COMPUTER

高等院校计算机技术

“十二五”规划教材

Visual C#.NET 程序设计案例教程

◎ 梁曦 张运涛 吴建玉 编著

郭尚鸿 主审

ZHEJIANG UNIVERSITY PRESS

浙江大学出版社

高等院校计算
浙江省高等教

Visual C#.NET 程序设计案例教程

编著　梁　曦　张运涛　吴建玉
主审　郭尚鸿

ZHEJIANG UNIVERSITY PRESS
浙江大学出版社

图书在版编目（CIP）数据

Visual C#.NET 程序设计案例教程 / 梁曦，张运涛，吴建玉编著. —杭州：浙江大学出版社，2012.6
ISBN 978-7-308-10033-5

Ⅰ. ①V… Ⅱ. ①梁… ②张… ③吴… Ⅲ. ①C 语言－程序设计－教材 Ⅳ. ①TP312

中国版本图书馆 CIP 数据核字（2012）第 108857 号

Visual C#.NET 程序设计案例教程

编著 梁 曦 张运涛 吴建玉

责任编辑	周卫群
封面设计	刘依群
出版发行	浙江大学出版社
	（杭州市天目山路 148 号 邮政编码 310007）
	（网址：http://www.zjupress.com）
排　　版	杭州中大图文设计有限公司
印　　刷	浙江省良渚印刷厂
开　　本	787mm×1092mm 1/16
印　　张	23
字　　数	589 千
版 印 次	2012 年 6 月第 1 版 2012 年 6 月第 1 次印刷
书　　号	ISBN 978-7-308-10033-5
定　　价	40.00 元

前　言

本书内容主要集中在如何解决软件开发项目中所涉及的技术工具、技术框架、开发流程和编码调试经验等方面。重点讲解了企业中最常用的技能点，对于不常用的技能点，教材都未讲解，力争做到以用为本、学以致用。教材内容的安排是以案例为中心开展，在真实项目的基础上，经过精心设计将项目分解为多个既独立又具有一定联系的教学案例，由相关的教学案例引出技术内容。学生学习案例的过程，就是学习 C#.NET 开发知识和技能的过程。在案例的选择上，我们在考虑案例的实用性同时，也尽可能地提高案例的趣味性，并加强与日常生活中遇到的问题和现象的联系。通过这种案例教学的方式，学生不会迷失在浩如烟海的知识中，同时会具备更多的行业知识和项目经验。

全书共分 13 章。第 1 章重点讲述 NET Framework 框架的体系结构和开发工具 VS.NET 2008 的使用；第 2 章讲述 C# 语法的基础知识；第 3 章讲述 WinForms 的窗体编程的基础知识；第 4 章讲述面向对象的一些基本概念；第 5 章重点讲述 ADO.NET 数据库编程的知识；第 6 章讲述使用数据展示控件显示和操作后台数据库；第 7 章重点讲述调试、异常的处理及单元测试；第 8 章讲述数组、集合和泛型的概念；第 9 章重点讲述面向对象的一些高级概念；第 10 章重点讲述 WinForms 窗体编程的高级控件；第 11 章讲解文件读写与 XML 文件的操作类；第 12 章讲解利用三层结构优化数据库应用系统；第 13 章重点讲述抽象工厂模式的应用。

本书第 2 章、第 3 章、第 4 章由吴建玉编写，第 5 章、第 6 章由梁曦编写，第 1 章、第 7 至第 13 章由张运涛编写。

本书还将出版与本书配套的教学辅导材料、课件及项目源代码提供电子稿下载。

本书适用面广，内容满足 C#.NET 编程类课程的基本要求，可作为高等院校软件专业编程类课程的教材或教学参考书，也可以作为软件编程人员的参考书。

由于时间和水平有限，书中难免有不足之处，请各位读者批评指正，欢迎反馈用书信息。

编　者

2012 年 5 月

目 录

第1章　C#与.NET概述

【本章工作任务】

- 设置 VS. NET 2008 环境
- 使用 VS. NET 2008 的动态帮助
- 编写一个命令行 C# 程序

【本章技能目标】

- 理解. NET Framework 框架的体系结构
- 会使用 VS. NET 2008 环境进行 C# 程序开发
- 会使用 Console 类的方法
- 会使用命名空间

