

程序设计实例丛书

Visual Basic

6.0 实用程序 100 例

段 兴 主编
张敬韶 宋 雷 王玉亮 赵 鲁 编写

```
Call Ping(Ips(Indexit), ECHO)  
and my line  
Display the results from the ECHO structure  
turning off showing anything  
prints status code  
DON'T show status  
CPanel.Print GetStatusCode(ECHO.status)  
The address that actually replied  
CPanel.Print ECHO.Address  
Trip time (note: routers tend to lie. The time is only an estimate)  
CPanel.Print ECHO.RoundTripTime & " ms"  
prints data size (as specified)  
CPanel.Print ECHO.DataSize & " bytes"  
  
If Left$(ECHO.Data, 1) <> Chr$(0) Then  
pos = InStr(ECHO.Data, Chr$(0))  
If pos > 0 Then  
If pos > 0 Then print returned Data  
CPanel.Print Left$(ECHO.Data, pos - 1)  
mycode = If success then  
Picture(Indexit).Picture = Good Picture  
Else  
Picture(Indexit).Picture = Bad Picture  
End If  
If Indexit + 1 > totalips Then  
Picture(0).Picture = temp Picture  
Else  
Picture(Indexit + 1).Picture = temp Picture  
End If  
CPanel.Print ECHO.DataForUser  
Checkbox.Enabled = True  
Indexit = Indexit + 1  
Exit Sub  
Private Sub Form_Load()  
Indexit = 0  
Panel.Left = (Screen.Width - CPanel.Width) / 2  
Panel.Top = 20  
Indexit = 0
```

人民邮电出版社
www.pptph.com.cn



附光盘

程序设计实例丛书

Visual Basic 6.0 实用程序 100 例

段 兴 主编

张敬韶 宋雷 王玉亮 赵鲁 编写

人 民 邮 电 出 版 社

图书在版编目 (CIP) 数据

Visual Basic 6.0 实用程序 100 例 / 段兴主编, 张敬韶等编写. —北京: 人民邮电出版社, 2002.1
(程序设计实例丛书)

ISBN 7-115-09842-5

I. V... II. ①段...②张... III. BASIC 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2001) 第 084996 号

内 容 提 要

Visual Basic 6.0 是微软公司主推的可视化编程语言之一, 由于该语言可视化程度高, 因而受到广大用户的青睐。Visual Basic 6.0 是众多编程语言中的主流语言。

本书精心选取了利用 Visual Basic 6.0 开发的、具有代表性的 100 个编程实例, 这些实例包括窗体与界面、图形图像处理、多媒体、网络、数据库、文件与系统等几方面的内容。通过阅读本书, 读者可以了解和掌握 Visual Basic 6.0 常用的使用方法和应用技巧。

本书适用于有一定 Visual Basic 基础的读者阅读, 也可作为培训班的教材和参考用书。

程序设计实例丛书

Visual Basic 6.0 实用程序 100 例

- ◆ 主 编 段 兴
编 写 张敬韶 宋 雷 王玉亮 赵 鲁
责任编辑 邹文波
- ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@pptph.com.cn
网址 <http://www.pptph.com.cn>
读者热线: 010-67129212 010-67129211(传真)
北京汉魂图文设计有限公司制作
北京鸿佳印刷厂印刷
新华书店总店北京发行所经销
- ◆ 开本: 787×1092 1/16
印张: 25.75
字数: 630 千字 2002 年 1 月第 1 版
印数: 1-5 000 册 2002 年 1 月北京第 1 次印刷

ISBN 7-115-09842-5/TP·2591

定价: 39.00 元(附光盘)

