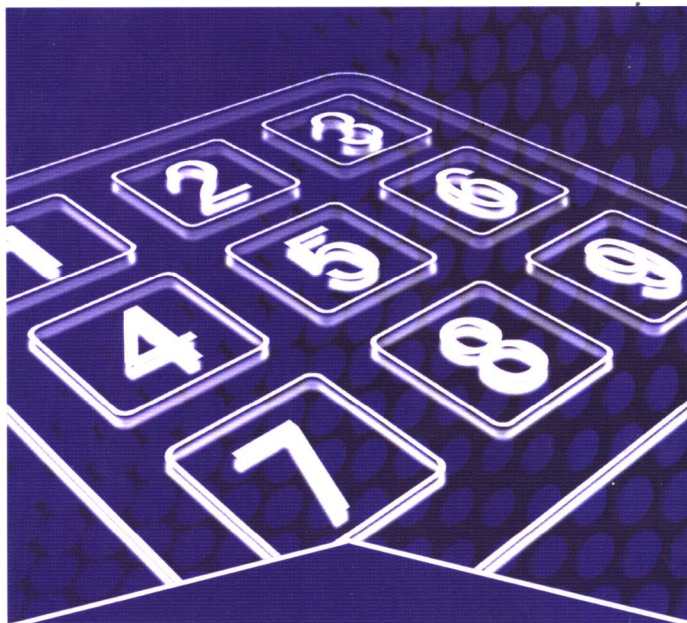


电脑编程实例导航丛书



C# 编程

典型实例解析

《电脑编程技巧与维护》杂志社 编著



中国水利水电出版社
www.waterpub.com.cn

电脑编程实例导航丛书

C# 编程典型实例解析

《电脑编程技巧与维护》杂志社 编著

中国水利水电出版社

内 容 提 要

本书根据 C# 的不同应用对象,精选了 48 个 C# 编程典型实例,分为 3 章。第 1 章为基础与应用编程实例,为初学者提供 C# 基础应用编程入门的实例;第 2 章为网络与通信应用编程实例,介绍使用 C# 实现网络与通信应用编程的方法与技巧;第 3 章为图形图像及计算机安全与维护编程实例,介绍使用 C# 实现图形图像及计算机安全与维护编程的方法与技巧。全书每一个实例解析都本着实用第一的原则,介绍读者感兴趣的 C# 编程应用开发的方法与技巧。书中每一个编程实例的讲解一般都是先给出设计目标,然后介绍实现该目标的基本思想和方法,最后详细给出其核心程序的源代码,对程序的关键部分进行讲解,并给出程序的运行效果,浓缩了 C# 应用程序设计的精华。

本书是 C# 编程人员的实践经验和智慧的结晶,通过一个个典型实例解析,总结了 C# 编程的难点和关键技术,详尽而具体地解析了使用 C# 进行应用程序开发的方法与技巧。该书对于有 C# 应用基础的编程人员和应用开发人员以及初学 C# 编程的新手都是一本很实用的参考书。

本书实例源代码可以从中国水利水电出版社网站免费下载,网址为: <http://www.waterpub.com.cn/softdown/>。

图书在版编目 (CIP) 数据

C# 编程典型实例解析/《电脑编程技巧与维护》杂志社编著. —北京:中国水利水电出版社,2007

(电脑编程实例导航丛书)

ISBN 978-7-5084-4238-9

I . C … II . 电… III . ①C 语言—程序设计
IV . TP312

中国版本图书馆 CIP 数据核字 (2006) 第 143334 号

书 名	C# 编程典型实例解析
作 者	《电脑编程技巧与维护》杂志社 编著
出版 发行	中国水利水电出版社 (北京市三里河路 6 号 100044) 网址: www.waterpub.com.cn E-mail: mchannel@263.net (万水) sales@waterpub.com.cn 电话: (010) 63202266 (总机)、68331835 (营销中心)、82562819 (万水)
经 售	全国各地新华书店和相关出版物销售网点
排 版	北京万水电子信息有限公司
印 刷	北京蓝空印刷厂
规 格	787mm × 1092mm 16 开本 20.75 印张 622 千字
版 次	2007 年 1 月第 1 版 2007 年 1 月第 1 次印刷
印 数	0001—4000 册
定 价	36.00 元