1.1　NET Framework 概述

1.1.1　Microsoft . NET 介绍

Microsoft . NET 是 Microsoft 的 XML Web 服务平台。. NET 包含了建立和运行基于 XML 的软件所需要的全部部件。Microsoft . NET 解决了下面这些当今软件开发中的一些核心问题：

- 互操作性(Interoperability)、集成性(Integration)和应用程序的可扩展性(extensibility)太难实现而且代价很高。Microsoft . NET 依靠 XML(一个由 World Wide Web Consortium(W3C)管理的开放标准)消除了数据共享和软件集成的障碍。
- 无数具有相当竞争力的私有软件技术使得软件的集成变得非常复杂。而 Microsoft . NET 建立在一个开放的标准上，它包含了所有编程语言。
- 当终端用户使用软件时，他们总觉得不够简便，有时甚至感到很沮丧，因为他们无法在程序之间方便地共享数据或是无法对能访问的数据进行操作。XML 使数据交换变得容易了，并且. NET 软件可以使得用户只要一得到数据就能对它们进行操作。
- 终端用户们在使用 Web 的时候，无法对自己的个人信息和数据进行控制，这导致了个人隐私和安全泄露问题。而 Microsoft . NET 提供了一套服务，使用户可以管理他们的个人信息，并且控制对这些信息的访问。
- . COM 公司和 Web 站点开发者们很难为用户们提供足够的有价值的数据，至少有一部分原因是由于他们的应用程序和服务无法很好地和其他程序和服务合作，只是一个不和外

界连接的信息孤岛。而 Microsoft .NET 的设计宗旨就是为了使来自于多个站点和公司的数据或服务能够整合起来。

对开发人员来说,.NET 平台中另一个重要的部分自然就是开发工具了。Microsoft .NET 程序员们设计编写的是 XML Web 服务,而不是服务器或客户端的独立应用程序。他们把这些服务组合成松耦合,相互协作的软件群,XML Web 服务之间使用 XML messaging 进行通讯。为了做到这一点,程序员需要:

- 一个软件平台,用于建立一种新的完整的个人用户经验。
- 一个编程模型和工具,用以建立和整合 XML Web 服务。
- 一套能为应用程序和服务提供基础的可编程的服务。

Microsoft .NET 包括:

(1).NET 平台,这是一套编程工具和基本构架,用来创建、发布、管理和整合 XML Web 服务。

(2).NET 体验,这是终端用户用以和.NET 交互的手段。

Microsoft .NET 的平台是由用于创建和运行 XML Web 服务组成的。它包含了下面四个组件:

- .NET 框架和 Visual Studio.NET:这些是开发人员用来生成 XML Web 服务的工具。.NET 框架是 Microsoft .NET 平台核心中的一套编程接口;Visual Studio.NET 是一套多语言系列的编程工具。
- 服务器基本结构(Server Infrastructure):.NET 的服务器基本结构是一系列用于生成、发布和操作 XML Web 服务的基础程序,包括 Windows 和各种.NET 企业服务器。
- Building Block Services:Building Block Services 是一套以用户为中心的 XML Web 服务,它把用户数据的控制权从应用程序移到了用户手上,这些服务包含了 Passport(用于用户身份验证)、服务之间的消息传递、文件存储、用户个性设置的管理、日历管理和其他一些功能。
- 智能设备(smart device):.NET 利用软件使智能设备,诸如手提电脑、轻便电话、游戏操纵台等都能够在.NET 世界中得以使用。

终端用户是通过.NET 体验访问 XML Web 服务的,这和现有的独立应用程序有点类似,但在下列这些重要的方面是不同的:

- .NET 体验可使用于多种设备:我们无需为可能使用的每一个设备编写一个不同的 XML Web 服务和不同的.NET 体验,.NET 体验能够读取用户选取设备的特征,给出一个恰当界面。
- .NET 体验使用 XML Web 服务:当.NET 体验连入网络后就能有效地利用 XML Web 服务为用户带来额外的价值,以更好地解决问题。
- .NET 体验是以用户为中心的:.NET 体验的焦点在终端用户,使用基于身份验证的块构建服务来为用户验证、参数设定、通知机制和用户数据提供服务。

.NET 的好处:

