



求是科技

实效编程百例

Visual Basic

实效编程百例

★求是科技 刘 韬 骆 娟 等编著

第二版

- ★学习编程技巧
- ★积累编程经验
- ★剖析功能模块
- ★突出应用实效



源代码光盘
CD-ROM



人民邮电出版社
POSTS & TELECOM PRESS

实效编程 百例

Visual Basic

实效编程百例

★求是科技 刘 韬 骆 娟 等编著 (第二版)

图书在版编目 (CIP) 数据

Visual Basic 实效编程百例 (第二版) / 刘韬, 骆娟编著. —北京: 人民邮电出版社, 2004.8
ISBN 7-115-12428-0

I. V... II. ①刘...②骆... III. BASIC 语言—程序设计 IV. TP312
中国版本图书馆 CIP 数据核字 (2004) 第 071375 号

内容提要

本书通过 100 多个精选的例子讲解了利用 Visual Basic 进行应用程序开发的各个方面, 涵盖了界面外观、多媒体控制与图像处理、时间控制、操作系统、程序控制、磁盘文件、数据库、网络与通信、鼠标和键盘、数学算法和程序发布等方面的内容。每章通过经典的例子重点讲述如何利用 Visual Basic 以及第三方控件、API 函数完成特定的功能。

本书内容突出了实用性, 85% 以上的实例模仿较常见的优秀软件的相关功能, 余下实例大多为帮助读者理解重点、难懂概念所做。本书的另一个特点在于给出了不少实用性很强的“方案实例”(以往的百例图书内容基本上都属于“功能实例”), 其内容多为典型或通用的功能模块的解决方案, 包括界面设计、操作流程以及代码控制等内容。

本书适用于已经初步掌握 Visual Basic 编程概念、方法的读者阅读, 本书可以帮助读者迅速掌握实际应用中的各种经验、技巧。

实效编程百例

Visual Basic 实效编程百例 (第二版)

- ◆ 编 著 求是科技 刘 韬 骆 娟 等
责任编辑 张立科
- ◆ 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@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
读者热线 010-67132692
北京汉魂图文设计有限公司制作
北京顺义振华印刷厂印刷
新华书店总店北京发行所经销
- ◆ 开本: 787×1092 1/16
印张: 22.5
字数: 551 千字 2004 年 8 月第 2 版
印数: 17 001-23 000 册 2004 年 8 月北京第 5 次印刷

ISBN7-115-12428-0/TP·4074

定价: 35.00 元 (附光盘)

本书如有印装质量问题, 请与本社联系 电话: (010) 67129223

第二版序

在近几年的计算机图书市场上，编程小实例（通常称为百例）类的图书已经很普遍，而且为很多读者所接受。经调查发现，绝大多数百例类的图书，主要面向了最初级读者，其内容在实用程度上相对较差，经常有很多纯粹娱乐性或很偏门的小技巧性的案例。

我们通过网络调研、召开读者座谈会等途径，进行调研，结果表明很多已经具备了一定开发能力的人（且以实际开发人员居多），他们也非常喜欢和需要这种百例类的图书，只是多数图书内容在实用性方面满足不了他们的需要。

为此，我们于 2002 年初策划，并于当年 6 月份前后出版了这套实效百例丛书的第一批，一套 4 本，主要针对当时最主流的开发工具（Visual Basic 6.0、Visual C++ 6.0、Visual FoxPro 6.0 以及 Delphi 6.0），书中非常强调实例的实用功效。

在内容安排上，本套丛书按照程序的完成功能进行案例分类，尽量将相关的内容归纳在一章中。方便读者在有实际需求的时候，能快速找到解决方法（参见书中相关实例）。

在每个实例的讲解过程中，基本都分为 3 个步骤：

实例目的——讲解本例的功能所在，指出本例要到达的目的和效果，让读者做到心中有数。

实现方法——讲解技术原理/设计思路，给出技术原理的合理解释、规范的算法和流程描述，便于读者阅读代码、学习程序设计方法。

程序代码——给出具体的实现过程，包括界面设计、编写代码和注释，读者可参照实现。

自出版以来，本套丛书中取得了不错的销售效果，在当前众多百例类图书中处于佼佼者的地位。承蒙广大读者的厚爱，给我们反馈了很多非常宝贵的意见和建议，除了指出图书中存在的一些问题、部分案例在功能存有不足之外，还针对当前软件开发的特点，提出了不少好的建议和期望。为此，我们决定将第一期的 4 本图书进行深入修订、改版：