凡购买我社图书,如有缺页、倒页、脱页的,本社营销中心负责调换

版权所有·侵权必究

丛书序

《电脑编程技巧与维护》杂志是为从事电脑编程、系统应用开发的人员创办的专业性和实用性都很强的技术刊物。自1994年创刊以来，始终以“实用第一，智慧密集”为宗旨，坚持“质量第一”、“读者第一”的原则，为广大电脑编程爱好者、软件开发人员和专业计算机系统维护人员提供第一手技术资料、编程技巧和维护经验；紧紧跟踪计算机软硬件技术发展和应用趋势，不断求变创新，针对软件开发过程中的许多关键技术问题着重提供各类解决方案，在业内获得一致好评，是广大编程和维护人员的首选刊物。在栏目内容上，选题覆盖面广，涉及技术领域宽、信息量大，帮助程序员开阔视野；在技术水平上，始终把握计算机技术发展的大方向，提供先进、详尽、准确的技术指导，并在长期工作中与国际性大公司建立了良好的合作关系，为读者提供全球最新、最全的实用信息；在实用性上，稿源来自于专业开发和维护人员的实践经验，是普通书籍难以获得的编程心得、体会与技巧。

2006年是《电脑编程技巧与维护》创刊十二周年，为了最大限度地开发和利用本刊宝贵而丰富的资源，更好地服务和真诚回报多年来一直关爱和支持本刊的广大读者，《电脑编程技巧与维护》杂志社和中国水利水电出版社共同策划出版了这套“电脑编程实例导航丛书”。

这套丛书包括《Visual C/C++ 系统开发典型实例解析》、《Visual C/C++ 图形图像与游戏编程典型实例解析》、《Visual Basic 编程典型实例解析》、《Delphi 编程典型实例解析》、《C#编程典型实例解析》、《Java 编程典型实例解析》、《计算机系统安全与维护编程典型实例解析》、《计算机网络与通信编程典型实例解析》、《编程疑难问题解析126例》，一套9册共684个典型实例。每册书的编程实例均依不同的编程应用分成若干章，条目清晰可查，使用极为方便。

这套丛书选编了《电脑编程技巧与维护》杂志近两年发表的和一部分尚未发表而又极为实用、精彩的典型编程实例。该套书的特点是：其各册内容来自编程高手的智慧和经验总结，其中不少文章的作者是业界资深程序员和技术专家，内容有深度、思路有新意、讲解深入浅出，编程技巧新颖实用，构思巧妙；丛书中的实例都是作者从实际项目提炼出的开发范例，实例讲解部分先给出设计目标，然后介绍实现目标的基本思想和方法，最后详细给出其核心程序的源代码，对程序的关键部分进行讲解并给出程序的运行效果；丛中每一个实例

的程序源代码都经过上机调试通过，对编程中的疑难问题进行了深入解答，给程序开发人员移植源代码和学用编程带来了方便，加快了编程应用的步伐。全套书既讲究内容的深入性、专业性、权威性和实用性，同时兼顾轻松、通俗易懂、时效性强的特点。

这套丛书是《电脑编程技巧与维护》资源的二次深入开发，浓缩了当前主流编程语言 Visual C/C++、Visual Basic、Delphi、Java、C#等程序设计的精华，其目的是力求为读者建造一个真正的知识整合、编程思想、编程技术、技巧交流的平台，让读者从中学习到编程高手的诀窍，丰富读者的编程技巧，拓宽读者的编程思路，迅速提升读者的程序开发能力。对电脑编程人员来说，程序开发能力的提高，除了对语言和算法的不断钻研学习、不断实践、不断总结提高，练好基本功，打好基础外，还要集思广益，善于学习，善于借鉴参考别人的经验，深入透彻地理解其中的精髓，然后溶入到自己的设计方案中去，这无疑是一条有效的学习途径，对于自身编程能力的增强和编程水平的迅速提高十分重要，这也正是我们编写这套丛书想要达到的目的。

