

第 2 章 Excel 窗口和应用程序

2.1 设置 Excel 窗口状态

```
Sub SetWindowState()  
    Application.WindowState = xlMaximized '/xlMinimized  
    With ActiveWindow  
        .WindowState = xlNormal  
        .Left = 0  
        .Top = 0  
        .Height = Application.UsableHeight  
        .Width = Application.UsableWidth  
    End With  
End Sub
```

2.2 设置 Excel 全屏显示

```
Sub ShowFullScreen()  
    Application.DisplayFullScreen = True  
    With ActiveWindow  
        .DisplayHorizontalScrollBar = False  
        .DisplayVerticalScrollBar = False  
        .DisplayWorkbookTabs = False  
        .DisplayHeadings = False  
    End With  
End Sub
```

2.3 限定工作簿窗口大小

```
Option Explicit  
Sub SetGameWindow()  
    Dim sngWidth As Single, sngHeight As Single  
    Dim rngView As Range  
    Set rngView = Range("A1:I9")  
    With ActiveWindow  
        .DisplayHeadings = False  
        .DisplayHorizontalScrollBar = False  
        .DisplayVerticalScrollBar = False  
        .DisplayWorkbookTabs = False  
        .WindowState = xlNormal  
        sngWidth = .Width - .UsableWidth  
        sngHeight = .Height - .UsableHeight  
        .Width = rngView.Width + sngWidth  
        .Height = rngView.Height + sngHeight  
        .ScrollRow = 1  
        .ScrollColumn = 1  
        .ActiveSheet.ScrollArea = rngView.Address  
        .EnableResize = False  
    End With  
    Set rngView = Nothing  
End Sub
```

```
Sub RestoreWindow()  
    Dim rngView As Range  
    Set rngView = Range("A1:I9")  
    With ActiveWindow  
        .DisplayHeadings = True
```

```
.DisplayHorizontalScrollBar = True  
.DisplayVerticalScrollBar = True  
.DisplayWorkbookTabs = True  
.EnableResize = True  
.WindowState = xlMaximized  
.ActiveSheet.ScrollArea = ""
```

```
End With  
Set rngView = Nothing
```

```
End Sub
```

2.4 隐藏 Excel 主窗口

```
Option Explicit  
Sub MoveAppMainWindow()  
    Application.WindowState = xlNormal  
    Application.Left = 10000  
    frm_Login.Show  
End Sub
```

2.5 利用状态栏显示提示信息

```
Option Explicit  
Sub SetStatusBar()  
    Dim blnPreStatus As Boolean, i As Long  
    With Application  
        blnPreStatus = .DisplayStatusBar  
        .DisplayStatusBar = True  
        For i = 1 To 10000  
            .StatusBar = "正在写入第" & i & "个数据,共  
10000 个数据,请稍候..."  
            Cells(i) = Rnd()  
        Next i  
        MsgBox "写入数据完成。", vbInformation + vbOKOnly,  
"完成"  
        .StatusBar = False  
        .DisplayStatusBar = blnPreStatus  
    End With  
End Sub
```

2.6 Excel 中的“定时器”

```
Option Explicit  
Dim mdteTime As Date  
Sub RunTimer()  
    mdteTime = Now() + TimeValue("00:00:01")  
    Application.OnTime mdteTime, _  
        Procedure:="my_Procedure"  
End Sub  
Sub my_Procedure()  
    Range("A1") = Format(Time(), "h:mm:ss")  
    Call RunTimer  
End Sub  
Sub KillTimer()  
    Application.OnTime mdteTime, _  
        Procedure:="my_Procedure", Schedule:=False  
End Sub
```

第 2 章 Excel 窗口和应用程序

2.7 精美的数字秒表

```
Option Explicit

Public Declare Function SetTimer Lib "user32" ( _
    ByVal hWnd As Long, ByVal nIDEvent As Long, _
    ByVal uElapsed As Long, _
    ByVal lpTimerFunc As Long) As Long

Public Declare Function KillTimer Lib "user32" ( _
    ByVal hWnd As Long, ByVal nIDEvent As Long) _
    As Long

Public glngTimerID As Long, gsngTimeX As Single

Public Sub OnTimer()
    gsngTimeX = gsngTimeX + 0.1
    frm_time.lblShowTime.Caption = Format(gsngTimeX,
"0.0")
End Sub

Sub ShowUserForm()
    frm_time.Show
End Sub
```

2.8 暂停宏代码的运行

2.8.1 使用 Wait 方法

```
Option Explicit

Sub OpenForm()
    Frm_Wait.Show
End Sub
```

2.8.2 使用 Sleep API 方法

```
Option Explicit

Private Declare Sub Sleep Lib "kernel32" (ByVal dwMilliseconds
As Long)

Sub TypeDemo()
    Dim strTest As String, i As Integer
    strTest = "这是 Sleep API 函数的一个简单演示。"
    For i = 1 To Len(strTest)
        Range("A1").Value = Left(strTest, i)
        Sleep 200
    Next i
End Sub
```

