



电脑伴我行

计算机应用宝典

Visual C# 7.0 编程不求人



香港恒明出版有限公司授权
广州网垠科技发展公司改编



珠海出版社

计算机应用宝典

VC#7.0 编程不求人

香港恒明出版有限公司授权

广州网垠科技发展公司改编



珠海出版社

图书在版编目(CIP)数据

电脑伴我行·网坝 编著. — 珠海: 珠海出版社,
2002. 1

ISBN7 - 80607 - 819 - 3

I. 电… II. 网… III. 电子计算机 - 基本知识 IV. TP8

作 者 ■ 网 坝

责任编辑 ■ 雷良波

出版发行 ■ 珠海出版社

社 址 ■ 珠海香洲银桦新村 47 栋 A 座二层

电 话 ■ 2515348 邮政编码 ■ 519001

印 刷 ■ 广东科普印刷厂

开 本 ■ 850/1168mm 1/64

印 张 ■ 234 字数 ■ 5180 千字

版 次 ■ 2002 年 1 月第 1 版

2002 年 1 月第 1 次印刷

印 数 ■ 1 - 5000 册

书 号 ■ ISBN7 - 80607 - 819 - 3/TP. 8

定 价 ■ 全 36 册印刷版总定价: 360.00 元

带光盘版总定价 460.80 元

版权所有·翻印必究

前 言

一次次逛电脑城、书店、报刊亭，总感觉广大水平越来越高的电脑爱好者，能够惊喜、能够酷毙、能够帅呆了、能够巴不得马上拥有的东西越来越少了。

这批书盘的面市或许会给广大电脑高手一种全新的感受。

首先是新：《Visual Basic7.0 编程不求人》、《Visual Foxpro7.0 数据库不求人》、《Visual C#7.0 编程不求人》、《Delphi 6.0 编程不求人》、《SQL 数据库不求人》、《AutoCAD2000 辅助设计不求人》、《Windows NT 网络技术不求人》、《CoreIDRAW10 广告设计不求人》、《3DS MAX4.0 三维动画不求人》、《MAYA4.0 影视动画不求人》、《Word 书刊排版不求人》、《Office XP 办公应用不求人》这 12 本书，包含当今最流行的软件，全面学习，融会贯通。

其次是实。全书以一个个案例为主线，现场直播、全程体验。基础知识，一点就明。附录收集的英汉速查、快捷键、应用技巧等，实用而解恨。

再其次是便利。一本书、一张光碟，拿在手上，揣在口袋里，随学随用，随用随查，轻松极了。

还有一个最大的特点，就是便宜。每套一书一碟，内容价值上千元，才卖 12.80 元。纯粹正版，印制精美，比用翻版碟还划算。

不再啰嗦了。这么好的好东东，你要是不买，才怪呢！

目 录

第一章 应用程序案例新鲜出炉	1
1.1 标准窗体属性程序举例	1
1.2 标准窗体方法程序举例	6
1.3 菜单应用程序举例	23
1.3.1 绘图板程序设计	24
1.3.2 分析程序	82
1.4 对话框应用程序举例	83
1.4.1 按钮控件应用程序举例	97
1.4.2 列表框、组合框与文本框综合应用程序举 例	106
1.4.3 列表框、组合框与文本框应用程序举例	115
1.4.4 进程控件综合应用程序举例	125
1.4.5 时间控件应用程序举例	132
1.4.6 创建工具栏应用举例	136
1.4.7 创建状态栏应用举例	141
1.4.8 进度条控件应用程序举例	145

1.5	VISUAL C#.NET 7.0 综合应用程序举例	148
第二章	认识 VISUAL C#.NET	166
2.1	VISUAL C#.NET 7.0 的发展	167
2.1.1	对网络窗体和网络服务支持	168
2.1.2	对数据库发展的支持	171
2.1.3	对编程语言发展的支持	172
2.1.4	对工程调试发展的支持	175
2.1.5	对创建组件和控件的支持	176
2.1.6	Windows 窗体	177
2.2	如何安装 VISUAL STUDIO.NET 7.0	178
2.2.1	安装 Visual Studio.net 7.0 的系统要求	178
2.2.2	安装 Visual Studio.net 7.0	179
2.2.3	删除 Visual Studio.net 7.0	199
2.3	VISUAL STUDIO.NET 7.0 的启动与退出	200
2.3.1	Visual Studio.net 7.0 的启动	200
2.3.2	Visual Studio.net 7.0 的退出	201
2.4	如何使用帮助系统	201
2.4.1	MSDN For Visual Studio.net 7.0 查阅器的使用	202