这套丛书可作为高等院校学生进行课程项目开发、毕业项目设计的参考书，也可作为软件从业人员及编程爱好者的珍藏宝典，还可作为高等培训学校的实例教程。

《电脑编程技巧与维护》杂志社
中国水利水电出版社
2006年5月

前 言

C#语言是微软公司针对 .NET 框架专门开发的一种面向对象的编程语言，它继承和发展了 C/C++ 的语言特征和面向对象的开发思想，吸收了 Delphi 和 Visual Basic 的优点，同时也克服了运用 C/C++ 进行项目开发时难度大和周期长的缺点。它是以 Internet 为中心面向组件设计的全新语言。在 .NET 平台上运行 C# 可以很便捷地进行企业大型应用开发，特别是分布式应用。

为了让更多的电脑编程人员比较集中地学习和参考 C# 应用编程的实践经验、心得体会和编程技巧，在“电脑编程实例导航丛书”中，《C# 编程典型实例解析》一书精选了《电脑编程技巧与维护》杂志近两年共 24 期已发表的精彩编程实例 48 例。根据 C# 的不同应用对象，将精选的 48 个应用实例分为 3 章。第 1 章为基础与应用编程实例，为初学者提供 C# 基础应用编程入门的实例；第 2 章为网络与通信应用编程实例，介绍使用 C# 实现网络与通信应用编程方法和技巧；第 3 章为图形图像及计算机安全与维护编程实例，介绍使用 C# 实现图形图像及计算机安全与维护的方法与技巧。全书每章都本着实用第一的原则，通过一个个应用实例介绍使用 C# 进行应用程序开发的方法与技巧。

本书的主要特色如下：第一，每章都是通过一个个的实例解析归纳和总结 C# 应用编程方法和技巧，避免了枯燥、空洞的理论，并且每个实例都具有很强的实用性和代表性。在实例的讲解上一般都是先给出设计目标，然后介绍实现该目标的基本思想和方法，最后详细给出其核心程序的源代码，并对程序的关键部分进行讲解，给出程序的运行效果。第二，所选的每个实例都是从事 C# 应用编程人员的经验总结，具有很强的实用性，其中很多编程方法和技巧可供读者借鉴。第三，每个实例的程序源代码都经过上机调试，给程序开发人员移植源代码带来方便。本书提供实例中的源程序代码，读者可以从中国水利水电出版社网站免费下载，网址为：<http://www.waterpub.com.cn/softdown/>。

本书结构清晰、层次分明、实例典型而实用，是《电脑编程技巧与维护》杂志丰富资源

的二次开发，浓缩了 C# 程序设计的精华。该书对于有 C# 应用基础的编程人员和应用开发人员以及初学 C# 编程的新手都有很好的参考价值。书中疏漏甚至错误之处在所难免，恳请广大读者批评指正。

《电脑编程技巧与维护》杂志社

2006 年 12 月

目 录

丛书序

前言

第 1 章 基础与应用编程实例

例题 1	在 C# 中用函数实现 Excel 单元格纵向合并	2
例题 2	在 Visual C# 中运用 API 函数编写多功能关机程序	7
例题 3	在 C# 中实现高效的集合类	13
例题 4	C# 中消息的处理	18
例题 5	通过 C# 实现集合类纵览 .NET Collections 及相关技术	20
例题 6	利用平台调用技术在 C# 中调用 DLL 函数	28
例题 7	C# 开发 ASP. NET 验证控件	33
例题 8	设计 C# 定制控件实现贝塞尔曲线	42
例题 9	文本关键词醒目显示和智能提示的 C# 编程实现	48
例题 10	开发财务专用数字的掩码控件方法与技巧	51
例题 11	C# 对通用组件的编程	60
例题 12	Delphi 与 C# 的混合编程	64
例题 13	在 C# 中使用 Visual J# 类压缩文件和数据	70
例题 14	Visual C# 的文件批处理编程	77
例题 15	在 Visual C# 下用 FileSystemWatcher 监控作业	81
例题 16	Visual C# 文件编程之分割合并文件	84
例题 17	基于 C# 的 GML 文档解析	96
例题 18	C# 中动态数组设计原理与方法	106
例题 19	用 C#. NET 开发通用报表	109
例题 20	在 C# 中利用正则表达式实现字符串搜索功能	116
例题 21	C# 与 Fortran 的混合编程技术及实例	125
例题 22	用 C# 编程制作媒体播放器	129
例题 23	一种基于 C# 的企业库存管理系统的实现	141

