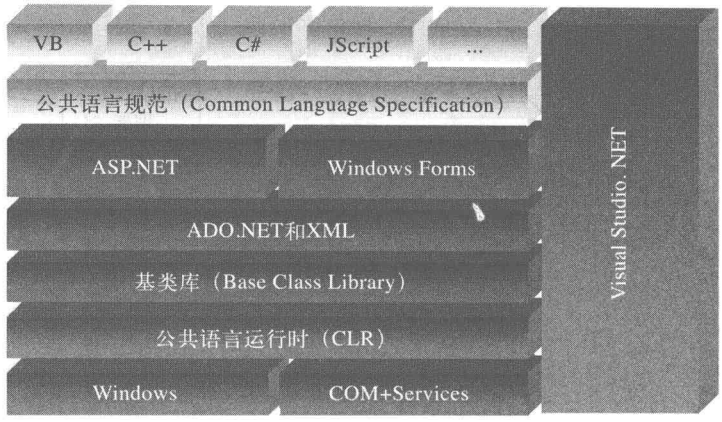

图1-1 .NET框架结构

.NET框架类库是以命名空间（namespace）的方式来组织类库的。命名空间与类库的关系就像文件系统中的目录与文件的关系一样，如用于处理文件的类属于System.IO命名空间。

在.NET框架基础上的应用程序主要包括ASP.NET应用程序和Windows Forms应用程序，其中ASP.NET应用程序又包含Web Forms和Web Service，它们组成了全新的因特网应用程序；而Windows Forms是全新的窗口应用程序。

在.NET框架之上，无论采用哪种编程语言编写的程序，都被编译成中间语言（IL），IL经过再次编译形成机器码，完成IL到机器码编译任务的是JIT（Just In Time）编译器。上述处理过程如图1-2所示。

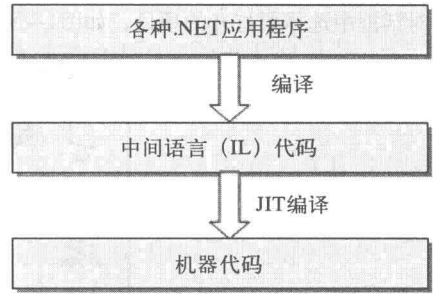

图1-2 .NET应用程序的编译过程

随着.NET技术的不断发展，.NET框架的发展也经历了几个阶段，从早期的.NET Framework 1.0、1.1发展到.NET FrameWork 2.0标志着.NET技术走向成熟，功能更加强大。2010年，随着Microsoft推出VS 2010开发平台，.NET框架由3.5更新为4，新版本增加了很多新特性，例如WCF、WPF、WF、LINQ和AJAX等，使.NET技术更加强大和成熟。在利用VS 2010进行项目开发时，可以根据所要使用的特性选择所采用的.NET框架版本，实现与早期版本的兼容。

1.2.2 VS 2010的集成开发环境

VS 2010是微软.NET平台上的一个功能强大的、集成了多种开发语言（包括VB.NET、VC++、VC#、VJ#）的软件开发工具。它可用于多种类型的程序开发，如基于Web的应用程序、基于WPF的应用程序、基于Windows的应用程序、控制台应用程序和移动应用程序等。本书中的第2～4章（C#基础）是以控制台应用程序介绍C#语言基础，其他章节介绍的主要是基于Windows的应用程序开发。此开发工具不仅可以实现.NET编程语言的快速开发，而且将程序编辑、调试、测试、打包、部署等操作集成在一起，大大提高了开发效率。

初次运行VS 2010将会出现“选择默认环境设置”对话框，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 Studio”按钮，经过配置后打开VS 2010主窗口，显示“起始页”，如图1-3所示。

