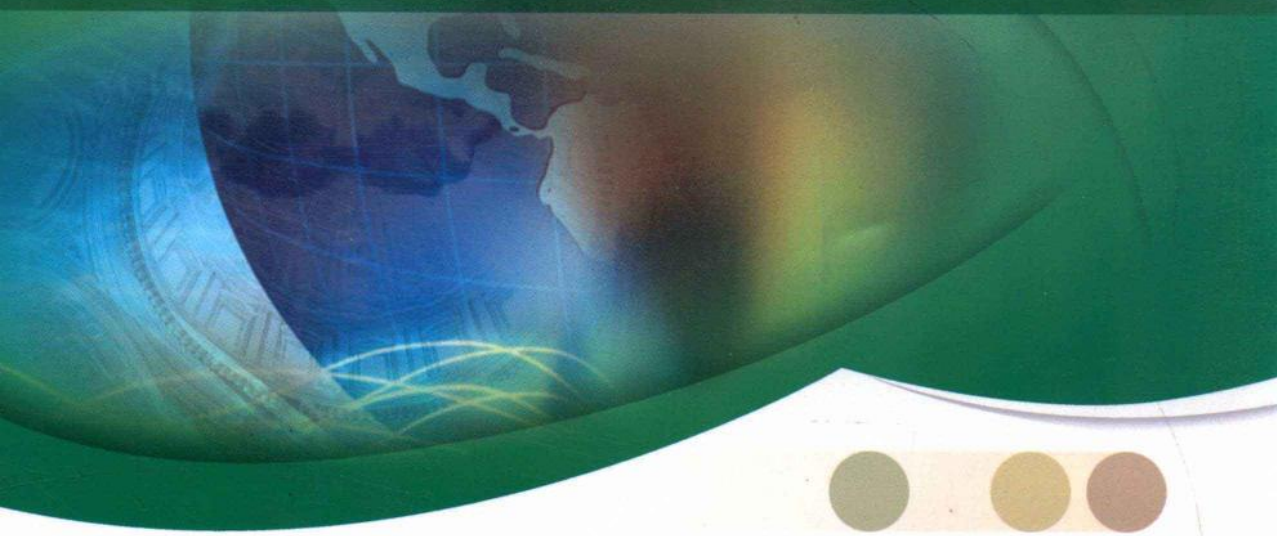
普通高等教育软件工程专业“十二五”规划教材

C#应用程序开发

主编　车战斌



科学出版社

普通高等教育软件工程专业“十二五”规划教材

C#应用程序开发

主　编　　车战斌

副主编　高　亮　薛海燕

科学出版社

北　京

内 容 简 介

本书是C#语言学习的提高篇，主要介绍C#应用开发过程中各种问题的解决方案。

本书采用问题驱动的编写方法，以软件设计中遇到的问题为线索，每章解决一个大的问题。在编写上，以实例为引导，通过解决实例问题、剖析解决过程、拓展解题思路，对每一个问题解决方案进行全面介绍。书中8～11章案例是按照学生学习编程时，先阅读、再模仿、再设计的学习规律编排设计的，旨在提高学生C#应用程序设计的总体能力。

本书共12章，其中1～7章按照问题驱动模式，分别介绍了数据库高级编程、报表设计工具、多线程高级编程、网络高级编程、系统架构、API编程基础等方面的高级编程技巧，以及辅助工具及技术。第8～11章针对不同类型系统提供实例，供学生阅读参考，其中第12章是一个综合实例，在本教材中没有给出代码，希望学生在前几个实例阅读的基础上，自己设计编写程序解决问题。