本书如有印装质量问题, 请与本社联系 电话: (010)67129223

前 言

Visual Basic 6.0 是微软公司主推的可视化编程语言之一，由于其可视化程度高，因而是最容易入门的编程语言。

Visual Basic 经过微软公司长期不断的完善、第三方资源的大量加入以及广大爱好者的大力支持，Visual Basic 6.0 已经成为功能强大、使用人数众多的主流编程语言。它提供了向导自动生成代码，支持拖放技术、属性检查、实时语法检查等内置自动化技术，大大降低了编写代码的工作量，同时它还提供了极其丰富的控件，使得 Visual Basic 6.0 成为最简捷、最有效的开发 Windows 应用程序工具。随着网络技术的不断发展，Visual Basic 6.0 的应用范围已从开发单机软件发展到开发网上分布式应用软件。Visual Basic 6.0 不但受到广大初、中级计算机爱好者的欢迎，也越来越受到高级程序员的欢迎。

本书精心选取了 100 个编程实例，共分为七篇。

第一篇 窗体界面，讲述如何利用 Visual Basic 6.0 设计窗体与界面。

第二篇 图形图像处理，讲述如何利用 Visual Basic 6.0 显示、处理图像。

第三篇 多媒体，讲述如何利用 Visual Basic 6.0 设计多媒体程序。

第四篇 网络，讲述如何利用 Visual Basic 6.0 设计网络应用程序。

第五篇 数据库，讲述如何利用 Visual Basic 6.0 设计数据库应用程序。

第六篇 文件与系统，讲述如何利用 Visual Basic 6.0 设计文件处理程序与系统控制程序。

第七篇 其他应用程序设计，讲述如何利用 Visual Basic 6.0 设计监视程序、回收站等应用程序。

本书着重讲解实例效果、制作步骤和有关参数调整的问题，这可以使得读者在学会应用实例的同时，对相关函数调用与参数配置也有相应的了解，起到触类旁通的作用。

本书的配套光盘中包含有本书的所有实例，全部经过笔者的精心调试，读者可以直接调用。

本书由三角洲数码科技开发中心策划，段兴主编，张敬韶、宋雷、王玉亮、赵鲁编写。

编者

stard001@sohu.com

目 录

第一篇 窗体与界面.....	1
实例 1 异形窗体.....	1
实例 2 图形窗体.....	5
实例 3 定制自己的提示栏.....	10
实例 4 个人助手界面的实现.....	17
实例 5 任意多边形窗体.....	23
实例 6 制作屏幕放大镜.....	27
实例 7 悬浮窗体.....	30
实例 8 将位图图标放置在菜单中.....	36
实例 9 将系统菜单变为自己的菜单.....	38
实例 10 将一个窗体始终放在最前面.....	43
实例 11 可以用鼠标切分的窗体.....	46
实例 12 将窗体压入系统托盘.....	50
实例 13 菜单的竖向分列.....	54
实例 14 动态增添和移除菜单.....	58
实例 15 在关闭程序对话框中隐藏运行程序名称.....	61
实例 16 在系统菜单里加上自己的菜单项.....	63
实例 17 设置整个程序的快捷键.....	66
实例 18 弹出窗口时出现动画效果.....	68
实例 19 在菜单中加入广播按钮.....	70
实例 20 将其他应用程序作为本程序的一个子窗口.....	73
实例 21 在菜单中显示最近打开的文件记录.....	77
实例 22 IE 式浮动工具栏.....	82
实例 23 隐藏 Windows 的开始按钮.....	84
实例 24 设置系统托盘的图标及提示.....	87
实例 25 使窗体的标题栏闪烁.....	93
实例 26 动画光标.....	94
实例 27 控制 MessageBox.....	97
第二篇 图形图像处理.....	101
实例 28 图标显示的不同方式.....	101
实例 29 提取可执行文件内部所有图标.....	107
实例 30 各种算法图形及画笔.....	112
实例 31 透明图片的设置及叠加.....	117
实例 32 可以与 Photoshop 滤镜媲美的各种图像滤镜.....	123

