

以完成作品为教学导向，真正整合理论与实务

Visual Basic 案例教程



郝春强 编著

- ▶ 编程没有多数人想象的那么困难，通过本案例教程可以学会从基本的窗体设计到复杂的实用程序设计是如何像搭积木一样完成的。
- ▶ 100多个案例都包括『案例说明』、『设计思想』、『设计步骤』等5个部分，真正做到了全面分析，详细讲解。
- ▶ 光盘包括每个案例的完整源代码和执行程序，另外还有200多个实用控件。



- ▶ 案例源文件导读
- ▶ 实用控件大派送

北京科海培训中心

Visual Basic 案例教程

郝 春 强 编著

中科多媒体电子出版社 出版

2001. 6

内 容 提 要

本书精选了 100 多个有独特设计思想和编程技巧的案例，通过对这些案例进行详细地分析和讲解，使读者能够快速提高自己的程序设计水平。

这 100 多个案例分门别类地归属于本书的 9 章中，每一章集中介绍了 Visual Basic 编程的某一个侧面，它们分别是：窗体设计、控件与菜单、图形与文本设计、系统控制、文件管理、数据库编程技术、多媒体应用程序、网络编程和实用程序。

本书是专为具有一定程序设计基础，而又迫切需要提高程序设计能力的读者编写的。对于其他编程爱好者以及程序设计人员也有一定的参考价值。

由于篇幅有限，书中没有完整地列出所有程序的源代码，详细的源代码请查阅光盘。

品 名：Visual Basic 案例教程
作 者：郝春强
责任编辑：马首鳌
出 版：中科多媒体电子出版社
印 刷 者：北京门头沟胶印厂
发 行：新华书店总店北京科技发行所
开 本：787×1092 1/16 印张：23 字数：540 千字
版 次：2001 年 6 月第 1 版 2001 年 10 月第 2 次印刷
印 数：5001~7000
盘 号：ISBN 7-900025-88-X
定 价：35.00 元（1CD）



前 言

Visual Basic 是 Microsoft 公司推出的一个集成开发环境，是 Microsoft Visual Studio 系列产品之一。Visual Basic 采用面向对象的程序设计技术，提供了开发 Windows 应用程序的最迅速、最简捷的方法。全世界有几百万的程序设计人员正在使用 Visual Basic 开发各种类型的软件。

目前，已出版的有关 Visual Basic 的书籍可谓不计其数，但以讲解 Visual Basic 功能的书居多，而案例型的书较少。事实上，编程是一门实践性很强的课程，在对语言本身以及开发环境较为熟悉后，还必须通过大量的编程实践来掌握编程技巧，提高程序设计的综合能力。为此，本书精选了 100 多个有独特设计思想和编程技巧的案例，试图通过对这些案例进行的详细分析和讲解，使读者能够快速提高自己的程序设计水平。

这 100 多个案例分门别类地归属于本书的 9 章中，每一章集中介绍了 Visual Basic 编程的某一个侧面。它们分别是：窗体设计、控件与菜单、图形与文本设计、系统控制、文件管理、数据库编程技术、多媒体应用程序、网络编程和实用程序。各章中所选案例均是按由易到难、由浅入深的方式编排的。

在案例的讲解上，全书采用了相同的方式，每个案例都包括说明、设计思想、设计步骤、代码分析和总结 5 个部分。在案例说明部分中，我们以图文结合的方式给出了案例的功能说明以及运行效果；设计思想部分介绍编写该案例的一些基本思想和技巧；设计步骤部分介绍该案例的设计过程并列出程序的源代码；代码分析部分对程序中一些关键的代码进行分析讲解；总结部分主要告诉读者本案例的可借鉴之处，以便使读者在学习过程中能够达到举一反三的效果。对一个案例通过上述 5 个部分的讲解，可谓是面面俱到，应有尽有。通过学习，相信读者可以在较短的时间内迅速提高自己的程序设计水平。

