

第 3 章 图形图像处理

图形和图像能为应用程序增加良好的视觉效果和艺术美感,对图形和图像的处理也是 Windows 应用程序的主要分支之一。Visual Basic 提供了丰富的图形图像处理功能,包括可以利用绘图函数绘制各种图形、对各种图像进行特效处理、利用控件设计统计图等,通过对本章的学习可以使得用户掌握图形图像处理的基本编程技巧。

实例 3.1 基本绘图板

【目的】

绘图之前需要首先绘制所用坐标系和设置绘制比例,这是绘制图形的基础。另外,Visual Basic 没有提供直接绘制曲线的方法,可以用画点的方法来实现绘制曲线。现在就来介绍 Visual Basic 的坐标系设置、设置绘制比例和绘制曲线的方法。

本例实现了一个绘图板程序,可更改当前坐标系,绘制正弦曲线。通过对本例的学习可了解 Visual Basic 的坐标系统和基本绘图函数。程序运行界面如图 3-1 所示。单击“坐标系”按钮,出现当前系统的 X 轴、Y 轴,单击“正弦曲线”按钮,可绘制正弦曲线。

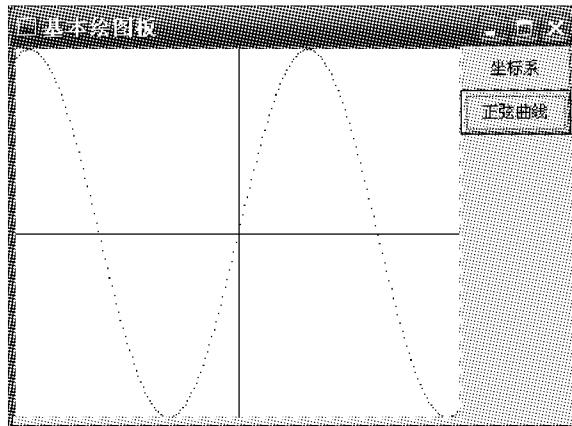


图 3-1 基本绘图板的程序运行界面图

【步骤】

(1) 新建“标准 EXE”工程。

(2) 建立程序用户界面。在窗体上添加 2 个命令按钮,分别修改其 Caption 属性为“坐标系”和“正弦曲线”;添加 1 个图片框。将“正弦曲线”命令按钮的 Enabled 属性设置为 False。

(3) 进入窗体代码窗口,添加如下代码:

```
Private Sub Command1_Click()  
    Picture1.Scale (-5, 5) - (5, -5)    '设置坐标系  
    Picture1.Line (-5, 0) - (5, 0)      '绘制直线,两个括号里的坐标值分别表示直线的两点  
    Picture1.Line (0, 5) - (0, -5)  
    Command2.Enabled=True  
End Sub  
  
Private Sub Command2_Click()  
    Dim i As Single  
    For i=-5 To 5 Step 0.05  
        Picture1.PSet (i, 5 * Sin(i))  
    Next i  
End Sub
```

【相关知识】

1. 坐标系

Visual Basic 的坐标系统是指在屏幕(screen)、窗体(form)、容器(container)上定义的表示图形对象位置的平面二维格线,一般采用数对(x,y)的形式定位。其中,x 值是沿 x 轴点的位置,最左端是默认位置 0。y 值是沿 y 轴点的位置,最上端是默认位置 0。

在 Visual Basic 坐标系中,沿坐标轴定义位置的测量单位,统称为刻度,坐标系统的每个轴都有自己的刻度。坐标轴的方向、起点和刻度都是可变的。

在 Visual Basic 中绘制图形或图像,需要画在一个容器中,在 Visual Basic 中这个容器可以是(窗体)或是 PictureBox(图片框)。并且不论移动还是改变容器的大小,图形或图像都会限制在容器内。

在进行绘图之前,需要首先了解绘图容器的坐标系统,是指在屏幕、窗体、其他容器(container)上定义的表示图形对象位置的平面二维格线,一般采用数对(x,y)的形式定位。

创建图形对象的坐标系统,一般有以下几种方法。

1) 使用系统默认定义

在系统默认状态下,Visual Basic 的坐标系以“缇”(Twips)为单位(1 缇的长度等于 1/1440 英寸、1/567 厘米、1/20 磅)。不过这些值指示的是图形对象打印尺寸的大小,而在计算机屏幕上的物理距离则要根据监视器的大小及分辨率的变化而变化。