实例 33	用鼠标旋转立方体.....	130
实例 34	将整个窗体铺满图片.....	134
实例 35	三维立方体演示.....	136
实例 36	下雪场景的制作.....	141
实例 37	渐变色彩展示.....	143
实例 38	将位图 (BMP) 转化为图标 (ICO)	146
实例 39	在系统托盘中设置动画图标.....	153
实例 40	形象的动态三维饼图.....	156
实例 41	为程序增加扫描功能.....	159
实例 42	将桌面图标文字背景设置为透明色.....	161
实例 43	让文字显示渐变色.....	163
实例 44	旋转文字.....	169
第三篇 多媒体.....		174
实例 45	播放 WAV 文件并显示波形.....	174
实例 46	袖珍 MP3 播放器.....	182
实例 47	测试声卡.....	184
实例 48	控制音量.....	189
实例 49	多光驱的 CD 播放器.....	193
实例 50	在系统托盘中增加一个光驱按钮.....	197
第四篇 网 络.....		200
实例 51	电子邮件发送程序.....	200
实例 52	电子邮件接收程序.....	205
实例 53	获取 IP 地址和计算机名称.....	212
实例 54	超级链接的制作.....	213
实例 55	发送电子邮件附件.....	215
实例 56	收发电子邮件.....	221
实例 57	实时监测局域网内部的计算机.....	230
实例 58	检测是否联网及联网方式.....	235
实例 59	新颖的拖放功能.....	238
实例 60	建立和断开网络映射驱动器.....	240
实例 61	飓风浏览器.....	243
实例 62	一个漂亮的浏览器.....	249
实例 63	点对点聊天 (服务器端)	255
实例 64	点对点聊天 (客户端)	260
实例 65	FTP 客户端程序.....	264
实例 66	从 VB 应用程序中发送 ICQ 信息	271
实例 67	拨号上网.....	275

实例 68	添加 URL 快捷方式到收藏夹、开始菜单和桌面上	277
实例 69	监视网络连接	281
实例 70	自动上网计时	285
第五篇	数据库	290
实例 71	一次 SQL 查询获得多个查询数值	290
实例 72	将多媒体文件保存到数据库中	292
实例 73	在程序运行阶段建立 MDB 数据库文件	302
实例 74	将数据库输出到一个文本文件中	304
实例 75	数据库打印	307
第六篇	文件与系统	311
实例 76	切换中文输入法	311
实例 77	发送脱机打印任务	315
实例 78	完全控制打印	321
实例 79	建立关联文件	324
实例 80	自动打开上次关机时未关闭的应用程序	326
实例 81	多功能的关机程序	329
实例 82	获取系统信息	335
实例 83	密码查看器	340
实例 84	枚举系统正在运行的程序	343
实例 85	获取驱动器序列号	349
实例 86	启动控制面板	352
实例 87	控制其他软件运行	354
实例 88	设置启动时自动被执行的程序	355
第七篇	其他类应用	358
实例 89	获取文件信息	358
实例 90	监视程序	361
实例 91	带图形的 ComboBox	366
实例 92	窗口渐变背景	370
实例 93	鼠标取词	373
实例 94	得到当前 Windows 的版本号及运行模式	377
实例 95	清空回收站	382
实例 96	在 treeview 的节点处使用图形	383
实例 97	拖放 treeview 树节点	387
实例 98	快速的全盘文件查找程序	390
实例 99	计算机加锁程序	394
实例 100	在程序中控制鼠标	399

第一篇 窗体与界面

实例 1 异形窗体

实例效果

程序运行后单击相应按钮，窗体将变为如图 1-1 所示的形状，该形状外围是圆角矩形，而中间是一个透明的椭圆，其别致的造型给人以耳目一新的感觉。当然同样也可以做成如图 1-2 所示的形状。

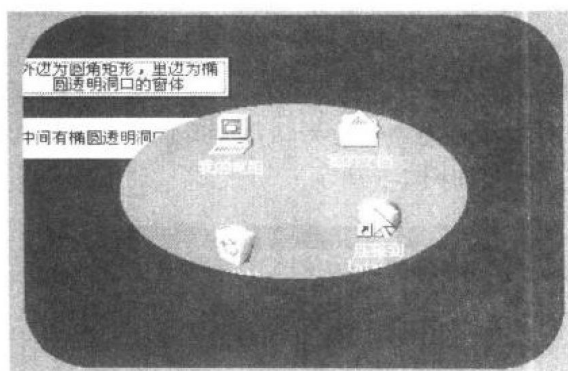


图 1-1 效果图 (1)

