

万水编程实例五十讲丛书

突破 C#

编程实例五十讲

刘 杨 等编著

 中国水利水电出版社
www.waterpub.com.cn

内 容 提 要

本书全面介绍了微软新推出的软件 C# 的语言基础和编程技巧。

全书分为 11 篇，共 53 讲。主要内容包括：Windows 窗体相关实例；使用 ADO.NET 访问数据库的相关实例；网络应用程序开发；使用 C# 处理图形图像的实例；ASP.NET 经典实例；多线程的相关实例；XML 在 C# 程序中的应用；系统安全性的实例等。

本书是按照实例的归类组织的，每一篇或实例都与前面没有太大关系，读者完全可以挑选自己感兴趣的部分以跳跃的方式阅读。在每讲后面的小结中，笔者都会对本实例进行一下总结。

全书内容翔实，结构合理，实例精彩而且丰富，技术剖析深入浅出，是广大计算机用户学习 C# 必备的入门参考书。

图书在版编目 (CIP) 数据

突破 C# 编程实例五十讲 / 刘杨等编著. — 北京: 中国水利水电出版社, 2002
(万水编程实例五十讲丛书)

ISBN 7-5084-1187-0

I. 突… II. 刘… III. C 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2002) 第 063920 号

书 名	突破 C# 编程实例五十讲
作 者	刘杨 等编著
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@public3.bta.net.cn (万水) sale@waterpub.com.cn
经 售	电话: (010) 68359286 (万水)、63202266 (总机)、68331835 (发行部) 全国各地新华书店
排 版	北京万水电子信息有限公司
印 刷	北京市天竺颖华印刷厂
规 格	787×1000 毫米 16 开本 25.5 印张 565 千字
版 次	2002 年 9 月第一版 2002 年 9 月北京第一次印刷
印 数	0001—5000 册
定 价	36.00 元

凡购买我社图书，如有缺页、倒页、脱页的，本社发行部负责调换

版权所有·侵权必究

《万水编程实例五十讲丛书》

编委会

主 编 万 博 王 龙 韩中领

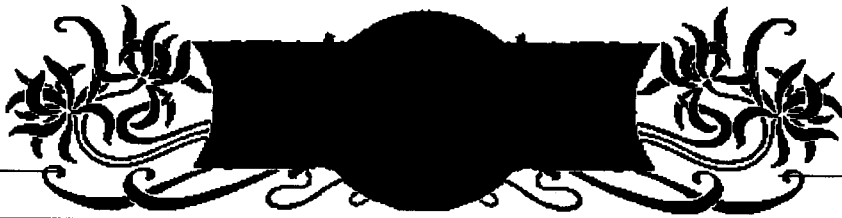
编 委 苏 瑞 索双有 袁 博 陈海亮 黎 骅 康通博

陈 飞 郭燕婷 关 宁 张美生 文 锋 姜俊杰

杨现青 胡晓冰 商 铮 特 铮 赵庆元 姜仁武

王 静 陈心怡 陈江龙 谢小卓 陈志华 顾卫华

吴志华 王志伟 邹振亚 刘 杨



一个优秀程序员和一个普通程序员的区别在什么地方？有一个答案是这样的：普通程序员花了一个星期，苦思冥想如何实现程序功能和排除 Bug。而优秀程序员只用了一天就做完了，因为他给已经做过类似东西的朋友打了个电话。

这段话多半是玩笑，但是也说明了交流的重要。很多读者在学习编程的过程中，早已摆脱了教条式的入门讲解。他们需要的是前人的经验和总结，这就需要交流，看更多的实例，学更多的实用技巧。

如何把前人摸索出来的经验和套路传授给急需它们的读者，如何使计算机的入门教育变成更具有艺术性的教育，是我们一直在寻求的突破口。为此，我们策划并编写了“编程实例五十讲”丛书，从写作到排版风格，从整体结构到每个小例子，我们都做了精心的安排。

在本丛书的编写过程中贯穿了如下的原则：

- (1) 由浅入深、由易到难。整本书结构清晰，适用于初级和中级读者。
- (2) 内容实用、可操作性强。书中对于实例程序行和操作过程的描述要求清楚透彻、简明扼要，编写风格则要求“功能性强”、“示范性强”。
- (3) 知识讲解与动手操作并行。每个实例都突出讲解了某些知识点和使用技巧。知识点与实例的结合是本书的一大特点。
- (4) 简单常用的基础实例与综合性较强的应用实例相结合。每本书中都讲解了一些经常会碰到的应用实例，给读者提供思路上的参考。
- (5) 不做简单的功能介绍。本丛书是侧重于应用的图书，我们假定丛书的读者对软件已有一定的了解，或是计算机基础较好，需要快速地在实践中掌握软件的应用。

