

Visual Basic 5.0

编程实例

戴剑彬 曹琦 冯峰 等编著

计算机编程及制作实例丛书

人民邮电出版社

11/21/12
DJE/1

计算机编程及制作实例丛书

Visual Basic 5.0 编程实例

戴剑彬 曹琦 冯峰 等编著

人民邮电出版社

内 容 提 要

本书是一本专门以实例的方式来介绍 Visual Basic 5.0 编程方法和技巧的图书。书中给出了 45 个有实用价值的编程实例, 这些实例涵盖了 Visual Basic 5.0 各种控件的应用方法。其中, 实例 1~15 主要介绍如何建立一些标准的 EXE 程序; 实例 16~30 主要介绍如何调用 Windows95 的 DLL, 来建立 ActiveX EXE、ActiveX DLL、ActiveX 文档的方法, 如何建立客户 / 服务器应用程序以及进行面向对象的编程; 实例 31~45 是综合应用, 包括如何建立数据库应用程序、媒体播放器程序、Internet 浏览器以及其它应用程序。

本书最大的特点是实用性强, 许多应用程序读者都可以直接引用。

本书可供所有运用 Visual Basic 5.0 编程工具进行编程的人员阅读和使用, 也可以作为相关院校师生的计算机编程教学辅助读物。

3537.5215

计算机编程及制作实例丛书 Visual Basic 5.0 编程实例

◆ 编 著 戴剑彬 曹 琦 冯 峰等
责任编辑 王晓明

◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
北京顺义振华印刷厂印刷
新华书店总店北京发行所经销

◆ 开本: 787 × 1092 1/16
印张: 45.5

字数: 1 155 千字 1998 年 9 月第 1 版
印数: 1 - 6 000 册 1998 年 9 月北京第 1 次印刷

ISBN 7-115-07331-7/TP·811

定价: 67.00 元

丛书前言

随着我国计算机应用的普及与发展,各种不同用途的应用软件不断地被开发出来,并迅速地为人们掌握和使用。目前,除了专业软件开发人员外,许多一般的计算机用户也已经开始自己开发一些应用程序。计算机的应用开发工作包括很多方面,例如:图像处理开发,动画制作、专项应用程序开发、通用应用程序开发、游戏软件的制作等等。开发这些项目需要使用各种各样的开发工具,以便能够大大地缩短开发周期并减少开发费用。因此,很多从事计算机应用开发工作的人员迫切需要了解 and 掌握各种计算机编程和制作的方法。为适应广大读者的这种需求,我们特地组织出版了这套《计算机编程及制作实例丛书》,以各种实例的方式,向读者介绍计算机编程和制作方面的知识。

这套丛书主要是针对应用程序开发者而编写的,力求使读者能够轻松地学习开发的方法并熟练地掌握使用相关的开发工具。对于需要编制应用程序或利用计算机进行开发的用户,这套丛书不仅是一个开发指南,而且还给出了其它图书很少涉及到的开发技巧。通过本丛书中介绍的一个个具体的例子,用户便可以比较容易地掌握各种开发软件的功能和使用技巧。

由于这套丛书是以实例的方式由浅入深地讲述编程和开发内容,所以不仅适用于那些刚刚开始从事编程或开发制作的计算机使用者,而且对已经有较多开发经验的计算机用户也同样有较大的帮助。易学易懂且实用性强是本丛书的特点,书中每一个例子中的程序或方法,读者都可以直接引用,相信会使广大读者受益匪浅。

为把这套丛书编得更好,我们真诚地希望广大读者提出宝贵的意见和建议。

前 言

Visual Basic 5.0是美国微软公司开发的一种可视化编程工具，具有简单易行的特点。使用Visual Basic 5.0，只需要通过鼠标在屏幕上的拖动操作，就能完成一些高级的Windows应用程序的设计和开发，例如数据库、多媒体、Internet等等。