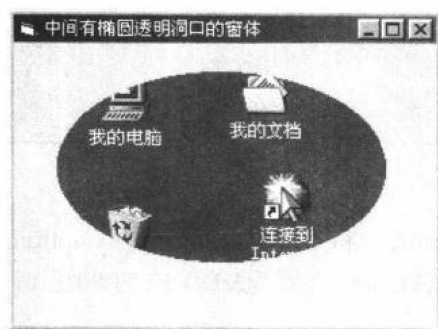


图 1-2 效果图 (2)

程序说明

普通的窗体都是矩形的，但使用 API 函数可以不同形状的窗体，比如圆角矩形的窗体等。

读者首先要理解一个重要的概念：区域。区域是描述设备场景的 GDI 对象，每个区域都有一个句柄。一个区域可以是矩形，也可以是复杂的多边形，甚至是几个区域组织在一起。窗体默认的区域就是用户看到的矩形，当然用户并非一定要用这个默认的区域，这就是做出各种各样窗体的问题核心。

本例首先在窗体上做一个圆角矩形区域，这是窗体的大概轮廓。然后在圆角矩形里再确定一个椭圆形的区域，最后把这两个区域组织成一个新的区域，并把这个新的区域设置为窗体区域。

程序运行后，在窗体上单击相关按钮，窗体即会变形，而双击窗体，程序结束。要注意的是，在卸载窗体时，用 `DeleteObject` 函数删除已定义的区域。

`CreateRoundRectRgn` 函数用于创建一个圆角矩形区域；`CreateEllipticRgn` 函数用于创建一个椭圆形区域；`CombineRgn` 函数用于将两个区域组合为一个新区域；`SetWindowRgn` 函数允许用户改变窗体的区域。

制作步骤

一、窗体界面的建立

1. 启动 Visual Basic 6.0，新建一个标准工程，如果 Visual Basic 6.0 已经运行，则单击菜单选项“文件/新建工程”，新建一个标准工程。

2. 将 Form1 属性中的 `Caption` 设置为“异形窗体”，在窗体中添加两个 `CommandButton`，并将其属性中的 `Caption` 分别设置为“外边为圆角矩形，里边为椭圆透明洞口的窗体”、“中间有椭圆透明洞口的窗体”，界面如图 1-3 所示。

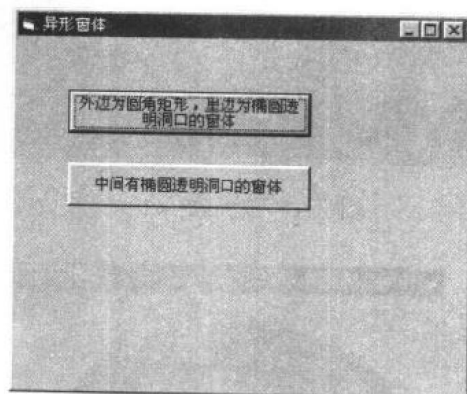


图 1-3 界面图

3. 添加一个新的窗体 Form2，将 Form2 属性中的 `Caption` 设置为“中间有椭圆透明洞口的窗体”，并将其属性中的 `BackColor` 设置为 `&H00FFFF80&`（浅绿色）。

二、源代码与注释

窗体 Form1 的代码与注释：

`Option Explicit`

' API 函数声明

`Private Declare Function CreateRectRgn Lib "gdi32" (ByVal X1 As Long, ByVal Y1 As Long, ByVal X2 As Long, ByVal Y2 As Long) As Long`

`Private Declare Function CreateRoundRectRgn Lib "gdi32" (ByVal X1 As Long, ByVal Y1 As Long, ByVal X2 As Long, ByVal Y2 As Long, ByVal X3 As Long, ByVal Y3 As Long) As Long`