愿本丛书能带领读者进入程序设计的广阔天地。



编者

2002年4月

我们寻求突破

前 言

欢迎使用本书。如果您选择本书是为了由 C# 的初学者迅速进阶，编写成熟的 .NET 应用程序，那么您真是选对了。在本书 53 讲的 C# 学习之旅中，您将学到 C# 各个重要方面的知识，并轻松应用于实践。

一本什么样的书

胡适先生说：“发表是最好的记忆。”在此这句话不妨套用为：“使用是最好的学习”。没有人可以通过听讲而真正掌握一门技术，实际应用是最好的导师。本书的目的，就是要带领您制作大批较为完整的实例，使您在应用的过程中，体会 C# 程序编写的方法；进而迅速由初学者进入只需查 MSDN 就可以轻松编写实用 C# 程序的阶段。

主要内容

本书分为 11 篇，共 53 讲。

第一篇是与 Windows 窗体相关的实例，此部分的实例多数都很基本，但是也有个别实例运用了较为复杂的技术。

第二篇是使用 ADO.NET 访问数据库的相关实例。

第三篇进入网络应用程序开发，其中的实例“聊天室”是一个经典的分布式程序，应给予相当重视。

第四篇为使用 C# 处理图形图像的实例，大部分的实例应用了 .NET 的新技术 GDI+，它比以前的版本更加简单易用。

第五篇是一些 ASP.NET 的经典例子，例如论坛和在线购物。

第六篇介绍了两个移动电话上网的例子。

第七篇是多线程的相关实例，只需理解其中主要的思想，就会发现其实很简单。

第八篇介绍了 XML 在 C# 程序中的应用。

第九篇是关于文件和文件夹操作的实例。

第十篇是有关系统安全性的实例。

第十一篇是其他无法归为上述 10 部分的实例。

本书是按照实例的归类组织的，每一篇或实例都与前面没有太大关系，读者完全可以挑选自己感兴趣的部分以跳跃的方式阅读。在每讲后面的小结中，笔者都会对本实例进行一下总结。

适合什么样的人

本书适合各种层面的读者。要完全理解本书设置的实例，读者必须具备一些 C#编程常识。

作为一本讲解实例的书籍，本书的重点始终放在实际问题的解决上。虽然在本书的每一讲中都会对本讲涉及到的知识进行详细的介绍，但是笔者建议读者在阅读本书之前，先读一本 C#入门的书，这样在读者体会实例中技术的应用时，就不会感到太吃力。

需要安装的软件

为完成书中的实例，需要读者使用 Windows 2000 (Server 或 Professional 均可) 或者安装有 Service Pack 6 的 Windows NT 操作系统。另外，需要安装 Visual Studio.NET 正式版。由于本书的所有实例都是在正式版下开发的，因此如果读者使用的是测试版，则可能存在问题。

读者若要实现第五篇“Web 窗体技术”中的实例，还需要安装一个 Web 服务器。可以是 IIS5.0 或者 Personal Web Server，但是必须在 Visual Studio.NET 之前安装，否则在发布 ASPX 文件时，会出现问题。

第六篇“手机上网程序设计”中的实例可以直接在 IE 中浏览，如果读者想要更加真实的效果，可以安装各种 Mobile 模拟器。

最后，为了使用书中与数据库相关的范例，还需要安装某种 OLE DB 兼容的数据库系统，如微软的 Access 或 SQL Server。

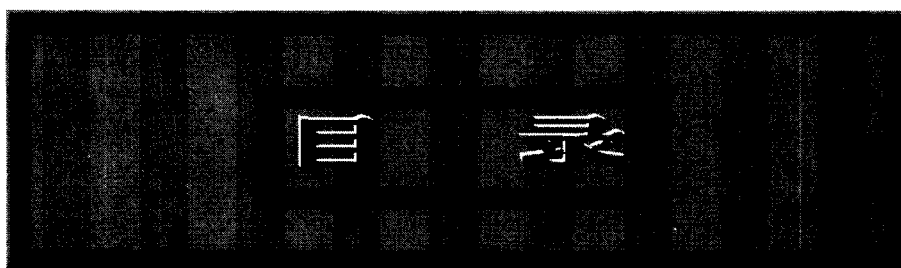
结束语

在 BBS 上与其他人交流也是学习 C#的一条捷径，国内较为成熟的.NET 论坛有水木清华的 DOTNET 版等，国外则有微软的.NET 站点<http://www.microsoft.com/net/> 和 C# Corner 网站<http://c-sharpcorner.com> 比较专业。