2.9 防止用户干预宏代码的运行

```
Option Explicit

Sub DataInput()
    Dim i As Long
    For i = 1 To 50000
        DoEvents
        ActiveCell.Value = i
    Next i
End Sub

Sub DataInputInteractive()
    Dim i As Long
    Application.Interactive = False
    For i = 1 To 50000
```

```
        DoEvents
        ActiveCell.Value = i
    Next i
    Application.Interactive = True
End Sub

2.10 调用变量名称指定的宏过程

Option Explicit '运用 Run 方法

Function strGetString(intNum As Integer, strName As String) As
String
    strGetString = strName & " " & intNum
End Function

Sub RunParameter()
    Dim strResult As String
    strResult = Application.Run("strGetString", 2016, "Excel")
    MsgBox strResult, vbOKOnly + vbInformation, "Run 传递
参数"
End Sub

Option Explicit 'mdlSelectCall

Sub SelectCall(intIndex As Integer)
    Select Case intIndex
        Case 1
            Call Macro1
        Case 2
            Call Macro2
        ,
        .....
        Case n
            Call Macron
    End Select
End Sub

Sub SelectRun(intIndex As Integer)
    Application.Run "Macro" & intIndex
End Sub

2.11 利用 OnKey 方法捕捉键盘输入

Option Explicit '捕捉/恢复 Ctrl+V

Sub CtrlV()
    Application.OnKey "^v", "CatchPaste"
End Sub

Sub CatchPaste()
    MsgBox "按下 Ctrl+V 组合键", vbInformation
End Sub

Sub RestoreCtrlV()
    Application.OnKey "^v"
End Sub

Option Explicit ' 禁止/恢复 F11

Sub OnkeyF11()
    Application.OnKey "{F11}", ""
End Sub

Sub RestoreF11()
    Application.OnKey "{F11}"
End Sub
```

第 2 章 Excel 窗口和应用程序

2.12 使用 SendKeys 方法模拟键盘输入

Option Explicit

Sub SendKeysDemo() ‘修改【信任对 VBA 工程对象模型的访问】

If Not Application.ShowDevTools Then

Application.ShowDevTools = True

End If

Application.SendKeys "/las%V~"

End Sub

Sub SelectSpecialChar() ‘选定单元格中指定字符

Range("A1").Select

Application.SendKeys "{F2}+{Left 4}"

End Sub

Sub SaveAsFilename() ‘显示【另存为】对话框

Application.SendKeys "ExcelHome.xlsm"

Application.Dialogs(xlDialogSaveAs).Show

End Sub

2.13 巧妙捕获用户中断

Sub ESCInterrupt() ‘运行过程 ESCInterrupt

Dim i As Long

Cells.ClearContents

For i = 1 To 100000

Cells(i, 1).Value = i

Next i

End Sub

Sub DisbledESC() ‘运行过程 DisbleESC

Dim i As Long

Dim strTip As String

strTip = "代码运行需要较长时间，是否继续？选择""是""执行，""否""取消。"

Application.EnableCancelKey = xlDisabled

If MsgBox(strTip, vbInformation + vbYesNo) = vbYes Then

Cells.Clear

For i = 1 To 100000

Cells(i, 1).Value = i

Next i

End If

End Sub

Sub ErrorHandleESC() ‘运行过程 ErrorHandleESC

Dim i As Long

Dim strTip As String

strTip = "代码运行需要较长时间，按 ESC_或 <Ctrl+Break> 可终止当前代码。"

On Error GoTo HandleCancel

Application.EnableCancelKey = xlErrorHandler

MsgBox strTip, vbExclamation

Cells.Clear

For i = 1 To 100000

Cells(i, 1).Value = i

Next i

Exit Sub

HandleCancel:

If Err.Number = 18 Then

MsgBox "用户终止了代码运行。", vbExclamation

End If

End Sub

2.14 使用 Application 级别事件（新建工作簿查看代码功能）

Option Explicit

Private WithEvents xlApp As Application

Private Sub Workbook_BeforeClose(Cancel As Boolean)

Set xlApp = Nothing

End Sub

Private Sub Workbook_Open()

Set xlApp = Application

End Sub

Private Sub xlApp_NewWorkbook(ByVal Wb As Workbook)

Wb.Close False

End Sub

第 3 章 工作簿和工作表

3.1 引用工作表

Sub GetActiveSheet()

With ActiveSheet

.Name = "ExcelHome"

Debug.Print "Name: " & .Name, "CodeName: " & .CodeName, "Index: " & .Index

End With

End Sub

3.2 添加新工作表

Sub AddWorksheet() ‘添加工作表

Sheets.Add After:=Sheets(Sheets.Count)

ActiveSheet.Name = "新添工作表"

End Sub

Sub AddNewWorksheet() ‘引用对象变量

Dim wksSht As Worksheet

With ThisWorkbook

Set

wksSht

= .Worksheets.Add(After:=.Sheets(.Sheets.Count))

End With

wksSht.Name = "新添工作表 2"

Set wksSht = Nothing

End Sub

Sub DeleteWorksheet() ‘删除其他工作表

Application.DisplayAlerts = False

Dim wksSht As Worksheet

For Each wksSht In Worksheets

If wksSht.Name <> "Sheet1" Then wksSht.Delete

Next

Set wksSht = Nothing

Application.DisplayAlerts = True

End Sub

第3章 工作簿和工作表

3.3 防止更改工作表的名称

```
Option Explicit
Private Sub Worksheet_SelectionChange(ByVal Target As Range)
If Me.Name <> "Excel Home" Then Me.Name = "Excel Home"
End Sub
```

3.4 判断工作簿中是否存在指定名称的工作表

```
Function blnSheetExist(ByVal strShtName As String) As Boolean
    Dim wksSht As Worksheet
    On Error Resume Next
    Set wksSht = Worksheets(strShtName)
    If Err.Number = 0 Then blnSheetExist = True
    Set wksSht = Nothing
End Function
```