首先，我们安排了富有开发经验的开发人员，对图书中所有案例程序进行了严格测试，找出问题，并仔细审阅图书文档，针对错误和不足之处，加以完善，让读者阅读起来更为轻松、顺畅。

其次，我们充分考虑读者建议，结合最新的技术手段以及当前软件开发特点，对原书中各案例进行了筛选，去掉哪些相对失去了实效性的案例。

最后，我们又进行了优秀案例的增补工作，引进了很多实效性好、技术性强、适用于当前实际开发的案例。

本书由求是科技策划，在本书写作过程中得到了郭瑞军、卞志强、郭琦、马伟勤、林陈雷、罗颂、刘敬、王作远、窦亮、陈艳华、陈林、张增强、张文毅、彭涛、王志伟、吴星、华剑锋、吴明辉、曹广鑫、常明、刘萌、黄超、孙炳文、童行行、张益贞、闫海荣、马方魁、罗建光、贾博、张双民、李廷文的帮助，在此表示感谢。

尽快我们进行了全面修订和升级，但由于水平有限，可能还会存于一定的问题和不足，恳请广大读者提出宝贵意见和建议。作者的网站 <http://www.cs-book.com>，欢迎访问。

编者

2004 年 6 月

CONTENTS

第 1 章 界面外观..... 1

- 实例 1 带图标的菜单.....2
- 实例 2 动画进度条.....3
- 实例 3 带提示文字的列表框.....7
- 实例 4 任意形状的窗体.....8
- 实例 5 无边框的窗体.....10
- 实例 6 自动排序的列表框.....13
- 实例 7 自动显示货币格式的文本框.....18
- 实例 8 文字特效.....19
- 实例 9 调色程序.....21
- 实例 10 窗体分割条.....24
- 实例 11 自动完成文本框的文本输入.....27
- 实例 12 控制窗体的系统菜单和按钮.....29
- 实例 13 产生 Windows 任务栏.....34
- 实例 14 自动隐藏菜单.....38

第 2 章 多媒体控制与图像处理..... 40

- 实例 15 旋转的 3D 物体.....41
- 实例 16 画贝塞尔曲线.....48
- 实例 17 获得鼠标位置的 RGB 数值.....50
- 实例 18 世界地图.....52
- 实例 19 自己的 3D 引擎（方案例）.....53
- 实例 20 自己的 VCD 播放器.....58
- 实例 21 不需控件的快速绘图方法.....61
- 实例 22 屏幕抓图.....65

- 实例 23 图片浏览.....70

- 实例 24 位图处理.....74

- 实例 25 图像放大.....77

第 3 章 时间控制..... 79

- 实例 26 绘制动态 SIN 曲线.....80

- 实例 27 定时计时工具.....82

- 实例 28 同步网络时间.....84

- 实例 29 日期安排工具.....88

- 实例 30 动画文件制作工具（方案例）.....92

第 4 章 操作系统..... 97

- 实例 31 获取系统注册所有文件后缀.....98

- 实例 32 获得 BIOS 信息.....99

- 实例 33 编写屏保程序.....100

- 实例 34 设置墙纸.....103

- 实例 35 调整屏幕分辨率.....105

- 实例 36 获取系统硬件信息.....107

- 实例 37 获得 CPU 使用情况.....110

- 实例 38 编辑注册表信息.....112

- 实例 39 注销、重新启动和关闭电脑.....117

- 实例 40 获取 Windows 版本及目录.....118

- 实例 41 枚举系统可用字体.....121

- 实例 42 打开系统打印机对话框.....122

- 实例 43 打印文件.....127

- 实例 44 改变打印方向.....130

第 5 章 程序控制.....132

- 实例 45 注册/注销 ActiveX 控件.....133
- 实例 46 系统托盘程序.....136
- 实例 47 可以自动隐藏的 begin 菜单.....137
- 实例 48 隐藏程序不被“关闭程序”发现.....140
- 实例 49 枚举系统中所有进程.....141
- 实例 50 任务管理器.....144
- 实例 51 程序生成 EXCEL 表格.....147
- 实例 52 调用系统中的一些功能对话框.....148
- 实例 53 多线程方式同时进行多项任务.....150
- 实例 54 调用 DLL.....157
- 实例 55 用 Tab 键切换 SSTAB 控件内容.....160
- 实例 56 在 VB 中运行 FLASH 动画.....162
- 实例 57 显示和隐藏任务条.....164
- 实例 58 热键启动程序.....165

