高等学校**应用型特色**规划教材

Visual C# .NET

程序设计与应用开发（第2版）

赠送电子教案

郑广成　沈蕴梅　虞　勤　主　编
顾蓬蓬　沈　晔　戴锐青　副主编

清华大学出版社

高等学校应用型特色规划教材

Visual C# .NET 程序设计与应用开发

(第 2 版)

郑广成　沈蕴梅　虞　勤　主　编
顾蓬蓬　沈　晔　戴锐青　副主编

清华大学出版社
北 京

内 容 简 介

C#作为.NET 框架中的主流编程语言，深受专业爱好者和从业人员的青睐。本书采用理论知识与实例操作相结合的方式，由浅入深、循序渐进地介绍 Visual C#编程语言的相关知识。包括面向对象编程知识，以及基于数据库的 Windows 应用程序开发知识。最后给出一个综合性的实战项目，全面讲述以数据库为基础的应用系统的开发全过程。

本书学以致用，注重能力，以“基础理论→实用技术→实训”为主线编写，在讲解技术方法的过程中贯穿实例，在实训项目中巩固技术方法。课后附有习题，且每一章都设置了“案例实训”，使读者能够掌握该章的重点及提高实际操作能力。

本书还将提供配套教学课件和各单元的源代码程序，以供读者参考。本书既可作为大中专院校的教材，也可作为各类培训班的培训教程。

图书在版编目(CIP)数据

Visual C# .NET 程序设计与应用开发/郑广成，沈蕴梅，虞勤主编. --2 版. --北京：清华大学出版社，2014
(高等学校应用型特色规划教材)
ISBN 978-7-302-35468-0

Ⅰ. ①V…　Ⅱ. ①郑…　②沈…　③虞…　Ⅲ. ①C 语言—程序设计—高等学校—教材　Ⅳ. ①TP312

中国版本图书馆 CIP 数据核字(2014)第 023015 号

责任编辑：章忆文
封面设计：杨玉兰
责任校对：周剑云
责任印制：刘海龙

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn，http://www.wqbook.com
地　　址：北京清华大学学研大厦 A 座　　**邮　　编**：100084
社 总 机：010-62770175　　**邮　　购**：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质 量 反 馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
课 件 下 载：http://www.tup.com.cn,010-62791865

印 刷 者：北京密云胶印厂
装 订 者：北京市密云县京文制本装订厂
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　张**：20.75　　**字　数**：503 千字
版　　次：2008 年 5 月第 1 版　2014 年 3 月第 2 版　　**印　次**：2014 年 3 月第1 次印刷
印　　数：1～3000
定　　价：38.00 元

产品编号：051847-01

前　言

Visual Studio .NET 已成为面向对象程序开发的主流平台，它可以开发 Windows 应用程序、Web 应用程序、嵌入式软件应用程序、游戏程序等各种软件，深受广大专业人士和学习者的青睐。

本书主要介绍 Visual C# 2012 编程的基础知识，共分为 15 章，包括 Visual C#简介、变量与表达式、流程控制、数组与字符串、函数字段与属性、面向对象编程、绘图、程序部署与调试、ASP.NET、综合训练等内容，内容编排由浅入深，并采用理论知识结合实例操作的方式进行详尽的讲解，本书的主要内容如下。

第 1 章：介绍.NET 框架，并且对 C#语言的特点进行描述。

第 2 章：介绍 C#应用程序的基础知识。包括变量、数据类型、表达式，以及变量的声明、使用方法和注意事项等。

第 3 章：介绍选择结构、循环结构的设计。

第 4 章：介绍数组和字符串处理的基础知识。

第 5 章：介绍函数的定义、使用，以及几种参数传递的不同和注意事项，还将介绍属性和字段的概念及使用方法。

第 6 章：介绍程序调试的方法，以及常见的几种调试方法的演示。

第 7 章：介绍面向对象编程思想在 Visual C#中的应用，并依次讲解类与对象的建立，构造函数、析构函数，以及继承、多态、代理等面向对象编程常用的手段。

第 8 章：介绍 Windows 应用程序常用的控件及其相关的属性、方法和事件。

第 9 章：介绍 ADO.NET 向用户提供的数据集、数据适配器、数据连接、Windows 窗体等组件。

第 10 章：介绍 System.Drawing 命名空间中的一些类，介绍颜色的设置以及 GDI+中的坐标的分类以及 GDI+中的几种绘图对象。

第 11 章：介绍 ASP.NET 的特点以及 IIS 的安装，这些内容都是学习 ASP.NET 编程之前的前期工作。

第 12 章：介绍如何通过 File 类和 Directory 类进行目录和文件的操作，以及如何采用 StreamReader、StreamWriter、BinaryReader、BinaryWriter 类进行文本模式和二进制模式的文件读写操作。

