



Visual Basic 6.0/.NET 实例编程宝典



新科教育

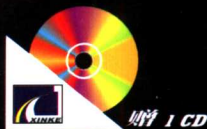
Visual Basic 顶尖高手现身说法
详细的讲解, 深入的剖析
全面掌握创作的观念与技巧
权威的 *Visual Basic* 工具书

新编中文

Visual Basic 6.0/.NET

精彩编程 150 例 I

本书编委会 编



西北工业大学出版社



新编中文
Visual Basic 6.0/.NET
精彩编程 150例 (I)

本书编委会 编

西北工业大学出版社

【内容提要】本书通过 150 个实例由浅入深地讲解了 Visual Basic 6.0/.NET 程序开发的方法和技巧,展示了 Visual Basic 6.0/.NET 中文版的强大功能。全书共分七篇:界面外观、文字及文件处理、时钟控件、图形及图像处理、多媒体及游戏、操作系统和数据库的编程方法和技巧。

本书每个实例都列举了创建的全部过程和完整的代码,因此本书既可作为不同层次培训班的教材,也可供 Visual Basic 6.0/.NET 程序开发的读者参考使用。

为配合读者使用,书中附有程序源代码光盘,读者可以结合所附光盘中的源代码进行学习。

图书在版编目(CIP)数据

新编中文 Visual Basic 6.0/.NET 精彩编程 150 例(I) / 《新编中文 Visual Basic 6.0/.NET 精彩编程 150 例(I)》编委会编. —西安:西北工业大学出版社, 2003.12

ISBN 7-5612-1705-6

I. 新… II. 新… III. BASIC 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2003)第 099822 号

出版发行:西北工业大学出版社

通信地址:西安市友谊西路 127 号

邮编:710072

电话:029-88493844

网 址:www.nwpup.com

印刷者:陕西天元印务有限公司

开 本:787 mm×1 092 mm 1/16


印 张:124

字 数:3 341 千字

版 次:2004 年 7 月第 1 版

2004 年 7 月第 1 次印刷

定 价:150.00 元(共 3 册,本册 38.00 元)(本册含 1CD)



前言

Visual Basic 6.0/.NET 是微软公司推出的可视化编程语言。可视化编程为我们提供了一条便捷之路，它走出了过去的 DOS 下那种既枯燥又复杂的编程方式。

在各种可视化编程语言中，Visual Basic 是最容易学习的一种编程语言。它提供了生成向导、拖放技术、属性检查以及丰富的功能控件，是目前可视化程度最高的开发工具，适于开发任何 Windows 下的应用程序。Visual Basic 6.0/.NET 还提供了强大的实时语法检查功能，在调试运行之前就可以排除许多语法错误，大大提高了编程效率。

然而，编程的学习不是一件很简单的事情。Visual Basic 6.0/.NET 很容易入门，但精通不易。必须通过不断的实践，亲手编写程序，在实践中不断体会，才能深入领悟 Visual Basic 6.0/.NET 的精髓。

本书通过丰富的实例，全面系统地讲解了各种编程技巧。全书共分七个部分：第一篇：界面外观；第二编：文字及文件处理；第三篇：时钟控件；第四篇：图形及图像处理；第五篇：多媒体及游戏；第六篇：操作系统；第七篇：数据库。全书有 150 个经典的实例，系统地介绍了 Visual Basic 6.0/.NET 的编程技巧与使用方法。结构安排合理，由易到难，由简入繁，循序渐进，每个实例都是用“实例说明”+“编程思路”+“操作步骤”的形式过程。