`Private Declare Function CreateEllipticRgn Lib "gdi32" (ByVal X1 As Long, ByVal Y1 As Long, ByVal X2`

As Long, ByVal Y2 As Long) As Long

Private Declare Function CombineRgn Lib "gdi32" (ByVal hDestRgn As Long, ByVal hSrcRgn1 As Long, ByVal hSrcRgn2 As Long, ByVal nCombineMode As Long) As Long

Private Declare Function SetWindowRgn Lib "user32" (ByVal hWnd As Long, ByVal hRgn As Long, ByVal bRedraw As Boolean) As Long

Private Declare Function DeleteObject Lib "gdi32" (ByVal hObject As Long) As Long

'常数声明

Private Const RGN_DIFF = 4

'目标区域被设置为两个区域不相交的部分

'模块级变量声明

Private OutRgn As Long

'外边的圆角矩形区域

Private InRgn As Long

'里边的椭圆区域

Private MyRgn As Long

'圆角区域剪切掉椭圆区域后的区域,也是窗体最终的形状

Private Sub Command1_Click()

If OutRgn <> 0 And InRgn <> 0 And MyRgn <> 0 Then Exit Sub

Dim w As Long, h As Long

w = ScaleX(Form1.Width, vbTwips, vbPixels)

h = ScaleY(Form1.Height, vbTwips, vbPixels)

MyRgn = CreateRectRgn(0, 0, 0, 0)

'创建一个圆角矩形区域

OutRgn = CreateRoundRectRgn(30, 30, w - 30, h - 30, 100, 100)

创建一个椭圆区域

InRgn = CreateEllipticRgn(100, 100, w - 100, h - 100)

Call CombineRgn(MyRgn, OutRgn, InRgn, RGN_DIFF)

'改变窗口区域为新建区域

Call SetWindowRgn(Form1.hWnd, MyRgn, True)

Form1.BackColor = QBColor(4)

End Sub

Private Sub Command2_Click()

'加载窗体 Form2

Form2.Show

End Sub

'双击卸载窗体

```
Private Sub Form_DblClick()
    Unload Form1
End Sub
```

'加载初始化窗体

```
Private Sub Form_Load()
    OutRgn = 0
    InRgn = 0
    MyRgn = 0
    Form1.Width = 7800
    Form1.Height = 6000
End Sub
```

'卸载窗体时清除已定义的区域

```
Private Sub Form_Unload(Cancel As Integer)
    If MyRgn <> 0 Then DeleteObject MyRgn
    If OutRgn <> 0 Then DeleteObject OutRgn
    If InRgn <> 0 Then DeleteObject InRgn
End Sub
```

窗体 Form2 的代码与注释:

Option Explicit

'API 函数声明

```
Private Declare Function CreateRectRgn Lib "gdi32" (ByVal X1 As Long, ByVal Y1 As Long, ByVal X2 As Long, ByVal Y2 As Long) As Long
```

```
Private Declare Function CreateEllipticRgn Lib "gdi32" (ByVal X1 As Long, ByVal Y1 As Long, ByVal X2 As Long, ByVal Y2 As Long) As Long
```

```
Private Declare Function CombineRgn Lib "gdi32" (ByVal hDestRgn As Long, ByVal hSrcRgn1 As Long, ByVal hSrcRgn2 As Long, ByVal nCombineMode As Long) As Long
```

```
Private Declare Function SetWindowRgn Lib "user32" (ByVal hWnd As Long, ByVal hRgn As Long, ByVal bRedraw As Long) As Long
```

