



求是科技

实效编程百例

Visual Basic 6.0

实效编程百例

★求是科技 刘 韬 骆 娟 编著

- ★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 & TELECOMMUNICATIONS PRESS

实效编程百例

Visual Basic 6.0

实效编程百例

★求是科技 刘 韬 骆 娟 编著

NBJK16/03

人民邮电出版社

图书在版编目 (C I P) 数据

Visual Basic 6.0 实效编程百例/刘韬, 骆娟编著. —北京: 人民邮电出版社, 2002.7
ISBN 7-115-10387-9

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 (2002) 第 042831 号

实效编程百例

Visual Basic 6.0 实效编程百例

- ◆ 编 著 求是科技 刘 韬 骆 娟
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 010-67180876
北京汉魂图文设计有限公司制作
北京顺义向阳胶印厂印刷
新华书店总店北京发行所经销
- ◆ 开本 787×1092 1/16
印张 22
字数 549 千字 2002 年 7 月第 1 版
印数 1-6 000 册 2002 年 7 月北京第 1 次印刷

ISBN7-115-10387-9/TP • 2938

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

内容提要

本书通过 120 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。

前 言

精确掌握编程语言的语法概念并不意味着可以完成功能强大的应用程序。相反，这才是万里长征的第一步。编程水平的提高需要在实际应用中积累点滴经验。本书通过 120 个新颖别致、风格各异的应用实例详细讲解了如何利用 Visual Basic 的强大功能以及 API 函数、第三方控件开发应用程序。希望能为读者提高 Visual Basic 的编程水平有一些帮助。

本书实例的分类按照完成功能来划分，尽量将相关的内容归纳在一章中。通过每章的例子讲解有关界面外观设计、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、数学算法与程序发布等内容，以使读者对计算机应用的各个方面有所了解，本书是广大 Visual Basic 开发和使用人员积累编程经验、拓展视野的得力助手。本书强调实用性，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”，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。

每个实例的讲解分为 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。

本书的主要作者包括刘韬、骆娟等。此外，以下人员也参与了本书的资料搜集和写作工作，他们是刘冰玉、杨柯岚、王保东、杨珏、赵昊彤、程凡、程卫峰、范桂山、莫微、宋征、吴频、邢庆子、闫华文、袁亚玎、岳进、钟明、黄成昆、王远、吴晓超、肖永顺、钱力鹏、骆小来、李岩松、王雅琴、王芳、王国红等。以上人员对本书的顺利完成付出了辛勤的汗水和心血，在此一并表示感谢。

由于时间、水平限制，缺点和不足之处在所难免，敬请读者批评指正。

编者

目 录

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 颜色渐变进度条	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 渐变的窗体背景	13
实例7 透明的按钮	15
实例8 会动的窗体	19
实例9 调色程序	22
实例10 窗体分割条	25
实例11 自动完成文本框的文本输入	29
实例12 控制窗体的系统菜单和按钮	31
实例13 产生 Windows 任务栏	36
实例14 自动隐藏菜单	40
第2章 多媒体控制与图像处理	42
实例15 获得声卡设置	43
实例16 调节系统音量	45
实例17 播放 WAV 和 MIDI 文件	51
实例18 再现 Windows 的 CD 播放机	54
实例19 自己的 MP3 播放器	57
实例20 自己的 VCD 播放器	60
实例21 获得显卡支持的 DirectX 显示模式	63
实例22 屏幕抓图	64
实例23 图片浏览	69
实例24 将彩色位图转化为灰度图	73
实例25 翻转图片	75
第3章 时间控制	78
实例26 设计打字效果	79
实例27 定时计时工具	81
实例28 同步网络时间	83
实例29 日期安排工具	86
实例30 动画文件制作工具(方案例)	91
第4章 操作系统	96
实例31 修改计算机名称	97
实例32 获得 BIOS 信息	97
实例33 编写屏保程序	98
实例34 设置墙纸	101
实例35 调整屏幕分辨率	103
实例36 获取系统硬件信息	105
实例37 获得 CPU 使用情况	108

