

用 VB 开发图形化组态的编辑界面

郭巧菊 (郑州电力高等专科学校动力系 450004)

王红达 (河南建材研究设计院 450002)

摘要:本文主要介绍了采用 VB 开发图形编辑软件的基本思路及具体实现方法,并提供了主要功能的实现代码供读者使用。

关键词:可视化界面 图形编辑 Toolbar 控件 工具类

一、引言

在当今的软件开发过程中,可视化的编程及应用软件已经成为潮流,很多编程爱好者同样也想在自己开发的软件中能提供类似的图形编辑功能(如画笔工具),作者在开发“控制系统通用仿真软件”时,就实现了用图形化的组态方法来搭建控制系统的功能,下面就本人的开发体会介绍一下用 VB 如何实现图形编辑功能的基本思路。

二、图形组态的界面设计

在进行具有编辑功能的软件开发时,一般主窗体大都采用多文档窗体,再利用两个子窗体分别作为工具窗及编辑窗。三个窗体对象的有关属性设置如下:

对象	属性	设置
MDIForm	Name	frmMain
	Caption	主窗体
	WindowState	Maximized
Form	Name	frmTool
	Caption	工具窗体
	WindowState	normal
	MDIChild	Ture
	BorderStyle	Fixed dialog
Form	Name	frmEdit
	Caption	编辑窗体
	WindowState	normal
	MDIChild	Ture
	BorderStyle	Fixed dialog

多文档窗体作为应用软件的主窗体,可以通过 VB 提供的菜单编辑器工具及 Toolbar 控件,根据需要设计出美观、实用的主菜单及工具栏,通过编写代码使其生效,在此不再做详细介绍。

当运行时,主窗体加载时子窗体同时加载,并初始化各子窗体,代码如下:

```
Sub MDIForm_Load()  
    With frmTool  
        .Top = 0  
        .Left = 0  
        .Show  
    End With  
    With frmEdit  
        .Top = 0  
        .Left = frmTool.Width  
        .Width = frmTool.Width - 200  
        .Height = Height - 500  
        .Show  
    End With  
End Sub
```

三、工具窗体的界面设计

我们在工具窗体中增加一个 Toolbar 控件及 ImageList 控件。利用 Toolbar 控件来做我们的工具箱。Toolbar 控件中包含一个 button 对象集合,利用 button 对象就可创建不同的图形工具。为了保证同一时刻只能选用唯一的图形工具,设计时可将工具按钮(button)的“样式”属性设置成 tbrbuttongroup(“组按钮”),同时还可利用工具按钮(button)的“工具提示文本”属性为不同功能的工具进行文本提示(当鼠标在工具按钮上暂停时显示工具提示文本)。利用 ImageList 控件存放我们所要创建工具的图象标识(位图),并将 ImageList 控件与 Toolbar 控件进行绑定,即将 Toolbar 控件的 ImageList 属性设置为 ImageList 控件名。这样,在设计时就可用其 Index 属性或 Key 属性的值来引用 ImageList 控件的 ListImage 对象了。我们在设计时将其 Index 赋给 Toolbar 控件中对应的 button 对象的 Image 属性,这样在工具箱中就出现了具有位图标识的工具对象。到此,我们就设计好了类似于 VB 中所提供的工具箱式的工具箱。

四、图形编辑的具体实现

为了使同一图形工具能重复选用,且能使其产生的对象具有不同的属性值,需为每一个图形工具定义一个“工具类”,在该类中,为该图形工具定义其所应具有的属性,并且通过动态数组的形式为每一个工具类定义一个对象数组,使得该图形工具选中一次则自动产生一个该类的“新实例”(对象),通过在编辑界面上双击该对象,使其显示其属性窗口,允许用户修改其属性值。下面通过一个简单“画圆/椭圆”工具来具体说明其实现过程。

1. 工具按钮的设计

首先,利用 Windows 提供的画图工具设计一个代表圆或椭圆的位图,将其存盘为 circle.bmp。然后,在 ImageList 控件的“自定义”属性中通过增加图象的方式将该图象增加到 ImageList 控件的 ListImage 对象集合中。最后,在 Toolbar 控件的“自定义”属性中通过增加按钮的方式增加工具按钮,并将其“样式”属性设置成 tbrbuttongroup(“组按钮”),将其“工具提示文本”属性设置为“画圆/椭圆”,将其“图象”属性设置为 ImageList 控件的 ListImage 对象的 Index 值。到此,我们便设计好了“画圆/椭圆”工具的图符按钮,如下图所示。