本书可作为高等院校相关专业C#应用程序开发课程的教材，也可作为C#程序设计培训机构的教材或参考书。

图书在版编目(CIP)数据

C#应用程序开发 / 车战斌主编. —北京：科学出版社，2013
普通高等教育软件工程专业“十二五”规划教材
ISBN 978-7-03-037566-7

Ⅰ. ① C… Ⅱ. ①车… Ⅲ. ① C语言－程序设计 Ⅳ. ① TP312

中国版本图书馆CIP数据核字(2013)第110083号

责任编辑：贾瑞娜 / 责任校对：朱光兰
责任印制：闫 磊 / 封面设计：迷底书装

科学出版社 出版
北京东黄城根北街16号
邮政编码：100717
http://www.sciencep.com

北京华正印刷有限公司 印刷

科学出版社发行 各地新华书店经销

*

2013年7月第 一 版 开本：787×1092 1/16
2013年7月第一次印刷 印张：26
字数：683 000

定价：53.00元

（如有印装质量问题，我社负责调换）

普通高等教育软件工程专业“十二五”规划教材

编　委　会

主任委员

李占波　郑州大学软件技术学院副院长

副主任委员

车战斌　中原工学院软件学院院长
刘黎明　南阳理工学院软件学院院长
刘建华　华北水利水电大学软件学院院长
乔保军　河南大学软件学院副院长

委　　员

高　岩　河南理工大学计算机科学与技术学院副院长
邓璐娟　郑州轻工业学院软件学院副院长
史玉珍　平顶山学院软件学院副院长
周文刚　周口师范学院计算机科学与技术学院副院长
席　磊　河南农业大学信息与管理科学学院系主任
陈建辉　郑州航空工业管理学院计算机科学与应用系副主任
张永强　河南财经政法大学计算机与信息工程学院副院长
郑延斌　河南师范大学计算机与信息工程学院副院长
谭营军　河南职业技术学院信息工程系副主任
赵素萍　洛阳师范学院信息技术学院软件工程系主任
潘　红　新乡学院计算机与信息工程学院院长

《C#应用程序开发》编委会

主　编　车战斌

副主编　高　亮　薛海燕

编　委　（以姓氏笔画排序）

车战斌　刘　洁　朱彦松　余雨萍　李妍琰

赵　冬　高　亮　韩玉民　薛海燕

此为试读，需要完整PDF请访问：www.ertongbook.com

前　言

Visual C#是 Microsoft 公司的 Visual Studio.NET 平台上一种面向对象的编程语言。它继承了 C/C++和 Java 语言的很多优点，简单易学、面向对象且类型安全，已成为当今计算机程序设计的主流编程语言之一。

“C#程序设计”课程在高校中开设已经有很多年，在软件学院等专门学习软件开发的专业中，基本形成了由“C#程序设计”、“C#应用程序开发”和“ASP.Net Web 程序设计”三门课构成的 C#系列课程。其中“C#程序设计”课程是一门编程的基础课，主要介绍 C#语法及基本的编程框架，“C#应用程序开发”课程介绍应用开发过程中各种问题在 C#中的解决方案。

本书在总结“C#应用程序开发”课程现有教材编写优点的基础上，吸收了 Visual C#发展的最新成果，结合作者多年的教学经验编写而成。采用问题驱动的编写方法，以软件设计中遇到的问题为线索，每章解决一个大的问题。在编写上，以实例为引导，通过解决实例问题、剖析解决过程、拓展解题思路，对每一个问题解决方案进行全面介绍。通过较全面的实例学习，引导学生提高 C#应用程序设计的总体能力。本书第 8～11 章中的例子，按照学生学习编程时先阅读、再模仿、再设计的学习规律编排设计，学生能通过这些案例的阅读和模仿，全面提高解决实际问题的能力。

本书共 12 章，其中第 1～7 章按照问题驱动模式，分别介绍了数据库高级编程、报表设计工具、多线程高级编程、网络高级编程、系统架构、API 编程基础等方面的高级编程技巧，以及辅助工具及技术。第 8～11 章针对不同类型系统提供实例，供学生阅读参考，其中第 12 章是一个综合实例，在本教材中没有给出代码，希望学生在前几个实例阅读的基础上，自己设计编写程序解决问题，对学生来说，既是一个综合训练，也是对自己学习的一个总结。

本书每章后都配有实训题目，大部分都是书中问题案例的扩展，有助于读者实践。因为篇幅原因，第 8～12 章的大部分源代码并没有全部印在书上，可访问网址 http://soft.zzti.edu.cn/booksourcecodes/CsharpApplicationDevelop.rar 下载阅读。本书代码力求规范易读，变量名、类名等都依照规范设计，代码部分尽量给予注释，读者通过仔细研读这些代码并且通过实训操作，可以迅速培养和提高独立解决实际编程问题的能力。