本书的目的并不仅仅在于将读者引入Visual Basic 5.0的精彩世界，更主要的是向读者提供一条用Visual Basic 5.0实现复杂应用的途径。本书提供了许多精彩的、全新的并可立即使用的应用实例，它们几乎涵盖了Visual Basic 5.0各种控件的应用。书中结合这些实例，不仅讲述各种控件的用法，还对如何建立一个标准的EXE、ActiveX EXE、ActiveX DLL以及ActiveX控件进行了详细的阐述。本书还向读者介绍了大量的编程技巧，因此特别适合那些从事计算机应用程序设计和开发的人员阅读使用，读者可以参考书中提供的代码并可直接引用，这样可以使读者尽快建立起自己的Visual Basic 5.0应用程序。

本书内容安排由浅入深，前15个例子基本上是Visual Basic 5.0的一些标准控件的应用，主要是介绍如何用Visual Basic 5.0建立一些标准的EXE程序，并熟悉Visual Basic 5.0的常用控件的用法。中间15个例子属于Visual Basic 5.0的高级应用，其中包括如何调用Windows95的DLL，建立ActiveX EXE、ActiveX DLL、ActiveX 文档的方法，如何用Visual Basic 5.0建立客户/服务器应用程序，用Visual Basic 5.0进行面向对象编程，以及优化所编写的Visual Basic代码，使其达到最高效率。最后15个例子是具体的应用，向读者展示了Visual Basic 5.0缤纷多彩的应用领域，让读者能建立自己的数据库应用程序、媒体播放器、Internet浏览器以及在其他方面的应用。这样的编排相信从Visual Basic 5.0的初学者到经验丰富的高级程序员都能从本书中获益。

本书的所有代码都经过详细测试，运行环境是Windows95的Visual Basic 5.0。为方便读者引用相关程序代码，本书附有全部实例程序的磁盘供使用。

参加本书的编写并提供例子的人员包括：戴剑彬、冯峰、蔡晓、王雪乾、张东江、曹琦、程金璞、刘力、李宝胜、张彩彬、张海慧、于英、刘海涛、高华、周宏宝、杜书远、丁永春、孙云、刘一江、王飞、陈洁、李旭群、林以真、阎辉、张毅、孙源、李勇勇、杨洁、白建军、曾东、杨彬、赵栋伟、王宏军、梁辉、赵宇明、程卫国、缪银昌、李古暄和刘鹏。

本书由姜汉卿、王爱民等人审校。

由于时间仓促，加上作者本身的水平有限，书中的错误之处在所难免，在这里我们真诚希望广大读者予以指正。

作 者

1998年7月

目 录

实例 1	单文档界面.....	1
实例 2	多文档界面.....	17
实例 3	鼠标应用	36
实例 4	标准控件	46
实例 5	CommonDialog 控件	62
实例 6	MSFlexGrid 控件演示.....	72
实例 7	DBGrid 控件的非绑定模式例程.....	84
实例 8	ListBox 和 ComboBox 的数据绑定	95
实例 9	OLE 容器控件演示	102
实例 10	PictureClip 控件	109
实例 11	DataTree	116
实例 12	SysInfo 控件	136
实例 13	RTF 文本编辑器	143
实例 14	控件组装程序.....	151
实例 15	工具栏图标管理.....	171
实例 16	错误处理	182
实例 17	DLL 调用	190
实例 18	Active DLL 例程	211
实例 19	获取系统信息.....	230
实例 20	选项卡设置	243
实例 21	创建自己的集合类.....	264
实例 22	缓冲池管理器.....	294
实例 23	优化代码	311
实例 24	Active EXE 例程	346
实例 25	ActiveX 文档创建.....	370
实例 26	对象多态性实例.....	390
实例 27	引发并处理类中事件例程.....	413
实例 28	客户/服务器之间的呼叫/响应	427
实例 29	友元属性和方法例程.....	438
实例 30	网络浏览器	451
实例 31	秒表	473
实例 32	计算器	480
实例 33	调色板应用程序.....	486

实例 34	ATM 自动取款机	493
实例 35	地理信息处理程序	502
实例 36	文件搜索器	512
实例 37	文献数据库浏览器	522
实例 38	MCI 设备管理程序	535
实例 39	VCR——模拟录像机	549
实例 40	图像演示	561
实例 41	VB 终端	575
实例 42	图书销售	599
实例 43	Microsoft Jet 访问 ODBC 数据例程	619
实例 44	DAO 访问数据实例	655
实例 45	电子邮件	692