实例 38	编辑注册表信息	110
实例 39	注销、重启和关闭计算机	115
实例 40	获取 Windows 版本信息及系统目录	116
实例 41	枚举系统可用字体	119
实例 42	打开系统打印机对话框	120
实例 43	打印文件	125
实例 44	改变打印方向	128
第 5 章 程序控制		130
实例 45	密码验证登录 (方案例)	131
实例 46	系统托盘程序	132
实例 47	程序显示在最前面	134
实例 48	隐藏程序不被“关闭程序”发现	135
实例 49	枚举系统中所有进程	136
实例 50	任务管理器	139
实例 51	启动其他应用程序	142
实例 52	调用系统中的一些功能对话框	142
实例 53	多线程方式同时进行多项任务	144
实例 54	建立程序的快捷方式	151
实例 55	得到剪贴板 (clipboard) 上的信息	153
实例 56	把数据写入自身 (exe 文件) 中	154
实例 57	显示和隐藏任务条	158
实例 58	热键启动程序	159
第 6 章 磁盘文件		162
实例 59	获取驱动器信息	163
实例 60	控制光驱开关	167
实例 61	简易资源管理器 (方案例)	169
实例 62	获得文件属性	172
实例 63	文件比较	176
实例 64	查找文件 (方案例)	179
实例 65	清空回收站	186
实例 66	拷贝、删除、重命名和移动文件	187
实例 67	全面控制 INI 文件	189
实例 68	按照 16 进制方式显示文件	193
实例 69	格式化磁盘	196
实例 70	自己的压缩程序	197
第 7 章 数据库		201
实例 71	输入控制与格式化数据 (方案例)	202
实例 72	在数据库中存取图像字段	205
实例 73	获得数据库的信息	209
实例 74	数据库用户管理	211
实例 75	用户的数据库权限管理	216
实例 76	数据库加密与解密	219
实例 77	代码连接到 ODBC	221
实例 78	RDO 方式连接数据库	223
实例 79	DAO 方式数据库示例	227
实例 80	ADO 连接 SQL 数据库函数	230

实例 81	典型的数据查询（方案例）	232
实例 82	增删改典型操作模块（方案例）	237
实例 83	记录的锁定	241
实例 84	表的锁定	243
实例 85	快速制作报表	245
实例 86	报表输出类型控制	250
实例 87	文本文件转换到数据库（方案例）	252
第 8 章	网络与通信	259
实例 88	获得网卡物理地址	260
实例 89	获得指定机器的 IP 地址	263
实例 90	连续批量 Ping 测试	266
实例 91	映射和断开网络驱动器	270
实例 92	获得网络邻居所有信息（方案例）	272
实例 93	使用 UDP 协议发送信息	276
实例 94	局域网上发送信息	277
实例 95	网络聊天	278
实例 96	检测联网信息	283
实例 97	电话拨号上网	284
实例 98	给 ICQ 发送信息	285
实例 99	自定义网页浏览器	287
实例 100	获得指定网页源代码	290
实例 101	获得路由关系	291
实例 102	导入导出 IE 收藏夹	295
实例 103	收发电子邮件	296
实例 104	FTP 下载	299
实例 105	端口扫描	302
第 9 章	鼠标与键盘	305
实例 106	鼠标位置跟踪	306
实例 107	鼠标控制	307
实例 108	模拟鼠标的单双击	310
实例 109	动画指针	311
实例 110	鼠标记录器	313
实例 111	获得键盘信息	314
实例 112	键盘记录器	316
第 10 章	数学算法与程序发布	318
实例 113	进制转换	319
实例 114	制造随机数	322
实例 115	七种常用的排序算法	324
实例 116	加密——一种加密的算法	330
实例 117	加密——对文本文件的加密	333
实例 118	GB 码与 BIG5 码之间的转换	334
实例 119	产生程序序列号	337
实例 120	建立关联文件	341