由于作者、编者的水平和经验有限，书中难免存在不足之处，敬请读者批评指正。

本书编委会



02511/06

目 录

第一篇 界面外观.....	1
实例 1 菱形界面	2
实例 2 不规则的动画窗口	4
实例 3 无边框的窗体	9
实例 4 任意形状的窗体	14
实例 5 动态半透明窗体	20
实例 6 渐变颜色的窗体标题栏	24
实例 7 实现渐变实例的背景	41
实例 8 平铺背景	44
实例 9 软件封面	46
实例 10 百叶窗效果	48
实例 11 滚动字幕	51
实例 12 读写器	53
实例 13 滚动显示的窗体背景	55
实例 14 窗体文字效果	58
实例 15 边框调整	60
实例 16 历史记录	63
实例 17 Form 始终放在最前	68
实例 18 表格视窗展示技巧	70
实例 19 将鼠标锁定在指定区域的程序	75
实例 20 将窗口放在屏幕中心	79
实例 21 进度条	82
实例 22 动画进度条	85
实例 23 变颜色的菜单	91
实例 24 图形菜单	93
实例 25 自动隐藏菜单	96
实例 26 显示多列菜单	98
实例 27 在文本框中设置新的系统功能菜单	103
实例 28 智能窗体放大器	105
第二篇 文字及文件处理.....	109
实例 29 字型展示	110

实例 30 特殊文字效果	112
实例 31 在窗体中垂直显示文字	114
实例 32 设计打字效果	120
实例 33 字符串反转	123
实例 34 任意旋转的文本	125
实例 35 自动完成文本框的文本输入	130
实例 36 用手工拖放复制文本	133
实例 37 打开文本文件	136
实例 38 拷贝文件	138
实例 39 文件定位	145
实例 40 列表显示文件	147
实例 41 批量改变多层文件夹中文件的属性	158
实例 42 批量删除某类文件	161
实例 43 撤消上次操作	163
实例 44 判断文件是否被改动	165
实例 45 快速全盘文件查找	167
实例 46 批量重命名文件	171
实例 47 应用资源文件	173
实例 48 记事本（一）	175
实例 49 记事本（二）	180
实例 50 仿 Word 文档（一）	188
实例 51 仿 Word 文档（二）	191
实例 52 插入 BMP 图像	195
实例 53 插入 Word 文档	198
实例 54 插入 Excel 图表	201
第三篇 时钟控件	203
实例 55 时 钟（一）	204
实例 56 时 钟（二）	207
实例 57 电子时钟	210
实例 58 秒表计时器	216
实例 59 台 历	218
实例 60 倒计时牌	220
实例 61 时速表	223
实例 62 雷 达	225

第四篇 图形及图像处理.....	227
实例 63 绘制曲线	228
实例 64 画彩虹	230
实例 65 画工资曲线	232
实例 66 图形处理	236
实例 67 绘制图表	238
实例 68 万花规	240
实例 69 图表控件演示	242
实例 70 温度计	244
实例 71 画 图	247
实例 72 弹动的小球	252
实例 73 位图颜色处理	255
实例 74 设计动画	263
实例 75 太阳、地球和月亮	266
实例 76 图形魔术变换	268
实例 77 获取图像的像素值	271
实例 78 图标显示状态的设置	274
实例 79 雨滴图片	278
实例 80 图片剪切	281
实例 81 图像大小、图像格式的批量转换	284
第五篇 多媒体及游戏.....	287
实例 82 点击即播的多媒体播放器	288
实例 83 CD 播放器（一）	292
实例 84 CD 播放器（二）	294
实例 85 MP3 播放器（一）	301
实例 86 MP3 播放器（二）	304
实例 87 超级 VCD 播放器	307
实例 88 播放 Flash 动画	312
实例 89 播放指定的 AVI-MID-WAV 文件	314
实例 90 播放光驱内所有 VCD 节目	317
实例 91 拖放小游戏	322
实例 92 火焰字	327
实例 93 小蜜蜂射击游戏	335

