

自适应窗体尺寸调整的 VB 程序界面的设计

陈健 (吉林四平师范学院 136000)

凡用过 VB 编写 Windows 应用程序的用户都可能有过这样的经历: 当一个经过精心设计的应用程序运行后, 如果用户重新调整了窗体的大小, 则控制在窗体中的相对位置、控件与窗体的大小比例均会严重失调, 程序的界面变得面目全非。

一个好的 Windows 应用程序的界面, 自适应窗体尺寸改变的能力是必不可少的。笔者在这方面做了一些探讨, 希望能对 VB 编程爱好者提供一些启发和帮助。

1. 按照窗体尺寸缩放比例自动调整控件的大小

窗体和控件的大小由窗体和控件的 Width 属性和 Height 属性确定。所以当用户界面设计完成之后, 窗体及其内部的各控件的 Width、Height 属性便随之确定下来; 从而窗体相对于每一个控件, 它们的宽度之比、高度之比均被确定。如果窗体 Form1 内的一个文本框 Text1 的宽度 (即 Text1 的 Width 属性值) 为 3610, 高度 (即 Text1 的 Height 属性值) 为 1935; 而窗体 Form1 的上述两个值分别为 4890 和 3615, 则它们的宽度之比和高度之比分别为: 3610/4890、1935/3615。当用户在程序启动后调整了窗体的尺寸, 窗体的宽度和高度将分别为 Form1.ScaleWidth、Form1.ScaleHeight; 此时应该按上述比例来调整文本框 Text1 的高度和宽度值。即:

$$\text{(调整后的 Text1 的 Width 属性值)} / (\text{Form1. ScaleWidth}) = 3610/4890$$
$$\text{(调整后的 Text1 的 Height 属性值)} / (\text{Form1. ScaleHeight}) = 1935/3615$$
$$\text{所以调整后的 Text1 的 Width 属性值} = (3610/4890) * \text{Form1. ScaleWidth}$$
$$\text{调整后的 Text1 的 Height 属性值} = (1935/3615) * \text{Form1. ScaleHeight}$$
 对于一般控件来说, 应该有:
$$\text{调整后的控件的 Width 属性值} = (\text{控件原 Width 属性值} / \text{窗体原 Width 属性值}) * \text{窗体. ScaleWidth}$$
$$\text{调整后的控件的 Height 属性值} = (\text{控件原 Height 属性值} / \text{窗体原 Height 属性值}) * \text{窗体. ScaleHeight}$$

按照上述方法确定窗体缩放后控件 Width 和 Height 属性值, 则当窗体尺寸被调整后, 控件的大小将按比例得到相应的调整。

2. 按照窗体尺寸缩放比例自动调整控件在窗体中的相对位置

控件在窗体中的位置由该控件的 Left 和 Top 属性确定。程序启动后如果窗体被缩放, 只要按照缩放的比

例来重新调整窗体内各控件的 Left 和 Top 属性值即可。所以根据 1 中介绍的原理,在窗体被缩放之后,只要按照下面的关系来设置控件的 Left 和 Top 属性值即可。

调整后控件的 Left 属性值 =

(控件原 Left 属性值/窗体原 Left 属性值) * 窗体.ScaleWidth

调整后控件的 Top 属性值 =

(控件原 Top 属性值/窗体原 Top 属性值) * 窗体.ScaleHeight

3. 实现按窗体缩放比例调整控件大小及位置的具体方法

我们知道,Resize 事件是窗体响应的一个事件。当窗体第一次显示或当窗体的状态被改变时(如窗体被最大化、最小化或被还原时)该事件发生。所以只要将对控件大小、位置的调整代码写入窗体的 Resize 事件中便可达到目的。具体可在窗体的 Resize 事件过程中使用对象的 Move 方法:

```
Private Sub Form_Resize()
```

```
Object.move Newleft, Newtop, NewWidth, NewHeight
End Sub
```

其中:

Object 是需要调整大小和位置的任意一个控件的名称:

NewLeft、NewTop、NewWidth 与 NewHeight 分别是窗体缩放后控件应有的新的 Left、Top、Width 与 Height 的属性值,它们确定了当窗体被缩放后控件的大小和位置。NewWidth、NewHeight 的值可由 1 中的两个公式得到;NewLeft、NewTop 的值可由 2 中的两个公式得到。

4. 窗体的多控件的处理

当一个窗体中有多个控件时,如果对每一个控件都用上述方法处理显然比较繁琐。为此,以下给出一个通过编程来处理窗体内多个控件的一般方法(注意:如果窗体内有不可见的控件,则应该在程序中加入相应的代码以避免检查这些控件的尺寸,否则将出错)。

(1)自定义一个数据类型

在标准模块的声明段写入:

```
Type cp
```

```
wp As Single
```

```
hp As Single
```

```
tp As Single
```

```
lp As Single
```

```
End Type
```

(2)定义一个窗体级数组

在窗体模块的声明段写入:

```
Dim ap() As cp
```

(3)定义一个通用过程:

```
Sub ai()
```

```
Dim i As Integer
```

```
For i = 0 To Controls.Count - 1
```

```
With ap(i)
```

```
.wp = Controls(i).Width/Form1.ScaleWidth
```

```
.hp = Controls(i).Height/Form1.ScaleHeight
```

```
.lp = Controls(i).Left/Form1.ScaleWidth
```

```
.tp = Controls(i).Top/Form1.ScaleHeight
```

```
End With
```

```
Next i
```

```
End Sub
```

(4)在窗体的 Load 事件过程中写入代码:

```
Private Sub Form_Load()
```

```
ReDim ap (0 To Controls.Count - 1)
```

```
ai
```

```
End Sub
```

(5)在窗体的 Resize 事件过程中写入代码:

```
Private Sub Form_Resize()
```

```
Dim i As Integer
```

```
For i = 0 To Controls.Count - 1
```

以下三行需在一行内写完

```
Controls(i).Move ap(i).lp * Form1.ScaleWidth, ap
```

```
(i).tp *
```

```
Form1.ScaleHeight, ap(i).wp * Form1.ScaleWidth,
```

```
ap(i).hp *
```

```
Form1.ScaleHeight
```

```
Next i
```

```
End Sub
```

不难看出,上面的代码具有普遍意义:即对于任何 VB 程序,只要将上面给出的代码按照要求写入程序中,则无论该 VB 程序的一个窗体中具有多少控件,均能使窗体内的控件自适应于窗体尺寸缩放,从而解决了控件与窗体在尺寸和位置两方面的比例失调问题。

(来稿时间:1998年9月)