'窗体大小变化时透明洞口也相应变化

```
Private Sub Form_Resize()
```

```
    Const RGN_DIFF = 4
```

```
    Dim outer_rgn As Long
```

```
    Dim inner_rgn As Long
```

```
    Dim combined_rgn As Long
```

```
    Dim wid As Single
```

```

Dim hgt As Single
Dim border_width As Single
Dim title_height As Single
If WindowState = vbMinimized Then Exit Sub
'创建区域.
wid = ScaleX(Width, vbTwips, vbPixels)
hgt = ScaleY(Height, vbTwips, vbPixels)
outer_rgn = CreateRectRgn(0, 0, wid, hgt)
border_width = (wid - ScaleWidth) / 2
title_height = hgt - border_width - ScaleHeight
'创建一个椭圆区域
inner_rgn = CreateEllipticRgn( _
border_width + ScaleWidth * 0.1, _
title_height + ScaleHeight * 0.1, _
ScaleWidth * 0.9, ScaleHeight * 0.9)
'将区域组合为一个新区域
combined_rgn = CreateRectRgn(0, 0, 0, 0)
CombineRgn combined_rgn, outer_rgn, _
inner_rgn, RGN_DIFF
'改变窗口的区域
SetWindowRgn hWnd, combined_rgn, True
End Sub

```

实例 2 图形窗体

实例效果

图形窗体可以说是异形窗体中最有特色的一种，它的形状由图形决定，可以任意设置，效果如图 2-1 所示，鼠标所指气球状即为图形窗体。

程序说明

创建异形窗体需要使用 API 函数：CreateRectRgn、CombineRgn、SetWindowRgn、DeleteObject 函数等，而创建图形窗体还需要多用一个 GetPixel 函数。

CreateRectRgn 函数用于创建一个区域，CombineRgn 函数用于将两个区域组合为一个新区域，SetWindowRgn 函数将新建区域设置为窗体区域，DeleteObject 函数用于删除已定义的区域。

GetPixel 函数用于取得一点（单位为像素）的颜色值，然后与设定的背景色相比（本例为绿色 BGCOLOR = 65280），用一循环结构得到图形边界，并用 CombineRgn 函数设置组合为新区

域, SetWindowRgn 函数将新建区域设置为窗体区域, 形成图形窗体。

为了更明显地表示图形窗体, 本例设置了图形窗体由屏幕底部向上运动, 单击鼠标右键可以改变运动方向。

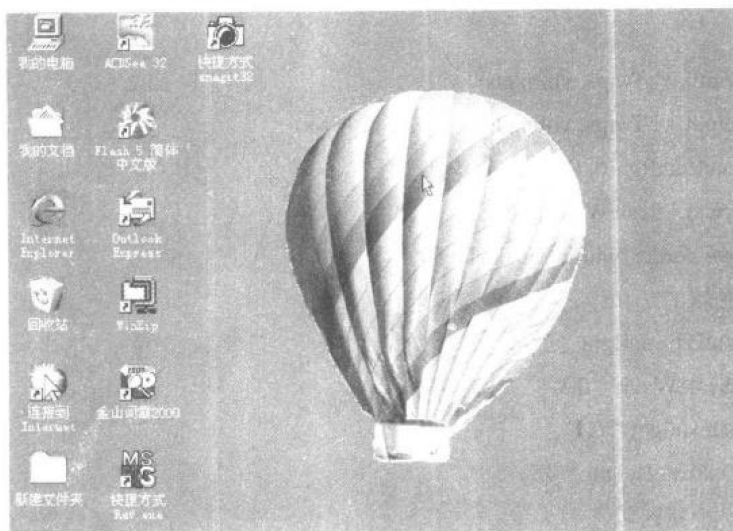


图 2-1 效果图

制作步骤

一、窗体界面的建立

1. 启动 Visual Basic 6.0, 新建一个标准工程, 如果 Visual Basic 6.0 已经运行, 则单击菜单选项“文件/新建工程”, 新建一个标准工程。

2. 将 Form1 属性中的 Caption 设置为“正在加载...”, MaxButton、MinButton 设置为“False”。在窗体中添加一个 Label, 并将其属性中的 Caption 设置为“正在创建图形(异形)窗口, 请稍候...”, 界面如图 2-2 所示。