2) 选择系统标准刻度

除了默认的 Twips 坐标系外,用户还可以通过对象的 ScaleMode 属性来设置其他的坐标刻度,具体取值如表 3-1 所示。

表 3-1 ScaleMode 属性取值

取值	含 义	说 明
0	User	用户自定义
1	Twips 缁	系统默认设置
2	Point 磅	每英寸约为 72 磅
3	Pixel 像素	像素是监视器或打印机分辨率的最小单位。每英寸中像素的数目由系统设备的分辨率决定
4	Character 字符	打印时,一个字符高为 1/6 英寸,宽为 1/12 英寸
5	Inch 英寸	每英寸为 2.54 厘米
6	Millimeter 毫米	
7	centimeter 厘米	

在上述取值中,除了 0 和 3 以外,其他所有模式都是打印机所打印的单位长度。例如,设某个对象长为 3 个单位,则把 ScaleMode 属性设为 7 时,打印时就是 3 厘米长。

3) 创建自定义坐标系统

当 ScaleMode 属性为 0,即为用户自定义模式时,可设置对象的相应属性来创建所需的坐标系统。这些属性包括: ScaleLeft——设置对象左边距值;ScaleTop——设置对象上边距值;ScaleWidth——设置对象宽度;ScaleHeight——设置对象高度。

例如给出如下设置代码:

```
ScaleLeft=100
ScaleTop=100
ScaleWidth=300
ScaleHeight=200
picture1.ScaleLeft=50
picture1.ScaleTop=50
```

以上代码定义窗体左上角坐标为(100,100),定义窗体内图形对象 picture1 距窗体左边距离为 50,上边距离为 50,ScaleWidth 和 ScaleHeight 语句定义窗体内部宽度的 1/300 为水平坐标单位,当前窗体内部高度的 1/200 为垂直坐标单位。如果窗体的大小以后被调整,这些单位将保持原状。也就是说,ScaleWidth 和 ScaleHeight 是按照对象的内部尺寸来定义单位的,并且这些尺寸不包括边框厚度或菜单标题的高度。ScaleWidth 和 ScaleHeight 是指对象内的可用空间的大小,决定了对象本身的坐标系统。这有别于内部尺寸和外部尺寸(由 Width 和 Height 属性指定)的定义,Width 和 Height 总是按照容器的坐标系统来表示。另外,以上刻度属性都可包括分数,也可是负数。如果将 ScaleWidth 和 ScaleHeight 属性设置为负数即可改变坐标系统的方向。

另外,还可以利用 Scale 方法定义坐标系统,具体格式如下:

```
[Object .] Scale (x1, y1)-(x2, y2)
```

其中, Object 为窗体或图片框名, (x1, y1) 指所规定矩形区域的左上角坐标, (x2, y2) 指所规定区域的右下角坐标。x1 和 y1 的值, 决定了 ScaleLeft 和 ScaleTop 属性的设置值。x2-x1 的差值和 y2-y1 的差值, 分别决定了 ScaleWidth 和 ScaleHeight 属性的设置值。

例如可以定义坐标系如下:

```
Scale (100, 100)-(200, 200)
```

该语句定义等同于以下属性设置:

```
ScaleTop=100:ScaleLeft=100:ScaleWidth=100:ScaleHeight=100
```

在定义其他坐标系后, 如果需要将坐标系统恢复为默认的 Twips 坐标系, 可以使用不含参数的 Scale 方法, 例如语句: picture1. Scale, 将图形对象的坐标系统恢复为默认, 其左上角坐标为 (0, 0)。

2. PSet 方法

PSet 方法用来在相应对象上的某个点设置成某种颜色, 默认颜色为对象的前景颜色, 这时看上去好像在对象上画了个点。具体格式如下:

```
[Object .] PSet [Step] (X, Y) [,Color]
```

其中, Object 为窗体或图片框名, X、Y 所画点表示坐标, Color 表示所画点颜色, Step 是可选项, 如果选用则 (X, Y) 是指相对于 CurrentX、CurrentY 的相对坐标。