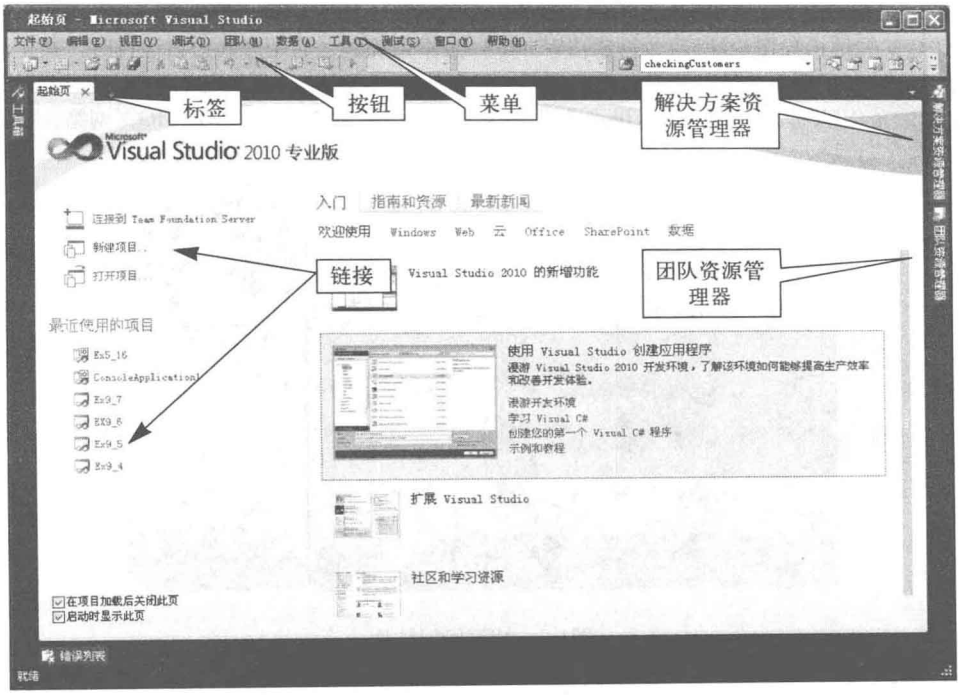

图1-3 VS 2010主窗口

在“起始页”中，允许用户打开或新建项目。若要打开已有项目，可单击最近使用的项目列表中的某个项目名称；也可以依次单击菜单“文件”→“打开”→“项目/解决方案”，在弹出的“打开项目”对话框中选择要打开的项目，如图1-4所示。

图1-4 “打开项目”对话框

1.2.3 第一个C#程序

让我们从经典的“Hello world”程序开始C#之旅。在控制台中输出“Hello world!”字样。读者可以发现，基于.NET框架强大的类库进行应用开发是多么简单与快捷。

【例1.1】在控制台窗口中输出“Hello world!”字样。

若要新建项目，可单击“新建项目”，将弹出“新建项目”对话框，如图1-5所示。首先，在“项目类型”栏中选择要建立的项目类型，然后在右侧的“模板”栏中选择某个模板类型。选择模板后，在“名称”栏中输入项目的名称，在“位置”栏中输入保存项目的路径，在“解决方案名称”栏中输入解决

方案的名称，单击“确定”按钮即可进入项目集成开发环境。新建立的项目都放在设定的解决方案中，一个解决方案可以含有一个或多个项目。默认情况下，解决方案的名字与项目名称相同，而且存放项目和解决方案的文件夹名就是项目名称。如果要将新建的项目添加到当前打开的解决方案中，在“解决方案名称”栏中选择“添加到解决方案”选项，单击“确定”按钮后，将把新建立的项目添加到打开的解决方案中，如图1-6所示。