最后，祝你在 C#学习的这一段时间中，一帆风顺！

编者

2002 年 8 月



丛书序
前言

第一篇 Windows 窗体技术

第一讲	开发你的第一个 MDI 应用程序.....	2
第二讲	转轮控件	7
第三讲	注册表浏览器	19
第四讲	使用菜单	28
第五讲	进程跟踪器	38
第六讲	消息队列的简单实例.....	42
第七讲	进度条、状态栏和计时器的实例.....	48
第八讲	把 Excel 数据读入 ListView.....	52
第九讲	打开文件夹对话框.....	56
第十讲	打印	61
第十一讲	浮动的窗体	68
第十二讲	计算器实例程序.....	69
第十三讲	实现 C#到 VB.NET 源代码的转换.....	75
第十四讲	使远程主机重新启动.....	78
第十五讲	给多行文本加行号.....	83

第二篇 数据库程序设计

第十六讲	访问 Oracle 数据库.....	86
第十七讲	在 SQL Server 中存取图片.....	89

第三篇 网络程序设计

第十八讲	文件下载器	94
第十九讲	远程处理入门.....	99
第二十讲	端口扫描器	105

第二十一讲	聊天室	111
第二十二讲	使用 C#编写 Ping 程序.....	123
第二十三讲	Email 提醒器.....	136
第二十四讲	Web 服务器	148
第二十五讲	浏览器	163
第二十六讲	发送 Email.....	166

第四篇 图形图像程序设计

第二十七讲	时钟	172
第二十八讲	玩转三维空间.....	188
第二十九讲	绘制液晶显示的数字.....	205
第三十讲	使用拖拽	209

第五篇 Web 窗体技术

第三十一讲	登录用户控件.....	216
第三十二讲	C#+HTML 打造服务器端 Web 应用程序	223
第三十三讲	使用缓存的实例.....	225
第三十四讲	在线投票.....	232
第三十五讲	在线论坛.....	264
第三十六讲	HTTP 处理程序.....	288
第三十七讲	使用 Application Center Test.....	291

第六篇 手机上网程序设计

第三十八讲	开发一个简单的 Mobile 应用程序	296
第三十九讲	手机跟踪股票信息.....	305

第七篇 高级论题

第四十讲	一个多线程的简单实例.....	313
第四十一讲	播放声音.....	316
第四十二讲	使用资源文件.....	318
第四十三讲	关于反射的例子.....	322

第八篇 XML 程序设计

第四十四讲	使用 XmlTextWriter 写 XML 文件	327
第四十五讲	使用 XmlDocument.....	332

第四十六讲 XML 注释.....	339
-------------------	-----

第九篇 文件系统程序设计

第四十七讲 文件搜索器.....	344
------------------	-----

第四十八讲 文件分块器.....	349
------------------	-----

第十篇 系统安全性

第四十九讲 聊天室扩展之一——Web 服务提供登录校检.....	356
----------------------------------	-----

第五十讲 关注 ASP.NET 的安全性.....	362
---------------------------	-----

第十一篇 综合实例

第五十一讲 使用文件系统监视器.....	371
----------------------	-----

第五十二讲 聊天室扩展之二——使用 Windows 服务.....	378
-----------------------------------	-----

第五十三讲 聊天室扩展之三——使用 MS Agent.....	389
---------------------------------	-----

第一篇

Windows窗体技术

本篇导读

Windows 窗体技术主要是指 System.Windows.Forms 名称空间中的各类的运用。在此名称空间中包含所有的 Windows 窗体控件类，使用它们可以设计美观、实用的用户界面 (UI)。