第 13 章：介绍多项目操作以及 MDI 开发环境项目编程技术。

第 14 章：介绍 Windows 应用程序的部署方法，训练程序项目的应用程序制作技术。

第 15 章：通过一个综合实训项目，从软件工程的角度进行设计与开发。

本书由郑广成、沈蕴梅、虞勤担任主编，顾蓬蓬、沈晔、戴锐青担任副主编，郑广成负责统稿。此外，参与本书编写的还有王珊珊、周海霞、卢振侠、石雅琴、陈海燕、缪静文、马新兵、何光明、钱妍池、赵梅、周汉、崔丹、冯勇、韩雪等。作为学习 Visual C# 2012 的一本实用的书籍，作者充分考虑了读者的习惯，在讲解理论知识的过程中插入了适当的实例，让读者能轻松、快速地进入 Visual C# 2012 编程世界。

本书适合以应用能力为本位的高职高专、应用型本科的教学训练要求。由于编者水平有限，书中难免有错误和疏漏之处，敬请广大读者批评指正。

目　录

第 1 章　Visual C#简介

本章要点

- .NET Framework 4.5 介绍
- Visual C#程序设计语言的优点
- Visual Studio 2012 开发平台的展示

本章主要是对 C#语言基础知识的介绍，其中包括 C#特点的介绍，以及 Visual Studio 开发环境的介绍。最后将给出一个简单的示例，初步熟悉窗体应用程序的编写方法。

1.1　.NET Framework 4.5 介绍

目前，.NET 框架的主流版本是 4.5，.NET 框架是微软为开发应用程序而创建的一个富有革命性的平台。.NET 框架发布的第一个版本是运行在 Windows 操作系统上的，以后随着技术的成熟和更新，其他操作系统，如 Linux、FreeBSD，甚至个人数字助理(PDA)类设备，都将有运行在其上的.NET 框架版本。

.NET 框架是.NET 的核心部分。.NET 应用程序运行时，所需的所有核心服务都是由.NET 框架提供的。.NET 框架的核心是公共语言运行时(CLR)，另外还包括了.NET 框架类库。

在.NET 框架中，CLR 是一个公共语言运行环境。它为.NET 应用程序运行提供了各种必要的服务。所有符合公共语言规范的语言——包括 Microsoft Visual Basic、Microsoft Visual C++和其他微软的编程语言，以及针对.NET 平台推出的第三方语言，都可以使用这些服务。公共语言运行时解决了语言的集成问题。在逐渐以网络计算为重点的今天，公共语言运行时显得尤为重要。

.NET 框架从底层的内存管理和构件装载一直到前端的用户界面，在各个层次上为用户提供了所有可能的支持。图 1.1 中给出了.NET 框架的主要组成部分。

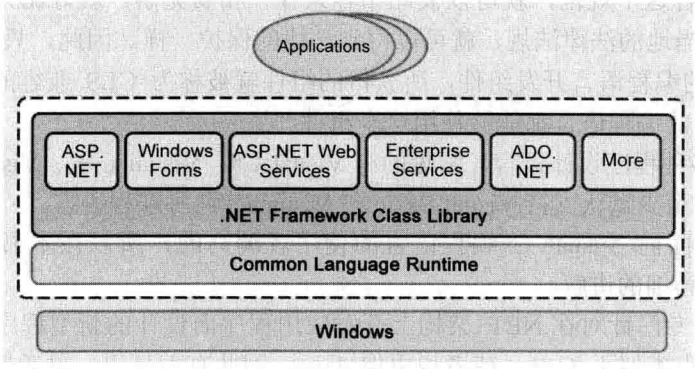

图 1.1　.NET 框架的组成

.NET 框架的最底层是公共语言运行时 CLR。它是.NET 框架的核心，也是其关键的功能引擎。CLR 为所有语言和环境提供了一个通用基础，使得跨语言集成成为可能。CLR 还负责内存的分配和管理、代码的即时编译、代码的装载、对象的引用计数，以及垃圾回收等操作。