2. 定义工具类

我们在此为该“画圆/椭圆”工具定义四个基本属性(读者可根据需要定义其他有关属性):CenterX(中心横坐标), CenterY(中心纵坐标), LengthX(横轴长度), LengthY(纵轴长度)。并使该四个属性均可读写。具体实现如下:

首先在 VB 的“工程”菜单中,选择“添加类模块”子菜单,并在其属性窗体中定义其 Name 属性为 clsCircle。然后,为其属性过程编写代码。在编写其属性过程代码时,首先要在其代码声明部分定义私有数据成员,供该类的属性代码中使用。如下所示:

```
'类名称:    clsCircle  
'创建日期:  98-12-25  
'作者:      郭巧菊  
  
Option Explicit  
'私有数据成员  
Private myCenterX As Single
```

```
Private my- CenterY As Single
```

```
Private my- LengthX As Single
```

```
Private my- LengthY As Single
```

下面是定义属性 CenterX 的属性过程程序代码, 其他属性过程与此类似, 在此不再一一列出。

```
'公有属性过程(可写性)
```

```
Public Property Let CenterX (X As Single)
```

```
my- CenterX = X
```

```
End Property
```

```
'公有属性过程(可读性)
```

```
Public Property Get CenterX () As Single
```

```
CenterX = my- CenterX
```

```
End Property
```

3. 图形工具的选中

在工具箱中单击图符按钮时选中该工具。实现该功能的程序代码如下:

```
首先定义模块级变量
```

```
'定义对象数组
```

```
Global myCircle() As clsCircle
```

```
'定义允许画图变量(1 - 允许; 0 - 不许)
```

```
Global AllownDrawPict As Integer
```

```
'定义选中变量(2 - 选中"画圆"工具)
```

```
Global SeltPict As Integer
```

```
'定义选中次数
```

```
Global repNb2 As Integer
```

然后, 编写 Toolbar 控件的 ButtonClick 事件代码, 代码如下:

```
Private Sub Tlbar-ButtonClick(ByVal Button As Com-  
ctlLib.Button)
```

```
AllownDrawPict = 1 '允许画图
```

```
Select Case Button.Key
```

```
Case "circle" '选中"画圆"工具
```

```
SeltPict = 2
```

```
Case - - - '选中其他工具
```

```
- - -
```

```
End Select
```

```
'改变编辑窗体鼠标形状("十字"形)
```

```
FrmEdit.MousePointer = 2
```

```
End Sub
```

4. 图形的绘制

当在工具箱内选中某一画图工具后, 便可在编辑窗体内绘制该类工具所示的图形了, 这时编辑界面的鼠标形状呈“十字”形, 单击编辑窗体便可在编辑界面绘出该类图形。该功能由 frmEdit 对象的MouseDown 事件来实现, 代码如下:

```
Private Sub Form-MouseDown (Button As Integer,  
Shift As Integer, x As Single, y As Single)
```

```
If AllownDrawPict = 1 Then '允许画图
```

```
Select Case SeltPict
```

```
Case 2 '画圆
```

```
RepNb2 = repNb2 + 1
```

```
ReDim myCircle(repNb2)
```

```
Set myCircle(repNb2) As New clsCircle
```

```
frmEdit.Circle (x + 200, y + 200)
```

```
myCircle(repNb2).CenterX = x + 200
```

```
myCircle(repNb2).CenterY = x + 200
```

```
myCircle(repNb2).LengthX = 400
```

```
myCircle(repNb2).LengthY = 400
```

```
Case - - - '选中其它工具
```

```
- - -
```

```
End Select
```

```
End If
```

```
End Sub
```

当鼠标抬起时, 画图结束, 将变量 AllownDrawPict 置 0, 鼠标形状还原, 其他对应属性恢复原状, 具体由 frmEdit 对象的 MouseUp 事件来实现, 代码从略。

5. 对象属性的修改

为了使用户在编辑时能对所绘制的图形对象进行修改, 需要设计一属性窗口, 当在编辑窗体双击某对象时, 该对象被选中, 同时弹出对应属性窗口, 使用户可以修改对象属性。该功能由 frmEdit 对象的 DoubleClick 事件来实现, 代码从略。

(来稿时间: 1999 年 1 月)