本书由车战斌担任主编，高亮和薛海燕担任副主编，第 1、8 章由薛海燕编写，第 2、3 章由朱彦松编写，第 4、10 章由余雨萍编写，第 5、6 章由车战斌编写，第 7、9 章由赵冬编写，第 11 章由韩玉民编写，第 12 章由高亮编写，全书由车战斌负责统稿。

本书是 C#语言学习的提高篇，读者应具有 C#基本知识。建议读者遵循书中介绍的方法和步骤实际建立案例程序，然后对案例程序进行修改或扩展，并通过对案例代码进行分析和讨论掌握案例背后所包含的概念、原理、知识点和方法等。本书可作为高等院校相关专业 C#应用程序开发课程的教材，也可作为 C＃程序设计培训机构的教材或参考书。

本书是按照问题驱动模式进行教材编写的一个尝试，书中难免存在疏漏之处，望读者多提宝贵意见，以便我们继续改进。本书编写过程中得到了很多老师的帮助和建议，在此一并致谢。

编　者

2013 年 5 月

本书导读

一、本书的编写思路

本书是为“C#应用程序开发”课程编写的教材。作为教材，一方面，它是教师实施教学过程的主要参考，另一方面，它也是学生学习活动的主要参考。

我们认为“C#应用程序开发”课程的教学目的有两个方面：① 学习和掌握软件开发常见问题在 C#语言中的解决方案；② 通过综合项目案例，对学生进行综合程序设计训练，以提高综合能力。

通过在本门课程之前学习的“C#程序设计”课程，读者已经学习了程序设计的通用技术，具备了使用 C#进行程序设计的基本能力。但是，在真正编写实际的应用程序时，还有很多问题需要解决，本书将其归纳为数据库编程、报表设计、多线程程序设计、网络编程、系统架构设计、API 编程与使用等六个方面。这些问题应用性很强，如何让读者快速高效地掌握这些问题的解决方案，是教学过程及教材编写共同的问题。

“从问题入手，循着解决问题的过程，逐渐引出解决问题的方法，进而使读者在解决问题的过程中掌握方法”，这是本书前 6 章的主要编写思路。教师在作为教材使用时，也最好按照这一思路进行教学。

很多读者虽然学习了 C#语言，但是当真的面对一个任务时却不知从何下手。我们认为学习编程的过程，总是要经历三个阶段：① 阅读代码阶段，就是看懂范例代码，理解别人解决问题的过程；② 模仿阶段，就是完全模仿范例的架构、方法，能够解决类似的问题；③ 自主解决问题阶段，通过阅读和模仿，积累了解决问题的经验，就可以自主解决问题了。

本书从第 8 章开始，每章都是一个综合案例，其中第 8～11 章的案例，分别侧重于不同的技术方向，适合学生进行代码阅读训练。第 12 章的案例，希望读者模仿前面的案例，自行编写出代码。

二、本书内容及使用建议

本书共分 12 章三部分。第一部分 1～7 章，主要介绍 C#编程主要技术原理和工具。第二部分 8～11 章，剖析典型应用实例的实现方法。第三部分第 12 章，提供一个综合实用项目的完整的需求与设计描述，由读者自行实现。

第 1 章学习 C#数据库高级编程。介绍常规的数据库访问架构 ADO.NET，以及新的 LINQ 数据库访问方法。本章内容中的 ADO.NET 部分一些内容在“C#程序设计课程”中已经有所介绍，可根据实际情况删减。

第 2 章讲解水晶报表和 RDLC 报表的相关知识及具体应用，学习在应用程序中嵌入报表的方法。通过本章的学习，读者应重点掌握水晶报表和 RDLC 报表的设计与制作过程，并能应用到具体项目中。