CLR 之上是.NET 框架的基本类库(Base Class Library，BCL)。BCL 实现了运行时的各种功能，并通过各种命名空间为开发者提供了所需的各种高级服务。例如，Collections 命名空间包括了链表、哈希表等集合类型；System.IO 命名空间包含了输入/输出服务；BCL 是.NET 语言共享的标准类库，任何遵从.NET 规范的编程语言都可以使用它，这些服务都在.NET 框架的控制之下，为所有的语言提供了统一的类库支持。

ADO.NET 中主要使用 DataSet(数据集)在内存中处理数据。在 DataSet 中，数据表格可以单独存放，也可以通过 DataRelation 对象在表格之间建立关联，包括一对多、多对多、甚至是同一表格的自我关联(同一表格的外键关联到同一表格的主键)，大大增加了数据处理的灵活性，再加上 DataSet 中的所有数据都是离线的，也就是说，数据是直接存储在内存中的，因此可以降低后台 DBMS(数据库管理系统)的负担，特别适合用在 Web 数据库应用程序的开发上。

除了使用数据库的存取方式进行数据处理外，.NET 上还支持 XML 文档的操作，通过 XmlDataDocument 类就可以进行 XML 文档的存取。而要使用这个类，就应该添加对命名空间 System.Xml 的引用。

.NET 依旧支持对 Windows 应用程序的开发，以前我们在 Visual Basic 中经常使用的 ActiveX 控件，现在则由基类库中的 System.Windows.Forms 命名空间下的类取代(当然 ActiveX 控件还可以继续沿用)。新一代的 Web 应用程序开发则使用了 ASP.NET 技术，相比于原来的 ASP，它延续了容易使用的优点，而且对其进行了改进。

ASP.NET 在 2001 年由微软公司推出，在结构上，与前面的版本相比大不相同，几乎完全是基于组件和模块化的。Web 应用程序的开发人员使用这个开发环境，可以实现更加模块化的、功能更加强大的应用程序。

要使各种不同的程序语言在同一个软件平台上运行，这看起来有点不可思议。.NET 框架就实现了这个看似无解的难题。通用语言规范(Common Language Specification，CLS)包含了函数(类的方法)调用方式、参数传递方式、数据类型、异常处理方式等规则，任何编程语言只要符合这个规范，就可以彼此相容共存、相安无事。就好比人们不论是如何赚钱的，只要符合当地的法律法规，就可以得到法律的保护一样。因此，只要遵循 CLS 和只使用 CLS 兼容的编程语言开发组件，所获得的组件就被称为 CLS 兼容的组件，就可以保证其他支持 CLS 的编程语言都能够使用这个组件。

.NET 支持多种程序设计语言，常见的有 Visual C#、Visual C++、Visual Basic、Visual J#等。.NET 架构至少默认支持 Visual Basic 和 Visual C#两种编程语言。

本书主要是面向 Visual C#编程语言的使用者编写的，所以在本书的内容中，只对 Visual C#内容做详细的讲解。

Visual C#是专门针对在.NET 架构上开发应用程序而设计的新型程序设计语言，就程序语法来说，有点类似于 C++，或者说更像 Java。因此有着易用、灵活性大的特点，拥有完整的面向对象支持，是.NET 平台上最常用的语言之一。

1.2　Visual C#介绍

1.2.1　Visual C#的由来

最近 20 年，C 和 C++一直被商用软件开发者普遍使用。C#的出现，为开发者提供了一个快速建立应用程序的开发平台。微软对 C#的定义是“一种类型安全、现代、简单、由 C 和 C++衍生出来的面向对象的编程语言，它是牢牢植根于 C 和 C++语言之上的，并可立即被 C 和 C++的使用者所熟悉。Visual C#的目的就是综合 Visual Basic 的高生产率和 C++的行动力”。

Visual C#是一种强大的语言，在 C++中能完成的任务利用 Visual C#也能完成。但在 Visual C#中，与 C++比较高级的功能等价的功能(例如直接访问和处理系统内存)，只能在标记为“不安全”的代码中使用。这种高级编程技术是非常危险的，因为它可能覆盖系统中重要的内存块，导致严重的后果。因此本书将不讨论这方面的特殊内容。

