

# 利用 VB 的传址参数制作只读文本框

况正谦 (西北工业大学 710022)

文本框(TextBox, 或称编辑框 Edit)是 Windows 环境中最常用的控件之一,这是用户通过键盘与应用程序接口的重要方式。应用程序可以通过文本框显示文本信息,同时用户还可以通过文本框输入信息。许多应用程序利用能够显示多行文本的文本框来编辑或浏览文本文件。我们常用的记事本程序就用到了文本框。另外许多软件也直接调用记事本程序来显示 Readme 文件。在显示 Readme 文件这样的应用中,往往不希望用户任意改动文件内容。这时就希望文本框能够具有只读属性。标签(Label)作为另一种常用的控件能够显示文本,而且具有只读的特性,但它没有多行属性 Multi-Line,对于较大的文本无法容纳,另外在标签中也无法移动光标,不能选定文本进行拷贝。因此,为文本框增加只读功能非常具有实用价值。

在 Visual Basic 开发环境下,为文本框增加只读特性的思路是把修改文本框内容的输入键屏蔽掉。用户按的键包括:ASCII 字符键和非 ASCII 字符键两类。非 ASCII 字符键包括 Shift 键、Ctrl 键、Alt 键、控制光标移动的上下左右箭头键和 Home、End 键以及其他各种控制键、功能键等等。对于只读文本框来说,需要屏蔽的是所有的 ASCII 字符键、擦除 BackSpace 和删除键 Delete。BackSpace 键属于 ASCII 字符键,而灰键 Delete 和数字键区的白键 Del 属于非 ASCII 字符键。除去这些之外,其他的非 ASCII 字符键都不需要屏蔽。尤其是控制光标移动的键必须保留,否则用户无法在文本框中通过移动光标来滚动文本和选定文本。

文本框 TextBox 有两个事件驱动过程可以用于处理用户按键,一个是 KeyDown,另一个是 KeyPress。KeyDown 过程对所有的按键敏感,而 KeyPress 过程只对 ASCII 字符键敏感。这两个过程的编程接口如下:

```
Private Sub Text1_KeyDown(KeyCode As Integer, Shift As Integer)
```

```
'事件处理代码
```

```
End Sub
```

```
Private Sub Text1_KeyPress(KeyAscii As Integer)
```

```
'事件处理代码
```

```
End Sub
```

KeyDown 过程中,参数 KeyCode 传递用户按键的键码。VB 中的键码不同于键盘扫描码,其定义可以通过 VB 的对象浏览器 Object Browser 在对象库 VBRUN 的 KeyCodeConstants 类中查到。KeyDown 过程的 Shift 参数传递 Shift、Ctrl 和 Alt 键的状态。具体而言,Shift 的最右位(位 0)是否为 1 对应 Shift 键是否按下,右边倒数第二位(位 1)对应 Ctrl 键,右边倒数第三位(位 2)对应 Alt 键。KeyPress 过程的 KeyAscii 参数传递用户按键的 ASCII 码。

在 VB 中,过程和函数的参数传递方式有传值和传址两种。参数前面有关键字 ByVal 的都是传值调用,否则为传址调用。对于传值调用的参数,在对过程中对参数值的修改只在本过程中有效,不会影响调用过程的实际参数值。而传址参数则不同,对传址参数的修改会直接改变调用过程的实际参数。

KeyDown 过程中的参数 KeyCode 和 KeyPress 过程中的参数 KeyAscii 都是传址型参数。因此,可以通过修改这两个参数来屏蔽某些按键。屏蔽的办法是首先判断 KeyCode 和 KeyAscii 是否要屏蔽,如果是则把 KeyCode 和 KeyAscii 修改为 0。0 是无效的键值,修改为 0 后按键就被屏蔽掉了。