本书是以 Visual Basic 6.0 中文版为背景编写的，书中所有程序都在 Visual Basic 6.0 以及 Windows 98 系统下调试通过。

本书是专为具有一定程序设计基础，而又迫切需提高程序设计能力的读者编写的。对于其他编程爱好者以及程序设计人员也有一定的参考价值。

作 者

2001 年 4 月

光盘使用说明

运行环境:

- Pentium 166以上的处理器
- VGA显卡
- 光盘驱动器
- Windows 95/98/NT/2000操作系统
- IE 4.0及以上版本浏览器
- 800×600分辨率
- 16位真彩色以上显示模式

操作方法:

- 一般情况下, 将本光盘放到光驱中后, 光盘就会自动运行。
- 如果本光盘没有自动运行, 请双击光盘根目录下的Start.exe或Index.htm文件即可启动运行。
- 如果计算机上没有安装Winzip解压软件, 请先运行光盘根目录下的Winzip80.exe进行安装。

光盘内容:

- 案例源文件和一些实用控件 (注意有些控件使用前需要注册)。

目 录

第 1 章 窗体设计	1
案例 1.1 渐变背景	1
案例 1.2 窗体背景图案的实现	2
案例 1.3 拖动没有标题栏的窗口	5
案例 1.4 奇形怪状的窗体	6
案例 1.5 创建字型窗体	9
案例 1.6 创建具有动画效果的窗体标题	12
案例 1.7 创建垂直的窗体标题栏	14
案例 1.8 窗体启动特效	18
案例 1.9 使窗体的“关闭”按钮无效	20
案例 1.10 软件封面的制作	22
案例 1.11 制作屏幕保护程序	25
案例 1.12 制作 About 窗体	28
案例 1.13 自动调整控件的位置与大小	32
案例 1.14 为窗体添加透明阴影	34
案例 1.15 向窗体的控制菜单栏添加菜单项	38
案例 1.16 编写托盘程序	40
第 2 章 控件与菜单	48
案例 2.1 聪明的按钮	48
案例 2.2 制作浮动按钮	50
案例 2.3 制作动态按钮	53
案例 2.4 创建只读文本框	55
案例 2.5 设置文本框的滚动区域	57
案例 2.6 滚动文本框中的内容	59
案例 2.7 文本框的输入限制	61
案例 2.8 实现 UNDO 功能	63
案例 2.9 获取多行文本框信息	65
案例 2.10 屏幕取词	67
案例 2.11 使列表框中不同项目显示不同的提示	70
案例 2.12 更换文本框的快捷菜单	72
案例 2.13 在窗体间拖动控件	75
案例 2.14 在运行时使用代码生成控件	78
案例 2.15 菜单标记	80

案例 2.16 创建快捷菜单.....	82
案例 2.17 创建位图菜单.....	85
第 3 章 图形与文本设计	88
案例 3.1 调色板	88
案例 3.2 五彩缤纷的清屏效果.....	90
案例 3.3 滚动显示图片	93
案例 3.4 绘制柱状图表	95
案例 3.5 “百叶窗”图形特效.....	99
案例 3.6 图片切换特效	101
案例 3.7 实现屏幕变暗的效果.....	105
案例 3.8 旋转位图	107
案例 3.9 缩放位图	110
案例 3.10 显示背景透明的图片.....	112
案例 3.11 演示 3D 旋转效果.....	115
案例 3.12 滚动字幕	122
案例 3.13 实现打字效果.....	124
案例 3.14 在 VB 中使用 Word 的“艺术字”	126
案例 3.15 创建 3D 文字效果.....	129
案例 3.16 创建旋转文本效果.....	133
案例 3.17 创建蒙板字	135
第 4 章 系统控制	138
案例 4.1 重新启动计算机.....	138
案例 4.2 获取磁盘空间信息.....	140
案例 4.3 获取磁盘信息	142
案例 4.4 获取磁盘的序列号.....	144
案例 4.5 获取与设置计算机名.....	146
案例 4.6 获取系统目录	148
案例 4.7 显示动画鼠标指针.....	149
案例 4.8 限制鼠标的移动.....	152
案例 4.9 定义热键	154
案例 4.10 屏蔽 Ctrl+Alt+Del 功能键.....	156
案例 4.11 改变屏幕的设置.....	158
案例 4.12 定时启动程序.....	161
案例 4.13 开机时自动更换墙纸.....	165
案例 4.14 自动记录应用软件的工作时间.....	166
案例 4.15 设置启动程序.....	170
案例 4.16 隐藏与显示任务栏.....	175
案例 4.17 变化的“开始”按钮.....	177