第 6 章 磁盘文件.....169

- 实例 59 获取驱动器信息.....170
- 实例 60 控制光驱开关.....174
- 实例 61 简易资源管理器（方案例）.....176
- 实例 62 获得文件属性.....179
- 实例 63 文件比较.....183
- 实例 64 查找文件（方案例）.....186
- 实例 65 制作 PDF 文档.....193
- 实例 66 拷贝、删除、重命名和移动文件.....197
- 实例 67 自己的翻译软件.....199
- 实例 68 按照 16 进制方式显示文件.....202
- 实例 69 格式化磁盘.....204
- 实例 70 自己的压缩程序.....205

第 7 章 数据库.....209

- 实例 71 输入控制与格式化数据（方案例）.....210

- 实例 72 在数据库中存取图像字段.....213
- 实例 73 获得数据库的信息.....217
- 实例 74 数据库用户管理.....219
- 实例 75 用户的数据库权限管理.....224
- 实例 76 数据库加密与解密.....227
- 实例 77 代码连接到 ODBC.....229
- 实例 78 RDO 方式连接数据库.....231
- 实例 79 DAO 方式数据库示例.....235
- 实例 80 ADO 连接 SQL 数据库函数.....238
- 实例 81 典型的数据查询（方案例）.....240
- 实例 82 增删改典型操作模块（方案例）.....245
- 实例 83 控制表格控件.....249
- 实例 84 列出系统中所有 DSN 和数据库驱动...252
- 实例 85 快速制作报表.....253
- 实例 86 报表输出类型控制.....259
- 实例 87 文本文件转换到数据库（方案例）.....261

第 8 章 网络与通信.....268

- 实例 88 获得网卡物理地址.....269
- 实例 89 获得指定机器的 IP 地址.....272
- 实例 90 创建自己的 Telnet 服务器.....275
- 实例 91 映射和断开网络驱动器.....278
- 实例 92 获得网络邻居所有信息（方案例）.....281
- 实例 93 使用 UDP 协议发送信息.....284
- 实例 94 局域网上发送信息.....285
- 实例 95 网络聊天.....287
- 实例 96 网上活动监视.....291
- 实例 97 电话拨号上网.....294
- 实例 98 给 ICQ 发送信息.....295
- 实例 99 自定义网页浏览器.....297
- 实例 100 获得指定网页源代码.....300

实例 101	获得路由关系.....	301
实例 102	导入导出 IE 收藏夹.....	305
实例 103	收发电子邮件.....	306
实例 104	FTP 下载.....	309
实例 105	端口扫描.....	311

第 9 章 鼠标和键盘.....314

实例 106	鼠标位置跟踪.....	315
实例 107	鼠标控制.....	316
实例 108	模拟鼠标的单双击.....	319
实例 109	动画指针.....	320
实例 110	鼠标记录器.....	322

实例 111	获得键盘信息.....	323
实例 112	键盘记录器.....	325

第 10 章 数学算法与程序发布.....327

实例 113	完整的文本编辑器（方案例）.....	328
实例 114	自己的计算器.....	332
实例 115	俄罗斯方块游戏（方案例）.....	335
实例 116	加密——一种加密的算法.....	339
实例 117	加密——对文本文件的加密.....	342
实例 118	GB 码与 BIG5 码之间的转换.....	343
实例 119	产生程序序列号.....	346
实例 120	操作提示.....	350



第 1 章 界面外观



-  带图标的菜单
-  动画进度条
-  带提示文字的列表框
-  任意形状的窗体
-  无边框的窗体
-  自动排序的列表框
-  自动显示货币格式的文本框
-  文字特效
-  调色程序
-  窗体分割条
-  自动完成文本框的文本输入
-  控制窗体的系统菜单和按钮
-  产生 Windows 任务栏
-  自动隐藏菜单