如前所述,在只读型文本框中,需要屏蔽的是所有的 ASCII 字符键和灰键 Delete 及数字键区的白键 Del。擦除键 BackSpace 已包括在 ASCII 字符键内了,不需要单独处理。KeyPress 过程对所有的 ASCII 字符键敏感,因此可以在 KeyPress 过程中把接收到的所有 KeyAscii 参数修改为 0,从而屏蔽掉所有的 ASCII 字符键。KeyDown 过程对所有的按键,包括 ASCII 字符键和非 ASCII 字符键敏感,因此在 KeyDown 过程中要首先判断一下:如果键码 KeyCode 对应的是 Delete 键或 Del 键,则把 KeyCode 修改为 0 从而屏蔽掉,如果是其他键则不予处理。Delete 键和 Del 键的键码都是 46。KeyDown 过程和 KeyPress 过程的事件处理代码如下:

```
Private Sub Text1_KeyDown(KeyCode As Integer, Shift As Integer)
```

```
    If KeyCode = 46 Then
```

```
        KeyCode = 0
```

```
    End If
```

```
End Sub
```

```
Private Sub Text1_KeyPress(KeyAscii As Integer)
```

```
    KeyAscii = 0
```

```
End Sub
```

下面给出一个完整的实例,说明如何制作只读文本框。在下例中,窗体 Form1 中有编辑框 Text1 和复选框 Check1 两个控件。Check1 的标题设置为“Read Only”。程序运行时,用户通过标记 Check1 和清除 Check1 的标记来设定和取消 Text1 的只读特性。在 Check1 的事件驱动过程 Click 中,通过修改模块级(module level)变量 ReadOnly 来建立和取消只读标志。Text1 的 KeyDown 和 KeyPress 过程查看 ReadOnly 标志,决定是否屏蔽按键。以下是 Form1 模块的源程序文本:

```
VERSION 5.00
Begin VB.Form Form1
    Caption       = "Form1"
    ClientHeight  = 3195
    ClientLeft    = 60
    ClientTop     = 345
    ClientWidth   = 4680
    BeginProperty Font
        Name       = "MS Sans Serif"
        Size       = 30
        CharSet    = 0
        Weight     = 700
        Underline  = 0   False
        Italic     = 0   False
        Strikethrough = 0  False
    EndProperty
    LinkTopic     = "Form1"
    ScaleHeight   = 3195
    ScaleWidth    = 4680
    StartUpPosition = 3  Windows Default
    Begin VB.CheckBox Check1
        Caption       = "Read Only"
        BeginProperty Font
            Name       = "MS Sans Serif"
            Size       = 8.25
            CharSet    = 0
            Weight     = 400
            Underline  = 0   False
            Italic     = 0   False
            Strikethrough = 0  False
        EndProperty
        Height       = 495
        Left        = 1740
```

```
End Sub

    KeyCode = 0

    End If

End Sub

Private Sub Text1_KeyPress(KeyAscii As Integer)
    KeyAscii = 0
End Sub

Dim ReadOnly As Boolean
'窗体 VB Form1 的属性
'窗体 VB Form1.Enabled = True
'窗体 VB Form1.Visible = True
'窗体 VB Form1.Caption = "Form1"
'窗体 VB Form1.Parent = .Form1.
End

Enum
    Width      = 1212
    Top        = 1380
    Text       = ".15411."
    TabOrder   = 0
    Left       = 1170
    Height     = 702

    Enabled:
        Checked = 0   False
        Name    = 0   False
        Visible = 0   False
        Width   = 700
        Height  = 0
        Size    = 8.25
        Name    = ".15228882411."
    BeginProperty Font
        Name VB TextBox Text1
    End
    Width      = 1212
    Top        = 1080
    TabOrder   = 1
```

以上程序在 Microsoft Windows 95 环境下的 Visual Basic Enterprise Edition Version 5.0 中生成,并在相同环境下调试运行成功。程序中运用的方法在 Windows 3.x 下的低版本 Visual Basic 中同样适用。

在 Delphi 环境下,系统提供的编辑控件 Edit 已经有了只读属性 ReadOnly。只要把该属性从缺省值 False 改为 True 就能实现文本框的只读特性。遗憾的是在 VB 中,直到目前最新的版本 5.0 为止,也没有为文本框控件 TextBox 增加只读属性,用户只能自己想办法解决。但愿在未来新推出的 VB 版本中,微软能够注意到这个小小的但是非常实用的功能。

(来稿时间:1997年10月)