案例 4.18 将程序从 Windows 的系统任务列表中隐藏.....	180
案例 4.19 查看窗口信息.....	182
第 5 章 文件管理	186
案例 5.1 读写 INI 文件	186
案例 5.2 VB 中的 APP 对象及其应用	189
案例 5.3 加密文件	192
案例 5.4 拷贝文件	195
案例 5.5 删除文件到回收站.....	198
案例 5.6 创建文件夹列表框.....	200
案例 5.7 显示文件的属性.....	202
案例 5.8 利用递归方法查找文件.....	205
案例 5.9 获取目录的大小.....	209
第 6 章 数据库编程技术	212
案例 6.1 用数据库保存程序的用户设置.....	212
案例 6.2 从 Excel 中读取数据到数据库.....	214
案例 6.3 非绑定模式下的 DbGrid 控件	217
案例 6.4 将数据库转换为文本文件.....	223
案例 6.5 使用图表显示数据.....	226
案例 6.6 制作通讯录	230
第 7 章 多媒体应用程序	234
案例 7.1 使用 MMControl 控件制作 CD 播放器.....	234
案例 7.2 使用 API 函数制作 CD 播放器.....	237
案例 7.3 使用 MMControl 控件播放媒体文件	242
案例 7.4 Flash 动画播放器.....	244
案例 7.5 RealPlayer 播放器	252
案例 7.6 获取光驱的盘符.....	268
案例 7.7 测试声卡	270
案例 7.8 弹动的小球	275
第 8 章 网络编程	280
案例 8.1 创建超级链接	280
案例 8.2 简单的拨号程序.....	283
案例 8.3 发送电子邮件	285
案例 8.4 收取邮件	288
案例 8.5 制作自己的浏览器.....	293
案例 8.6 拖放网页的图像与链接.....	297

第 9 章 实用程序	301
案例 9.1 编写一个计算器	301
案例 9.2 Hanoi 塔问题	308
案例 9.3 内存统计程序	314
案例 9.4 指针式时钟	316
案例 9.5 刮卡	320
案例 9.6 打字练习程序	322
案例 9.7 “井字”游戏	326
案例 9.8 将阿拉伯数字转换成汉字数字	341
案例 9.9 密码登录程序	344
案例 9.10 进制转换器	350