3.5 按名称排序工作表

```
Option Explicit
Sub SortShtByName() '按名字排序
    Dim intSheetCount As Integer
    Dim i As Integer, j As Integer
    Dim strName As String, astrSheet() As String
    Application.ScreenUpdating = False
    intSheetCount = Worksheets.Count
    ReDim astrSheet(1 To intSheetCount)
    For i = 1 To intSheetCount
        astrSheet(i) = Worksheets(i).Name
    Next i
    For i = 1 To intSheetCount - 1
        For j = i + 1 To intSheetCount
            If astrSheet(i) < astrSheet(j) Then
                strName = astrSheet(i)
                astrSheet(i) = astrSheet(j)
                astrSheet(j) = strName
            End If
        Next j
    Next i
    For i = 1 To intSheetCount
        Worksheets(astrSheet(i)).Move
    Before:=Worksheets(1)
    Next i
    Application.ScreenUpdating = True
End Sub
```

```
Sub SortByNumber() '按数字部分排序
    Dim i As Integer, intSheetCount As Integer
    Dim aintArr1() As Integer, aintArr2() As Integer
    Application.ScreenUpdating = False
    intSheetCount = Worksheets.Count
    ReDim aintArr1(1 To intSheetCount)
    ReDim aintArr2(1 To intSheetCount)
    For i = 1 To intSheetCount
        aintArr1(i) = Val(Worksheets(i).Name)
```

```
Next i
    For i = 1 To intSheetCount
        aintArr2(i) =
Application.WorksheetFunction.Large(aintArr1, i)
    Next i
    For i = 1 To intSheetCount
        Worksheets(aintArr2(i) & "月份").Move
    Before:=Worksheets(1)
    Next i
    Application.ScreenUpdating = True
End Sub
```

3.6 限制工作表滚动区域

```
Private Sub Workbook_Open()
    Worksheets(1).ScrollArea = "A1:F10"
End Sub
Sub SetScrollArea() '动态设置允许滚动区域
    With Sheet2
        .Activate
        .ScrollArea = ActiveWindow.VisibleRange.Address
    End With
End Sub
Sub DisbleScrollAreaSetup() '取消允许滚动区域
    Worksheets(1).ScrollArea = ""
End Sub
```

3.7 操作受保护的工作表

```
Sub ChangeValue()
    With ActiveSheet
        .Unprotect
        .Range("A1").Value = "Excel Home 欢迎您!"
        .Protect
    End With
End Sub
```

3.8 在指定单元格区域中禁止显示右键菜单

```
Option Explicit
Private Sub Worksheet_BeforeRightClick _
    (ByVal Target As Range, Cancel As Boolean)
    If Not Application.Intersect(Target, Range("A1:F10")) _
        Is Nothing Then Cancel = True
End Sub
```

3.9 选中所有工作表

```
Option Explicit
Sub SelectedAllSheets() '全选工作表 1
    Dim wksSht As Worksheet
    For Each wksSht In Worksheets
        wksSht.Select False
    Next wksSht
    Set wksSht = Nothing
End Sub
Sub SelectedSheetsA() '全选工作表 2
    Worksheets.Select
End Sub
```

第 3 章 工作簿和工作表

3.10 在 VBA 中使用工作表函数

```
Sub GetSum() ‘输入公式计算 B1:B10 单元格区域的和
    Range("D2") = "=SUM(B1:B10)"
End Sub

Sub EvaluateSum() ‘直接计算 B1:B10 单元格区域的和
    Range("F2") = Evaluate("=SUM(B1:B10)")
End Sub

Sub GetCountif() ‘统计 A1:A10"ExcelHome"出现次数
    Range("H2") = Application.WorksheetFunction.CountIf _
        (Range("A1:A10"), "ExcelHome")
    Range("H3") = WorksheetFunction.CountIf _
        (Range("A1:A10"), "ExcelHome")
    Range("H4") = Application.CountIf(Range("A1:A10"),
    "ExcelHome")
End Sub
```

3.11 判断是否存在指定名称的工作簿

```
Function blnWBExist(ByVal strWbName As String) As Boolean
    Dim wkbName As Workbook
    For Each wkbName In Workbooks
        If wkbName.Name = strWbName Then
            blnWBExist = True
            Exit For
        End If
    Next wkbName
    Set wkbName = Nothing
End Function

Function blnExistWB(ByVal strWbName As String) As Boolean
    Dim wkbName As Workbook:On Error Resume Next
    Set wkbName = Workbooks(strWbName)
    If Err.Number = 0 Then blnExistWB = True
    Set wkbName = Nothing
End Function
```

3.13 新建工作簿

```
Sub CreateNewWorkbook()
    Workbooks.Add
    ActiveWorkbook.SaveAs ThisWorkbook.Path &
    "\NewBook1.xlsx"
End Sub
```

3.14 导入文本文件中的数据

```
Sub UserOpenText()
    Workbooks.OpenText _
        Filename:=ThisWorkbook.Path & "\Students.txt", _
        StartRow:=2, _
        DataType:=xlDelimited, _
        Other:=True, _
        OtherChar:="|", _
        FieldInfo:=Array(Array(1, 9), Array(2, 1), _
        Array(3, 1), Array(4, 5))
    ActiveWorkbook.Sheets(1).Columns("A:C").AutoFit
End Sub
```

```
附：Students.txt
序号|姓名|出生日期
#0001|孙悟空|19971223
#0002|唐僧|19980604
#0003|猪八戒|19920512
#0004|沙和尚|19911002
#0005|太上老君|19901225
#0006|二郎神|19970622
#0007|菩萨|19930318
#0008|佛祖|19901105
#0009|白龙马|19970715
```