【编程要点】

任何曲线都是由一系列的点构成的, 利用循环的方法, 画一系列密集的点, 就可使这些有序的密集的点看起来像一条曲线。本例首先利用 Scale 函数重新定义图片框的坐标系, 然后利用 PSet 函数逐点绘制正弦曲线。根据方程 $y = \sin(x)$, 调用函数 PSet(x, Sin(x))。

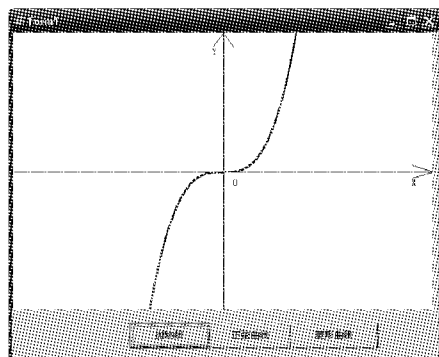
实例 3.2 动态曲线的绘制

【目的】

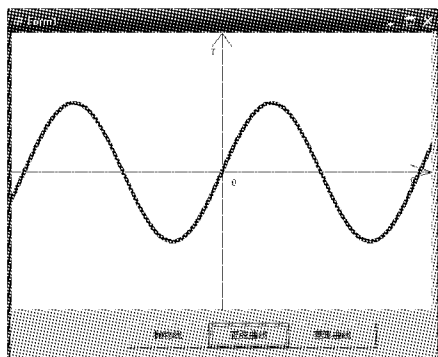
前面已经了解了 Visual Basic 的坐标系和基本的绘制曲线的方法, 现在来为绘制的曲线添加一些动态效果。动态效果的实现主要是结合 For 循环和绘制点的函数, 将步长设置得足够小, 使其在显示时有逐步绘制曲线的效果。

本例是一个函数绘图程序, 用于动态绘制不同函数的曲线。通过对本例的学习可了

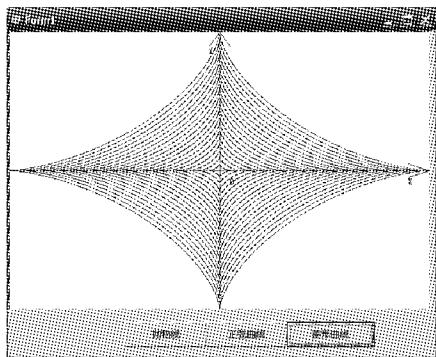
解 Visual Basic 的基本绘图函数。程序运行时如果单击“抛物线”、“正弦曲线”和“菱形曲线”按钮,可显示不同的函数曲线,如图 3-2 所示。



(a) 抛物线运行界面



(b) 正弦曲线运行界面



(c) 菱形曲线运行界面

图 3-2 动态绘制不同函数的曲线界面

【步骤】

- (1) 新建“标准 EXE”工程。
- (2) 建立程序用户界面。在窗体上添加 3 个命令按钮,分别修改其 Caption 属性为“抛物线”、“正弦曲线”和“菱形曲线”,然后添加 1 个图片框。
- (3) 进入窗体代码窗口,添加如下代码:

```
Const pi=3.14159
Dim a
Private Sub Command1_Click()      '绘制抛物线
    init
    Picture1.DrawWidth=2
    For a=-10 To 10 Step pi / 10000
        Picture1.PSet (a, a^3 / 17), vbRed
    Next a
```

```
End Sub

Private Sub Command2_Click()          '绘制正弦曲线
    init
    Picture1.DrawWidth=4
    For a=-7 * pi To 7 * pi Step pi / 8000
        Picture1.PSet (a, Sin(a / 3) * 10), vbRed
    Next

End Sub

Private Sub Command3_Click()          '绘制菱形曲线
    init
    For a=0 To 20 Step 1
        For th=0 To 2 * 3.1415926+0.1 Step 3.1415926 / 1600
            x=a * Cos(th)^3
            y=a * Sin(th)^3
            Picture1.PSet (x, y), vbRed
        Next th
    Next

End Sub

Sub init()                            '初始化子过程,设置并绘制绘图坐标系,设定当前坐标
    Picture1.Cls
    Picture1.ScaleMode=3
    Picture1.Scale (-20, 20)-(20, -20)
    Picture1.DrawWidth=1
    Picture1.Line (-20, 0)-(20, 0), vbBlue
    Picture1.Line (18, 1)-(20, 0), vbBlue
    Picture1.Line - (18, -1), vbBlue
    Picture1.ForeColor=vbBlue
    Picture1.Print "X"
    Picture1.Line (0, 20)-(0, -20), vbBlue
    Picture1.Line (1, 18)-(0, 20), vbBlue
    Picture1.Line - (-1, 18), vbBlue
    Picture1.Print "Y"
    Picture1.CurrentX=1
    Picture1.CurrentY=-1
    Picture1.Print "O"
End Sub
```