第1章 窗体设计

设计一个友好且富有特色的用户界面是所有开发人员所希望的。本章精选了 16 个与窗体设计有关的典型案例，相信会使读者大开眼界、耳目一新。

案例 1.1 渐变背景

【案例说明】

许多 Windows 应用程序的安装界面都是一个颜色由蓝至黑的渐变背景，如图 1.1 所示。本例将介绍如何在 VB 中实现窗体背景的渐变效果。

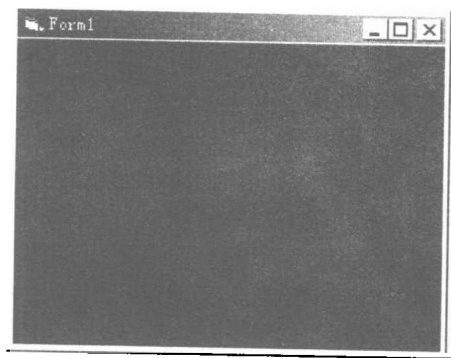


图 1.1 窗体背景的渐变效果

【设计思想】

使用 VB 产生渐变效果很简单，其基本思想是：在窗体中从上至下依次绘制多个矩形，只要有足够多的矩形，同时它们的填充色从蓝变化到黑，就能很好地模拟出渐变效果了。

【设计步骤】

- (1) 新建一个工程，将窗体的 **Autoredraw** 属性设置为 **True**。
- (2) 在代码窗口中编写一个用来实现窗体背景渐变的子过程，代码如下：

```
Private Sub Gradient(TheObject As Object, Redval, Greenval, Blueval)
    Dim Step, i, T, L, R, B
    Step = (TheObject.Height / 60)
    T = 0
    L = 0
    R = TheObject.Width
    B = T + Step
```

```
'使用循环在窗体上从上至下依次绘制 60 个矩形
For i = 1 To 60
    TheObject.Line (L, T)-(R, B), RGB(Redval, Greenval, Blueval), BF
    Redval = Redval - 4
    Greenval = Greenval - 4
    Blueval = Blueval - 4
    If Redval <= 0 Then Redval = 0
    If Greenval <= 0 Then Greenval = 0
    If Blueval <= 0 Then Blueval = 0
    T = B
    B = B + Step
Next
End Sub
```

(3) 在窗体的 **Resize** 事件过程中调用子过程 **Gradient**, 代码如下:

```
Private Sub Form_Resize()
    Gradient Form1, 0, 0, 255
End Sub
```

【代码分析】

我们首先定义了一个名为 **Gradient** 的子过程, 再在该子过程中使用 **For** 循环语句来实现在窗体中从上至下依次绘制 60 个矩形。其中的 **Line** 方法用来绘制矩形, 变量 **L** 与 **T** 为矩形左上顶点的坐标, **R** 与 **B** 为矩形右下顶点的坐标, 变量 **Redval**、**Greenval** 和 **Blueval** 分别表示红、绿、蓝三原色的值。

在窗体的 **Resize** 事件中, 以参数 **Form1** (窗体名) 和颜色值 (0, 0, 255) 调用子过程 **Gradient**, 运行程序后, 即可看到窗体具有由蓝至黑的渐变背景。若以其他颜色值调用该函数, 则可得到其他颜色的渐变背景。改变窗体的大小, 渐变色会随着充满整个窗体。

【总结】

通过在窗体上绘制各种形式和颜色的图形, 可以创建出具有各种渐变颜色和各種渐变形式背景的窗体。例如, 若在窗体上绘制的不是矩形, 而是一系列颜色渐变的同心圆, 则窗体背景将呈现出径向渐变的效果。

案例 1.2 窗体背景图案的实现

【案例说明】

一般商业软件的窗体都是以一种非常美丽的图案为背景的, 并且在改变窗体大小后, 图案能自动填充整个窗体, 如图 1.2 所示。本例将介绍如何在 VB 中制作出该窗体。

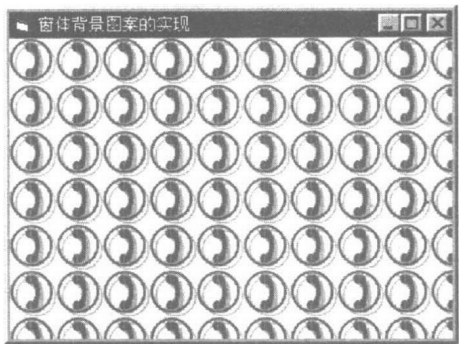


图 1.2 具有背景图案的窗体

【设计思想】

采用 VB 进行窗体设计时,要想实现立体感很强的纹理图案,一般是采用窗体的 `Picture` 属性调用位图文件。使用这种方法实现有许多缺点,其一是窗体大小受位图大小的限制,调整起来麻烦,不具备通用性和灵活性;其二是浪费资源,因为窗体背景基本上是由图案重复排列而成的,采用与窗体同样大小的位图是一种浪费。实际上在 VB 中可以使用 API 函数来制作出以图案为背景的窗体。基本思想是:事先在窗体的左上角放置一个图片框,再在图片框中放置一个图案,如图 1.3 所示。在程序运行时,在代码中使用 API 函数 `BitBlt` 将该图片拷贝在窗体的其他部分。