3.15 保存工作簿

```
Sub SaveThisBook() ‘Save
    ThisWorkbook.Save
End Sub

Sub SaveAsWorkbook() ‘SaveAs
    ThisWorkbook.SaveAs Filename:=ThisWorkbook.Path & _
        "\Backup.xlsm", Password:="123"
End Sub

Sub SaveWorkbookCopy() ‘SaveCopyAs
    ThisWorkbook.SaveCopyAs ThisWorkbook.Path & "\副本.xlsm"
End Sub
```

3.16 保存指定工作表到新的工作簿文件

```
Sub SavedSheetAsWorkbook() ‘保存活动工作表
    ActiveSheet.Copy
    ActiveWorkbook.Close SaveChanges:=True, _
        Filename:=ThisWorkbook.Path & "\SheetBackup.xlsx"
End Sub

Sub SaveSheetsAsWorkbook() ‘保存工作表组
    Worksheets(Array("Sheet1", "Sheet2")).Copy
    ActiveWorkbook.SaveAs Filename:= _
        ThisWorkbook.Path & "\SheetsBackup.xlsx"
End Sub

Sub DynamicSaveSheets() ‘动态保存工作表组
    Dim wksSht As Worksheet
    Dim intCount As Integer
    Dim astrArr() As String
    For Each wksSht In Worksheets
        If VBA.InStr(1, wksSht.Name, "月份",
        vbTextCompare) > 0 Then
            intCount = intCount + 1
            ReDim Preserve astrArr(1 To intCount)
            astrArr(UBound(astrArr)) = wksSht.Name
        End If
    Next wksSht
    If intCount > 0 Then
        Worksheets(astrArr).Copy
        ActiveWorkbook.SaveAs Filename:= _
            ThisWorkbook.Path & "\SheetsBackup.xlsx"
    End If
    Set wksSht = Nothing
End Sub
```

第 3 章 工作簿和工作表

3.17 禁止工作簿文件另存

Option Explicit '在 ThisWorkbook 里操作

```
Private Sub Workbook_BeforeSave _  
    (ByVal SaveAsUI As Boolean, Cancel As Boolean)  
    If SaveAsUI = True Then  
        Cancel = True  
        MsgBox "本文件另存操作已被禁止！", vbOKOnly +  
vbCritical, "警告"  
    End If  
End Sub
```

3.18 关闭工作簿不显示保存对话框

Sub CloseWithoutSave() '不保存关闭 1

 ThisWorkbook.Close False

End Sub

Sub NotSaveAndClose() '不保存关闭 2

 ThisWorkbook.Saved = True

 ThisWorkbook.Close

End Sub

Sub SaveAndClose() '保存后关闭

 ThisWorkbook.Save

 ThisWorkbook.Close

End Sub

3.19 限制工作簿只能通过代码关闭

Option Explicit

Dim mblnCloseFlag As Boolean

Private Sub Workbook_BeforeClose(Cancel As Boolean)

 If mblnCloseFlag = False Then

 Cancel = True

 MsgBox "文件已被限制只能通过代码关闭。",

vbExclamation, "提示"

 End If

End Sub

Sub CloseWorkbook()

 mblnCloseFlag = True

 Me.Close

End Sub

3.20 打开启用宏的工作簿时禁用宏

Option Explicit

Sub OpenWithoutMacro()

 With Application

 .AutomationSecurity =

msoAutomationSecurityForceDisable

 Workbooks.Open ThisWorkbook.Path &

"\Macro1.xlsm"

 .AutomationSecurity = msoAutomationSecurityLow

 End With

End Sub

Option Explicit

Sub DemoTest()

 Range("a1").Select

End Sub

3.21 打开工作簿时禁止更新链接

Option Explicit

Sub OpenWithoutUpdateLinksWindows() '禁止弹出提示框

 Application.AskToUpdateLinks = False

 Workbooks.Open ThisWorkbook.Path & "\Macro1.xlsx"

 Application.AskToUpdateLinks = True

End Sub

Sub OpenWithoutUpdateLinks() '禁止更新链接

 Workbooks.Open Filename:=ThisWorkbook.Path _

& "\Macro1.xlsx", UpdateLinks:=0

End Sub

3.22 定义隐藏的名称

Option Explicit

Sub AddHiddenName() '定义隐藏名称

 ThisWorkbook.Names.Add Name:="EH", _

 RefersTo:="Excel Home", Visible:=False

End Sub

Sub ShowName() '显示名称

 ThisWorkbook.Names("EH").Visible = True

End Sub

Sub HideAllName() '隐藏所有名称

 Dim objName As Name

 For Each objName In ActiveWorkbook.Names

 objName.Visible = False

 Next objName

 Set objName = Nothing

End Sub

3.23 实现工作簿“自杀”功能

Option Explicit

Sub KillThisWorkbook()

 With ThisWorkbook

 .Saved = True

 .ChangeFileAccess xlReadOnly

 Kill .FullName

 .Close

 End With

End Sub

3.24 限制工作簿的使用次数

Option Explicit

Sub AddCustomDocumentProperties()

 ThisWorkbook.CustomDocumentProperties.Add _

 Name:="OpenTimes", _

 LinkToContent:=False, _

 Type:=msoPropertyTypeNumber, _

 Value:=0

End Sub

第 4 章 使用 Range 对象

4.1 引用单元格区域

Sub SelectData() '选择数据区域

```
Dim rngRange As Range
Set rngRange = Range("A1").CurrentRegion
With rngRange
    .Offset(1, 0).Resize(.Rows.Count - 1, _
        .Columns.Count).Select
End With
Set rngRange = Nothing
```