第 2 章 网络与通信编程实例

例题 24	C# 利用 WMI 类中的 Win32_NetworkAdapterConfiguration 实现网卡 IP 地址的自动配置	146
例题 25	Visual C# 实现 Ping 命令	150
例题 26	Visual C# 创建 Tracert 命令	160

例题 27	用 C#编写多线程搜索引擎	172
例题 28	Visual C#套接字编程	181
例题 29	用 C#实现 Http 代理服务器	190
例题 30	Visual C#使用 POP3 协议构建客户端邮件接收程序	200
例题 31	使用 .NET Remoting 开发分布式应用程序	210
例题 32	用 C#实现中英文手机短信 PDU 编程解码	215
例题 33	基于 C#语言的自动气象信息发布程序开发	222
例题 34	用 C#开发手机短信收发程序	225
例题 35	C#实现点对点通信	230
例题 36	利用 C#编程保护你的雅虎通聊天	239
例题 37	用 C#实现 BT 体育发布区种子资源的自动下载功能	243

第 3 章 图形图像与游戏及计算机安全与维护编程实例

例题 38	Visual C#图形图像编程基础及基本绘图方法	250
例题 39	Visual C#编程捕获屏幕	258
例题 40	Visual C#编程处理图片	266
例题 41	基于 C#的 3D 动画编程	280
例题 42	经典游戏俄罗斯方块的数据建模和 C#实现	284
例题 43	用 Visual C#实现对注册表的操作	294
例题 44	用 C#调用 Microsoft. Win32 命名空间中的类实现对注册表的读取管理	298
例题 45	用 C#实现文件寄生隐藏	304
例题 46	用 C#中的类实现数据表的插入与更新	308
例题 47	RC5 加解密算法的 C#实现	314
例题 48	Visual C#中文验证	318

第 1 章

基础与应用编程实例

例题 1 在 C# 中用函数实现 Excel 单元格纵向合并

在软件开发中，做报表工具有很多种，如水晶报表、Cell 组件、数巨报表 (Max Reports)、Infor-Report 等，这些报表都有自己的特色和优势，都能够解决某一方面的问题。但是在实际开发中，用户却更希望能够用 Office 作为报表工具，特别是把数据导出到 Excel 中。当然如果不需要进行报表头统计、纵横向合并等操作，直接把数据导入 Excel 中，实现起来是相当容易的。但是在实际开发中，大多数报表头需要实现数据信息统计，且统计信息需填入合并行单元格中，还需对导入 Excel 中的一些内容相同的单元格信息进行纵向的合并。根据笔者做项目时对各个报表的具体要求，结合用 VS.NET 的开发实际，用 C# 编写一个实现单元格纵向合并的通用函数。下面通过一个实例来说明该函数的用法。

一、问题的提出

有这样的一个数据信息表，它有日期、区域名称、组别、队号 4 个字段，表的数据信息见表 1-1。

表 1-1

日期	区域名称	组别	队号
2005.6.26	重庆	重庆 1 组	323 队
2005.6.26	新疆	新疆 1 组	479 队
2005.6.26	重庆	重庆 1 组	837 队
2005.6.26	兰州	兰州 1 组	321 队
2005.6.26	新疆	新疆 1 组	475 队
2005.6.26	兰州	兰州 3 组	324 队
2005.6.26	新疆	新疆 2 组	322 队
2005.6.26	新疆	新疆 3 组	354 队
2005.6.26	重庆	重庆 1 组	400 队
2005.6.26	兰州	兰州 3 组	789 队
2005.6.26	兰州	兰州 1 组	478 队
2005.6.26	新疆	新疆 3 组	355 队