图1-5 “新建项目”对话框

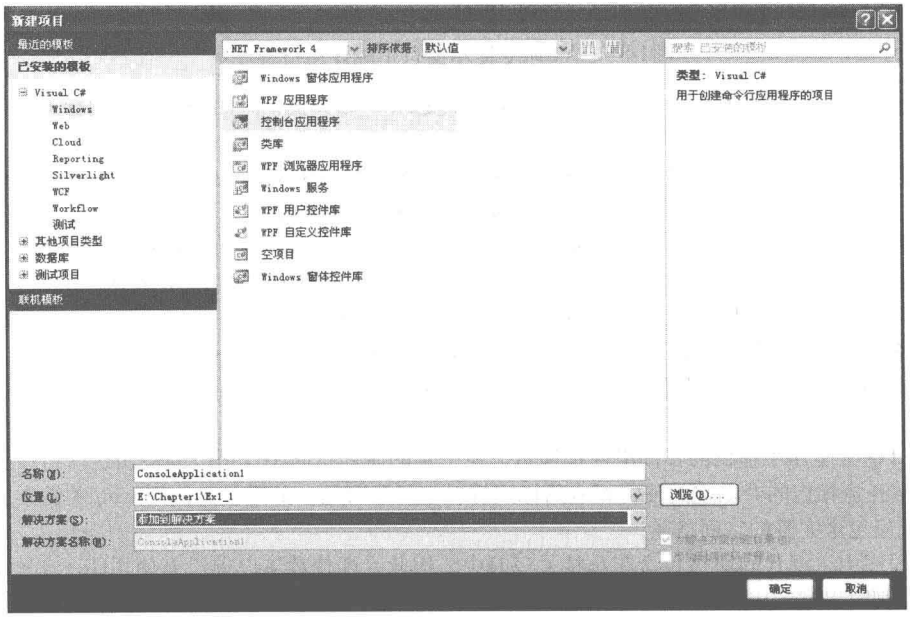

图1-6 添加到解决方案

在.NET开发环境中新建一个控制台应用程序项目后，在源代码文件中输入如下语句：

```
using System;
class HelloWorld
{
      public static void Main()
      {
            Console.WriteLine("Hello world!");
      }
}
```

此项目命名为“Ex1_1”，然后选择菜单“调试”→“启动”或直接按F5键运行此程序。可以看到运行结果为出现控制台窗口，并且在窗口中显示出“Hello world!”字样，如图1-7所示。

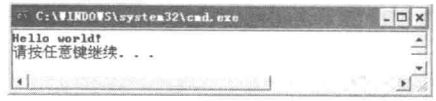

图1-7 控制台版本程序运行结果

读者也可以不使用.NET环境，直接用文本编辑工具输入以上代码，并且保存到Ex1_1.cs文件中，再通过开始菜单中的“所有程序”→“Microsoft Visual Studio 2010”→“Visual Studio Tools”→“Visual Studio 命令提示（2010）”打开命令行窗口，在命令窗口中输入：

```
csc /target:exe Ex1_1.cs
```

以上假设Ex1_1.cs文件在“C:\Program Files\Microsoft Visual Studio 10.0\VC”目录中。编译器编译该程序后，就可以输入“Ex1_1”来运行该程序了。程序运行的结果如图1-8所示。

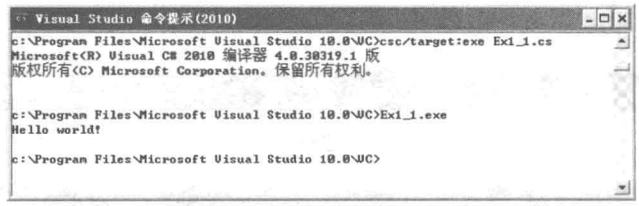

图1-8 在VS命令提示窗口中编译运行程序

下面通过上面的代码来认识一下C#。代码最前面是以using关键词开始的命名空间导入语句，然后是使用class关键字对类Helloworld的定义。

命名空间是为了防止相同名字的不同标识符发生冲突而设计的隔离机制。比如你开发了一个二维的图形组件，将类命名为Point，而你的同事开发的一个三维图形组件恰好也命名为Point，你们的用户需要在应用程序中使用你们的组件，在编译时编译器将无法判断引入哪一个版本的组件，通过将类的命名放在不同的命名空间中就可以加以区别，要使用哪一个类，通过using关键字打开其所在的命名空间即可。在C#中（确切地说是在.NET框架类库中）使用了一种树状的类似于中国→江苏→南京这样的地址编码方式来对命名空间进行管理，通过引入命名空间MyClass和YourClass就可以用MyClass.Point和YourClass.Point这样的方式对相同名称的标识进行识别了，即使是同时使用这两个组件也不会使编译器迷惑。