End Sub

Sub AreasDemo() 'Area 属性

```
Dim rngRange As Range, i As Integer
Set rngRange = Range("A1,B2:D5,G3:I8")
For i = 1 To rngRange.Areas.Count
    MsgBox "将选择第" & i & "个 Areas 区域: " & _
        rngRange.Areas(i).Address
    rngRange.Areas(i).Select
Next i
Set rngRange = Nothing
```

End Sub

4.2 取得最后一个非空单元格

Sub LastRow() '最后一行

```
Dim rngLastRow As Range
Set rngLastRow = Range("a" & Rows.Count).End(xlUp)
MsgBox rngLastRow.Row: Set rngLastRow = Nothing
```

End Sub

Sub LastColumn() '最后一列

```
Dim rngLastColumn As Range
Set rngLastColumn = Cells(1, Columns.Count).End(xlToLeft)
MsgBox rngLastColumn.Column
Set rngLastColumn = Nothing
```

End Sub

4.3 随心所欲复制单元格区域

Sub RangeCopy() '简单复制

```
Range("A1").CurrentRegion.Copy Destination:= _
    Sheets("Sheet2").Range("A1")
```

End Sub

Sub CopyWithSameColumnWidths() '复制列宽

```
Sheets("Sheet1").Range("A1").CurrentRegion.Copy
With Sheets("Sheet2").Range("A1")
    .PasteSpecial xlPasteColumnWidths
    .PasteSpecial xlPasteAll
End With
Application.CutCopyMode = False
```

End Sub

4.4 仅复制数值到另一区域

Sub Copy_PasteSpecial() '选择性粘贴

```
Sheets("Sheet1").Range("A1").CurrentRegion.Copy
Sheets("Sheet2").Range("A1").PasteSpecial
Paste:=xlPasteValues
Application.CutCopyMode = False
```

End Sub

Sub Get_Value() '直接赋值

```
Sheets("Sheet2").Range("A5:B7").Value = _
    Sheets("Sheet1").Range("A2:B4").Value
```

End Sub

Sub GetValue_Resize() 'Resize 方法

```
With Sheets("Sheet1").Range("A1").CurrentRegion
    Sheets("Sheet2").Range("A10").Resize(.Rows.Count, _
        .Columns.Count).Value = .Value
End With
```

End Sub

4.5 对行进行快速分组

Sub RowsGroup() '快速分组

```
Dim i As Long
For i = 1 To 21 Step 3
    Rows(i & ":" & i + 1).Group
Next i
ActiveSheet.Outline.ShowLevels RowLevels:=1
```

End Sub

Sub AllUnGroup() '取消分组

```
On Error Resume Next
ActiveSheet.Outline.ShowLevels RowLevels:=2
Rows("1:21").UNGroup
```

End Sub

4.6 获取两个单元格区域的交叉区域

Sub IntersectRng(rng1 As Range, rng2 As Range)

```
Dim rngTarget As Range
Set rngTarget = Application.Intersect(rng1, rng2)
If rngTarget Is Nothing Then:MsgBox "不存在交叉区域。"
Else:MsgBox "交叉区域地址为: " & rngTarget.Address
End If
Set rngTarget = Nothing
```

End Sub

Sub TestIntersect()

```
Call IntersectRng(Range("C3:D8"), Range("B5:F6"))
```

End Sub

4.7 联合多个单元格区域

Sub UnionSelect()

```
Dim rngCell As Range, rngUnion As Range
For Each rngCell In Range("A1:F10")
    If rngCell.Value = "H" Then
        If rngUnion Is Nothing Then
            Set rngUnion = rngCell
        Else
            Set rngUnion = Application.Union(rngUnion, rngCell)
        End If
    End If
Next rngCell
```

```
If Not rngUnion Is Nothing Then rngUnion.Select
Set rngCell = Nothing
Set rngUnion = Nothing
```

End Sub

第 4 章 使用 Range 对象

4.8 判断一个区域是否包含在另一个区域中

```
Function blnInclude_Intersect(ByVal rngCell1 As Range, _  
    ByVal rngCell2 As Range) As Boolean  
    Dim rngIntersect As Range  
    Set rngIntersect = Application.Intersect(rngCell1, rngCell2)  
    If Not rngIntersect Is Nothing Then  
        If rngIntersect.Address = rngCell1.Address Then  
            blnInclude_Intersect = True  
        End If  
    End If  
End Function
```

```
Function blnInclude_Union(ByVal rngCell1 As Range, _  
    ByVal rngCell2 As Range) As Boolean  
    Dim rngUnion As Range  
    Set rngUnion = Application.Union(rngCell1, rngCell2)  
    If Not rngUnion Is Nothing Then  
        If rngUnion.Address = rngCell2.Address Then  
            blnInclude_Union = True  
        End If  
    End If  
    Set rngUnion = Nothing  
End Function
```

```
Sub Include_1()  
    MsgBox blnInclude_Intersect(Range("C5:D6"),  
    Range("B3:H8"))  
End Sub  
Sub Include_2()  
    MsgBox blnInclude_Union(Range("C5:D6"),  
    Range("B3:C4"))  
End Sub
```

4.9 设置字符格式

```
Sub CellCharacter()  
    With Range("A1")  
        .Clear  
        .Value = "Y=X2+1"  
        .Characters(4, 1).Font.Superscript = True  
        .Characters(1, 1).Font.ColorIndex = 3  
        .Font.Size = 20  
    End With  
End Sub  
Sub ShapeCharacter()  
    If Range("A3").Comment Is Nothing Then  
        Range("A3").AddComment Text:=""  
    End If  
    With Range("A3").Comment  
        .Text Text:="Microsoft Excel 2016"  
        .Shape.TextFrame.Characters(17).Font.ColorIndex = 3  
    End With  
End Sub
```