需求如下：调用 Excel 报表模板实现报表输出，以日期作为查询条件，把表中的内容导入到 Excel 中，把 Excel 中区域名称列和组列单元格中内容相同项进行合并，输出结果采用如表 1-2 的形式。

上述题目就是典型的调用 Office 中的 Excel 作为报表工具，并且用程序实现报表纵横向的合并，面对这样的问题，在网上进行搜索，但大多数都是直接导出到 Excel 表中，很少有实现纵横向合并方面的资料。在网上找到了几天没有找好的处理方式，自己静下心来想一想，在 Excel 中能够实现合并，何不使用宏来看看呢！考虑宏代码的过程顿时豁然开朗，想到通过参数传递来实现纵横向的合并。

二、实现原理及过程

1. 数据库表设计

为了能够通过实例来解决上述问题，设计如下的数据信息表，表名为：生产信息表 (SCXXB)，

具体字段为：rq（日期）、qy（区域名称）、zb（组别）、dh（队号），采用日期和队号为主键，其具体的表信息如图 1-1 所示。

rq	dh	qy	zb
2005-6-26	321队	兰州	兰州1组
2005-6-26	322队	新疆	新疆2组
2005-6-26	323队	重庆	重庆1组
2005-6-26	324队	兰州	兰州3组
2005-6-26	354队	新疆	新疆3组
2005-6-26	355队	新疆	新疆3组
2005-6-26	400队	重庆	重庆1组
2005-6-26	475队	新疆	新疆1组
2005-6-26	478队	兰州	兰州1组
2005-6-26	479队	新疆	新疆1组
2005-6-26	789队	兰州	兰州3组
2005-6-26	837队	重庆	重庆1组

图 1-1

表 1-2

2005. 6. 26		
区域名称	组别	队号
重庆	重庆 1 组	323 队
		837 队
		400 队
兰州	兰州 1 组	321 队
		478 队
	兰州 3 组	789 队
新疆	新疆 1 组	475 队
		479 队
	新疆 2 组	322 队
	新疆 3 组	354 队
		355 队

2. 实现原理

要进行同列中内容相同单元格的合并，无论纵向还是横向，都是找内容相同的单元格通过程序来实现这些单元的合并。可以通过传递 Excel 属性页、导入到 Excel 页中的起始行、总行数、需要合并列单元格的初始信息编写一个函数实现合并。通过在函数内设记录合并字符串变量 qy1、记录当前行单元格的位置 me1、记录上条合并的单元格位置 ms1，通过 Excel.Range 定义 rg 和 rg1，用 rg 获得下一行的填充内容，用 rg1 选取上条合并位置和当前行的合并区域，用 strtemp 获得当前区域的内容并进行合并区域信息的传递。通过一个以总行数作为循环总次数，从起始行到终止行进行循环，以当前行单元格内容是否与上行单元格内容是否相同，以及当前行单元格位置和上条单元格合并位置是否相同为条件，一行一行地查找，直到当前单元格且上行单元格内容不同且当前单元格位置与上条单元格合并的位置不同，把这些相同的单元进行合并，把下一个合并单元格的字符串区域的值通过 strtemp 传递给 qy1，记录当前行单元格的位置 me1，记录上条合并的单元格位置 ms1，然后再继续向下查找，这样反复循环，直至结束。其具体过程见下面的函数及备注。