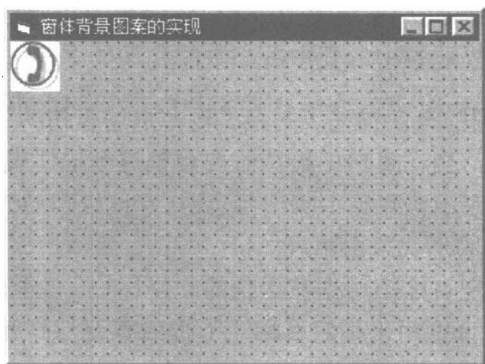


图 1.3 窗体的设计

【设计步骤】

- (1) 新建一个工程,在窗体的左上角放置一个图片框,设置其名称为 `Pic`。
- (2) 在图片框中放置一个图片。
- (3) 在窗体的声明段声明 API 函数 `BitBlt`,代码如下:

```
Private 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
```

(4) 编写一个设置窗体背景图案的子过程 fillpic, 代码如下:

```
Public Sub fillpic(w As Object, pic As PictureBox)
    Dim i, j As Integer
    For i = 0 To w.Height Step pic.ScaleHeight
        For j = 0 To w.Width Step pic.ScaleWidth
            '使用 BitBlt 函数将图片框中的位图拷贝到窗体上
            BitBlt form1.hDC, i, j, pic.ScaleWidth, pic.ScaleHeight, _
                pic.hDC, 0, 0, vbSrcCopy
        Next
    Next
End Sub
```

(5) 在 Form_Paint 事件过程中调用 fillpic 子过程, 代码如下:

```
Private Sub Form_Paint()
    fillpic form1, Picture1
End Sub
```

【代码分析】

本例中我们主要使用到了 API 函数 BitBlt, 该函数的功能是将位图从一个内存设备描述表复制到另一内存设备描述表中。其中各参数的含义如下:

- HdestDC: 目标位图的设备描述表句柄。设备描述表句柄 (hDC) 用来标识一个称为设备描述表的对象, 当在窗口中绘图时要用到这个描述表。hDC 句柄用于窗体、图片框和打印机。当在这些对象上输出信息时, 可以把 hDC 作为它们的属性, 即可以以“对象名.hDC”的形式来引用。
- x、y: 分别用来指定目标矩形框左上角的 x 坐标与 y 坐标。
- nWidth、nHeight: 分别用来指定目标矩形框的宽度与高度。
- hSrcDC: 源位图的设备描述表句柄。
- xSrc、ySrc: 分别用来指定源位图左上角的 x 坐标与 y 坐标。
- dwRop: 光栅操作码。当其值为 vbSrcCopy 时, 表示把源位图拷贝到目标位图。

【总结】

在本例中, 我们使用到了 API 函数。API 是应用程序接口 (Application Program Interface) 的简称, 它是 Windows 环境中可供任何 Windows 应用程序访问和调用的一组函数集合, 这些函数是用 C 语言编写的。Windows 操作系统中包含了 1000 多个功能强大, 经过严格测试的 API 函数, 可供程序开发人员编程时直接调用。Visual Basic 提供了访问和调用这些 API 函数的能力, 充分利用这些 API 函数, 可大大增强 VB 的编程能力, 并可实现一些用 Basic 语言本身不能实现的特殊功能。在以后的编程中, 我们还将多次用到 API 函数, 希望

读者能给予 API 函数足够的重视。

案例 1.3 拖动没有标题栏的窗口

【案例说明】

许多 Windows 应用程序的用户界面被设计成一个非常美观的小面板，它们的共同特点是没有了标题栏。但是，在没有标题栏以后，相应的问题也就出现了，即不能再使用鼠标来拖动窗体。

