

实验 7 二次函数曲线演示器

7.1 实验目的

1. 熟悉绘图坐标系统。
2. 掌握绘图语句 Pset、Line、Cls 的使用。

7.2 相关知识

7.1.1 坐标系统

当通过 Visual Basic 作图时，往往要通过一定的能够容纳图形的容器（如：窗体、图片框）并依靠在这些容器上的坐标系统才能表现出来。

坐标系统是一个二维网格，用来定义屏幕上、窗体中或图片框上的位置。例如窗体中的坐标，可以使用(x, y)来表示。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，如图 7-1 所示。其他容器上的坐标系统和窗体类似。实际上，坐标系统不仅用在作图上，它也用来定义窗体和控件在应用程序中的位置。

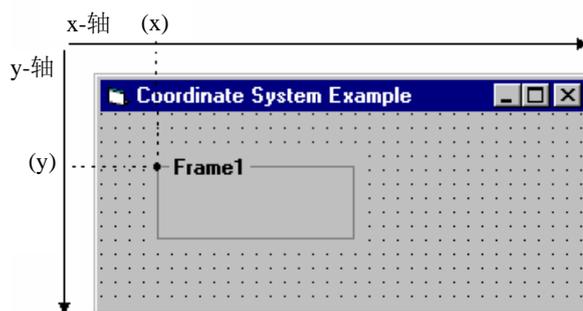


图 7-1 窗体上的坐标系统

7.1.2 绘图方法

在 Visual Basic 中可以使用作图方法构造图形，主要方法有：Cls、Pset、Point、Line、Circle。

1. 使用 PSet 方法画点

语法格式如下：

```
[object.]PSet (x, y) [, color]
```

其中，x 和 y 参数是单精度参数，它们可以接受整数或分数的输入。输入可以是任何数值表达式。color 用来设置所画点的颜色，如果没有包括 color 参数，PSet 将像素设置为前景色 (ForeColor)。

例如，下述语句在代码所连接的当前窗体、MyForm 窗体、picPicture1 图片框上画点。

```
PSet (50, 75), RGB(0, 0, 255)
MyForm.PSet (230, 1000)
picPicture1.PSet (1.5, 3.2)
```

2. 使用 Line 方法画直线

语法格式如下：

```
[object.]Line [(x1, y1)]-(x2, y2) [, color]
```

其中，Object 是可选的，若省略，此方法将在代码所连接的当前窗体上画线。第一对坐标 (x1, y1) 也是可选的，如果省略，将把该对象的 x, y 当前位置 (画图坐标) 作为端点。

例如，以下语句通过三点连接画出一个三角形。

```
CurrentX = 1500           '设置起点的 x 坐标
CurrentY = 500           '设置起点的 y 坐标
Line -(3000, 2000)       '向起点的右下方画一直线
Line -(1500, 2000)       '向当前点的左方画一直线
Line -(1500, 500)        '向右上画一直线到起点
```

3. 使用 Line 方法绘制方框

可以使用 Line 方法绘制并填充方框，这时要用到该方法的 Step 关键字。

例如，下列语句能画出一个边长为 1000 缇的方框。

```
Line (500, 500)-Step(1000, 0)
Line -Step(0, 1000)
Line -Step(-1000, 0)
Line -Step(0, -1000)
```

4. 使用 Circle 方法画圆

使用 Circle 方法可以画出圆形和椭圆形。另外，Circle 方法还可以画出圆弧 (圆的一部分) 和楔形饼块。使用变化的 Circle 方法，可以画出多种曲线。

画一个理想圆的语法如下：

```
[object.]Circle [Step](x, y), radius [, color]
```

其中，x 参数和 y 参数是圆心坐标，radius 参数是圆的半径。

例如，画一个以 (1200,1000) 为圆心、750 为半径的圆。

Circle (1200, 1000), 750

5. 使用 Circle 方法画椭圆

通过圆的方位比，可以决定它是圆还是被拉长的椭圆。

Circle 方法画椭圆的完整语法如下：

```
[object.]Circle [Step](x, y), radius, [color], [start], [end] [, aspect]
```

其中，start 参数和 end 参数是可选的。参数可以省略，但参数之间的逗号不能省略。另外，aspect 参数指定了水平长度和垂直长度比，它是正浮点数。aspect 参数较大时，椭圆沿垂直轴线拉长；aspect 参数较小时，椭圆沿水平轴线拉长。

7.3 实验内容

1. 实验要求

使用绘图语句建立一个演示二次函数曲线的窗体程序，窗体设计如图 7-2 所示。其中，三个文本框控件 txtA、txtB、txtC 用来输入二次函数的参数，图片框 Picture1 用来显示二次函数曲线，命令按钮 cmdXsqx 用来启动二次函数曲线的绘画，命令按钮 cmdCls 用来清除二次函数曲线。