【编程要点】

动态绘制效果的实现。本例主要是应用上例讲述的坐标系设置和 PSet 函数来完成

的。自定义过程 init 主要进行初始化处理,使用 Scale 方法设定用户坐标系,将坐标原点设置在 Picture1 中心;绘坐标系的 X、Y 轴及箭头线;指定位置显示原点 O。在绘制曲线的按钮的单击事件中首先调用 init 过程进行初始化,然后用 For 循环绘点,使其按抛物线或正弦规律变化。将步长值设置得很小,使其形成动画效果。

实例 3.3 鼠标拖曳的实现

【目的】

树型控件经常用来表示从属关系,比如 Windows 资源管理器就使用该结构来显示。资源管理器的拖曳属性使用户的移动文件操作很方便,这里利用微软公司提供的树型控件来完成利用鼠标的拖曳,实现项目的移动。

本例使用 Visual Basic 提供的拖曳事件、方法和属性完成鼠标拖曳功能的实现,通过学习本例可了解鼠标拖曳的具体方法,程序运行界面如图 3-3 所示。选择一个子项,按下鼠标左键,拖动鼠标到某个该子项的上一级项目,该子项即被拖动到新选择的上一级项目。

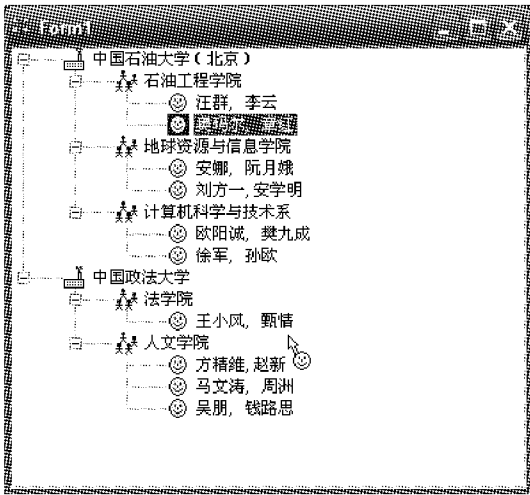


图 3-3 鼠标拖曳的程序运行界面

【步骤】

- (1) 新建“标准 EXE”工程。
- (2) 建立程序用户界面。在窗体上添加 1 个树型控件、1 个 ImageList 控件和 9 个 Image 控件。
- (3) 进入代码窗口中,编写如下事件过程:

```
Option Explicit
Private Enum ObjectType
```

```

        otNone= 0
        otUniv= 1
        otDepartment= 2
        otstudent= 3
        otUniv2= 4
        otDepartment2= 5
        otstudent2= 6
End Enum

Private SourceNode As Object
Private SourceType As ObjectType
Private TargetNode As Object

'返回节点类型
Private Function NodeType(test_node As Node) As ObjectType
    If test_node Is Nothing Then Exit Function
    Select Case Left(test_node.Key, 1)
        Case "u"
            NodeType=otUniv
        Case "d"
            NodeType=otDepartment
        Case "s"
            NodeType=otstudent
    End Select
End Function

' 初始化 ImageList 和 TreeView 控件
Private Sub Form_Load()
    Dim i As Integer
    Dim Univ As Node, Department As Node, Student As Node

    For i=1 To 6
        TreeImages.ListImages.Add , , TreeImage(i).Picture
    Next i
    Tree.ImageList=TreeImages

    ' 创建节点
    Set Univ=Tree.Nodes.Add(, , "u", "中国石油大学(北京)", otUniv, otUniv2)
    Set Department=Tree.Nodes.Add(Univ, tvwChild, "d1", "石油工程学院", otDepartment,
otDepartment)
    Set Student=Tree.Nodes.Add(Department, tvwChild, "s11 ", "汪群, 李云", otstudent,
otstudent2)
    Set Student=Tree.Nodes.Add(Department, tvwChild, "s12", "奥玛尔, 章红", otstu-
dent, otstudent2)