4.10 单元格区域添加边框

```
Sub AddBorders()  
    Dim rngCell As Range  
    Set rngCell = Range("B2:F8")  
    With rngCell.Borders  
        .LineStyle = xlContinuous  
        .Weight = xlThin  
        .ColorIndex = 5  
    End With  
    rngCell.BorderAround xlContinuous, xlMedium, 5  
    Set rngCell = Nothing  
End Sub
```

```
Sub BordersIndexDemo()  
    Dim rngCell As Range  
    Set rngCell = Range("B2:F8")  
    With rngCell.Borders(xlInsideHorizontal)  
        .LineStyle = xlDot  
        .Weight = xlThin  
        .ColorIndex = 5  
    End With  
    With rngCell.Borders(xlInsideVertical)  
        .LineStyle = xlContinuous  
        .Weight = xlThin  
        .ColorIndex = 5  
    End With  
    rngCell.BorderAround xlContinuous, xlMedium, 5  
    Set rngCell = Nothing  
End Sub
```

```
Sub Restore() '清除边框线  
    Columns("B:F").Borders.LineStyle = xlNone  
End Sub
```

4.11 高亮显示单元格区域

```
Private Sub Worksheet_SelectionChange(ByVal Target As  
    Range)  
        ThisWorkbook.Names.Add "ActRow", ActiveCell.Row  
        ThisWorkbook.Names.Add "ActCol", ActiveCell.Column  
    End Sub
```

4.12 将单元格公式转换为数值

```
Sub SpecialPaste()  
    With Range("Data")  
        .Copy  
        .PasteSpecial xlPasteValues  
    End With  
End Sub  
Sub UseValue()  
    Selection.Value = Selection.Value  
End Sub  
Sub UseFormula()  
    Range("A1").Formula = Range("A1").Value  
End Sub  
Sub Restore()  
    Range("A1") = "=B1+C1"  
End Sub
```


第4章 使用 Range 对象

4.13 动态设置单元格数据验证序列

```
Sub SheetsNameValidation() '设置数据有效性
    Dim i As Integer, strList As String
    Dim wksSht As Worksheet
    For Each wksSht In Worksheets
        If wksSht.Name <> "Office 2016" Then
            strList = strList & wksSht.Name & ","
        End If
    Next wksSht
    With Worksheets("Office 2016").Range("C3").Validation
        .Delete
        .Add Type:=xlValidateList, Formula1:=strList
    End With
    Set wksSht = Nothing
End Sub

Sub DeleteValidation() '删除数据有效性
    Range("C3").Validation.Delete
End Sub
```

4.14 判断单元格公式是否存在错误

```
Sub FormulasError()
    If VBA.IsError(Range("A1").Value) = True Then
        MsgBox "A1 单元格错误类型为:" & Range("A1").Text
    Else
        MsgBox "A1 单元格公式结果为:" &
Range("A1").Value
    End If
End Sub
```

4.15 批量删除所有错误值

```
Sub DeleteError() '执行代码，删除错误值_1
    Dim rngRange As Range
    Dim rngCell As Range
    Set rngRange = Range("a1").CurrentRegion
    For Each rngCell In rngRange
        If VBA.IsError(rngCell.Value) = True Then
            rngCell.Value = ""
        End If
    Next rngCell
    Set rngCell = Nothing
    Set rngRange = Nothing
End Sub

Sub DeleteAllError() '执行代码，删除错误值_2
    On Error Resume Next
    Dim rngRange As Range
    Set rngRange = Range("a1").CurrentRegion.SpecialCells _
(xlCellTypeConstants, xlErrors)
    If Not rngRange Is Nothing Then
        rngRange.Value = ""
    End If
    Set rngRange = Nothing
End Sub
```

Sub BackUp() '恢复初始状态

Range("L1").CurrentRegion.Copy Range("A1")

End Sub

4.16 返回指定列的列标

```
Function strGetColumnLetter(ByVal intCol As Integer) As String
    strGetColumnLetter = VBA.Split(Cells(1, intCol).Address,
"$")(1)
End Function

Sub GetLetter()
    MsgBox "第" & Range("A1").Value & "列对应的列标为" & _
strGetColumnLetter(Range("A1").Value)
End Sub
```

4.17 判断单元格是否存在批注

```
Function blnComment(ByVal rngRange As Range) As Boolean
    If rngRange.Cells(1).Comment Is Nothing Then
        blnComment = False
    Else: blnComment = True
    End If
End Function
```

```
Sub CheckComment()
    MsgBox blnComment(Selection)
End Sub
```

4.18 为单元格添加批注

```
Sub Comment_Add() '添加批注
    With Range("B5")
        If .Comment Is Nothing Then
            .AddComment Text:=.Text
            .Comment.Visible = True
        End If
    End With
End Sub
```

Sub Comment_Delete() '删除批注

```
    With Range("B5")
        If Not .Comment Is Nothing Then
            .Comment.Delete
        End If
    End With
End Sub
```

End Sub

4.19 编辑批注文本

```
Sub Show_Change_Comment() '编辑批注文本
    With Range("B5")
        If Not .Comment Is Nothing Then
            MsgBox .Comment.Text
            .Comment.Text Text:="Excel Home"
        End If
    End With
End Sub
```