第六篇 操作系统.....	349
实例 94 获取屏幕上点的颜色	350
实例 95 判断桌面大小	352
实例 96 自动更换墙纸	354
实例 97 设置墙纸	356
实例 98 流星雨屏保	359
实例 99 孔雀开屏式屏保	362
实例 100 逐渐展开式屏保	365
实例 101 模拟的显示器属性	370
实例 102 剪贴板的应用	372
实例 103 Windows 任务栏	375
实例 104 禁止出现“快捷方式”	381
实例 105 在任务栏中禁用右键快捷菜单	383
实例 106 资源管理器	385
实例 107 Windows 控制面板	388
实例 108 判断驱动器属性	391
实例 109 驱动器信息搜索器	393
实例 110 控制光驱的弹开与关闭	396
实例 111 进入 Windows 98 前发出警告	398
实例 112 实现 Windows 的重新启动	400
实例 113 去掉开始菜单中的关闭系统项	402
实例 114 清空回收站	405
实例 115 隐藏、显示开始按钮	407
实例 116 获得 Windows 和 System 路径	409
实例 117 隐藏、显示 Windows 任务栏	411
实例 118 用列表显示系统正在运行的程序	413
实例 119 将程序设置成自动执行的程序	416
实例 120 在不同的模式下实现定时关闭计算机	419
实例 121 启动控制面板	422
实例 122 获得 Windows 98 的注册码	425
实例 123 获取磁盘序列号	428
实例 124 判断键盘按下的键值	430
实例 125 限制鼠标移动区域	432
实例 126 为程序添加快捷键	434
实例 127 隐藏、显示桌面	436

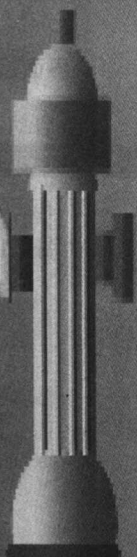
实例 128	获取鼠标在任意点上的位置	438
实例 129	创建、删除桌面快捷方式	440
实例 130	用键盘控制动画	442
实例 131	禁止使用网上邻居	444
实例 132	检测有无映射驱动器	446
实例 133	获取计算机串口的数量	448
实例 134	用屏幕保护设置密码	450
实例 135	用软盘给程序加密码	452
实例 136	如何调用 EXE 文件并使其获得焦点	454
实例 137	获取汉字的拼音简码	456
实例 138	利用 WinRAR 压缩、解压缩文件	461
实例 139	提取并保存 EXE 文件的图标	463
第七篇	数据库	465
实例 140	创建数据库	466
实例 141	数据库的基本操作（一）	470
实例 142	数据库的基本操作（二）	475
实例 143	浏览数据库中所有表	478
实例 144	数据库浏览器	480
实例 145	数据表编辑器	483
实例 146	使用 ADO 控件连接 Access 数据库	485
实例 147	SQL 计数	487
实例 148	DAO 方法	490
实例 149	人员管理系统	495
实例 150	将 Access 数据库转换为 Html 表格文档	498

第一篇

界面外观

Visual Basic 作为一种可视化计算机编程语言，在对用户的界面外观上日渐趋于友好，更充分地发挥了它简捷、易懂的特点。界面的处理简单且功能齐全使其成为界面中的佼佼者。

本篇通过二十多个精典的实例，讲解了在界面外观上的 Visual Basic 处理技巧和方法，使你学完之后真正达到事半功倍的效果。



实例 1 菱形界面

实例说明

本例将制作一个菱形界面。

运行程序后, 单击 **菱形** 按钮, 方形窗体即变成菱形窗体。同时 **菱形** 按钮变成 **恢复** 按钮。如果此时再单击 **恢复** 按钮, 菱形窗体又将变成方形窗体, 如图 1.1.1 所示。

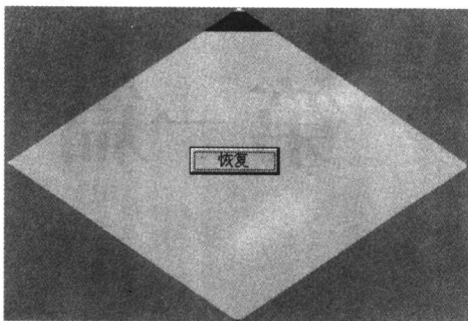


图 1.1.1 效果图

编程思路

变幻的窗体形状经常在图形处理中出现, 这样的界面给大家的一种比较新奇的感觉, 所以对图形处理来说这是一种很特别的界面。

本程序中用了 API 声明的两个函数, 再加上我们非常熟悉的 IF 语句, 便轻松地写出这个简洁的代码。

操作步骤

1. 界面设计

(1) 启动 Visual Basic 6.0, 打开一个新的标准工程。如果 Visual Basic 6.0 已经运行, 单击 **文件(F)** → **新建工程(N)** **Ctrl+N** 命令, 打开一个新的标准工程。