```



```

        Student.EnsureVisible
        Set Department=Tree.Nodes.Add(Univ, tvwChild, "d2", "地球资源与信息学院", otDepartment, otDepartment2)
        Set Student=Tree.Nodes.Add(Department, tvwChild, "s21", "安娜, 阮月娥", otstudent, otstudent2)
        Set Student=Tree.Nodes.Add(Department, tvwChild, "s22", "刘方一, 安学明 ", otstudent, otstudent2)
        Student.EnsureVisible
        Set Department=Tree.Nodes.Add(Univ, tvwChild, "d3", "计算机科学与技术系", otDepartment, otDepartment2)
        Set Student=Tree.Nodes.Add(Department, tvwChild, "s31", "欧阳诚, 樊九成", otstudent, otstudent2)
        Set Student=Tree.Nodes.Add(Department, tvwChild, "s32", "徐军, 孙欧", otstudent, otstudent2)
        Student.EnsureVisible

        Set Univ=Tree.Nodes.Add(, , "u2", "中国政法大学", otUniv, otUniv2)
        Set Department=Tree.Nodes.Add(Univ, tvwChild, "d4", "法学院", otDepartment, otDepartment2)
        Set Student=Tree.Nodes.Add(Department, tvwChild, "s41", "王小风, 甄情", otstudent, otstudent2)
        Student.EnsureVisible
        Set Department=Tree.Nodes.Add(Univ, tvwChild, "d5", "人文学院", otDepartment, otDepartment2)
        Set Student=Tree.Nodes.Add(Department, tvwChild, "s51", "方精维, 赵新 ", otstudent, otstudent2)
        Set Student=Tree.Nodes.Add(Department, tvwChild, "s52", "马文涛, 周洲", otstudent, otstudent2)
        Set Student=Tree.Nodes.Add(Department, tvwChild, "s53", "吴朋, 钱路思", otstudent, otstudent2)
        Student.EnsureVisible

    End Sub

    Private Sub Form_Resize()
        Tree.Move 0, 0, ScaleWidth, ScaleHeight
    End Sub

    Private Sub Tree_MouseDown(Button As Integer, Shift As Integer, x As Single, y As Single)
        Set SourceNode=Tree.HitTest(x, y)
    End Sub

    Private Sub Tree_MouseMove(Button As Integer, Shift As Integer, x As Single, y As Single)

```

```

gle)
    If SourceNode Is Nothing Then Exit Sub
    If Button=vbLeftButton Then
        ' 开始拖曳
        SourceType=NodeType (SourceNode)
        ' 若没有节点高亮显示,则把此节点作为选中节点
        Set Tree.SelectedItem= SourceNode
        ' 设置拖曳图标
        Tree.DragIcon= IconImage (SourceType)
        Tree.Drag vbBeginDrag
    End If
End Sub
'正在拖曳过程中,检查是否有效
Private Sub Tree_DragDrop(Source As Control, x As Single, y As Single)
    If SourceNode Is Nothing Then Exit Sub
    If Not (Tree.DropHighlight Is Nothing) Then
        ' 如果拖曳有效,设置此节点为源节点的父节点
        Set SourceNode.Parent=Tree.DropHighlight
        Set Tree.DropHighlight=Nothing
    End If

    Set SourceNode=Nothing
    SourceType=otNone
End Sub

Private Sub Tree_DragOver(Source As Control, x As Single, y As Single, State As Integer)
    Dim target As Node
    Dim highlight As Boolean

    If SourceNode Is Nothing Then Exit Sub

    ' 找到拖曳经过的节点
    Set target=Tree.HitTest(x, y)
    ' 若与上一次一致,退出
    If target Is TargetNode Then Exit Sub
    Set TargetNode=target
    highlight=False
    If Not (TargetNode Is Nothing) Then
        ' 检查目标节点类型
        If NodeType (TargetNode)+1=SourceType Then highlight=True
    End If
    If highlight Then
        Set Tree.DropHighlight=TargetNode
    Else

```