2.4.2 利用 Visual Studio.net 7.0 联机链接方式	205
---	-----

第三章 VISUAL C#.NET 集成开发环境(IDE)207

3.1 集成开发环境的窗口组成	207
3.1.1 主窗口的组成	209
3.1.2 窗体(Form)的使用	219
3.1.3 代码窗口	221
3.1.4 工具箱(Toolbox) 窗口	223
3.1.5 属性窗口的使用	228
3.1.6 工程资源浏览器窗口 (Solution Explorer)	231
3.1.7 对象浏览器窗口(Object Browser)的使用	236
3.1.8 输出窗口(Output Window)	241
3.1.9 其他窗口简介	242
3.2 自定义工作环境	243
3.2.1 工具栏的自定义	243
3.2.2 设置自己的代码编辑器	252
3.3 菜单系统的使用	262

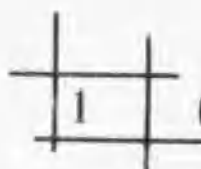
3.3.1 文件菜单项(File)	262
3.3.2 编辑菜单项(Edit)	266

第四章 C#编程设计基础269

4.1 C#语言有关的概念.....	269
4.1.1 声明的作用.....	269
4.1.2 成员的种类.....	275
4.1.3 成员的访问.....	277
4.1.4 签名与重载	287
4.1.5 作用域的规则	289
4.1.6 名字屏蔽的组成	294
4.1.7 名字空间和类型名的指定	298
4.2 数据类型的使用.....	300
4.2.1 值类型的种类.....	301
4.2.2 引用类型的种类	312
4.2.3 加框转换与消框转换.....	313
4.3 变量的使用	317
4.3.1 各种变量类型的定义.....	317
4.3.2 缺省值的使用	321
4.3.3 定义赋值的的使用.....	321

4.3.4	如何引用变量	326
4.4	类型的转换	327
4.4.1	隐式转换.....	327
4.4.2	显式转换.....	331
4.4.3	标准转换.....	336
4.4.4	自定义转换.....	336
4.5	用户定义的数据类型	338
4.5.1	结构的定义.....	338
4.5.2	数组的定义.....	353
4.5.3	枚举的定义.....	364
4.6	程序流控制语句.....	370
4.6.1	终点及其可达性	371
4.6.2	block 语句	374
4.6.3	空语句的使用	375
4.6.4	标签语句的使用	376
4.6.5	声明语句的使用	377
4.6.6	表达式语句.....	380
4.6.7	选择语句.....	381
4.6.8	循环语句.....	394

4.6.9 跳转语句.....	403
4.6.10 try 语句.....	411
4.6.11 溢出检查语句.....	415
4.6.12 锁定语句.....	416
光盘内容指南	418
1.1 技巧与教程	418
VC 编程技巧集锦.....	418
VC ++ 编程技巧.....	418
VC 编程技巧.....	419
1. 2 源码.....	419
数据库.....	419
DLL.....	421
文件	423
网络	427
1.3 控件.....	430
文件	430
图像	431
多媒体.....	437



第 1 章 应用程序案例新鲜出炉

1.1 标准窗体属性程序举例

为了帮助读者理解和使用标准窗体编程技术,下面给出了一个综合使用窗体的属性进行编程的实例。

在这个例子中,当用户按 ESC 键时,会执行默认的退出操作。用户在这里不必用鼠标单击退出键也可以关闭当前窗口。而且该操作并不影响窗体中的数据控制。

创建程序的步骤:

- 用鼠标单击文件(File)菜单下的新建(New)菜单从中选择新建工程(New Project)选项
- 在出现的窗体设计器中设置其 Text 属性和其它窗体的属性
- 用鼠标单击视图(View)菜单下的工具箱(ToolBox)菜单,在其中选择按钮(Button)控件,并在窗体编辑其中用鼠标拖出两个按钮控件

不求人

- 打开属性窗口对两个按钮的属性进行编辑
- 用鼠标双击两个按钮控件在弹出的代码编辑器中加入代码

入代码

- 从 Build 菜单中选择 Build

本程序创建了一个窗体实例。利用了 ShowDialog 方法显示一个对话框窗体这个例子使用 BorderStyle、AcceptButton、CancelButton、MaximizeBox、MaximizeBox、StartPosition 来改变对话框窗体的外观和功能。这个程序使用 Add 方法向窗体控件集合中添加了两个按钮控件。

程序运行后，在计算机屏幕中央显示一个固定边框的对话框窗体，使用 VC#编程技术在窗体上生成一个接收按钮和一个取消按钮。运行此程序后，相信读者会对 VC#窗体的新属性有了更深的理解。

例程代码：

```
public void CreateMyForm()  
{  
    Form form1 = new Form();  
    Button button1 = new Button ();  
    Button button2 = new Button ();  
    button1.Text = "OK";
```

```
button1.Location = new Point (10,
10);
button2.Text = "Cancel";
button1.Location
    = new Point (button1.Left, button1.
Height + button1.Top + 10);
form1.Text = "My Dialog Box";
form1.HelpButton = true;
form1.BorderStyle = FormBorderStyle.
FixedDialog;
form1.MaximizeBox = false;
form1.MinimizeBox = false;
form1.AcceptButton = button1;
form1.CancelButton = button2;
form1.StartPosition =
FormStartPosition.CenterScreen;
form1.Controls.Add(button1);
form1.Controls.Add(button2);
form1.ShowDialog();
}
```