然而光有用户界面并不能称为一个应用程序。在本篇的实例中，笔者尽力将一些其他技术与窗体技术整合在一起，使得实例本身可以完成一些简单的功能。

这些“其他技术”包括 MDI 程序的编写、用户控件的制作、访问注册表、控制进程、使用消息队列、操作 Excel 文件、使用 Web 服务和使用 WMI 技术等。

通过本篇的学习，希望读者能够对 .NET 的面向组件的程序设计思想有初步领会。

第一讲 开发你的第一个 MDI 应用程序

【实例说明】

本讲实例将讨论多文档界面应用程序（Multiple Document Interface ——MDI applications）的开发。MDI 应用程序可以同时程序中呈现多个文档，每个文档显示在各自的独立窗口中。在 C# 中，提供了为实现 MDI 程序设计的很多功能。本文就是通过一个具体例子来详细介绍在 Visual C# 中的 MDI 编程。


【编程思路】

多文档界面就是所谓的 MDI，是从 Windows 2.0 下的 Microsoft Excel 电子表格程序开始引入的，这是因为 Excel 电子表格用户有时需要同时操作多份表格，MDI 正好为这种操作多表格提供了很大的方便。在 Windows 3.1 版本中，MDI 得到了更大范围的应用。其中系统中的程序管理器和文件管理器都是 MDI 程序。

MDI 编程主要就是要在主窗体中能够新建一个 MDI 窗体，并且能够对主窗体中的所有 MDI 窗体实现层叠、水平平铺和垂直平铺。虽然这些操作比较基本，但却是程序设计中的要点和重点。下面讲解设计 MDI 程序的步骤。

【步骤】

1. 生成 MDI 父窗体（PARENT FORMS）

说明：多文档用户界面（MDI）的基础是 MDI 父窗体。MDI 父窗体是所有 MDI 子窗体的容器（CONTAINER），每个 MDI 应用程序只有一个父窗体，而根据需要可以有一个到多个子窗体。运行时若父窗体关闭或最小化，则其所有的子窗体也都会相应的关闭或最小化。

生成一个 MDI 父窗体，只需向项目中添加一个窗体（FORM）然后加入下列代码即可：

```
this.IsMDIContainer = true;
```

这行代码指定当前窗口作为子窗口的容器。

2. 生成 MDI 子窗体（MDI CHILD FORMS）

在 MDI 应用程序中起实质作用的是 MDI 子窗体，要生成 MDI 子窗体，请在你的程序中输入如下代码：

```
Form frmchild=new Form();  
frmchild.MDIParent=this;  
frmchild.Show();
```

3. 确定激活的 MDI 子窗体

在 MDI 应用程序众多的子窗体中，很多时候我们需要控制正在与用户进行交互的那个窗体，也就是激活的子窗体。因此，微软的工程师为我们提供了 `ActiveForm` 属性，当它的值为 `true` 时，表示当前窗体已被激活。

任何一个子窗体，只要已经被加载 (`loaded`) 且可见 (`visible` 属性为 `true`)，都可以通过设置 `ActiveForm` 属性使其成为激活窗体。相反，如果访问 `ActiveForm` 属性之前上述两个条件不全具备，就会产生一个异常。

4. 子窗体的排布方式

也许读者可能会注意到，很多 MDI 应用程序的菜单中有平铺 (`Tile`)、层叠 (`Cascade`) 和排布 (`Arrange`) 等命令。同样，在 C# 中也可以通过调用父窗体的 `LayoutMDI` 方法，并设置其为 `MDILayout` 枚举类型的成员的方法，来重新排布子窗体。

【步骤】

在下面的实例中，我们将会演示一个简单的 MDI 应用程序的开发过程。

1. 新建项目

选中【文件】|【新建】|【项目】，在新建项目对话框中选择项目类型为“Visual C#项目”、模板为“Windows 应用程序”，然后修改项目名称为“MDIApplication”。将默认生成的 `Form1.cs` 文件改名为“`MDI.cs`”，文件中其他出现 `Form1` 的地方，也一律改为“`MDI`”。