过去，人们改进、开发了许多语言，以提高软件生产的效率。但是这些或多或少都以牺牲 C 和 C++程序员所需要的灵活性为代价。这样的解决方案给程序员套上了太多的枷锁，限制了他们能力的发挥，且无法很好地与原有的系统兼容。更为头痛的是，那些语言并不总是与当前的 Web 应用结合得很好。

理想的解决方案是将快速的应用开发与对底层平台所有功能的访问紧密结合在一起。程序员们需要一种环境：它与 Web 标准完全同步，并且具备与现存程序方便地进行集成的能力。除此之外，程序员们希望这种开发环境允许自己在需要时使用底层代码。

针对该问题，微软的解决方案就是推出了 Visual C#，它是一种现代的、面向对象的程序开发语言，它使得程序员能够在新的微软.NET 平台上快速开发种类丰富的应用程序。.NET 平台提供了大量的工具和服务，能够最大限度地发掘和使用计算及通信能力。

由于其一流的面向对象的设计，从构建组件形式的高层商业对象，到构建系统级应用程序，我们都会发现，Visual C#将是最合适的选择。

使用 Visual C#语言设计的组件能够用于 Web 服务，这样通过 Internet，就可以被运行于任何操作系统上的任何编程语言所调用。

任何面向对象语言的核心在于支持对类的定义和处理。类定义了新的类型，可以扩展语言，以创造更适合于解决具体问题的模型。Visual C#中有声明新的类及其方法和性质的关键字，含有对实现面向对象编程的三大支柱封装、继承和多态的支持。

在 Visual C#中，与类的定义有关的一切都可在声明本身中找到，C#的类定义并不需要独立的头文件或 IDL(接口定义语言)文件。而且，Visual C#支持新的 XML 风格的内嵌文档，大大简化了软件的在线和印刷参考文档的制作工作。

Visual C#还支持接口(Interface)，一种与其所指定的服务的类订立合同(Contract)的方式。在 Visual C#中，类只能从一个父类继承，但可以实现多个接口。在实现接口时，C#类实际上也承诺了要提供接口所规定的功能。

Visual C#还提供了对结构体(Struct)的支持，但此概念的含义与 C++有显著的不同。在 C#中，结构体是有严格限制的轻量级类型，实例化时，比传统的类对操作系统和内存的需

求都小得多。结构体不能从类继承，也不能被类继承，但它可以实现接口。

Visual C#提供了面向组件的特性，如属性(Property)、方法、事件和称为特性信息(Attribute)的声明性结构。面向组件编程是通过 CLR 将元数据(Metadata)与类的代码一起保存而实现的。元数据负责描述类，包括其方法和性质，以及安全要求和其他属性信息，如是否可以序列化(Serialize)；代码则包含执行功能所必需的逻辑流程。因此已编译的类是自成一体的独立单位。这样宿主环境只需能够识别类的元数据和代码，无需其他信息(如类的注册信息)，就可以使用它。使用 Visual C#和 CLR，可以通过自定义特性信息来给类添加自定义元数据。同样也可以用支持反射的 CLR 类型阅读类的元数据。

程序集(Assembly)是文件的集合，对编程人员而言，就是 DLL 或者 EXE 文件。在.NET 中，程序集是重用、版本协调、安全性和部署的基本单位。CLR 提供了大量处理程序集的类。

使用 Visual C#开发应用程序比使用 C++更简单，因为 C#的语法更简单、Visual Studio 2012 平台的可视化功能更加直观，编辑方法更加灵活和方便。如前所述，Visual C#还支持使用 C++式的指针和关键字直接访问内存，不过这种操作都被归入不安全的范畴，并且会警告 CLR 无法使用内存回收器，在指针所引用的对象被释放前不进行回收。

1.2.2 C# 4.5 新增的功能

C# 4.5 新增的功能主要体现在如下一些方面。

(1) 异步读取和写入 HTTP 请求和响应：ASP.NET 4.5 可以异步读取、编写并刷新流。此异步性可以将增量数据发送到客户端，而不必占用操作系统线程。

(2) 当“请求验证”启用时，对读取 unvalidated 请求数据提供支持：ASP.NET 4.5 提供用于读取 unvalidated 请求数据的支持，以便允许用户选定字段或页的标记。

(3) 为 WebSockets 协议提供支持：在新 System.Web.WebSockets 命名空间中的方法提供 WebSockets 协议支持，可让我们读取和写入字符串及二进制数据。