实例1 单文档界面



主要内容

? 单文档界面的文字编辑器

建立一个单文档界面(SDI)应用程序。用此编辑器可以编辑、打开和保存一个文本文件。

? Visual Basic 的界面样式

用户界面样式主要有两种：单文档界面(SDI)和多文档界面(MDI)。SDI界面的一个例子是Microsoft Windows中的WordPad(记事本)应用程序。在WordPad中，只能打开一个文档，想要打开另一个文档时，必须先关上已打开的文档。

像Microsoft Excel和Microsoft Word for Windows这样的应用程序就是MDI界面；它们允许同时显示多个文档，每一个文档都显示在自己的窗口中。按照是否包含有能在窗口或文档之间进行切换的子菜单的Windows菜单项，就能识别MDI应用程序。

要决定哪种界面样式最好，需要看应用程序的目的。一个处理保险索赔的应用程序可能要用多文档界面MDI样式，一个职员很可能会同时处理一个以上索赔，或者需要对两个索赔进行比较。另一方面，日历程序最好设成SDI(单文档界面)样式，因为不大需要同时打开一个以上日历；在极少情况下，可以再打开一个单文档界面应用程序的实例。

SDI样式更常用，大多数实例都采用SDI应用程序。关于创建MDI应用程序有些需单独考虑的问题和技巧，在下面一个例子“多文档界面实例”中将进一步介绍设计方法。



实例过程

✎ 建立窗体 frmSDI

frmSDI窗体是SDI记事本实例程序界面，有以下四个菜单：

- “文件”，包括“新建”、“打开”、“保存”、“另存为”和“退出”五个菜单项。
- “编辑”，包括“剪切”、“复制”、“粘贴”、“删除”、“全选”和“时间/日期”六个菜单项。
- “搜索”，包括“查找”和“查找下一个”两个菜单项。
- “选项”，包括“工具栏”、“字体”和“创建新实例”三个菜单项。

添加各菜单项，然后添加ToolBar控件picToolBar以及CommonDialog控件CMDialog1。一个建立好的SDI记事本应用程序窗体如图1-1所示。

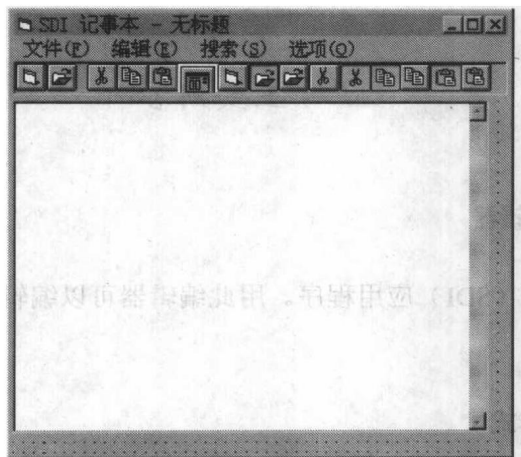


图 1-1 SDI 记事本应用程序窗体

添加各菜单项和程序加载时的代码

这一部分代码主要实现frmSDI窗体的各个菜单项及工具栏的功能。

Option Explicit

Private Sub Form_Load()

 '计数器变量。

 Dim i As Integer

 '应用程序从这里开始(启动窗体的Load事件)。

 Show

 '总是将工作目录设到应用程序所在目录。

 ChDir App.Path

 FState.Dirty = False

 '读系统注册表并适当地设置最近使用的菜单文件列表控件数组。

 GetRecentFiles

 '设置决定函数FindIt搜索方向的全局变量gFindDirection。

 gFindDirection = 1

 '将第一种字体名赋值给“字体”菜单项，再循环系统字体集合，添加到菜单。

 mnuFontName(0).Caption = Screen.Fonts(0)

 For i = 1 To Screen.FontCount - 1

 Load mnuFontName(i)

 mnuFontName(0).Caption = Screen.Fonts(i)