2. 添加代码

在代码窗口中键入如下代码：

```
using System;
using System.Drawing;
using System.Collections;
using System.ComponentModel;
using System.Windows.Forms;
using System.Data;

namespace MDIApplication
{
    public class MDI: System.Windows.Forms.Form
    {
        private MainMenu mainMenu;        //主菜单控件
        private int Count=0;              //子窗体数目
        private System.ComponentModel.Container components = null;
```

```
public MDI()
{
    InitializeComponent();

    this.IsMdiContainer=true;    //第一步,指定当前窗口作为子窗口的容器
    this.Text="Mdi Demo";
    //设置父窗体的标题栏为"Mdi Demo"(MDI 演示)
    mainMenu = new MainMenu();
    //创建主菜单的实例
    MenuItem File = mainMenu.MenuItems.Add("&File");
    //"File"菜单
    File.MenuItems.Add(new MenuItem("&New",new EventHandler (this.
FileNew_clicked),Shortcut.CtrlN));
    //"File"菜单中填入 New 命令
    File.MenuItems.Add(new MenuItem("&Active Child",new EventHandler
(this.FindActive_clicked),Shortcut.CtrlA));
    //"File"菜单中填入 Active Child 命令
    File.MenuItems.Add(new MenuItem("-"));
    //"File"菜单中填入分隔线
    File.MenuItems.Add(new MenuItem("&Exit",new EventHandler (this.
FileExit_clicked),Shortcut.CtrlX));
    //"File"菜单中填入 Exit 命令
    MenuItem Arrange = mainMenu.MenuItems.Add("&Arrange");
    //"Arrange"菜单
    Arrange.MenuItems.Add(new MenuItem("&Cascade",new EventHandler
(this.Cascade_clicked),Shortcut.F1));
    //"Arrange"菜单中填入 Cascade(层叠)命令
    Arrange.MenuItems.Add(new MenuItem("&Horizontal",new
EventHandler (this.Horizontal_clicked),Shortcut.F2));
    //"Arrange"菜单中填入 Horizontal(水平平铺)命令
    Arrange.MenuItems.Add(new MenuItem("&Vertical",new EventHandler
(this.Vertical_clicked),Shortcut.F3));
    //"Arrange"菜单中填入 Vertical(垂直平铺)命令
    this.Menu=mainMenu;
    //将刚刚设计好的主菜单加入 MDI 窗体
    mainMenu.GetForm().BackColor = Color.Indigo ;    //背景色
}
//程序的入口点
static void Main()
{
    Application.Run(new MDI());
}
//点击"Exit"命令时,程序关闭
```

```
private void FileExit_clicked(object sender, EventArgs e)
{
    this.Close();
}
//当点击"激活"的子窗体时
private void FindActive_clicked(object sender, EventArgs e)
{
    MessageBox.Show(this.ActiveMdiChild.Text, "Mdi FORM");
}
//点击"File" (文件) 菜单中的"New" (新建) 命令时
private void FileNew_clicked(object sender, EventArgs e)
{
    Form frmchild=new Form(); //第二步, 生成 MDI 子窗体对象
    frmchild.MdiParent=this; //子窗体的父窗体的当前窗体
    frmchild.Show(); //显示子窗体
    frmchild.Text="Child Form" + Count.ToString(); //子窗体的标题栏
    Count++; //子窗体的数目加 1
}

private void pop_Clicked(object sender, EventArgs e)
{
    MessageBox.Show("Popupmenu", "MENU_CREATION");
}
//点击 Cascade (层叠) 命令时
private void Cascade_clicked(object sender, EventArgs e)
{
    this.LayoutMdi(MdiLayout.Cascade );
}
//点击 Horizontal (水平平铺) 命令时
private void Horizontal_clicked(object sender, EventArgs e)
{
    this.LayoutMdi(MdiLayout.TileHorizontal);
}
//点击 Vertical (垂直平铺) 命令时
private void Vertical_clicked(object sender, EventArgs e)
{
    this.LayoutMdi(MdiLayout.TileVertical);
}
}
}
```

[[实例效果]]

编译执行, 程序显示效果如图 1-1 和图 1-2 所示。

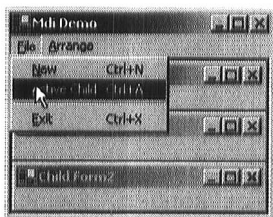


图 1-1 水平平铺

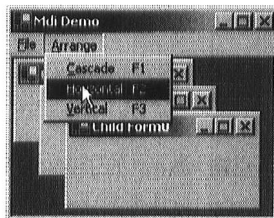


图 1-2 层叠



说明：上面的插图中标题栏颜色为蓝色的子窗体就是“激活”的窗体。

【小结】

本文主要介绍如何用 C#来实现对 MDI 窗体的编程。即通过以下步骤可以轻松开发 MDI 多文档界面程序。

- (1) 生成 MDI 父窗体。
- (2) 生成 MDI 子窗体。
- (3) 设置激活的子窗体。
- (4) 设置子窗体的排布方式（层叠、水平平铺和垂直平铺）。