(2) 在 Form1 中添加 1 个命令按钮 (CommandButton), 并将它属性中的 Caption 设置为“菱形”, 其效果如图 1.1.2 所示。

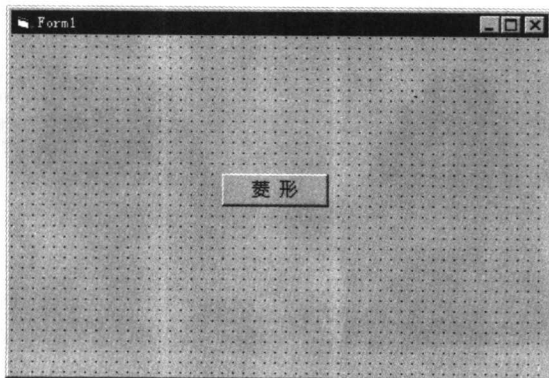


图 1.1.2 窗体界面

2. 代码及注释

Option Explicit

```
Private Declare Function CreatePolygonRgn Lib "gdi32" (lpPoint As POINTAPI, ByVal nCount As _  
Long, ByVal nPolyFillMode 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 Type POINTAPI
```

```
    x As Long
```

```
    y As Long
```

```
End Type
```

```
Private Sub Command1_Click()
```

```
    Dim ap(4) As POINTAPI
```

```
    Dim hx As Integer
```

```
    Dim hy As Integer
```

```
    Dim reRgn As Integer
```

```
    If Command1.Caption = "恢复" Then
```

```
        Command1.Caption = "菱形"
```

```
        SetWindowRgn Me.hWnd, 0, True           '恢复窗口
```

```
    Else
```

```
        Command1.Caption = "恢复"
```

```
        hy = Me.Height \ (2 * Screen.TwipsPerPixelY)
```

```
        hx = Me.Width \ (2 * Screen.TwipsPerPixelX)
```

```
        ap(0).x = 0: ap(0).y = hy                '设置 Polygon 为一个以 Form1 为依据的菱形
```

```
        ap(1).x = hx: ap(1).y = 0
```

```
        ap(2).x = Me.Width \ Screen.TwipsPerPixelX: ap(2).y = hy
```

```
        ap(3).x = hx: ap(3).y = Me.Height \ Screen.TwipsPerPixelY
```

```
        ap(4) = ap(0)
```

```
        reRgn = CreatePolygonRgn(ap(0), 5, 1)
```

```
        SetWindowRgn Me.hWnd, reRgn, True       '设置窗口为菱形
```

```
    End If
```

```
End Sub
```

实例 2 不规则的动画窗口

实例说明

本实例将制作一个运动的椭圆形窗口，而且该窗口没有标准的窗口部件，如标题栏、系统菜单等。程序运行时，窗口上显示一个蓝色的椭圆；用鼠标右键单击窗体，则窗口消失，只有椭圆在不断地变形和运动，此时，若用鼠标右键单击可退回到开始时的情形，只是椭圆的形状已变化了；在椭圆上按下鼠标左键并保持可拖动整个窗口移动，拖动时椭圆停止变形和运动，如图 1.2.1 所示。

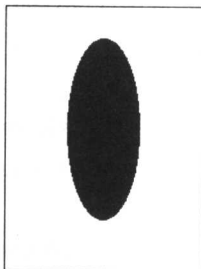


图 1.2.1 效果图

编程思路

本实例主要是通过修改窗口的属性及移动图形来实现效果的。窗口属性的修改以及图形移动的计算过程用 API 函数实现，具体的过程是：首先根据椭圆在窗口中的位置，计算出生成椭圆的矩形的两个端点，然后在坐标系中绘制图形并填充。再根据鼠标按键，确定什么时候取消所有的窗口部件。

操作步骤

1. 界面设计

(1) 启动 Visual Basic 6.0，打开一个新的标准工程。如果 Visual Basic 6.0 已经运行，单击 **文件(F)** → **新建工程(N)** **Ctrl+N** 命令，创建一个新的标准工程。