(4) 客户端脚本的绑定和缩减：ASP.NET 4.5 通过合并，能更快加载不同的 JavaScript 文件绑定(通过移除不必要的字符减少 JavaScript 和 CSS 文件的大小)。

(5) 对于异步模块和处理程序支持：新 async 和 await 关键字可以方便地编写异步 HTTP 模块和异步 HTTP 处理程序。以异步开发的更新包括如下几个。

- ClientDisconnectedToken：异步通知应用程序的 CancellationToken。
- TimedOutToken：异步通知应用程序的请求超时时的 CancellationToken。
- ThreadAbortOnTimeout：如果希望应用程序控制计时请求行为，应将此特性设置为 false。
- Abort：使用此方法在应用程序中强制停止请求的基础 TCP 连接。

(6) 集成反 XSS 编码例程：反 XSS(脚本的跨站点)核心编码例程集成于 ASP.NET 4.5 之中。这些实例仅使用过去提供的外部库。

(7) 为 OAuth 和 OpenID 提供支持：OAuth 和 OpenID 可以创建允许用户登录其他站点的凭据，包括 Google、雅虎、Facebook 和 Windows Live 的站点。

1.3　Visual C#语言的特点

Visual C#语言的特点可以归结为以下几种：

- 简洁的语法。
- 精细的面向对象设计架构。
- 与 Web 紧密结合。
- 完善的安全性与错误处理。
- 灵活的版本处理技术。
- 更好的灵活性与兼容性。

1.3.1　简洁的语法

在默认的情况下，Visual C#的代码在.NET 框架提供的“可操控”环境下运行，不允许直接内存操作。这与 C++不同，C++中会出现大量的“->”、“::”操作符，这些在 Visual C#中已经不再出现，Visual C#只支持“.”操作符，对于我们来说，现在需要理解的仅仅是名称嵌套而已。

语法的冗余是 C++常见的问题，比如“Const”和“#Define”、各种各样的字符类型等。Visual C#对此做了简化，只保留了常见的形式，而别的冗余形式已经从它的语法结构中清除出去。

1.3.2　精细的面向对象设计架构

面向对象的话题从 Smalltalk 开始就缠绕着任何一种现代程序设计语言，众多开发人员也越来越沉浸于面向对象编程思想给编程人员带来的便利以及给人们带来的快乐。

Visual C#语言具有面向对象语言所应有的一切特性：封装、继承、多态，这是很正常的。然而，通过精细的面向对象设计架构，对于高级商业目标和系统级应用来说，Visual C#已经成为建造各种组件的最佳选择。

Visual C#的类型系统可分为值类型和引用类型，引用类型是对象，值类型可通过一个叫作“装箱与拆箱”的机制来完成与引用类型之间的转换操作，这在以后的章节中将会进行更为详细的介绍。

Visual C#中只允许单继承，即每个类只允许有一个父类(亦称基类)，从而避免了类型定义的混乱。同时，Visual C#不存在全局函数、全局变量，也不存在全局常数。所有的东西都必须封装在类之中，这样做的好处是，代码将有更好的可读性，并且命名冲突的问题也迎刃而解。

整个 Visual C#的类模型是建立在.NET 虚拟对象系统(Visual Object System，VOS)的基础之上的，其对象模型是.NET 基础架构的一部分，而不再是其本身的组成部分。

1.3.3 与 Web 紧密结合

Web 编程是当今编程的一大趋势和潮流，在.NET 中，新的程序开发模型越来越多地需要与 Web 标准相结合、相统一。例如使用超文本标记语言(Hypertext Markup Language，HTML)和 XML。由于历史的原因，现存的一些开发工具不能与 Web 紧密地结合。SOAP 的使用使得 Visual C# .NET 克服了这一缺陷，大规模深层次的分布式开发从此成为可能。

由于有了 Web 服务框架的帮助，对于程序员来说，网络服务看起来就像是 C#的本地对象。程序员们能够方便地为 Web 服务，并允许服务通过 Internet 被运行在操作系统上的任何语言调用。举例说，XML 已经成为网络中结构化数据传送的标准，为了提高效率，Visual C#允许直接将 XML 数据映射为结构，这样就可以有效地处理各种数据了。