在.NET框架类库中提供的不同组件都被包含在一定的命名空间中，所以要使用这些组件也必须通过using关键字开放相应的命名空间使得相应的标识符对编译器可见。如果没有使用using关键字，那么相应的标识符就应包含完整的空间路径。

在C#程序中，源代码块被包含在花括号{和}中，因此，即使读者以前没有使用C++或Java编程的经验，也可以判断出类的内容包括在最外面的花括号中，其中Main函数是类的方法，方法的代码又包括在里面的花括号中。

由于C#是一种完全的面向对象的语言，所以不会有独立于类的代码出现，应用程序的入口也必须是类的方法。C#规定命名为“Main”的方法作为程序的入口（注意与C/C++语言不同，“Main”的第一个

字母是大写的，而且C#是一种大小写敏感的语言，所以不要写错了)。“public”、“static”关键字是对方法的修饰，其含义在后续章节中将详细介绍。我们要知道的是“public”使得这个方法能被外部访问（不然程序如何被操作系统执行？），“static”使得这个方法在类的实例被建立之前就可以调用（在程序入口的时候自然还不会有任何类的实例生成)。“Main”前面的关键字“void”代表该方法没有返回值，这与C/C++和Java是一样的。

方法中的代码“Console.WriteLine(“Hello world!”);”调用了.NET框架类库中对象的方法来向控制台或消息对话框输出信息。可以看出，本程序的核心代码所实现的功能全部来自.NET框架类库，而C#只是提供了一个语法框架。所以这里要再次强调学习.NET类库的重要性，C#开发实际上是用C#语言将.NET框架类库中的组件加以组织，实现应用程序的业务逻辑。

到这里读者已经跟随我们进入了C#编程的大门并且亲自实现了一个程序，通过本章的学习，读者应该掌握.NET的基本架构以及C#与.NET之间的关系，并且对C#程序的结构有一个大概的认识。正确理解.NET与C#的关系对以后的学习非常重要，因为C#开发的很多方面与.NET分不开。

1.3 Visual C#开发环境

1.3.1 新建Windows窗体应用程序

【例1.2】新建Windows窗体应用程序，输出含有“Hello world!”字样的对话框。

按照例1.1中新建控制台应用程序的方法新建Windows窗体应用程序，只是在“新建项目”对话框中要选择的是“Windows窗体应用程序”模板。单击“确定”按钮后，将进入基于C# Windows编程开发环境，如图1-9所示。从图中可以看出，屏幕被分成若干个部分，包括标题栏、菜单栏、工具栏、窗体设计器窗口、工具箱窗口、解决方案资源管理器窗口、属性窗口、输出窗口等。这些部分的使用将在下面的各节中逐一介绍。