实例 1 带图标的菜单

实例目的

使用 Visual Basic 创建自己的菜单是一件非常容易的事情，但用户创建的一般都是文字菜单。这个例子将讲解如何加入图标，让菜单更加色彩斑斓，从而使应用程序更有个性。

实现方法

使用 CreateCompatibleDC 函数为图标创建内存 DC (Device Context, 通常叫设备上下文)，然后使用 CreateCompatibleBitmap 函数为图标产生一个大小一致的位图对象，接下来使用 SelectObject 函数为位图对象指定内存 DC。使用 BitBlt 函数将位图信息拷贝到指定的内存 DC 中，通过 GetSubMenu 和 GetMenu 函数获得对菜单的控制，最后将位图信息写到指定的菜单中。

由于 API 函数定义都是采用 C 格式定义的，因此要在 Visual Basic 中使用需要重新定义，代码如下：

```
Private Declare Function GetMenu Lib "user32" (ByVal hwnd As Long) As Long
Private Declare Function GetSubMenu Lib "user32" (ByVal hMenu As Long, ByVal nPos As Long) As Long
Private Declare Function GetMenuItemID Lib "user32" (ByVal hMenu As Long, ByVal nPos As Long) As Long
Private Declare Function ModifyMenu Lib "user32" Alias "ModifyMenuA" (ByVal hMenu As Long, ByVal nPosition As Long, ByVal wFlags As Long, ByVal wIDNewItem As Long, ByVal lpString As Any) As Long
Private Declare Function CreateCompatibleDC Lib "gdi32" (ByVal hdc As Long) As Long
Private Declare Function CreateCompatibleBitmap Lib "gdi32" (ByVal hdc As Long, ByVal nWidth As Long, ByVal nHeight As Long) As Long
Private Declare Function SelectObject Lib "gdi32" (ByVal hdc As Long, ByVal hObject As Long) As Long
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
Private Declare Function DeleteDC Lib "gdi32" (ByVal hdc As Long) As Long
'常量定义
Const SRCCOPY = &HCC0020
Const MF_BITMAP = &H4&
Const MF_BYPOSITION = &H400&
```

程序代码

创建如图 1-1 所示的窗体，其中创建一个简单的菜单，然后加入一个图片控件并放入一个图标（后缀为.ico），具体内容参考源代码。

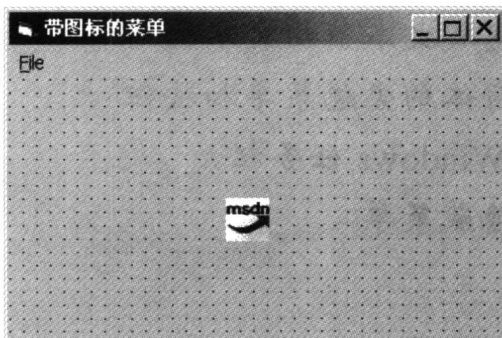


图 1-1 程序的窗体

在窗体的 Load 事件中加入如下代码：

```
Private Sub Form_Load()
    Dim nLoopCtr As Integer
    Dim IResult As Long
```

```

Dim hTempDC As Long
Dim nWidth As Long
Dim nHeight As Long
Dim lTempID As Long
Dim hMenuID As Long
Dim lItemCount As Long
Dim hBitmap As Long
'将图像宽度转化为以像素为单位的宽度
nWidth = Pic.Width \ Screen.TwipsPerPixelX
'将图像高度转化为以像素为单位的高度
nHeight = Pic.Height \ Screen.TwipsPerPixelY
'获得子菜单的标识
hMenuID = GetSubMenu(GetMenu(Me.hwnd), 0)
'为位图创建内存 DC
hTempDC = CreateCompatibleDC(Pic.hdc)
For i = 0 To 1
    '创建位图对象
    hBitmap = CreateCompatibleBitmap(Pic.hdc, nWidth, nHeight)
    '将位图对象指定到内存 DC
    lTempID = SelectObject(hTempDC, hBitmap)
    '拷贝位图内容到内存中
    lResult = BitBlt(hTempDC, 0, 0, nWidth, nHeight, Pic.hdc, 0, 0, SRCCOPY)
    lTempID = SelectObject(hTempDC, lTempID)
    M_Item1(i).Caption = ""
    '修改菜单的内容
    lResult = ModifyMenu(hMenuID, i, MF_BYPOSITION Or MF_BITMAP, GetMenuItemID(hMenuID, i), hBitmap)
Next i
lResult = DeleteDC(hTempDC)
End Sub