(2) 新建一个窗体 Form1，将其 Caption 属性设为“不规则的动画窗口”。在窗体 Form1 上添加 1 个时钟控件 (Timer)，将 Interval 属性设置为 1 000，Enabled 属性设置为 False。

(3) 单击 **工程(E)** → **添加模块(M)** 命令，为工程添加一个模块 Module1。

设计好的窗体界面如图 1.2.2 所示。

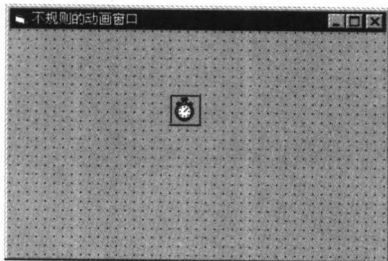


图 1.2.2 窗体界面

2. 代码及注释

(1) 模块 (Module1)。

```

Public Declare Function CreateEllipticRgn Lib "gdi32" (ByVal X1 As Long, ByVal Y1 As _
Long, ByVal X2 As Long, ByVal Y2 As Long) As Long
Public Declare Function SetWindowRgn Lib "user32" (ByVal hwnd As Long, ByVal hRgn As _
Long, ByVal bRedraw As Boolean) As Long
Public Declare Function CreateSolidBrush Lib "gdi32" (ByVal crColor As Long) As Long
Public Declare Function DeleteObject Lib "gdi32" (ByVal hObject As Long) As Long
Public Declare Function FillRgn Lib "gdi32" (ByVal hdc As Long, ByVal hRgn As Long, _
ByVal hBrush As Long) As Long
Public Declare Function GetSystemMetrics Lib "user32" (ByVal nIndex As Long) As Long
Public Declare Function SendMessage Lib "user32" Alias "SendMessageA" (ByVal hwnd As _
Long, ByVal wParam As Long, ByVal lParam As Long, lParam As Any) As Long
Public Declare Function ReleaseCapture Lib "user32" () As Long
Public Type RECT
    Left As Long
    Top As Long
    Right As Long
    Bottom As Long
End Type
Public Type POINTAPI
    X As Long
    Y As Long
End Type
Public Const SM_CYCAPTION = 4
Public Const SM_CXFRAME = 32
Public Const SM_CYFRAME = 33
Public Const WM_NCLBUTTONDOWN = &H81 '用来拖动无标题窗口的 API 函数中的常数
Public Const HTCAPTION = 2
Public scnPts(1) As POINTAPI '保存与椭圆形状有关的点的坐标
Public rgnPts(1) As POINTAPI
(2) 窗体 (Form1)。
Private Static Sub CalcRgnPoints() '计算生成椭圆区域的两点坐标
    Dim offSet As Long
    Dim thEta As Double
    Dim Radius1 As Long
    Dim X1 As Long
    Dim Y1 As Long
    Dim xOff As Long
    Dim yOff As Long
    Const Pi# = 3.14159265358979 '一些常用的常量

```

```

Const DegToRad# = Pi / 180
X1 = Me.ScaleWidth \ 2           '根据窗口的大小计算半径
Y1 = Me.ScaleHeight \ 2
If X1 > Y1 Then
    Radius1 = Y1 * 0.75
Else
    Radius1 = X1 * 0.75
End If
xOff = GetSystemMetrics(SM_CXFRAME) '窗口标题栏和边界的大小产生的偏移量
yOff = GetSystemMetrics(SM_CYFRAME) + GetSystemMetrics(SM_CYCAPTION)
thEta = (60 - offSet) * DegToRad   '根据 offSet 值计算偏角
scnPts(0).X = scnPts(0).X + xOff   '计算填充椭圆的两点
scnPts(0).Y = scnPts(0).Y + yOff
rgnPts(0).X = scnPts(0).X + xOff   '计算设置窗口范围的两点
rgnPts(0).Y = scnPts(0).Y + yOff
thEta = (240 - offSet) * DegToRad
scnPts(1).X = X1 + (Radius1 * (Sin(thEta)))
scnPts(1).Y = Y1 + (Radius1 * (Cos(thEta)))
rgnPts(1).X = scnPts(1).X + xOff
rgnPts(1).Y = scnPts(1).Y + yOff
If rgnPts(0).X = rgnPts(1).Y Then
    rgnPts(1).X = rgnPts(0).X + 5
End If
If rgnPts(0).Y = rgnPts(1).Y Then
    rgnPts(1).Y = rgnPts(0).Y + 5
End If
offSet = (offSet + 2) Mod 360
End Sub
Private Sub Form_Load()
    With Me
        .ScaleMode = vbPixels
        .Move (Screen.Width - .Width) \ 2, (Screen.Height - .Height) \ 2
        .Icon = Nothing
    End With
    Call CalcRgnPoints
    Call Form_Paint
End Sub
Private Sub Form_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
    Dim hRgn As Long