第 3 章讲解程序设计中的多线程高级编程问题。介绍什么情况下需要多线程程序、线程

的创建和执行，利用线程池完成多线程的管理。通过本章的学习，读者应该重点掌握多线程的创建及执行过程，并能应用到具体的项目中。

第 4 章通过一些典型案例来说明如何使用 C#语言实现网络通信，这些应用主要包括 Socket 通信、TCP 通信、UDP 通信、HTTP 通信、FTP 通信和 E-mail 通信等。每一种应用都有丰富的内容，本章基本覆盖了目前常用的网络编程核心技术。

第 5 章重点介绍 C#语言在应用程序设计时的基本架构，介绍三层架构及 C#语言环境下如何实现三层架构。本章内容对于提高读者代码质量具有重要作用。

第 6 章介绍 Windows API 的主要功能、API 调用机制及如何在 C#中调用 API。对于编写 Windows 应用程序来说，灵活应用 API 是一个必不可少的能力，特别是开发系统级的管理和控制程序时更是如此。本章内容教师可根据教学要求适当取舍。

第 7 章介绍 C#程序设计中的一些集成开发工具。主要介绍了数据库设计工具和版本控制工具。本章内容不是必须的，可以作为学生自学的参考内容。

第 8～10 章，每章介绍一个较完整的案例，分别侧重数据库编程、多线程编程和网络编程，与前面的相关章节有一定的对应关系。这几章的内容，主要作为学生阅读与训练内容，可少讲多练，建议教师可挑选 1～2 章，对系统架构和具体实现进行较详细的讲解，帮助学生理解，其余内容可不讲或少讲，让学生自行阅读。

第 11 章是 Windows 环境下典型界面特效设计与系统控制实例，本部分内容教师也可根据实际情况取舍。

第 12 章是一个综合训练案例，模拟软件公司中程序员工作设计的实例。给出了需求描述和设计，但没有给出实现代码。即不要求学生对系统进行设计，而是希望学生能在前面几章学习的基础上，通过阅读系统需求说明文档，独立地设计编程实现系统，并且调试完成系统开发。这里要求学生具有 UML 的基础，能读懂基本的软件设计图。教师可以简单介绍一下系统需求及设计，以帮助学生理解。本章实际上就是本教材的一个综合实训项目。

三、本书特色

本书作为基于案例的“C#应用程序开发”教材，具有如下特色。

第一，问题驱动，循序渐进。无论是关键技术原理的讲解，还是具体实例的剖析，都基于解决某一典型实际应用问题。同时，教材的三部分内容构成一条循序渐进的“原理理解→剖析模仿→实训提高”学习路线图。

第二，典型实例，躬行实践。软件开发能力培养过程是不断实践、不断提高的过程。本教材精选典型应用实例，读者掌握实现方法后，便可举一反三，灵活应用。所有实例的源代码都可以在本教材所提供的网址中下载，代码按规范标注详尽注释，便于理解。

第三，遵循规范，与时俱进。所有实例设计图表、程序范式、对象及标识符命名等都遵循软件设计规范，使读者在训练中培养规范化编程意识，掌握规范化编程方法。另外，开发环境采用 Visual Studio 2010，使读者可了解、掌握和利用.Net Framework 的新功能及技术。

目　录

此为试读，需要完整PDF请访问：www.ertongbook.com

第 1 章　数据库高级编程

数据库操作对于应用程序来说是一个很重要的主题。本章将从实用的角度重点介绍 ADO.NET 和 LINQ 两种技术。ADO.NET 提供两种数据库访问方式——连接式访问和断开式访问。本章将采用连接式访问实现数据的增、删、查、改，采用断开式访问实现多条件的模糊查询。同时介绍 LINQ 这一新技术，重点说明利用 LINQ to SQL 操作数据库进行增、删、查、改的方法。

1.1　为什么要用数据库

应用程序需要一个保存数据的地方，可以采取自定义格式的文件保存也可以采取数据库的方式保存，这取决于应用程序的具体情况。一般的自定义格式数据文件的格式是应用程序特定的，不具备通用性，需要自己定义访问数据的全部方法，但对特定程序来说可以提高运行效率。采用数据库方式主要有三方面的优点：① 数据库将数据管理从应用程序设计中清晰地分离出来，应用程序的设计者可只考虑数据的逻辑关系而不必考虑数据存储与管理的细节。② 采用数据库可以有效地对数据进行访问控制管理。③ 可以保证数据之间的一致性。基于上述优点，目前除特殊数据管理要求外，大部分应用程序都采用数据库方式管理数据。