```

程序运行的结果如图 1-2 所示。



图 1-2 带图标的菜单

实例 2 动画进度条

实例目的

这个例子讲解如何利用自己创建的控件来制作一个很酷的进度条。使用控件的人都知道，控件实际上是一个创建成功的类。控件使用的过程就是为类创建实例的过程。类包含某些数据结构、函数、方法和属性等，用户创建的控件应同样满足这些要求。

实现方法

创建控件，其中包括所有必需的参数、属性、函数、过程等，然后通过调用这个控件，来实现动画进度条颜色渐变。

程序代码

首先创建控件，如图 1-3 所示，具体属性设置参考源代码。

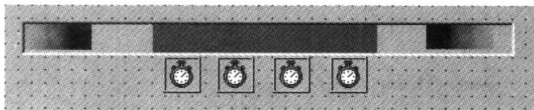


图 1-3 自己创建的控件

控件的代码如下：

```
'位置标志
Private Type flags
    STY As DisplayStyle '显示类型
    LOS As Long          '左边极限位置
    ROS As Long          '右边极限位置
    INC As Integer        '步长
    TMR As Integer        '时间间隔
    EXP As Boolean
End Type
'显示类型
Public Enum DisplayStyle
    Monic = 1
    Pacer = 2
    Bouncy = 3
    Pulse = 4
End Enum
'全局变量
Dim f As flags
'类型属性
Property Let Style(s As DisplayStyle)
    f.STY = s
End Property
'计时属性
Property Let Timing(milsec As Integer)
    f.TMR = milsec
End Property
'步长属性
Property Let Increment(twips As Integer)
    f.INC = twips
End Property
'Resize 过程
Private Sub UserControl_Resize()
    pBar.Width = UserControl.Width - 200
    pBar.Height = UserControl.Height - 200
    If pBar.Height < 315 Then pBar.Height = UserControl.Height - 200 _
    Else pBar.Height = 315
    Call UserControl_Initialize
End Sub
'初始化过程
Private Sub UserControl_Initialize()
    f.EXP = True
    tmrPace.Enabled = False
    tmrMonic.Enabled = False
```

```

tmrBouncy.Enabled = False
tmrPulse.Enabled = False
lblPulse.Visible = False
pLeft.Left = pBar.Left - pLeft.Width - 150
pRight.Left = pBar.Left + pBar.Width - 150
f.LOS = pLeft.Left
f.ROS = pRight.Left
End Sub
'结束显示过程
Sub EndDisplay()
    Call UserControl_Initialize
End Sub
'开始显示过程
Sub BeginDisplay()
    Call UserControl_Initialize
    '缺省设置
    If f.TMR = 0 Then f.TMR = 50
    If f.INC = 0 Then f.INC = 200
    If f.STY = Empty Then f.STY = Monic
    '显示类型选择
    If f.STY = Monic Then
        tmrMonic.Interval = f.TMR
        tmrMonic.Enabled = True
    ElseIf f.STY = Pacer Then
        tmrPace.Interval = f.TMR
        tmrPace.Enabled = True
    ElseIf f.STY = Bouncy Then
        tmrBouncy.Interval = f.TMR
        tmrBouncy.Enabled = True
    Else
        lblPulse.Visible = True
        lblPulse.Width = 0
        tmrPulse.Interval = f.TMR
        tmrPulse.Enabled = True
    End If
End Sub
'从左到右方式的计时器
Private Sub tmrMonic_Timer()
    pLeft.Left = pLeft.Left + f.INC
    If pLeft.Left >= f.ROS Then pLeft.Left = f.LOS
End Sub
'步进计时器
Private Sub tmrPace_Timer()
    If pLeft.Left < f.ROS Then
        pLeft.Left = pLeft.Left + f.INC
    ElseIf pLeft.Left >= f.ROS And pRight.Left > f.LOS Then
        pRight.Left = pRight.Left - f.INC
    Else
        pLeft.Left = f.LOS
        pRight.Left = f.ROS
    End If
End Sub
'单向反弹方式计时器
Private Sub tmrBouncy_Timer()
    If pLeft.Left < f.ROS Then
        pLeft.Left = pLeft.Left + f.INC
    Else
        pLeft.Left = f.LOS
    End If
    If pRight.Left > f.LOS Then