```

```
Static UsingPoly As Boolean
If Timer1.Enabled And Button = 1 Then
    Call ReleaseCapture
    Call SendMessage(Me.hwnd, WM_NCLBUTTONDOWN, HTCAPTION, 0&)
End If
If Button = 2 Then
    UsingPoly = Not UsingPoly
    If UsingPoly Then
        hRgn = CreateEllipticRgn(rgnPts(0).X, rgnPts(0).Y, rgnPts(1).X, rgnPts(1).Y)
        Call SetWindowRgn(Me.hwnd, hRgn, True)
    End If
    Timer1.Enabled = UsingPoly
End If
End Sub
Private Sub Form_Paint()
    Dim hBrush As Long
    Dim hRgn As Long
    hBrush = CreateSolidBrush(vbBlue)
    hRgn = CreateEllipticRgn(scnPts(0).X, scnPts(0).Y, scnPts(1).X, scnPts(1).Y)
    Call FillRgn(Me.hdc, hRgn, hBrush)
    Call DeleteObject(hRgn)
    Call DeleteObject(hBrush)
End Sub
Private Sub Form_Resize()
    With Me
        If .Visible Then
            CalcRgnPoints
            .Refresh
        End If
    End With
End Sub
Private Sub Timer1_Timer()
    Dim nRet As Long
    Dim hRgn As Long
    CalcRgnPoints
    hRgn = CreateEllipticRgn(rgnPts(0).X, rgnPts(0).Y, rgnPts(1).X, rgnPts(1).Y)
    nRet = SetWindowRgn(Me.hwnd, hRgn, True)
End Sub
```


实例 3 无边框的窗体

实例说明

本实例将通过创建一个没有边框的窗体学习如何控制窗体的边框。程序运行效果如图 1.3.1 所示。

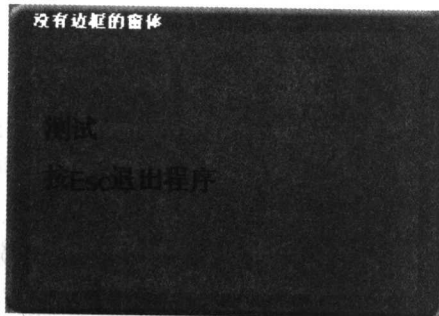


图 1.3.1 效果图

编程思路

在 Visual Basic 中，设置一个窗体的 `BorderStyle` 属性为 `None` 就可以获得一个没有边框的窗体。但是，这个窗体不能对其进行控制，如移动、改变大小等。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。

```
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 Declare Function CreateEllipticRgn Lib "gdi32" (ByVal X1 As Long, ByVal Y1 As Long, ByVal X2 As Long, ByVal Y2 As Long) 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 SetWindowRgn Lib "user32" (ByVal hwnd As Long, ByVal hRgn As Long, ByVal bRedraw As Boolean) As Long
```

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

```
Public Declare Sub ReleaseCapture Lib "user32" ()
```

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

操作步骤

1. 界面设计

(1) 启动 Visual Basic 6.0，打开一个新的标准工程。如果 Visual Basic 6.0 已经运行，单击 **文件(F)** → **新建工程(N)** **Ctrl+N** 命令，打开一个新的标准工程。