 Next

End Sub

Private Sub Form_QueryUnload(Cancel As Integer, UnloadMode As Integer)

 Dim strMsg As String

 Dim strFilename As String

 Dim intResponse As Integer

 '判断文本是否改变。

 If FState.Dirty Then

 strFilename = Me.Caption

```

strMsg = "[" & strFilename & "]" 中的文本已改变"
strMsg = strMsg & vbCrLf
strMsg = strMsg & "希望保存吗? "
intResponse = MsgBox(strMsg, 51, frmSDI.Caption)
Select Case intResponse
    Case 6                                '用户选择“是”。
        If Left(Me.Caption, 3) = "无标题" Then
            '文件尚未保存。
            strFilename = "无标题.txt"
            '获得trFilename，并调用保存过程和GetstrFilename。
            strFilename = GetFileName(strFilename)
        Else
            '窗体标题包含打开的文件名。
            strFilename = Me.Caption
        End If
        '调用保存过程。
        If strFilename <> "" Then
            '如果strFilename=Empty，用户在“另存为”对话框中选择“取消”；
            SaveFileAs strFilename
        End If
        '否则保存文件
    Case 7                                '用户选择“否”，卸载文件。
        Cancel = False
    Case                                  '用户选择“取消”，取消卸载。
        Cancel = True
End Select
End If
End Sub

Private Sub Form_Resize()
    '调用调整大小过程。
    ResizeNote
End Sub

Private Sub Form_Unload(Cancel As Integer)
    '调用当前最近使用的文件列表过程。
    GetRecentFiles
End Sub

Private Sub imgCopyButton_Click()
    '刷新图像。
    imgCopyButton.Refresh
    '调用复制过程。
    EditCopyProc
End Sub

Private Sub imgCopyButton_MouseDown(Button As Integer, Shift As Integer, X As Single, Y _
    As Single)
    '显示按下状态时图片。
    imgCopyButton.Picture = imgCopyButtonDn.Picture
End Sub

Private Sub imgCopyButton_MouseMove(Button As Integer, Shift As Integer, X As Single, _
    Y As Single)
    '如果按下按钮，判断鼠标是否在按钮上。
    Select Case Button
    Case 1
        If X <= 0 Or X > imgCopyButton.Width Or Y < 0 Or Y > imgCopyButton.Height Then