常见的操作海量数据的软件，如 QQ 聊天软件，以及各种管理系统，如股票软件、医疗系统、银行系统、超市收银系统、电子商务网站、铁路售票系统、航空售票系统等，都是数据库的典型应用。

我们身边的校园里也随处可见大量数据的使用，如高考报名及查分系统、教务管理系统、机房管理系统、图书馆管理系统和餐厅管理系统等。

本章将以机房管理系统为例，实现学生基本信息的管理，从而完成对数据库高级编程知识的学习。

1.2　如何访问数据库——ADO.NET

C#编写的应用程序如何获取存放在数据库中的数据？如何实现对数据的查询？如何实现对数据的添加、删除、更新呢？在.NET 中，对数据库的访问支持集中体现在 ADO.NET 技术上，本节将从多个实例引入 ADO.NET 操作数据库的常用方法，并介绍 ADO.NET 的组成结构和基本对象。

1.2.1　如何连接数据库

要想访问数据库，首先需要在应用程序和数据库之间建立连接。如何建立连接呢？

例 1.1　实现“机房管理系统”中的数据库连接。

首先，创建数据库。创建名为 mydb 的 SQL Server 数据库，其中包含 Student 表和 Department 表，表结构的定义分别如表 1-1 和表 1-2 所示。

表 1-1　Student 表的结构

字段名	数据类型	长　度	主　键	含　义
StuId	nvarchar	20	是	学号
StuName	nvarchar	10	否	姓名
StuAge	int	4	否	年龄
StuGender	nvarchar	2	否	性别
DepId	nvarchar	20	否	所属系部编号

表 1-2　Department 表的结构

字段名	数据类型	长　度	主　键	含　义
DepId	nvarchar	20	是	系部编号
DepName	nvarchar	20	否	系部名称

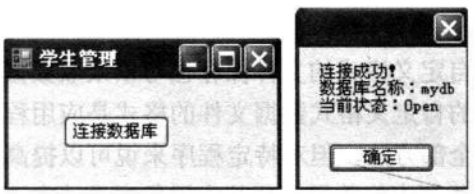


图 1-1　StudentManage 窗体的设计和运行结果

然后，创建 Windows 窗体应用程序。创建名为 StuManage 的 Windows 窗体应用程序，并添加 StudentManage 窗体，该窗体的设计界面和运行结果如图 1-1 所示，“连接数据库”按钮的 name 属性为 btnConnect，该按钮的单击事件处理方法如下。注意，需要在窗体中手动添加命名空间“using System.Data.SqlClient;”。

```
private void btnConnect_Click(object sender, EventArgs e)
                                                    //"连接数据库"按钮的单击
    {
        SqlConnection conn = new SqlConnection();//新建连接
        conn.ConnectionString = "server=.;database=mydb;uid=test;pwd=test";
                                                    //设置连接字符串
        conn.Open();//打开连接
        MessageBox.Show("连接成功!\n 数据库名称："+conn.Database+"\n 当前状态：
                        " + conn.State); //显示连接信息
        conn.Close();                               //关闭连接
    }
```

C#应用程序通过 ADO.NET 提供的 SqlConnection 类与 SQL Server 数据库建立连接。如本例中的代码：

```
SqlConnection conn = new SqlConnection();//新建连接
conn.ConnectionString = "server=.;database=mydb;uid=test;pwd=test";
                                                    //设置数据库连接字符串
```

“数据库连接字符串”包含要连接的数据库的信息，如 server 指定数据库服务器名称、database 指定数据库名称、uid 指定数据库账户、pwd 指定账户密码。

说明　server 指定数据库服务器名称，本例中的“.”表示本机，本机的写法也可以是“机器名”、“(local)”或“127.0.0.1”等。如果不是本机可以写对方数据库服务器的 IP 或服务器名称。以上两行语句也可以合并写为：

```
SqlConnection conn = new SqlConnection("server=.;database=Mydb;uid=test;pwd=test");
```