图 7-2 二次函数曲线演示器设计窗体视图

2. 实验过程

- (1) 新建“标准 EXE”工程，将该工程名设置为 EchsYsq。
- (2) 将窗体名设置为 frmEchs。
- (3) 从工具箱内拖入图片框、文本框、标签和命令按钮等控件到 frmEchs 窗体内，并

将控件按照图 7-2 所示的样式排列。

(4) 按照表 7-1 的要求设置控件属性。

表 7-1 frmEchs 窗体中控件属性值设置

对象	类型	属性	设置值	作用
frmEchs	窗体	Caption	二次函数曲线 ($Y=X^2+b*X$)	
Picture1	图片框			显示二次函数曲线
lblX	标签	Caption	100	坐标值
lblY	标签	Caption	100	坐标值
Label1	标签	Caption	Y=	显示函数表达式
Label2	标签	Caption	X^2+	显示函数表达式
Label3	标签	Caption	X+	显示函数表达式
txtA	文本框			输入函数 a 参数
txtB	文本框			输入函数 b 参数
txtC	文本框			输入函数 c 参数
cmdXsqx	命令按钮	Caption	显示函数曲线	启动函数曲线绘画
cmdCls	命令按钮	Caption	清除函数曲线	清除函数曲线

(5) 编写源代码。

(6) 运行、调试源代码。

(7) 保存窗体文件 frmEchs.frm、工程文件 EchsYsq.vbp。

(8) 编译生成可执行程序 EchsYsq.exe。

(9) 运行程序，图 7-3 所示是程序的运行结果。

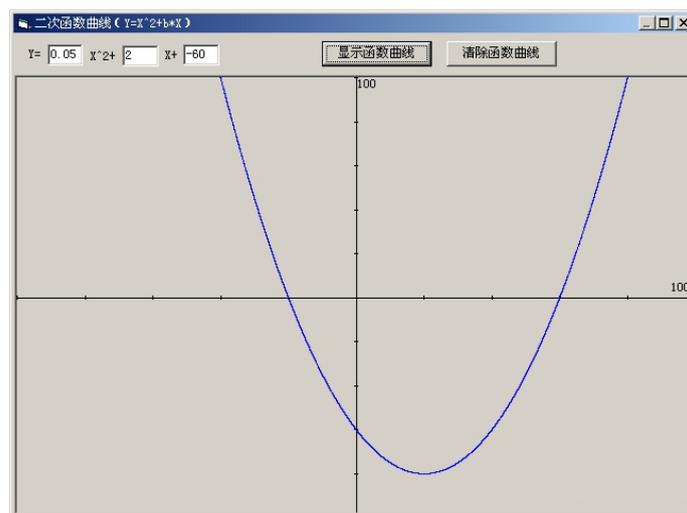


图 7-3 运行中的二次函数曲线演示器

7.4 实验参考源代码

```
Option Explicit
Dim bPat As Byte
Private Sub Form_Load()
    bPat = 0
End Sub

Private Sub Form_Resize()
    If Height < 6000 Then Height = 6000
    If Width < 8000 Then Width = 8000
    Picture1.Move 15, 585, Width - 160, Height - 1000
    lblY.Move Picture1.Width / 2 + 20, 0
    lblX.Move Picture1.Width - lblX.Width - 60, _
    Picture1.Height / 2 - lblX.Height - 60
    '当窗体发生改变时重画坐标、函数曲线
Picture1.Cls
    Pzb
    If bPat Then Pechs
End Sub

Private Sub cmdCls_Click()
    bPat = 0
    Picture1.Cls
    Pzb
End Sub

Private Sub cmdXsqx_Click()
    bPat = 1
    Picture1.Cls
    Pzb
    Pechs
End Sub

Private Sub Pechs()
    '画函数曲线
Dim x As Single, y As Single
Dim px As Long, py As Long
x = -100
Do Until x > 100
    y = Hsz(x)
    px = Picture1.Width - (Picture1.Width / 200) * (x + 100)
    py = Picture1.Height - (Picture1.Height / 200) * (y + 100)
    Picture1.PSet (px, py), RGB(0, 0, 255)
```

```
        x = x + 0.01
    Loop
End Sub

Private Sub Pzb()
    '画坐标
    Dim lnDw As Long
    Picture1.Cls
    Picture1.Line (0, Picture1.Height / 2) - (Picture1.Width, Picture1.Height / 2)
    Picture1.Line (Picture1.Width / 2, 0) -
    - (Picture1.Width / 2, Picture1.Height)
    For lnDw = 0 To Picture1.Height Step Picture1.Height / 10
        Picture1.Line (Picture1.Width / 2 - 30, lnDw) -
        - (Picture1.Width / 2 + 30, lnDw)
    Next

    For lnDw = 0 To Picture1.Width Step Picture1.Width / 10
        Picture1.Line (lnDw, Picture1.Height / 2 - 30) -
        - (lnDw, Picture1.Height / 2 + 30)
    Next
End Sub

Private Function Hsz(x As Single) As Single
    Hsz = Val(txtA) * x ^ 2 + Val(txtB) * x + Val(txtC)
End Function
```