End Sub

Sub AppendText() '添加批注文本

```
    With Range("B5").Comment
        .Text Text:=VBA.vbCrLf & [B5].Value, Start:=Len(.Text) + 1
    End With
End Sub
```

第 4 章 使用 Range 对象

4.20 修改批注外观

```
Sub ChangeCommentShapeType() ‘修改批注外观
    Range("B3").Comment.Shape.AutoShapeType = _
        msoShapeRoundedRectangle
End Sub

Sub BackUp() ‘恢复批注外观
    Range("B3").Comment.Shape.AutoShapeType = 1
End Sub
```

4.21 设置批注字体

```
Sub CommentFont() ‘执行代码
    Dim objComment As Comment
    For Each objComment In ActiveSheet.Comments
        With objComment.Shape.TextFrame.Characters.Font
            .Name = "微软雅黑"
            .Bold = msoFalse
            .Size = 14
            .ColorIndex = 3
        End With
    Next objComment
    Set objComment = Nothing
End Sub
```

```
Sub BackUp() ‘恢复
    Sheets("Sheet2").Range("A1:D9").Copy
    Sheets("Sheet1").Range("A1")
End Sub
```

4.22 显示图片批注

```
Sub ChangeCommentShapeType()
    With Range("B3").Comment
        .Shape.Fill.UserPicture _
            ThisWorkbook.Path & "\Logo.jpg"
    End With
End Sub

Sub BackUp()
    Range("L3").Copy Range("B3")
End Sub
```

4.23 快速判断单元格区域是否存在合并单元格

```
Sub IsMerge()
    If VBA.IsNull(Range("A1:E10").MergeCells) = True Then
        MsgBox "包含合并单元格"
    Else
        MsgBox "没有包含合并单元格"
    End If
End Sub

Sub IsMergeCell()
    If Range("A1").MergeCells = True Then
        MsgBox "包含合并单元格"
    Else
        MsgBox "没有包含合并单元格"
    End If
End Sub
```

4.24 合并单元格时连接每个单元格内容

```
Sub MergeValue()
    Dim strText As String, rngCell As Range
    If TypeName(Selection) = "Range" Then
        For Each rngCell In Selection
            strText = strText & rngCell.Value
        Next rngCell
        Application.DisplayAlerts = False
        Selection.Merge
        Selection.Value = strText
        Application.DisplayAlerts = True
    End If
    Set rngCell = Nothing
End Sub
```

4.25 取消合并时在每个单元格中保留内容

```
Sub UnMergeValue() ‘执行代码
    Dim strText As String, i As Long, intCount As Integer
    For i = 2 To Range("B1").End(xlDown).Row
        With Cells(i, 1)
            strText = .Value
            intCount = .MergeArea.Count
            .UnMerge
            .Resize(intCount, 1).Value = strText
        End With
        i = i + intCount - 1
    Next i
End Sub
```

```
Sub BackUp() ‘恢复
    Dim intRow As Integer
    Dim i As Long
    Application.DisplayAlerts = False
    With ActiveSheet
        intRow = .Range("A1").End(xlDown).Row
        For i = intRow To 2 Step -1
            If .Cells(i, 1).Value = .Cells(i - 1, 1).Value Then
                .Range(.Cells(i - 1, 1), .Cells(i, 1)).Merge
            End If
        Next i
    End With
    Application.DisplayAlerts = True
End Sub
```

4.26 按指定条件自动筛选数据

```
Sub DataFilter()
    Worksheets("Sheet1").Range("A1").AutoFilter _
        Field:=2, Criteria1:="男", Operator:=xlFilterValues
End Sub

Sub BackUp()
    Worksheets("Sheet1").Range("A1").AutoFilter
End Sub
```

第4章 使用 Range 对象

4.27 查找包含指定字符串的所有单元格

```
Sub FindCells() '选定包含指定字符串"BB"的所有单元格
    Dim rngCell As Range, rngResult As Range
    Dim strFirstAddress As String
    With Range("A1:E10")
        Set rngCell = .Find(What:="BB", After:=.Cells(1), _
            LookIn:=xlValues, LookAt:=xlPart)
        If Not rngCell Is Nothing Then
            strFirstAddress = rngCell.Address
            Set rngResult = rngCell
            Do
                Set rngResult = Application.Union(rngResult,
rngCell)
                Set rngCell = .FindNext(rngCell)
            Loop While rngCell.Address <> strFirstAddress
            rngResult.Interior.ColorIndex = 8
        End If
    End With
    Set rngCell = Nothing
    Set rngResult = Nothing
End Sub

Sub Backup() '恢复至初始状态
    Range("A1:E10").Interior.Pattern = xlNone
End Sub
```

4.28 合并计算多个工作表的数据

```
Sub ConsolidateSheets() '合并计算
    Dim astrArr(1 To 3) As String
    Dim i As Integer
    Dim rngRange As Range
    For i = 2 To 4
        astrArr(i - 1) = Sheets(i).Name & "!" & _
            Sheets(i).Range("A1").CurrentRegion.Address _
            (ReferenceStyle:=xlR1C1)
    Next i
    Set rngRange = Sheets("汇总表").Range("A1")
    With rngRange
        .Consolidate Sources:=astrArr, Function:=xlSum, _
            TopRow:=True, LeftColumn:=True
        .Value = "业务员"
    End With
    Sheets(1).Range("A1").Copy
    With rngRange
        .Resize(1, .CurrentRegion.Columns.Count) _
            .PasteSpecial xlPasteFormats
    End With
    Sheets(1).Range("A2").Copy
    With rngRange.CurrentRegion
        .Offset(1, 0).Resize(.Rows.Count - 1, .Columns.Count) _
            .PasteSpecial xlPasteFormats
```