```

```
        '否，则显示弹起时图片。
        imgCopyButton.Picture = imgCopyButtonUp.Picture
    Else
        '是，则显示按下时图片。
        imgCopyButton.Picture = imgCopyButtonDn.Picture
    End If
End Select
End Sub

Private Sub imgCopyButton_MouseUp(Button As Integer, Shift As Integer, X As Single, Y As Single)
    '显示弹起状态时图片。
    imgCopyButton.Picture = imgCopyButtonUp.Picture
End Sub

Private Sub imgCutButton_Click()
    '刷新图像。
    imgCutButton.Refresh
    '调用剪切过程。
    EditCutProc
End Sub

Private Sub imgCutButton_MouseDown(Button As Integer, Shift As Integer, X As Single, Y _
    As Single)
    '显示按下状态时图片。
    imgCutButton.Picture = imgCutButtonDn.Picture
End Sub

Private Sub imgCutButton_MouseMove(Button As Integer, Shift As Integer, X As Single, Y _
    As Single)
    '如果按下按钮，判断鼠标是否在按钮上。
    Select Case Button
    Case 1
        If X <= 0 Or X > imgCutButton.Width Or Y < 0 Or Y > imgCutButton.Height Then
            '否，则显示弹起时图片。
            imgCutButton.Picture = imgCutButtonUp.Picture
        Else
            '是，则显示按下时图片。
            imgCutButton.Picture = imgCutButtonDn.Picture
        End If
    End Select
End Sub

Private Sub imgCutButton_MouseUp(Button As Integer, Shift As Integer, X As Single, Y As Single)
    '显示弹起状态时图片。
    imgCutButton.Picture = imgCutButtonUp.Picture
End Sub

Private Sub imgFileNewButton_Click()
    '刷新图像。
    imgFileNewButton.Refresh
    '调用新建文件过程。
    FileNew
End Sub

Private Sub imgFileNewButton_MouseDown(Button As Integer, Shift As Integer, X As Single, Y _
    As Single)
    '显示按下状态时图片。
    imgFileNewButton.Picture = imgFileNewButtonDn.Picture
End Sub
```

```

Private Sub imgFileNewButton_MouseMove(Button As Integer, Shift As Integer, X As Single, Y _
    As Single)
    '如果按下按钮，判断鼠标是否在按钮上。
    Select Case Button
    Case 1
        If X <= 0 Or X > imgFileNewButton.Width Or Y < 0 Or Y > imgFileNewButton.Height Then
            '否，则显示弹起时图片。
            imgFileNewButton.Picture = imgFileNewButtonUp.Picture
        Else
            '是，则显示按下时图片。
            imgFileNewButton.Picture = imgFileNewButtonDn.Picture
        End If
    End Select
End Sub

Private Sub imgFileNewButton_MouseUp(Button As Integer, Shift As Integer, X As Single, Y _
    As Single)
    '显示弹起时图片。
    imgFileNewButton.Picture = imgFileNewButtonUp.Picture
End Sub

Private Sub imgFileOpenButton_Click()
    '刷新图像。
    imgFileOpenButton.Refresh
    '调用文件打开过程。
    FileOpenProc
End Sub

Private Sub imgFileOpenButton_MouseDown(Button As Integer, Shift As Integer, X As Single, _
    Y As Single)
    '显示按下时图像。
    imgFileOpenButton.Picture = imgFileOpenButtonDn.Picture
End Sub

Private Sub imgFileOpenButton_MouseMove(Button As Integer, Shift As Integer, X As Single, _
    Y As Single)
    '如果按下按钮，判断鼠标是否在按钮上。
    Select Case Button
    Case 1
        If X <= 0 Or X > imgFileOpenButton.Width Or Y < 0 Or Y > imgFileOpenButton.Height Then
            '否，则显示弹起时图片。
            imgFileOpenButton.Picture = imgFileOpenButtonUp.Picture
        Else
            '是，则显示按下时图片。
            imgFileOpenButton.Picture = imgFileOpenButtonDn.Picture
        End If
    End Select
End Sub

Private Sub imgFileOpenButton_MouseUp(Button As Integer, Shift As Integer, X As Single, Y _
    As Single)
    '显示弹起状态时图片。
    imgFileOpenButton.Picture = imgFileOpenButtonUp.Picture
End Sub

Private Sub imgPasteButton_Click()
    '刷新图像。
    imgPasteButton.Refresh
    '调用粘贴过程。