Microsoft .NET 为程序员、商业领导、IT 部门以及消费者带来了很多好处。

- 相对来说,程序员是比较缺乏的,雇用的费用也很高。然而 Microsoft .NET 使编程工作变得更加容易,开发投资的回报率也趋最大化。开发者们可以创建能重用的 XML Web 服务,而不再是一个单一的程序;这些 XML Web 服务易于编程和调试,彼此之间相互独立,通过 XML message 通讯及合作。所以对某一个服务的修改不会影响到其他的服务。

由于XML Web服务可以被很多.NET体验共同使用，所以对一个服务模块有效更新，也即更新了所有使用这个模块的.NET体验。任何编程语言都可以用来编写XML Web服务(如：C、C++、Visual Basic、COBOL、Perl、Python和Java等)，所以你的程序员可以选择他们最熟悉的语言来编程，这大大提高了开发效率。

- Microsoft .NET减少了程序员要写的代码量。一个XML Web服务能适用于所有的设备，不必再去为每一个设备编写一个不同的版本。另外，将显示特性与.NET体验分开以便以后加入新的接口技术，比如语音或手写识别，而不必去重写程序。
- Microsoft .NET开创了全新的商业模型，它使得一个公司可以用多种方法来把自己的技术商品化。举个例子来说，一个通讯公司可以使用XML Web服务的方式提供语音信件和呼叫者ID的访问，让用户从一个即时消息程序、电子邮件或用户所选的其他信息编译器中访问到上述信息。技术提供商可以把他们现有的软件包转变为XML Web服务，并把这些服务出售给需要这些功能的第三方，或是给.NET体验提供商，用以构建新的软件包。
- Microsoft .NET允许IT部门使用其他提供商的XML Web服务，减少内部研发的开销，并能提高工作效率。

提示：

XML Web服务是建立在XML数据交换基础上的软件模型，它帮助应用程序、服务和设备一起工作。用XML进行共享的数据，彼此之间独立，但同时又能够松耦合地连接到一个执行某特定任务的合作组。

1.1.2 .NET Framework概述

.NET Framework又称.Net框架，是由微软开发，一个致力于敏捷软件开发(Agile software development)、快速应用开发(Rapid application development)、平台无关性和网络透明化的软件开发平台。.NET框架是以一种采用系统虚拟机运行的编程平台，以通用语言运行库(Common Language Runtime)为基础，支持多种语言(C#、VB、C++、Python等)的开发。.NET也为应用程序接口(API)提供了新功能和开发工具，可以支持生成和运行下一个应用程序和XML Web Services。它的强大功能与新技术结合起来，用于构建具有视觉上引人注目的用户体验的应用程序，实现跨技术边界的无缝通信，并且能支持各种业务流程。

.NET框架非常强大，主要有以下几种体现：

- Microsoft .NET Framework提供了一个一致的面向对象的编程环境，而且无论代码是在本地还是在远程服务器上都可执行，提高了软件的可复用性、可扩展性、灵活性。
- Microsoft .NET Framework提供了对Web应用的强大支持，如今是互联网的时代，大量的网络应用程序发挥着重要的作用。例如世界上最大的PC供应商DELL，其官方网站www.dell.com就是由.NET开发的。面对如此庞大的用户群体的访问，它仍旧能够保证高效率，这与.NET平台的强大与稳定是分不开的。
- Microsoft .NET Framework提供了对Web Service(Web服务)的支持，Web Service是.NET非常重要的内容。Hotmail和MSN登录时都要使用Hotmail的账户，其实支持这个账户应用的就是一个Web服务。