1.3.4 完善的安全性与错误处理

语言的安全性和处理错误的能力，是衡量一种语言是否优秀的重要依据。任何人都会犯错误，即使是最熟练的程序员也不例外：忘记变量的初始化，对不属于自己管理范围的内存空间进行修改等。这些错误常常产生难以预见的后果。一旦这样的软件被投入使用，寻找与改正这些简单错误的代价也是让人难以忍受的。Visual C#的先进设计思想可以消除软件开发中的许多常见错误，并提供了包括类型安全在内的完整的安全性能。为了减少开发中的错误，Visual C#会帮助开发者通过更少的代码完成相同的功能，这不但减轻了编程人员的工作量，同时更有效地避免了错误的发生。

.NET 运行库提供了代码访问安全特性，它允许管理员和用户根据代码的 ID 来配置安全等级。当应用程序执行时，运行库将自动对它进行计算，然后给它一个权限集。根据应用程序获得的权限不同，应用程序或者正常运行，或者发生安全性异常，计算机上的本地安全设置最终决定代码所收到的权限。内存管理中的垃圾收集机制减轻了开发人员对内存管理的负担。.NET 平台提供的垃圾收集器(Garbage Collection，GC)将负责资源的释放和对象撤消时的内存清理工作。

变量是类型安全的。Visual C#中不能使用未初始化的变量，对象的成员变量由编译器负责将其置为 0，当局部变量未经初始化而被使用时，编译器将做出提醒；Visual C#不支持不安全的指针，不能将整数指向引用类型，例如对象，当进行向下转型时，Visual C#将自动验证转型的有效性；Visual C#中提供了边界检查和溢出检查功能。

1.3.5 灵活的版本处理技术

Visual C#中提供内置的版本支持来减少开发成本，使用 Visual C#将会使开发人员能够更加轻易地开发和维护各种商业应用。

升级软件系统中的组件(模块)是一件容易产生错误的工作。在代码修改过程中，可能对现存的软件产生影响，很有可能导致程序崩溃。为了帮助开发人员处理这些问题，Visual C#在语言中内置了版本控制功能。例如，函数重载必须被显式地声明，而不会像在 C++或者 Java 中经常发生的那样意外地被使用，这可以防止代码级错误和保留版本化的特

性。另一个相关的特性是对接口和接口继承的支持。这些特性可以保证复杂的软件能够被方便地开发和升级。

1.3.6　更好的灵活性和兼容性

在简化语法的同时，Visual C#并没有失去灵活性，尽管它不是一种无所不能的语言。比如，它不能用来开发硬件驱动程序，在默认的状态下没有指针等，但是，在学习过程中我们将发现，它仍然是那样的灵巧。

如果需要，Visual C#允许我们将某些类或者类的某些方法声明为非安全的，这样一来，我们将能够使用指针，并且调用这些非安全的代码不会带来任何其他的问题。此外，它还提供了代理(Delegate)来模拟指针的功能。又比如说，Visual C#不能支持类对多个类的继承，但是可以通过对多个接口的继承，来实现这一功能。

正是由于其灵活性，C#允许与 C 风格的需要传递指针型参数的 API 进行交互操作，DLL 的任何入口点都可以在程序中进行访问。Visual C#遵守.NET 公用语言规范(Common Language Specification，CLS)，从而保证了 Visual C#与其他语言组件间的互操作性。元数据(Metadata)概念的引入，既保证了兼容性，又实现了类型安全。

1.4　VS2012 开发环境介绍

1.4.1　VS2012 的界面

启动 Visual Studio 2012(简称 VS2012)，进入其集成开发环境，用户首先将可以看见如图 1.2 所示的界面窗口，这个界面窗口中包含进行 C#程序开发的基本工具。