第二讲 转轮控件

【实例说明】

本讲的内容是制作一个像图 2-1 金山影霸一样的转轮控件。

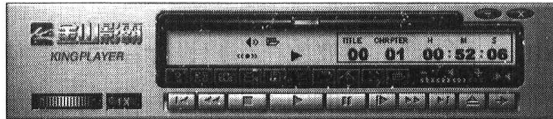



图 2-1 金山影霸的转轮控件

【编程思路】

用户控件提供了一种重复使用用户自己制作的图形界面的方法。它实际上是一个具有可视化表示形式的组件。我们可以用与放置其他控件 (如文本框和标签) 相同的方式将自定义的用户控件放置在 Windows 窗体上。在本讲的第一部分, 我们一起来创建一个名为 WheelCtrl 的转轮控件。在本讲的第二部分通过把此控件放置在 test 项目的窗体上来测试其功能。

【步骤】

1. 新建项目

选择【文件】|【新建】|【项目】以打开“新建项目”对话框, 如图 2-2 所示。从“C# 项目”列表中选择“Windows 控件库”项目模板, 然后在“名称”框中键入“WheelControl”, 点击【确定】按钮。可以看到在解决方案资源管理器中出现了一个用户控件文件“UserControl1.cs” (如图 2-3 所示), 图标  表示此文件中定义类是一个用户控件, 它继承自父类“System.Windows.Forms.UserControl”。

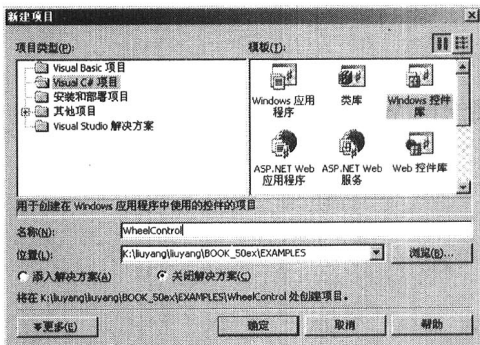


图 2-2 新建项目对话框



图 2-3 解决方案资源管理器



说明: UserControl 类提供所有用户控件所需的功能, 并实现标准方法和属性。

“用户控件设计器”如图 2-4 所示, 作为用户控件的设计图面, 用户控件设计器允许您使用与设计窗体相同的模型。例如, 当将按钮添加到窗体时, 会看到表示该元素的图标, 并且可以单击它以访问其属性, 还可以移动到代码编辑器。对于用户控件设计器来说同样如此: 可以通过与设计图面进行交互来添加和配置可视及非可视的元素, 并且可以对这些元素及其事件进行编程。

在将可视元素添加到用户控件时, 它被放在用户设计图面上, 使您可以调整该元素的可视表示形式和位置。在将非可视项 (如 DirectorySearcher 组件) 添加到用户控件时, 该项出现在设计器底部的组件栏中。

将文件 UserControl1.cs 改名为 “WheelCtrl.cs”, 将文件中所有出现 UserControl1 的地方都用 WheelCtrl 替换。用鼠标左键点击用户控件窗体中的任何位置, 在属性窗口中设置其 Size 属性为 “56, 8”。

为了显示转轮控件的滚动效果, 需要一组位图, 我们可以右击解决方案资源管理器的 “WheelControl” 项目, 在弹出的菜单中选择 “添加” | “添加现有项”, 选择项目文件夹中的 wheelh.bmp 文件将其添加到解决方案之中。图 2-5 为 wheelh.bmp 文件存储的图片, 它有 9 个不同转动位置的转轮构成。

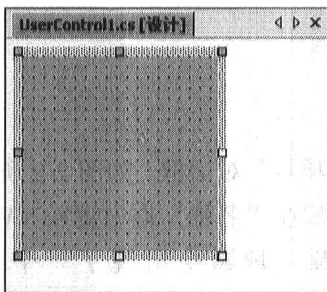


图 2-4 用户控件设计器

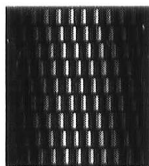


图 2-5 wheelh.bmp 文件存储的图片

2. 声明命名空间 WheelCtrl

下面就可以进入代码的编写, 首先是声明命名空间 WheelCtrl, 以便日后引用。

```
1 namespace WheelCtrl
{
3     /// <summary>
4     /// Summary EventHandlerArgs
5     /// </summary>
6     public class WheelEventHandlerArgs : EventArgs
7     {
8         private int nX;
```