其中 Excel 属性页 (Worksheet mySheet)、起始行 (startLine)、总行数 (RecCount)、要合并的列 (Col) 的具体实现过程及实现原理如下:

```
private void MergeCell(ref Worksheet mySheet, int startLine, int RecCount, string Col)
{
    string qy1 = mySheet.get_Range(Col + startLine.ToString(), Col + startLine.ToString())
        .Text.ToString(); //获得 起始行 合并列单元格的填充内容
    Excel.Range rg1, rg;
    int ms1, mel;
    string strtemp = "";
    int ntemp = 0;
    mel = startLine;
    //从起始行至终止行做循环
    for (int i = 1; i <= RecCount; i++)
    {
        ntemp = startLine + i;
        rg = mySheet.get_Range(Col + ntemp.ToString(), Col + ntemp.ToString());
        //获得下一行的填充内容
        strtemp = rg.Text.ToString().Trim();
        if (qy1.Trim() != strtemp.Trim())
            //内容不等于上条内容
            {
                ms1 = mel; //记录上条合并的位置
                mel = i + startLine - 1; //记录当前行的位置
                if (mel - ms1 > 0) //如果当前行和上条合并位置
                    //不相同,则存在合并项,合并
                {
                    rg1 = mySheet.get_Range(Col + ms1.ToString(), Col + mel.ToString());
                    //选取上条合并位置和当前行的合并区域
                    rg1.ClearContents(); //清空要合并的区域
                    rg1.MergeCells = true; //合并区域
                    if (Col == "A")
                        mySheet.Cells[ms1, 1] = qy1;
                    else if (Col == "B")
                        mySheet.Cells[ms1, 2] = qy1;
                }
                mel += 1; //当前位置向下移动
                strtemp = mySheet.get_Range(Col + mel.ToString(), Col + mel.ToString()).Text.ToString();
                if (strtemp.Trim() != "" || 1) //获得当前区域的内容
                    qy1 = strtemp;
            }
    }
}
```

因为该项目中只有纵向 A 和 B 中的单元格需要合并,只对这种情况进行了书写,如果有多行,可以通过传递过来的 COL 进行添加,如通过 else if (Col == "C") mySheet.Cells[ms1, 3] = qy1 来对纵向 C 中单元格中内容相同的单元格进行合并,其他与此类似,都可以通过该方法来实现。

3. 报表系统设计及实现过程

下面通过一个具体的项目来实现该操作过程,因为输出采用 Excel 作为报表,可以建立一个报表模板放在硬盘的一个具体的目录中,做报表输出时可以直接调用该模板,现把命名为 scrxxs.xls 和 test.xls 的 Excel 模板放在与项目在同一个目录的盘中。scrxxs.xls 模板具体形式如表 1-3 所示。

表 1-3

区域名称	组别	队号

数据报表的打印通过按钮事件来实现，因为开发平台是 VS. NET，通过在 VS. NET 中建立一个工程来实现其打印功能。在建立的 WebForm1.aspx 界面上添加一个按钮控件、一个 Label 控件，在 Label 控件的 text 属性中写上报表数据信息打印系统。在 Button 控件的属性上分别写上打印报表，因后台采用 SQL 作为后台数据库，在 using 的应用中添加上 using System. Data. SqlClient; 输出采用 Excel，在使用它之前必须把它的库函数引用到项目中，通过在 VS. NET 项目中添加 object library，然后应用 Excel，再将数据库的连接统一写在 Web.config 中，在项目中通过使用 Web.config 来对其进行引用，其数据库连接的具体写法如下：

```
< appSettings>
< add key = "连接字符串" value = "Password" = 123;
Persist Security Info = True; User ID = sa; Initial Catalog = Scxxglfxxt; Data Source = 10. 80. 45. 88" />
< /appSettings>
```

在 Web.config 添加以上代码，就完成数据库连接的书写，可以通过修改上面的密码参数、用户名参数、数据库名参数、机器的 IP 来设定不同机器的连接。

下面就在 WebForm1 中通过具体的代码实现调用 Excel 模板作报表纵横向的合并。在项目的 using 引用中添加上 using System. Data. SqlClient; using System. Configuration; using Excel; 通过 private static string Connstring 定义连接字符串，通过 private string mySelectQuery 定义查询字符串，通过 private string strrq 定义日期字符串。在 Page_Load 中可以通过如下方式来实现连接字符串的读取。

```
private void Page_Load(object sender, System.EventArgs e)
{
    if(!this.IsPostBack)
    {
        Connstring = ConfigurationSettings.AppSettings["连接字符串"]. ToString();
    }
}
```