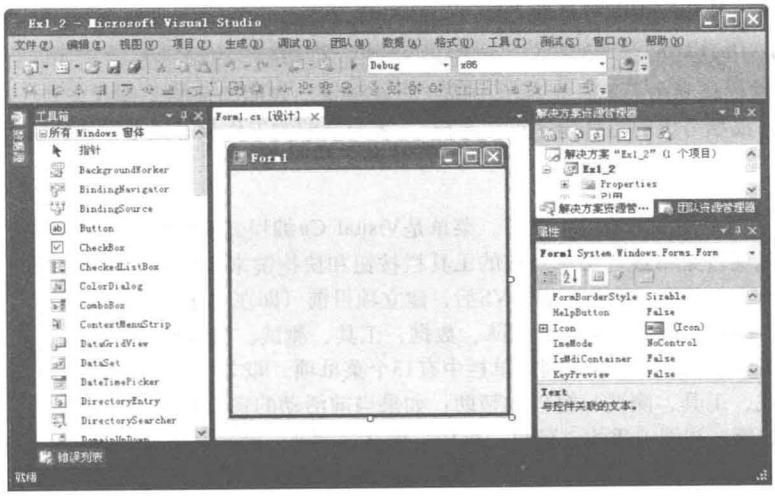

图1-9 C# Windows编程开发环境

从工具箱中选择“Button”按钮控件并将其拖曳到Form1窗体中，双击此按钮，在光标闪烁的地方添加如下代码：

```
using System;
using System.Windows.Forms;
namespace Ex1_2
{
    public partial class Form1 : Form
    {
```

```
        public Form1()
        {
            InitializeComponent();
        }
        private void button1_Click(object sender, EventArgs e)
        {
            MessageBox.Show("Hello World!");
        }
    }
}
```

按F5快捷键运行此程序，结果如图1-10所示。

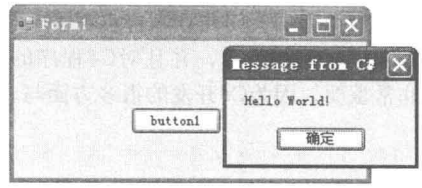

图1-10 例1.2的运行结果

1.3.2 标题栏

标题栏是VS 2010窗口顶部的水平条，它显示的是应用程序的名字。默认情况下，用户建立一个新项目后，标题栏显示如下信息：

```
WindowsApplication1 - Microsoft Visual Studio
```

其中，“WindowsApplication1”代表解决方案名称。随着工作状态的变化，标题栏中的信息也随之改变。当处于调试状态时，标题栏中的信息如下：

```
WindowsApplication1 (正在调试) - Microsoft Visual Studio
```

在上面的标题栏信息中，第一个括号中的“正在调试”表明当前的工作状态处于“调试阶段”。当处于运行状态时，该括号中的信息为“正在运行”，表明当前的工作状态处于“运行阶段”。

1.3.3 菜单栏

在标题栏的下面是集成环境的主菜单。菜单是Visual C#编程开发环境的重要组成部分，开发者要完成的主要功能都是通过菜单或与菜单对应的工具栏按钮和快捷键来实现的。在不同的状态下，菜单栏中的菜单项的个数是不一样的，例如，启动VS后，建立项目前（即在“起始页”状态下），菜单栏中有10个菜单项，即文件、编辑、视图、调试、团队、数据、工具、测试、窗口和帮助。而当建立或打开项目后，如果当前活动的窗口是窗体设计器，则菜单栏中有13个菜单项，即文件、编辑、视图、项目、生成、调试、团队、数据、格式、工具、测试、窗口和帮助；如果当前活动的窗口是代码窗口，则菜单栏中有13个菜单项，即文件、编辑、视图、重构、项目、生成、调试、团队、数据、工具、测试、窗口和帮助。

每个菜单包含若干个子菜单项，在子菜单中灰色的选项是不能使用的；菜单项中显示在菜单名后面“( )”中的字母为键盘访问键，菜单项后面显示的为快捷键。例如，“新建项目”的操作是先按Alt+F键打开“文件”菜单，再按N键，或直接按Ctrl+Shift+N键。