本例就来介绍如何使用鼠标拖动没有标题栏的窗体，该窗体如图 1.4 所示。

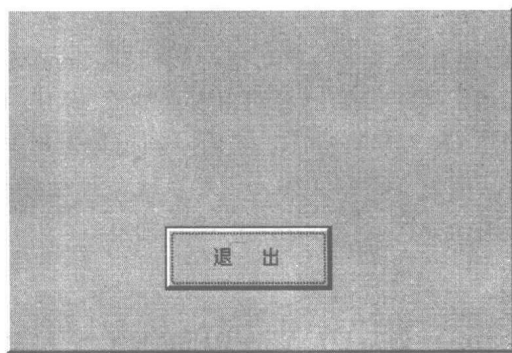


图 1.4 无标题栏的窗体

【设计思想】

使用 VB 本身来实现本例是较困难的，但如果借助 API 函数，则问题就变得非常容易。

【设计步骤】

(1) 新建一个工程，将窗体的 Caption 属性设置为空，ControlBox 属性的值设为 False，使之成为一个无标题的窗体。

(2) 在窗体上放置一个按钮控件，并设置其 Caption 属性为“退出”。

(3) 将以下几个 API 函数和常量的声明添加到窗体的声明段。

```
Private Declare Function SendMessage Lib "user32" Alias "SendMessageA" ( _  
    ByVal hWnd As Long, ByVal wParam As Long, ByVal lParam As Any) As Long  
Private Declare Function ReleaseCapture Lib "user32" () As Long  
Private Const WM_SYSCOMMAND = &H112  
Private Const SC_MOVE = &HF012
```

(4) 编写窗体的 MouseDown 事件过程，代码如下：

```
Private Sub Form_MouseDown(Button As Integer, Shift As Integer, _
```

```
X As Single, Y As Single)
ReleaseCapture
SendMessage Form1.hWnd, WM_SYSCOMMAND, SC_MOVE, 0
End Sub
```

(5) 编写“退出”按钮的 Click 事件过程，代码如下：

```
Private Sub Command1_Click()
    End
End Sub
```

【代码分析】

代码中 ReleaseCapture 函数是用来释放对鼠标的捕捉。SendMessage 的作用是用来向窗口发送消息。参数 hWnd 用来指定接受消息的窗口；参数 wParam 用来指定要发送的消息，本例中指定的消息为 WM_SYSCOMMAND，它表示向窗口发送系统命令；wParam 是消息的第一个参数，本例指定其值为 SC_MOVE，表示系统命令是移动窗口。lParam 是消息的第二个参数，本例中不使用该参数，因此将其值指定为 0。

【总结】

本例介绍的是如何在 VB 中使用鼠标拖动无标题的窗体，其实，对于在窗体中实现拖动控件的原理也是一样的。例如，将以下代码添加到按钮的 MouseDown 事件过程中，就可以实现在窗体上拖动按钮。

```
ReleaseCapture
SendMessage Command1.hWnd, WM_SYSCOMMAND, SC_MOVE, 0
```

需要说明的是，由于 ReleaseCapture 函数释放了系统对鼠标的捕捉，因此，按钮就不能再响应单击、双击等事件。

案例 1.4 奇形怪状的窗体

【案例说明】

普通的窗体都是正正方方的，使用 API 函数可以做出一些形状奇怪的窗体。例如，圆角矩形的窗体、椭圆型的窗体等，我们还可以在窗体中间挖一个洞。在本例中，我们制作一个外轮廓是椭圆，中间有一个方孔的窗体。该程序的界面设计如图 1.5 所示，单击“变形”按钮，窗体变形后的形状如图 1.6 所示。

【设计思想】

先要理解一个重要的概念——区域。区域是设备上的一块空间，它可以是任意形状。例如，一个区域可以是矩形、椭圆，也可以是多边形，甚至可以是几个区域合并而成的复杂形状。窗体默认的区域是矩形，每个区域都有一个句柄。