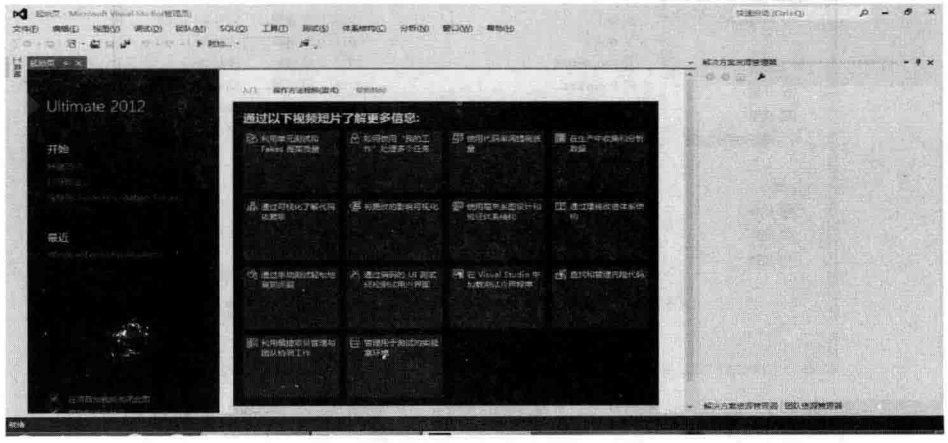

图 1.2　Visual Studio 2012(Visual C#的界面)

从界面窗口中，可以看到菜单栏、工具栏按钮、标题栏、类视图、解决方案资源管理器、属性视图窗口、代码编辑窗口。

可以单击“创建”按钮，进行新项目的创建；同样，我们也可以单击“打开”按钮，

从目录中寻找我们已经写好的程序。Visual C#的初始界面上也会显示最近 6 个我们曾经打开的程序，这很方便用户的操作。

1.4.2 菜单栏

菜单栏中列出了C#开发环境的各个主菜单项，单击某个菜单项，便会显示出下拉菜单中的各个命令项。单击某个命令，就可以完成相应的操作。

(1) C#菜单中的命令可以分为 4 类：普通命令、带有快捷键的命令、带有子菜单的命令和可弹出对话框的命令。下面分别介绍这几种命令的使用方法。

- 普通菜单命令：普通菜单命令的选用只需直接在下拉菜单上单击命令或键入命令中带有下划线的字母即可。如“生成”菜单中的“重新生成解决方案”菜单项、“窗口”菜单中的“隐藏”菜单项都属于普通菜单命令。
- 带有快捷键的命令：快捷键亦称热键，要选用该命令，除了用第一种方法之外，还可以直接按下快捷键。例如针对“编辑”菜单中的“复制”命令，直接按Ctrl+C 组合键即可，方法为先按住 Ctrl 键，然后再按住 C 键。用户应该多记住些快捷键，以节省操作时间。
- 可弹出对话框的命令：这类命令后面都带有 3 个小圆点的省略符号，如“项目”菜单中的“添加现有项...”命令，以及通过“文件”→“新建”菜单可以展开的“项目...”、“文件...”、“网站...”等命令，单击这种命令，就会弹出一个对话框。如图 1.3 所示为选择“项目”→“添加现有项...”菜单命令后弹出的对话框。

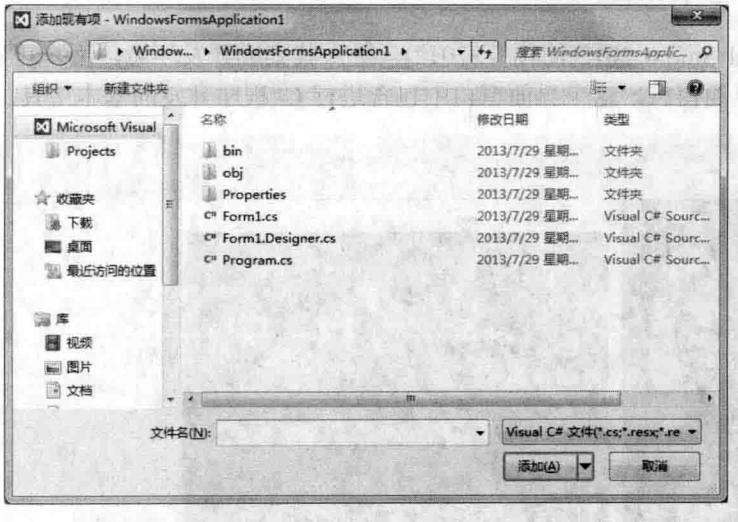

图 1.3 选择“项目”→“添加现有项...”菜单命令后弹出的对话框

- 带有子菜单的命令：C#的菜单命令中带有子菜单的命令很多，这些命令的最后都有一个向右的黑三角，表明有子菜单。如“文件”菜单中的“新建”、“打开”、“添加”等。

(2) C#主菜单为用户提供了开发、调试以及管理应用程序的各个功能和工具。

- 文件：包含用于打开、保存、新建应用程序文件的各种命令。