1. 文件菜单（File）

文件菜单用于对文件进行操作，如打开和新建项目，以及保存和退出等。文件菜单如图1-11所示，其对应的主要功能如表1-1所示。

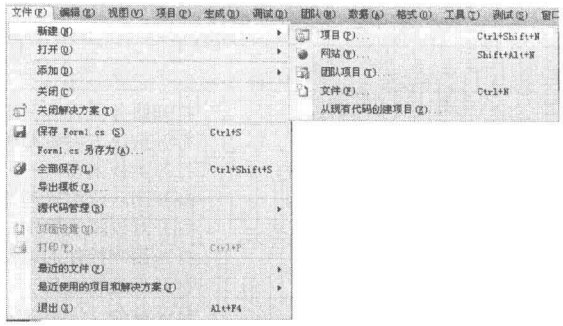

图1-11 文件菜单

表1-1 文件菜单功能表

下拉菜单	功 能
新建	包括新建项目、网站和文件等
打开	包括打开项目/解决方案、网站和文件等
添加	包括添加新建项目、新建网站和添加现有项目及现有网站
关闭	关闭当前项
关闭解决方案	关闭打开的解决方案
保存Form1.cs	保存对Form1.cs的修改，文件名不变
Form1.cs另存为	将Form1.cs另存为其他文件名
全部保存	保存当前打开的所有项目
导出模板	将项目或项导出为可用作将来项目的基础的模板
源代码管理	包括标签、在源代码管理中打开等
最近的文件	通过最近打开过的文件来打开相应的文件
最近使用的项目和解决方案	通过最近打开过的解决方案来打开相应的解决方案和项目
退出	退出VS 2008集成开发环境

2. 视图菜单（View）

视图菜单用于显示或隐藏各功能窗口或对话框。若不小心关闭了某个窗口，可以通过选择视图菜单项来显示该窗口。视图菜单还可以控制工具栏的显示。若要显示或关闭某个工具栏，只需单击“视图/工具栏”菜单项，找到相应的工具栏，选中或取消选中其前面的复选框即可。视图菜单如图1-12所示，其对应的主要功能见表1-2。

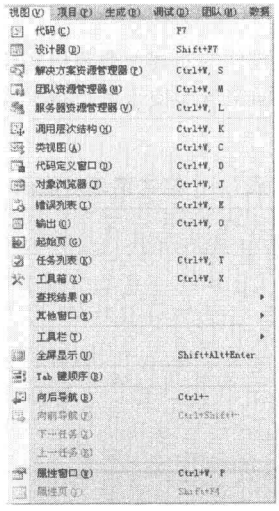

图1-12 视图菜单

表1-2 视图菜单功能表

下拉菜单	功 能
服务器资源管理器	打开服务器资源管理器窗口
解决方案资源管理器	打开解决方案资源管理器窗口
类视图	打开类视图窗口
团队资源管理器	打开团队资源管理器窗口
对象浏览器	打开对象浏览器窗口
属性窗口	打开用户控件的属性页
工具箱	打开工具箱窗口
其他窗口	打开命令、Web浏览器、起始页等其他窗口
工具栏	打开或关闭各种快捷工具栏

3. 项目菜单（Project）

项目菜单主要用于向程序中添加或移除各种元素，如窗体、模块、组件、类等。项目菜单如图1-13所示。项目菜单中的一般功能使用较简单，有两个重要功能见表1-3。

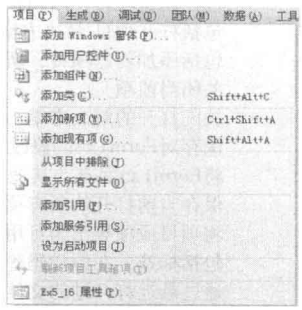

图1-13 项目菜单

表1-3 项目菜单功能表

下拉菜单	功 能
添加Windows窗体	向项目中添加新窗体
添加服务引用	添加一个Web服务引用或添加WCF服务引用

4. 格式菜单（Format）

格式菜单用于设计阶段窗体上各个控件的布局。它可以用于对所选定的对象调整格式，在设计多个对象时用来使界面整齐而便于进行统一操作。格式菜单如图1-14所示，其主要功能见表1-4。

图1-14 格式菜单

表1-4 格式菜单功能表

下拉菜单	功 能
对齐	将所有选中的对象对齐
使大小相同	将所有选中的对象按宽或高统一尺寸
水平间距	对所有选中的对象水平间距进行统一调整
垂直间距	对所有选中的对象垂直间距进行统一调整
在窗体中居中	使对象在窗体中居中对齐
顺序	使对象按前、后顺序放置
锁定控件	使所选中的控件锁定，不能调整位置