```

```
    EditPasteProc
End Sub

Private Sub imgPasteButton_MouseDown(Button As Integer, Shift As Integer, X As Single, Y _
    As Single)
'显示按下状态时图片。
    imgPasteButton.Picture = imgPasteButtonDn.Picture
End Sub

Private Sub imgPasteButton_MouseMove(Button As Integer, Shift As Integer, X As Single, Y _
    As Single)
'如果按下按钮，判断鼠标是否在按钮上。
    Select Case Button
    Case 1
        If X <= 0 Or X > imgPasteButton.Width Or Y < 0 Or Y > imgPasteButton.Height Then
            '否，则显示弹起时图片。
            imgPasteButton.Picture = imgPasteButtonUp.Picture
        Else
            '是，则显示按下时图像。
            imgPasteButton.Picture = imgPasteButtonDn.Picture
        End If
    End Select
End Sub

Private Sub imgPasteButton_MouseUp(Button As Integer, Shift As Integer, X As Single, Y As Single)
'显示弹起状态时图片。
    imgPasteButton.Picture = imgPasteButtonUp.Picture
End Sub

Private Sub mnuEditCopy_Click()
'调用复制过程。
    EditCopyProc
End Sub

Private Sub mnuEditCut_Click()
'调用剪切过程。
    EditCutProc
End Sub

Private Sub mnuEditDelete_Click()
'如果鼠标指针不在记事本末尾。
    If txtNote.SelStart <> Len(Screen.ActiveControl.Text) Then
        '如果选定为空，扩展选定长度为1。
        If txtNote.SelLength = 0 Then
            txtNote.SelLength = 1
            '如果鼠标指针在空行上，扩展选定长度为2。
            If Asc(txtNote.SelText) = 13 Then
                txtNote.SelLength = 2
            End If
        End If
        '删除选定文本。
        txtNote.SelText = ""
    End If
End Sub

Private Sub mnuEditPaste_Click()
'调用粘贴过程。
    EditPasteProc
End Sub
```

```

Private Sub mnuEditSelectAll_Click()
    '使用SelStart和SelLength选定文本。
    txtNote.SelStart = 0
    txtNote.SelLength = Len(txtNote.Text)
End Sub

Private Sub mnuEditTime_Click()
    '在光标处插入当前时间和日期。
    txtNote.SelText = Now
End Sub

Private Sub mnuFileExit_Click()
    '结束应用程序。
    Unload Me
End Sub

Public Sub mnuFileNew_Click()
    '调用新建窗体过程。
    FileNew
End Sub

Private Sub mnuFileOpen_Click()
    '调用文件打开过程。
    FileOpenProc
End Sub

Private Sub mnuFileSave_Click()
    '调用文件保存过程。
    FileSave
End Sub

Private Sub mnuFileSaveAs_Click()
    Dim strSaveFileName As String
    Dim strDefaultName As String
    '将窗体标题赋值给变量。
    strDefaultName = Right$(Me.Caption, Len(Me.Caption) - 10)
    '文件尚未保存。
    If Me.Caption = "SDI 记事本 - 无标题" Then
        '获取文件名并调用保存过程和strSaveFileName。
        strSaveFileName = GetFileName("无标题.txt")
        If strSaveFileName <> "" Then SaveFileAs (strSaveFileName)
        '更新“文件”菜单控件数组中最近打开的文件列表。
        UpdateFileMenu (strSaveFileName)
    Else
        '窗体标题包含打开的文件名。
        strSaveFileName = GetFileName(strDefaultName)
        If strSaveFileName <> "" Then SaveFileAs (strSaveFileName)
        '更新“文件”菜单控件数组中最近打开的文件列表。
        UpdateFileMenu (strSaveFileName)
    End If
End Sub

Private Sub mnuFontName_Click(Index As Integer)
    '将选定字体赋值给文本框fontname属性。
    txtNote.FontName = mnuFontName(Index).Caption
End Sub

Private Sub mnuOptions_Click()

```

```
'切换Checked属性来匹配Visible属性。
mnuOptionsToolbar.Checked = picToolbar.Visible
End Sub

Private Sub mnuOptionsLaunch_Click()
    Dim strApp As String
    '创建记事本的新实例。
    strApp = App.Path & "\" & App.EXENAME
    Shell strApp, 1
End Sub

Private Sub mnuOptionsToolbar_Click()
    '切换工具栏的visible属性。
    picToolbar.Visible = Not picToolbar.Visible
    '改变Checked属性来匹配当前状态。
    mnuOptionsToolbar.Checked = picToolbar.Visible
    '调用调整大小过程。
    ResizeNote
End Sub

Private Sub mnuRecentFile_Click(Index As Integer)
    '调用文件打开过程，传递一个对选定的文件名的引用。
    OpenFile (mnuRecentFile(Index).Caption)
    '更新“文件”菜单控件数组中最近打开的文件列表。
    GetRecentFiles
End Sub

Private Sub mnuSearchFind_Click()
    '如果文本框中有文本，把它赋值给“查找”窗体中的文本框，否则赋值上一次的搜索文本。
    If txtNote.SelText <> "" Then
        frmFind.txtFind.Text = txtNote.SelText
    Else
        frmFind.txtFind.Text = gFindString
    End If
    '设置全局变量从头开始。
    gFirstTime = True
    '设置区分大小写复选框匹配全局变量。
    If (gFindCase) Then
        frmFind.chkCase = 1
    End If
    '显示“查找”窗体。
    frmFind.Show vbModal
End Sub

Private Sub mnuSearchFindNext_Click()
    '如果全局变量不为空，调用查找过程；否则，调用“查找”菜单。
    If Len(gFindString) > 0 Then
        FindIt
    Else
        mnuSearchFind_Click
    End If
End Sub

Private Sub txtNote_Change()
    '设置全局变量，显示文本已经改变。
    FState.Dirty = True
End Sub
```