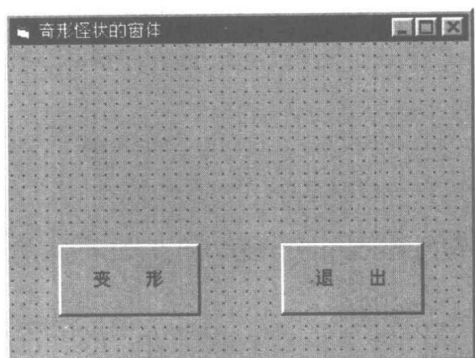


图 1.5 初始的窗体

可以通过以下 4 个步骤制作出图 1.6 所示的窗体:

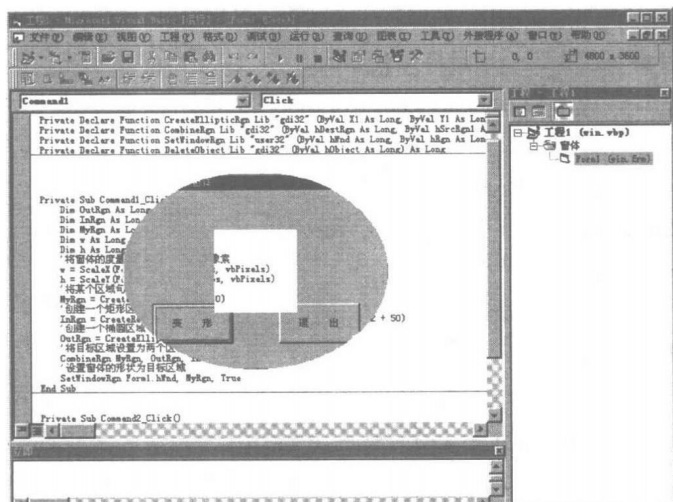


图 1.6 变形后的窗体

- (1) 创建一个椭圆区域, 这是窗体的外轮廓。
- (2) 在椭圆里再创建一个矩形区域。
- (3) 合并这两个区域, 使它成为一个外圆内方的区域, 这个区域就是最终的窗体形状。
- (4) 设置窗体的形状为这个合并出来的区域。

【设计步骤】

- (1) 新建一个工程，在窗体上放置两个按钮，并设置其 Caption 属性分别为“变形”和退出。
- (2) 在窗体的声明段声明几个程序中将要使用的 API 函数，代码如下：

```
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 Boolean) As Long
```

(3) 编写“变形”按钮的 Click 事件过程，代码如下：

```
Private Sub Command1_Click()  
    Dim OutRgn As Long  
    Dim InRgn As Long  
    Dim MyRgn As Long  
    Dim w As Long  
    Dim h As Long  
    '将窗体的度量单位由 Twips 转换为像素  
    w = ScaleX(Form1.Width, vbTwips, vbPixels)  
    h = ScaleY(Form1.Height, vbTwips, vbPixels)  
    '将某个区域句柄赋给 MyRgn  
    MyRgn = CreateRectRgn(0, 0, 0, 0)  
    '创建一个矩形区域  
    InRgn = CreateRectRgn(w / 2 - 50, h / 2 - 50, w / 2 + 50, h / 2 + 50)  
    '创建一个椭圆区域  
    OutRgn = CreateEllipticRgn(0, 0, w, h)  
    '将目标区域设置为两个区域不相交的部分  
    CombineRgn MyRgn, OutRgn, InRgn, RGN_XOR  
    '设置窗体的形状为目标区域  
    SetWindowRgn Form1.hWnd, MyRgn, True  
End Sub
```

(4) 编写“退出”按钮的 Click 事件过程，代码如下：

```
Private Sub Command2_Click()  
    End  
End Sub
```

【代码分析】

这里，我们对本例中使用到的几个 API 函数进行简要的说明。

CreateEllipticRgn 函数用于创建一个椭圆区域，它的返回值为所创建区域的句柄，如有错误则返回 Null。其中各参数的含义如下：

- X1、Y1：限定矩形左上角的坐标。