下面通过一个按钮事件的部分代码来实现报表的生成。在按钮事件中，通过程序来实现对 Excel 模板调用生成 Excel 报表的保存。

```
private void Button1_Click(object sender, System.EventArgs e)
{ Workbook myBook = null;
Worksheet mySheet = null;
Excel.ApplicationClass ExlApp = new ApplicationClass();
ExlApp.Visible = true;
object oMissiong = System.Reflection.Missing.Value;
string reqpath = this.Request.PhysicalPath;
int pos = reqpath.LastIndexOf("\\");
reqpath = reqpath.Substring(0, pos);
ExlApp.Workbooks.Open(reqpath + "\\scxx.xls",
oMissiong, oMissiong, oMissiong, oMissiong, oMissiong,
oMissiong, oMissiong, oMissiong, oMissiong, oMissiong,
oMissiong, oMissiong); //, oMissiong); oMissiong);
//Office 2000, 其需要 13 个参数, Office XP 则需要 15 个参数
myBook = ExlApp.Workbooks[1];
mySheet = (Worksheet)myBook.Worksheets[1];
//上述为定义和调用 Excel 的过程
Excel.Range rg; //定义 Excel 区域范围
strrq = DateTime.Today.ToString("yyyy-MM-dd");
mySheet.Cells[1, 1] = strrq.Year.ToString() + "年" + strrq.Month.ToString() + "月" + strrq.Day.ToString()
```

```

() + "日"; //把日期导入 mySheet.Cells[1, 1]中
mySelectQuery = "SELECT dh, qy, zb FROM SCXXB where SCXXB.rq = '" + strrq + "'";
SqlConnection myConnection = new SqlConnection(Connstring);
SqlCommand myCommand = new SqlCommand(mySelectQuery, myConnection);
myConnection.Open();
SqlDataReader myReader;
myReader = myCommand.ExecuteReader();
int recount = 0;
while (myReader.Read())
{recount = recount + 1;
} //通过 SQL 查询记录的总条数
myReader.Close();
myConnection.Close();
int DetailStartLine = 3; //定义导入的起始行
for(int i = 1; i < recount; i++)
{rg = mySheet.get_Range("A" + (DetailStartLine).ToString(), "C" + (DetailStartLine).ToString());
rg.Copy(oMissiong);
rg.Insert(XlInsertShiftDirection.xlShiftDown);
} //生成记录条数一样的行数, 使生成行与模板行
//的样式一样, 这样避免没有边框、单元格的形式不统一等问题
int Curhs = DetailStartLine; //给起始行赋值
下面通过查询把数据库中的信息读到 Excel 模板中:
mySelectQuery = "SELECT dh, qy, zb FROM SCXXB
where SCXXB.rq = '" + strrq + "' ORDER BY qy, zb"; //从数据表中取数据
myConnection = new SqlConnection(Connstring);
myCommand = new SqlCommand(mySelectQuery, myConnection);
myConnection.Open();
myReader = myCommand.ExecuteReader();
int Curhs = DetailStartLine;
while (myReader.Read())
{ mySheet.Cells[Curhs, 1] = myReader["qy"].ToString();
mySheet.Cells[Curhs, 2] = myReader["zb"].ToString();
mySheet.Cells[Curhs, 3] = myReader["dh"].ToString();
Curhs++;
} //数据库表的数据信息导入 Excel 中, 形成 Excel 文档
myReader.Close();
MergeCell(ref mySheet, 3, recount, "A");
//调用函数实现 A 列合并
MergeCell(ref mySheet, 3, recount, "B");
//调用函数实现 B 列合并
//下面为保存生成报表的过程, 适合于 Office2000
myBook.SaveAs(reqpath + "\\111.xls", oMissiong, oMissiong, oMissiong, oMissiong, oMissiong,
Excel.XlSaveAsAccessMode.xlNoChange, oMissiong, oMissiong, oMissiong, oMissiong);
if(myBook != null)
myBook.Close(true, reqpath + "\\111.xls", true);
if(mySheet != null)
System.Runtime.InteropServices.Marshal.ReleaseComObject(mySheet);
mySheet = null;
if(myBook != null)
System.Runtime.InteropServices.Marshal.ReleaseComObject(myBook);

```

```

myBook = null;
if(ExlApp != null)
{ ExlApp.Quit();
  System.Runtime.InteropServices.Marshal.ReleaseComObject ((object)ExlApp);
ExlApp = null;
}
GC.Collect();