```

```

pRight.Left = pRight.Left - f.INC
Else
    pRight.Left = f.ROS
End If
End Sub
'脉冲方式计时器
Private Sub tmrPulse_Timer()
    If f.EXP = True Then
        lblPulse.Width = lblPulse.Width + f.INC
        lblPulse.Left = pBar.Left + (pBar.Width / 2) - (lblPulse.Width / 2) - 300
        If lblPulse.Width > pBar.Width Then f.EXP = False
    Else
        If lblPulse.Width - f.INC < 0 Then
            f.EXP = True: Exit Sub
        Else
            lblPulse.Width = lblPulse.Width - f.INC
            lblPulse.Left = pBar.Left + (pBar.Width / 2) - (lblPulse.Width / 2) - 300
        End If
    End If
End Sub
End Sub

```

创建程序窗体如图 1-4 所示，加入用户创建的控件及其他控件。

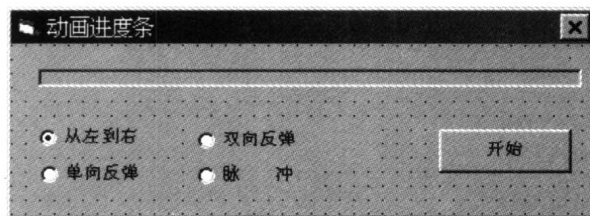


图 1-4 动画进度条主窗体

加入主程序的代码，如下所示：

```

'选项按钮响应事件
Private Sub Option1_Click(Index As Integer)
    If cmd.Caption = "停止" Then
        cmd.Caption = "开始"
        dbar1.EndDisplay
        Call cmd_Click
    End If
End Sub
'选择显示方式
Private Sub cmd_Click()
    If cmd.Caption = "停止" Then
        dbar1.EndDisplay
        cmd.Caption = "开始"
    Else
        If Option1(0).Value Then dbar1.Style = Monic
        If Option1(1).Value Then dbar1.Style = Pacer
        If Option1(2).Value Then dbar1.Style = Bouncy
        If Option1(3).Value Then dbar1.Style = Pulse
        dbar1.BeginDisplay
        cmd.Caption = "停止"
    End If
End Sub

```

实例3 带提示文字的列表框

实例目的

这里将实现一个带提示文字的列表框的例子。有些列表框中文字较多，不能全部显示，可以通过提示文字的方式显示。

实现方法

使用 `GetCursorPos` 函数获得光标在屏幕坐标系中的坐标，然后使用 `ScreenToClient` 函数将屏幕坐标系的坐标转换为指定的客户坐标系的坐标，最后使用 `SendMessage` 发送一个消息到指定的一个或者多个窗体。

由于 API 函数定义都是采用 C 格式定义的，因此要在 Visual Basic 中使用需要重新定义，代码如下：

```
Private Declare Function GetCursorPos Lib "user32" (lpPoint As POINTAPI) As Long
Private Declare Function ScreenToClient Lib "user32" (ByVal hWnd As Long, lpPoint As POINTAPI) As Long
Private Declare Function SendMessage Lib "user32" Alias "SendMessageA" (ByVal hWnd As Long, ByVal wParam As Long, ByVal lParam As Long, ByVal wMsg As Long) As Long
'数据结构
Private Type POINTAPI
    X As Long
    Y As Long
End Type
'消息定义，这个消息的作用是返回当前列表框中与当前光标位置最近的条目的索引，索引值从 0 开始
Private Const LB_ITEMFROMPOINT = &H1A9
```