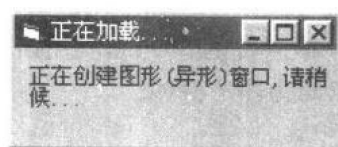


图 2-2 界面

3. 添加一个新的窗体 Form2, 将 Form2 属性中的 Caption 和 ControlBox 都设置为空, 以去掉窗口的标题栏, 在窗体中添加一个 Image、一个 Timer, 并将 Timer 属性中的 Interval 设置为“10”。

4. 添加一个模块。

二、源代码与注释

窗体 Form1 的代码与注释:

```

Public Face As Long
Private Sub Form_Load()
    Me.Show
    DoEvents
    '创建图形窗体
    Face = CreateRegionFromFile(App.Path & "\bill2.bmp")
    Form2.Show
    SetWindowRgn Form2.hwnd, Face, True
    ' 把窗体设置在最上层
    SetWindowPos Form2.hwnd, HWND_TOPMOST, 0, 0, 0, 0, SWP_NOMOVE + SWP_NOSIZE +
SWP_SHOWWINDOW
    Me.Visible = False
End Sub

```

窗体 Form2 的代码与注释:

```

Private Direction As String
Private First As Boolean
Private Sub Form_Load()
    Direction = "UP"
    First = True
End Sub

Private Sub Form_MouseDown(Button As Integer, Shift As Integer, x As Single, y As Single)
    Timer1.Enabled = False
    If Button = 1 Then
        ReleaseCapture
        ' 发送消息,在窗体上按下鼠标左键可以拖动窗口
        SendMessage hwnd, WM_NCLBUTTONDOWN, HTCAPTION, ByVal 0&
    ElseIf Button = 2 Then
        '在窗体上按下鼠标右键改变运动方向
        Direction = "DOWN"
    End If
    Timer1.Enabled = True
End Sub

Private Sub Timer1_Timer()
    ' 窗口运动状态
    If Direction = "UP" Then
        ' 在窗体向上运动至完全隐藏时改变运动状态
        If Me.Top > (-1 * Me.Height) Then
            Me.Top = Me.Top - 10

```

```

Else
    Me.Top = Screen.Height
    Me.Left = CLng(Rnd * (Screen.Width - Me.Width))
End If
Else
    ' 在窗体向下运动至完全隐藏时改变运动状态
    If Me.Top < Screen.Height Then
        Me.Top = Me.Top + 100
    Else
        '退出
        End
    End If
End If
End Sub

```

模块的代码与注释:

' 把窗口设置在最上层

```
Public Declare Function SetWindowPos Lib "user32" (ByVal hwnd As Long, ByVal hWndInsertAfter As Long,
ByVal x As Long, ByVal y As Long, ByVal cx As Long, ByVal cy As Long, ByVal wFlags As Long) As Long
```

```
Public Const HWND_TOPMOST = -1
```

```
Public Const SWP_NOMOVE = &H2
```

```
Public Const SWP_NOSIZE = &H1
```

```
Public Const SWP_SHOWWINDOW = &H40
```

' 区域句柄

```
Public Declare Function SetWindowRgn Lib "user32" (ByVal hwnd As Long, ByVal hRgn As Long, ByVal
bRedraw As Boolean) As Long
```

```
Public Declare Function CombineRgn Lib "gdi32" (ByVal hDestRgn As Long, ByVal hSrcRgn1 As Long,
ByVal hSrcRgn2 As Long, ByVal nCombineMode As Long) As Long
```

```
Public Declare Function CreateRectRgn Lib "gdi32" (ByVal X1 As Long, ByVal Y1 As Long, ByVal X2 As
Long, ByVal Y2 As Long) As Long
```

```
Public Const RGN_AND = 1
```

```
Public Const RGN_COPY = 5
```

```
Public Const RGN_DIFF = 4
```

```
Public Const RGN_OR = 2
```

```
Public Const RGN_XOR = 3
```

'图像句柄

```
Public Declare Function GetPixel Lib "gdi32" (ByVal hdc As Long, ByVal x As Long, ByVal y As Long) As
Long
```

```
Public Declare Function BitBlt Lib "gdi32" (ByVal hDestDC As Long, ByVal x As Long, ByVal y As Long,
```