程序运行效果

按 F5 键来运行这个工程，运行效果如图 1.1 所示。

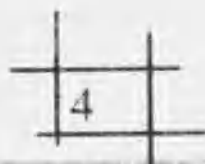


图 1.1 运行效果图

用鼠标右键单击 Cancel 按钮会出现如图 1.2 所示的结果，还原 Form1 设计时的窗体结构。本程序主要使用在用户自定义过程 myprocess 中通过设置窗体的 BorderStyle、AcceptButton、CancelButton、MaximizeBox、MaximizeBox、StartPosition 等属性来设置窗体的外观和功能。

BorderStyle 属性在本程序中被设置为 FormBorderStyle.FixedDialog，即固定对话框的边框样式。

AcceptButton 的属性值为 button1，表示从键盘敲入回车键就相当于使用鼠标左键选中了该按钮，然后执行该按钮的事件过程，由于本程序主要讲述窗体的属性，这里并没有编写处理按钮单击的事件过程，按钮单击的事件过程的处理将留到以后章节进行。

CancelButton 属性的使用同 AcceptButton 属性相似，

只是将回车键换成了 Esc 键，即从键盘敲入 Esc 键就相当于使用鼠标左键选中了该按钮。

MaximizeBox 属性表示是否允许用户使用窗体最大化按钮，该属性值为 True，表示窗体最大化按钮可以使用，即窗体执行时，用户可以使用最大化按钮将窗体最大化，该属性值为 False，窗体执行时，最大化按钮颜色为灰色，表示禁止用户使用窗体最大化按钮。**MaximizeBox** 属性的使用与 **MaximizeBox** 属性类似，只是它表示窗体的最小化按钮是否可用，该属性值为 True，程序执行时，用户可以使用最小化按钮，将窗体最小化。**StartPosition** 属性用来设置窗体执行时开始显示的位置，本例中这个属性值为 **FormStartPosition.CenterScreen**，表示窗体初始显示的位置为屏幕中央。

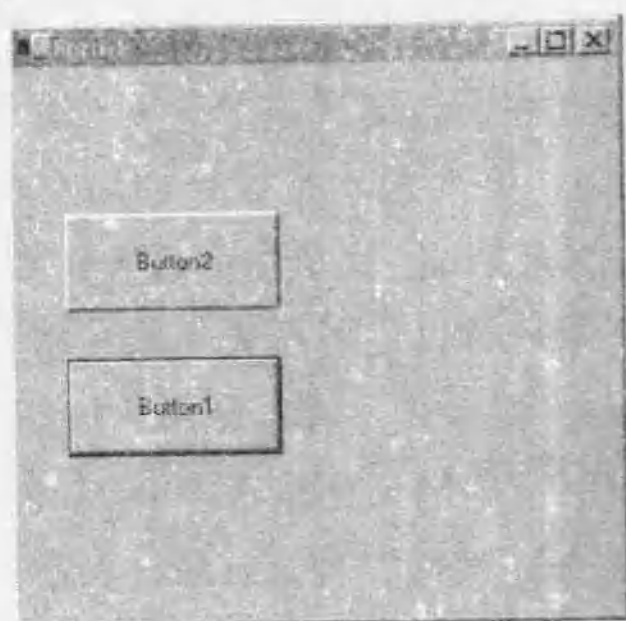


图 1.2 单击 Cancel 按钮的效果图

1.2 标准窗体方法程序举例

为了帮助读者理解和使用标准窗体编程技术,下面给出了一个使用窗体的方法进行编程的实例。在这个例子中,我们通过 Hide, Close, Remove, Add 等方法来隐藏、关闭、删除和添加当前窗体。为了使用不同的窗体方法,我们在窗体中添加了三个按钮。用户要运行此程序,可以点击相应的按钮,在按钮单击事件处理过程中调用窗体的方法,从而实现相应的功能。

具体步骤如下:

- 1.用鼠标选择工具栏的工具箱图标,从出现的工具列表中选择 Mainmenu 控件。
- 2.在菜单设计器中设置菜单项的属性。
- 3.用鼠标双击某一控件,在出现的代码窗口中添加程序代码。

程序代码如下:

```
namespace WindowsApplication36
{
    using System;
    using System.Drawing;
    using System.Collections;
    using System.ComponentModel;
    using System.Windows.Forms;
    using System.Data;
    public class Form1 :
        System.Windows.Forms.Form
    {
        private
        System.ComponentModel.Container
        components;
```