```

三、结论

通过运行程序，单击“打印”按钮，就可以做出如表 1-4 所示的报表。

表 1-4

数据信息管理表 (2005. 6. 26)		
区域名称	组别	队号
重庆	重庆 1 组	323 队
		837 队
		400 队
兰州	兰州 1 组	321 队
		478 队
	兰州 3 组	789 队
		324 队
新疆	新疆 1 组	475 队
		479 队
	新疆 2 组	322 队
	新疆 3 组	354 队
		355 队

从上面生成的报表可以看出，在 C# 中通过函数来实现报表纵向合并是可行的。

(谢军 侯康柱 郑雷清)

例题 2 在 Visual C# 中运用 API 函数编写多功能关机程序

熟悉 Windows 的用户对操作系统的关机程序一定很了解，你是不是觉得它的功能不是很多。许多软件都有自动关机功能，例如一些下载软件，这一功能能让你半夜踏踏实实地睡个好觉，而电脑却能按照你事先的设定自动关闭。你在使用电脑听音乐、看电影或下载一些资料时，是不是经常为忘记关机而心痛不已。现在用 Visual C# 来编写一个多功能的关机程序。该程序具有定时关机、倒计时关机、关机提醒、系统信息获取等 4 项功能，可设定关机时间精确到秒，并且让你很快掌握 Visual C# 中对 API 的操作程序。编写过程如下：

一、设计关闭 Windows 窗体

1. 界面的设计

新建一个标准工程，向工程中增加一个 Windows 窗体并向窗体中添加如表 2-1 所示的控件，并分别设置其属性。

表 2-1

控件名	类别	Text	控件名	类型	Text
CheckBox1	CheckBox	自动关机	GroupBox1	GroupBox	当前系统时间
CheckBox2	CheckBox	倒计时执行操作	GroupBox2	GroupBox	设定时间
CheckBox3	CheckBox	定时报警	TxtTime	TextBox	
ButCancel	Button	取消	SetupTime	DateTimePicker	
ButReOpen	Button	重新启动	SetupDate	DateTimePicker	
ButClose	Button	关机	Timer1	Timer	100
ButSysInfo	Button	系统信息	ButReLogin	Button	注销

Windows 窗体界面如图 2-1 所示。

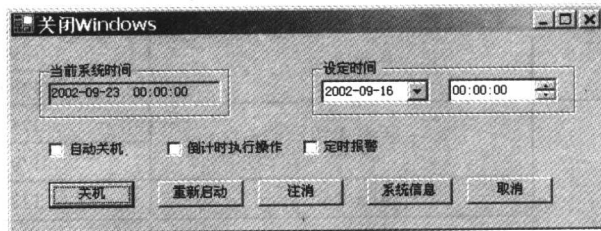


图 2-1

将窗体属性中的 Caption 设置为“关闭 Windows”，名称设置为“frmmain”。

2. 在窗体类中引用 API 函数

API 函数是构筑 Windows 应用程序的基石，是 Windows 编程的必备利器。每一种 Windows 应用程序开发工具都提供了间接或直接调用 Windows API 函数的方法，或者是调用 Windows API 函数的接口，也就是说具备调用动态链接库的能力。Visual C# 和其他开发工具一样也能够调用动态链接库的 API 函数。

在 Visual C# 中调用 API 的基本过程如下：

首先，在调用 API 之前，必须先导入 System. Runtime. InteropServices 命名空间。该命名空间包含了在 Visual C# 中调用 API 的一些必要集合，具体的方法如下：

```
using System. Runtime. InteropServices ;
using System. Text ;
```

在导入了命名空间后，要声明在程序中所要用到的 API 函数。程序主要是获取系统的相关信息，所以用到的 API 函数都是返回系统信息的。先给出在 Visual C# 中声明 API 的方法：

```
[DllImport("user32")]
public static extern long SetWindowPos(long hwnd, long hWndInsertAfter, long X, long y, long cx, long cy, long wFlags);
```

其中，DllImport 属性用来从不可控代码中调用一个方法，它指定了 DLL 的位置，该 DLL 中包含调