ByVal nWidth As Long, ByVal nHeight As Long, ByVal hSrcDC As Long, ByVal xSrc As Long, ByVal ySrc As Long, ByVal dwRop As Long) As Long

Public Declare Function SelectObject Lib "gdi32" (ByVal hdc As Long, ByVal hObject As Long) As Long

Public Declare Function CreateCompatibleBitmap Lib "gdi32" (ByVal hdc As Long, ByVal nWidth As Long, ByVal nHeight As Long) As Long

Public Declare Function CreateCompatibleDC Lib "gdi32" (ByVal hdc As Long) As Long

Public Declare Function DeleteObject Lib "gdi32" (ByVal hObject As Long) As Long

Public Declare Function DeleteDC Lib "gdi32" (ByVal hdc As Long) As Long

' 图像操作

Public Const SRCAND = &H8800C6 ' (DWORD) 目标 = 源 AND 目标

Public Const SRCCOPY = &HCC0020 ' (DWORD) 目标 = 源

Public Const MERGEPAINT = &HBB0226 ' (DWORD) 目标 = (NOT 源) OR 目标

' 移动无边框窗口

Public Const WM_NCLBUTTONDOWN = &HA1

Public Const HTCAPTION = 2

Public Declare Sub ReleaseCapture Lib "user32" ()

Public Declare Function SendMessage Lib "user32" Alias "SendMessageA" (ByVal hwnd As Long, ByVal wParam As Long, ByVal wMsg As Long, ByVal lParam As Any) As Long

' 其他

Public Declare Sub Sleep Lib "kernel32" (ByVal dwMilliseconds As Long)

' 变量声明

Private FW As Long

Private FH As Long

' 窗口背景色(即需要去掉的部分)

Private Const BGColor = 65280

Public Function CreateRegionFromFile(strFile) As Long

Dim T As Integer

Dim R As Integer

Form2.Image1.Picture = LoadPicture(strFile)

Form2.Width = Form2.Image1.Width

Form2.Height = Form2.Image1.Height

Form2.Image1.Visible = False

Form2.Picture = LoadPicture(strFile)

' 内部单位为像素

Form2.ScaleMode = 3

FW = Form2.ScaleWidth

FH = Form2.ScaleHeight

Dim DC As Long

Dim BMP As Long

```

Dim Pix As Long
Dim rgnInv As Long
Dim rgn As Long
Dim rgnTotal As Long
DC = CreateCompatibleDC(Form2.hdc)
BMP = SelectObject(DC, LoadPicture(strFile))
'创建新区域
rgnTotal = CreateRectRgn(0, 0, FW, FH)
rgnInv = CreateRectRgn(0, 0, FW, FH)
'将多个区域组合为一个新区域
CombineRgn rgnTotal, rgnTotal, rgnTotal, RGN_XOR
'设置图像边界
For T = 0 To FW
    For R = 0 To FH
        Pix = GetPixel(DC, T, R)
        If Pix = BGColor Then
            ' 创建区域
            rgn = CreateRectRgn(T, R, T + 1, R + 1)
            CombineRgn rgnTotal, rgnTotal, rgn, RGN_OR
            DeleteObject rgn
        End If
    Next R
Next T
'将多个区域组合为一个新区域
CombineRgn rgnTotal, rgnTotal, rgnInv, RGN_XOR
' 设置创建异形窗体函数
CreateRegionFromFile = rgnTotal
End Function

```

实例 3 定制自己的提示栏

实例效果

读者对提示栏都不会感到陌生,当鼠标移动到某个按钮上时,一个浅黄色的小提示窗口就会出现在鼠标指针旁,对控件的功能或用法给出提示。一般设计程序时提示栏都是由控件内置方法决定的,设计者无法自由设置。这个实例演示了可以显示多行文本的提示栏的设计方法,不但可以设置提示栏的文本图像,而且还可以设置提示栏的显示时间,改变提示栏背景颜色等,效果如图 3-1 所示。