程序代码

创建如图 1-5 所示的窗体，加入一个列表框控件。



图 1-5 带有提示的窗体

程序代码如下：

```
'载入窗体
Private Sub Form_Load()
    Dim iIndex As Integer
    '在列表框中添加内容
    For iIndex = 1 To 10
        List1.AddItem "从前有一座山，山上游一座庙，庙里有一个和尚，在讲故事。" & iIndex
    Next
End Sub
'鼠标移动
Private Sub List1_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    '当前光标位置的数据结构
    Dim tPOINT As POINTAPI
```

```

Dim iIndex As Long
'获得鼠标光标的位置
Call GetCursorPos(tPOINT)
'转换坐标从屏幕坐标系列列表框相对的坐标
Call ScreenToClient(List1.hWnd, tPOINT)
'判断鼠标移动到距离最近的一个条目的索引值
iIndex = SendMessage(List1.hWnd, LB_ITEMFROMPOINT, 0&, ByVal ((tPOINT.X And &HFF) Or (&H10000 *
(tPOINT.Y And &HFF))))
If iIndex >= 0 Then
    '展开索引值
    iIndex = iIndex And &HFF
    '设置列表框内容为指定条目的内容
    List1.ToolTipText = List1.List(iIndex)
End If
End Sub

```

程序执行结果如图 1-6 所示。

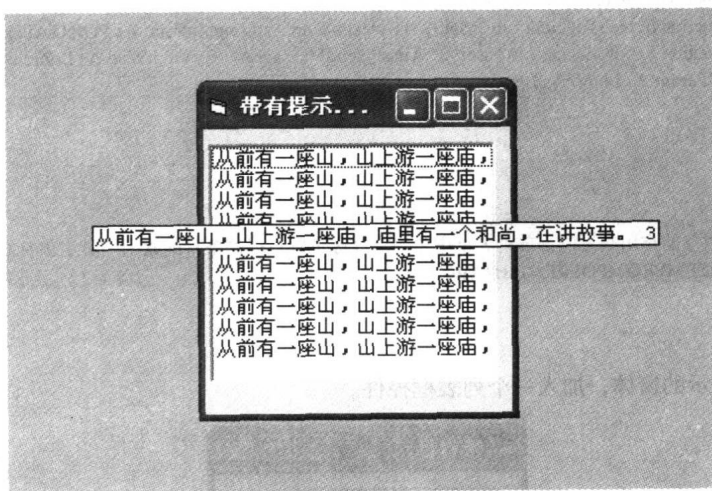


图 1-6 程序执行结果

实例 4 任意形状的窗体

实例目的

一般见到的窗体都是规矩的形状，其实 Windows 应用程序的界面还可以以任何需要的形状出现。这个例子将创建一个与图片形状一致的窗体，这里图片的形状是椭圆形的。

实现方法

首先用把图片放在资源文件 (.res) 中，然后用 LoadResData 函数将图片转化成一个数组，并保存到一个二进制文件中。将窗体的图片属性设置成当前图片。接下来用 ExtCreateRegion 函数获得当前图片的形状（函数返回值为当前图片形状信息），最后用 SetWindowRgn 函数将窗体形状设置为当前图片形状。

下面是程序中使用到的 API 函数在 Visual Basic 中的定义：

```

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 CombineRgn Lib "gdi32" (ByVal hDestRgn As Long, ByVal hSrcRgn1 As Long, ByVal hSrcRgn2

```

As

```

Long, ByVal nCombineMode As Long) As Long
Private Declare Function DeleteObject Lib "gdi32" (ByVal hObject As Long) As Long
Private Declare Function GetRegionData Lib "gdi32" (ByVal hRgn As Long, ByVal dwCount As Long, lpRgnData As Any)
As Long
Private Declare Function ExtCreateRegion Lib "gdi32" (lpXform As Any, ByVal nCount As Long, lpRgnData As Any) As
Long
Private Declare Function SetWindowRgn Lib "user32" (ByVal hWnd As Long, ByVal hRgn As Long, ByVal bRedraw As
Boolean) As Long
Public Declare Function GetPixel Lib "gdi32" (ByVal hdc As Long, ByVal X As Long, ByVal Y As Long) As Long
Public Declare Function GetTempPath Lib "kernel32" Alias "GetTempPathA" (ByVal nBufferLength As Long, _
ByVal lpBuffer As String) As Long
Public Declare Function PathAppend Lib "shlwapi.dll" Alias "PathAppendA" (ByVal pszPath As String, _
ByVal pMore As String) As Long
Declare Function GetOpenFileName Lib "comdlg32.dll" Alias "GetOpenFileNameA" _
(pOpenfilename As OPENFILENAME) As Long