1.2 NET Framework 的体系结构

1.2.1 NET Framework 结构

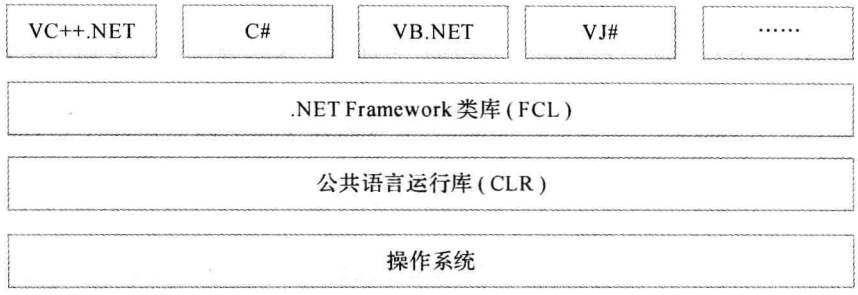

图 1.1 NET 框架体系结构

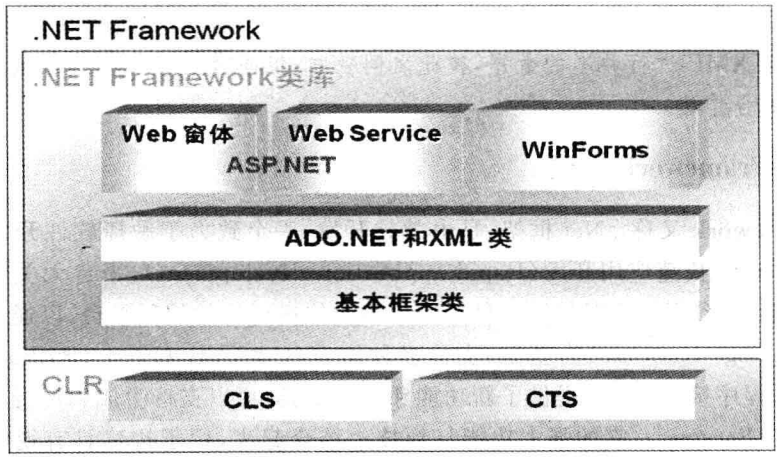

图 1.2 NET 框架结构中的组件

1.2.2 公共语言运行时(CLR)

公共语言运行时是.NET Framework 的基础。它负责在运行时管理代码的执行,并提供一些核心服务,如编译、内存管理、线程管理、代码执行、强制实施类型安全以及代码安全性验证。编译器以定义应用程序开发人员可用的基本数据类型的公共语言运行时为目标。由于公共语言运行时提供代码执行的托管环境,它提高了开发人员的工作效率并有利于开发可靠的应用程序。

CLR 也可以看作是一个在执行时管理代码的代理,管理代码是 CLR 的基本原则,能够被管理的代码成为托管代码,反之称为非托管代码。CLR 包含两个组成部分,CLS(公共语言规范)和 CTS(通用类型系统)。下面我们通过介绍.NET 的编程技术来具体了解这两个组件的功能。

1. CTS

C#和VB.NET都是公共语言运行时的托管代码，它们的语法和数据类型各不相同。CLR是如何对这两种不同的语言进行托管的呢？通用类型系统(Common Type System)用于解决不同语言的数据类型不同的问题，如C#中的整型是int，而VB.NET中是Integer，通过CTS我们把它们两个编译成通用的类型Int32。所有的.NET语言共享这一类型系统，在它们之间实现无缝互操作。

2. CLS

编程语言的区别不仅仅在于类型，语法或者说语言规范也都有很大的区别。因此.NET通过定义公共语言规范(Common Language Specification)，限制了由这些不同点引发的互操作性问题。CLS是一种最低的语言的标准，制定了一种以.NET平台为目标的语言所必须支持的最小特征，以及该语言与其他.NET语言之间实现互操作性所需要的完备特征。凡是遵守这个标准的语言在.NET框架下都可以实现互相调用。例如，在C#中命名是区分大小写的，而VB.NET不区分大小写，这样CLS就规定，编译后的中间代码必须除了大小写之外有其他的不同之处。

3..NET 编译技术

为了实现跨语言开发和跨平台的战略目标，.NET所有编写的应用都不是编译为本地代码，而是编译成微软中间代码MSIL(Microsoft Intermediate Language)。它将由JIT(Just In Time)编译器转换成机器代码。C#和VB.NET代码通过它们各自的编译器编译成MSIL，MSIL遵守通用的语法，CPU不需要了解它，再通过JIT编译器编译成相应的平台专用代码，这里所说的平台是指我们的操作系统。这种编译方式实现了代码托管，还能够提高程序的运行效率。

图1.3 .NET编译原理

1.2.3 .NET Framework 类库

.NET Framework包括可加快和优化开发过程并提供对系统功能的访问的类、接口和值类型。为了便于语言之间进行交互操作，大多数.NET Framework类型都符合CLS，因而可在编译器符合公共语言规范(CLS)的任何编程语言中使用。

.NET Framework类型是生成.NET应用程序、组件和控件的基础。.NET Framework包括的类型可执行下列功能：

- 表示基础数据类型和异常。
- 封装数据结构。
- 执行I/O。
- 访问关于加载类型的信息。
- 调用.NET Framework安全检查。