建立数据库连接后，则需要打开该连接。如本例中的代码：

```
conn.Open();
```

数据库连接使用完毕，则可以调用 Close 方法关闭该连接。如本例中的代码：

```
conn.Close();
```

SqlConnection 类的重要属性和方法如表 1-3 和表 1-4 所示。

表 1-3　SqlConnection 类的重要属性

属　性	说　明
ConnectionString	获取或设置用于打开 SQL Server 数据库的字符串
Database	获取连接打开的数据库名称
State	获取连接对象的状态

表 1-4　SqlConnection 类的重要方法

方　法	说　明
Open	打开数据库连接
Close	关闭数据库连接

1.2.2　如何查询数据

如何按姓名查询学生信息？如何按多个条件动态查询？能不能实现模糊查询？

例 1.2　实现“机房管理系统”中的“学生管理”界面的查询功能，要求按“学号”、“姓名”、“年龄”、“性别”或“系部”五个字段任意查询，并且由用户动态选择查询条件。

修改例 1.1 中的 StudentManage 窗体，其界面设计如图 1-2 所示。窗体中控件的设置，按 Tab 顺序描述如表 1-5 所示。

图 1-2　StudentManage 窗体的界面设计

表 1-5　StudentManage 窗体中的控件

Tab 顺序	控件类型	属性名	属性值
1	ComboBox	Name	cmbSearchField
		DropDownStyle	DropDownList
2	ComboBox	Name	cmbOperator
		DropDownStyle	DropDownList
3	TextBox	Name	txtSearchContent
4	Button	Name	btnSearch
		Text	查询
5	DataGridView	Name	dgvStuList
		AllowUserToAddRows	False
6	Button	Name	btnAdd
		Text	添加
7	Button	Name	btnUpdate
		Text	修改
8	Button	Name	btnDelete
		Text	删除

为了在 DataGridView 控件中显示中文列名，所以为 dgvStuList 添加列，如表 1-6 所示。

表 1-6　dgvStuList 列设置表

列头名	DataPropertyName 属性值
学号	stuId
姓名	stuName
年龄	stuAge
性别	stuGender
所属系部	depName

编写窗体的 Load 事件代码如下：

```
string currSearchContent = "";//该变量保存查询条件，当变量为空串时，代表查询所有记录
private void StudentManage_Load(object sender, EventArgs e)
{//窗体的 Load 事件
    cmbSearchField.Items.Add("学号");//设置查询字段的选项
    cmbSearchField.Items.Add("姓名");
    cmbSearchField.Items.Add("年龄");
    cmbSearchField.Items.Add("性别");
    cmbSearchField.Items.Add("系部");
    bindGridView(); //显示所有记录
}
public void bindGridView()
{  //显示查询结果
   SqlConnection conn = new SqlConnection("server=.;database=mydb;uid=test;
                                          pwd= test ");//新建连接
    //设置查询命令内容  本例实现 student 表和 department 表的组合查询
    string sql="select student.depid as depid,depname,stuid,stuname,stuage,stugender
              from student left join department on student.depid=department.depid";
    if (currSearchContent != "")//如果查询条件不空，则构造带条件的查询语句
       sql += " where " + currSearchContent;
    SqlDataAdapter da = new SqlDataAdapter(sql, conn);
                                                  //使用 DataAdapter 类存储命令
    DataSet ds = new DataSet();//新建数据集
    da.Fill(ds);//执行查询，并将结果填充至数据集
    dgvStuList.DataSource =ds.Tables[0];//将数据集中的查询结果显示在 DataGridview 控件中
}
```

双击“查询”按钮，编写该按钮的单击事件代码。

```
private void btnSearch_Click(object sender, EventArgs e)
   {//“查询”按钮的单击事件
    string searchField = "";//该变量用于保存待查找字段的名称
    switch (cmbSearchField.SelectedItem.ToString())
                          //将 cmbSearchField 中选中的项转换成对应的数据表字段名
    {
        case "学号":
            searchField = "stuid"; break;
        case "姓名":
            searchField = "stuname"; break;
```


此为试读，需要完整PDF请访问：www.ertongbook.com