```

程序代码

首先创建一个类，这个类中包括将获得图片信息以及获得的图片形状信息的函数，代码如下：

```

'获得图片的形状信息
Private Function RegionFromResource(ResID As Integer, ResType As String) As Long
    Dim b() As Byte '用来保存图片信息
    Dim dwCount As Long
    '从资源文件中获得图片
    b() = LoadResData(ResID, ResType)
    '获得数组的长度
    dwCount = UBound(b) - LBound(b) + 1
    '获得并返回图片的形状
    RegionFromResource = ExtCreateRegion(ByVal 0&, dwCount, b(0)) 'set region
End Function
'获得图片的图像信息
Private Function JpegFromResource(ResID As Integer, ResType As String) As String
    Dim b() As Byte
    Dim myFile As Long
    Dim tmpFile As String * 260
    '获得一个临时目录
    GetTempPath Len(tmpFile), tmpFile
    PathAppend tmpFile, "resjpg.tmp"
    '将图片的图像信息保存到一个数组中
    b = LoadResData(ResID, ResType)
    '出错将终止程序的运行
    On Error Resume Next
    Kill tmpFile
    '获得一个没有被使用的文件
    myFile = FreeFile
    '打开一个二进制的文件
    Open tmpFile For Binary Access Write As #myFile
    '将数组的信息保存到文件中
    Put #myFile, , b
    '关闭文件
    Close #myFile
    '返回保存图像信息的文件路径
    JpegFromResource = tmpFile
End Function
'设置窗体形状的函数
Public Sub FormShape(frm As Form, ResRgn As Integer, ResJpg As Integer)
    Dim tmpPic As String
    Dim tmpRgn As Long
    '获得图片的图像信息
    tmpPic = JpegFromResource(ResJpg, "JPEG")

```

```
'设置窗体的图像属性为当前图片
frm.Picture = LoadPicture(tmpPic)
'意外捕捉
On Error Resume Next
Kill tmpPic
'获得图片的形状
tmpRgn = RegionFromResource(ResRgn, "RGN")
'设置窗体形状
SetWindowRgn frm.hWnd, tmpRgn, True
```

End Sub

创建主窗体，并加入菜单。创建一个窗体，并把 BorderStyle 属性设置为 0。在菜单中加入如下代码：

```
Private Sub mnuSelect_Click(Index As Integer)
    Dim skn As New clsSkinForm
    Dim sknfrm As New frmSkin
    '根据图片编号显示相应的窗体
    skn.FormShape sknfrm, Index, Index
    sknfrm.Show
End Sub
```

程序执行的结果如图 1-7 所示，4 个不同形状的窗体同时出现。

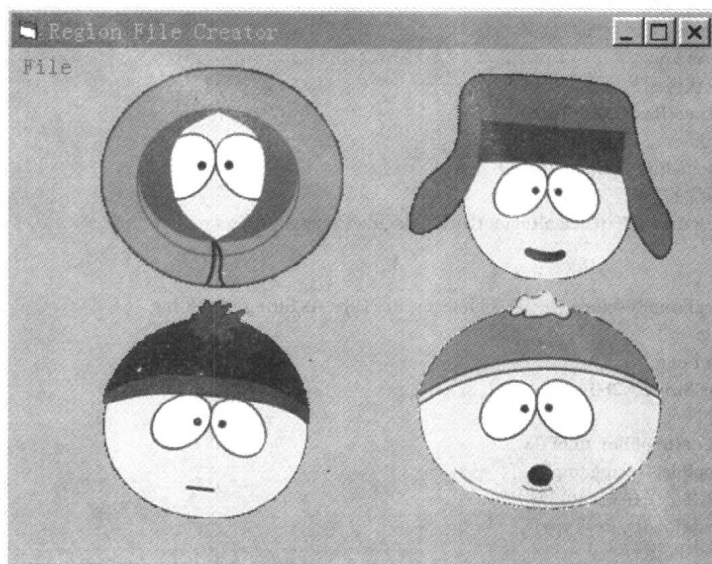


图 1-7 具有图片形状的窗体

实例 5 无边框的窗体

实例目的

这个例子通过创建一个没有边框的窗体，讲解如何控制窗体的边框。

实现方法

在 Visual Basic 中，设置一个窗体的 BorderStyle 属性为 None，可以获得一个没有边框的窗体。但是无法对窗体的行为（如移动、改变大小等）进行控制。这里将通过如下的 API 函数获得对窗体属性的设置和行为的控制：