建立对话框 frmFind

如图1-2所示的窗体为“搜索”菜单中的菜单项“查找”所弹出的对话框，用来搜索文本。编写此窗体代码，包括全局变量gFindCase（切换大小写匹配）、gFindString（搜索文本）、gFindDirection（切换搜索方向）、gFirstTime（切换从文本开头开始）。

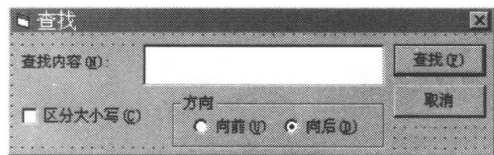


图 1-2 “搜索”菜单中“查找”菜单项所调用的对话框

Option Explicit

Private Sub chkCase_Click()

 '给全局变量赋值。

 gFindCase = chkCase.Value

End Sub

Private Sub cmdCancel_Click()

 '将值保存到全局变量。

 gFindString = txtFind.Text

 gFindCase = chkCase.Value

 '卸载“查找”对话框。

 Unload frmFind

End Sub

Private Sub cmdFind_Click()

 '将文本字符串赋值给全局变量。

 gFindString = txtFind.Text

 FindIt

End Sub

Private Sub Form_Load()

 '查找”按钮无效-还没有要查找的文本。

 cmdFind.Enabled = False

 '读全局变量并初始化选项按钮。

 optDirection(gFindDirection).Value = 1

End Sub

Private Sub optDirection_Click(Index As Integer)

 '给全局变量赋值。

 gFindDirection = Index

End Sub

Private Sub txtFind_Change()

 '设置全局变量。

 gFirstTime = True

 '如果文本框为空，“查找”按钮无效。

 If txtFind.Text = "" Then

 cmdFind.Enabled = False

 Else

 cmdFind.Enabled = True

```
End If
End Sub
```

✎ 建立模块 Module1

此模块为SDI记事本应用程序实例全局模块，代码如下：

```
Option Explicit
'存储关于子窗体信息的用户定义类型。
Type FormState
    Deleted As Integer
    Dirty As Integer
    Color As Long
End Type

Public FState As FormState           '用户定义型数组。
Public gFindString As String         '保存搜索文本。
Public gFindCase As Integer          '区分大小写标志。
Public gFindDirection As Integer     '搜索方向标志。
Public gCurPos As Integer            '保存当前光标位置。
Public gFirstTime As Integer         '起始位置。
Public Const ThisApp = "MDINote"     '登记 ThisApp 常数。
Public Const ThisKey = "Recent Files" '登记 ThisKey 常数。

Sub EditCopyProc()
    '复制选定文本到剪贴板。
    Clipboard.SetText frmSDI.txtNote.SelText
End Sub

Sub EditCutProc()
    '复制选定文本到剪贴板。
    Clipboard.SetText frmSDI.txtNote.SelText
    '删除选定文本。
    frmSDI.txtNote.SelText = ""
End Sub

Sub EditPasteProc()
    '将文本从剪贴板粘贴到活动控件。
    frmSDI.txtNote.SelText = Clipboard.GetText()
End Sub

Sub FileNew()
    Dim intResponse As Integer
    '如果文件已改变，则保存之。
    If FState.Dirty = True Then
        intResponse = FileSave
        If intResponse = False Then Exit Sub
    End If
    '清除textbox并更新标题。
    frmSDI.txtNote.Text = ""
    frmSDI.Caption = "SDI 记事本 - 无标题"
End Sub

Function FileSave() As Integer
    Dim strFilename As String
    '文件尚未保存。
```