第1章 界面外观



- ❏ 带图标的菜单
- ❏ 动画进度条
- ❏ 颜色渐变进度条
- ❏ 任意形状的窗体
- ❏ 无边框的窗体
- ❏ 渐变的窗体背景
- ❏ 透明的按钮
- ❏ 会动的窗体
- ❏ 调色程序
- ❏ 窗体分割条
- ❏ 自动完成文本框
- ❏ 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 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，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宽度
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。

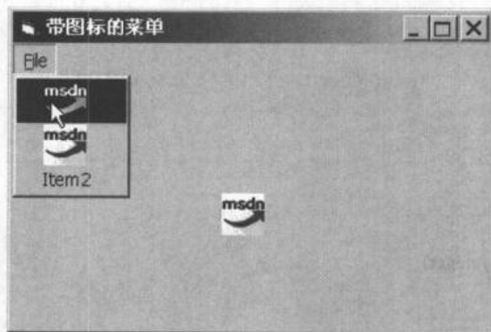


图 1-2 带图标的菜单

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
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
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
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。

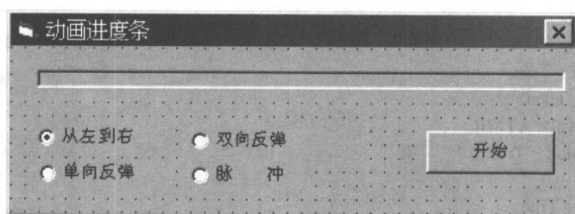


图 1-4 动画进度条主窗体

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 颜色渐变进度条

实例目的

这里将介绍一个简单但是非常实用的进度条。随着程序的进行，进度条的颜色由深变浅，并且用百分比显示进度。

实现方法

使用时间控件（Timer）和进程控件（Progress）。

程序代码

创建如图 1-5 所示的窗体，加入两个时间控件和两个进程控件。

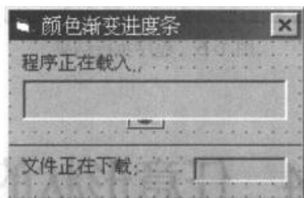


图 1-5 颜色渐变进度条主窗体

程序代码如下：

```
'申明所有变量均需要定义
Option Explicit
'载入窗体
Private Sub Form_Load()
'设置开始百分比为 0
Progress1.Percent = 0
'进程控件 1 开始初始化
Progress1.Refresh
'定义结束时的颜色
Progress1.EndColor = vbBlue
Progress2.EndColor = vbRed
'设置进程控件 2 开始时的百分比
Progress2.Percent = 10
'进程控件 2 开始初始化
Progress2.Refresh
End Sub
'进程控件 1 的时间控制
Private Sub Timer1_Timer()
'进程控件 1 百分比加 1
Progress1.Percent = Progress1.Percent + 1
'重画
Progress1.Refresh
If Progress1.Percent = 100 Then
'设置停止
Timer1.Enabled = False
lblLoading.Caption = "程序载入完成!"
End If
End Sub
'进程控件 2 的时间控制
Private Sub Timer2_Timer()
'进程控件 2 百分比加 1
Progress2.Percent = Progress2.Percent + 1
'重画
```

```
Progress2.Refresh
'更新数据显示
lblUpload.Caption = "文件正在下载: " & Progress2.Percent & "%"
If Progress2.Percent = 100 Then
    '设置停止
    Timer2.Enabled = False
    lblUpload.Caption = "文件下载结束"
End If
End Sub
```

程序执行结果如图 1-6 所示。

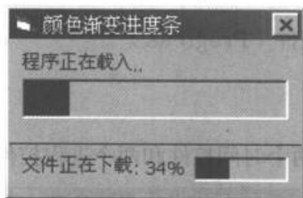


图 1-6 程序执行结果

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
    '返回保存图片信息的文件路径
    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

```