End With

Application.CutCopyMode = False

Set rngRange = Nothing

End Sub

Sub BackUp() '清除

Range("A1").CurrentRegion.ClearContents

Range("A1").Activate

End Sub

4.29 合并计算多个工作簿的工作表

```
Sub ConsolidateWorkbook() '合并计算
    Dim astrArray() As String
    Dim wkbBook As Workbook
    Dim wksSht As Worksheet
    Dim intCount As Integer, i As Integer
    intCount = Workbooks.Count
    ReDim astrArray(1 To intCount - 1)
    For Each wkbBook In Workbooks
        If Not wkbBook Is ThisWorkbook Then
            Set wksSht = wkbBook.Worksheets(1)
            i = i + 1
            astrArray(i) = "[" & wkbBook.Name & "]" & _
                wksSht.Name & "!" & wksSht.Range("A1"). _
                CurrentRegion.Address(ReferenceStyle:=xlR1C1)
        End If
    Next wkbBook
    Worksheets("汇总表").Range("A1").Consolidate _
        astrArray, xlSum, True, True
    Set wksSht = Nothing
    Set wkbBook = Nothing
End Sub

Sub BackUp() '恢复
    Range("a1").CurrentRegion.ClearContents
End Sub
```

4.30.1 合并内容相同的单列连续单元格

```
Sub MergeSameCells() '执行代码
    Dim intRow As Integer, i As Long
    Application.DisplayAlerts = False
    With ActiveSheet
        intRow = .Range("A1").End(xlDown).Row
        For i = intRow To 2 Step -1
            If .Cells(i, 1).Value = .Cells(i - 1, 1).Value Then
                .Range(.Cells(i - 1, 1), .Cells(i, 1)).Merge
            End If
        Next i
    End With
    Application.DisplayAlerts = True
End Sub
```

第 4 章 使用 Range 对象

4.30.1 合并内容相同的单列连续单元格

```
Sub BackUp()    '恢复
    Dim strText As String, i As Long, intCount As Integer
    For i = 2 To Range("B1").End(xlDown).Row
        With Cells(i, 1)
            strText = .Value
            intCount = .MergeArea.Count
            .UnMerge
            .Resize(intCount, 1).Value = strText
        End With
        i = i + intCount - 1
    Next i
    Range("C2:C13").Copy
    Range("A2:A13").PasteSpecial Paste:=xlPasteFormats
    Application.CutCopyMode = False
    Range("A1").Activate
End Sub
```

4.31 多条件筛选

```
Sub Filter_MoreCriteria()    '多条件筛选
    Application.ScreenUpdating = False
    With Worksheets("Sheet1")
        If .FilterMode = True Then .ShowAllData
        .Range("A1").AutoFilter Field:=2, Criteria1:="男"
        .Range("A1").AutoFilter Field:=3, Criteria1:=">=22", _
            Operator:=xlAnd, Criteria2:="<=26"
    End With
    Application.ScreenUpdating = True
End Sub
```

```
Sub BackUp()    '恢复
    Sheets("Sheet1").Range("A1").AutoFilter
End Sub
```

4.32 获取符合筛选条件的记录数量

```
Sub GetFilterRecordCount()
    Dim rngRange As Range, i As Long, lngCount As Long
    Dim lngAllCount As Long
    With ActiveSheet
        If .FilterMode Then
            lngAllCount = .AutoFilter.Range.Rows.Count - 1
            Set rngRange = .AutoFilter.Range.SpecialCells _
                (xlCellTypeVisible)
            For i = 1 To rngRange.Areas.Count
                lngCount = lngCount + _
                    rngRange.Areas(i).Rows.Count
            Next i
            Set rngRange = Nothing
            MsgBox "在 " & lngAllCount & " 条记录中找到 " & _
                lngCount - 1 & " 个"
        End If
    End With
End Sub
```

End With

End Sub

4.33 判断筛选结果是否为空

```
Sub DataFilter()    '自动筛选
    With Worksheets("Sheet1")
        If .FilterMode = True Then .ShowAllData
        .Range("A1").AutoFilter Field:=3, Criteria1:=">30", _
            Operator:=xlFilterValues
    End With
End Sub
```

```
Sub ShowData()    '取消自动筛选
    With Sheet1
        If .AutoFilterMode Then
            If .FilterMode = True Then .ShowAllData
        End If
    End With
End Sub
```

```
Sub FilterIsEmpty()    '筛选结果是否为空
    With ActiveSheet.AutoFilter.Range.SpecialCells _
        (xlCellTypeVisible)
        If .Areas.Count = 1 And .Rows.Count = 1 Then
            MsgBox "筛选结果为空"
        End If
    End With
End Sub
```

4.34 复制自动筛选后的数据区域

```
Sub DataFilter()    '自动筛选
    Worksheets("Sheet1").Range("A1").AutoFilter _
        Field:=2, Criteria1:="男", Operator:=xlFilterValues
End Sub
```

```
Sub CopyFilterResult()    '复制筛选结果
    With Worksheets("Sheet1")
        If .FilterMode Then
            .AutoFilter.Range.SpecialCells(xlCellTypeVisible).Copy _
                Worksheets("Sheet2").Range("A1")
        End If
    End With
End Sub
```

End Sub

4.36 删除空行

```
Sub RemoveDuplicates()    '获取不重复值
    Range("A1").CurrentRegion.Copy Range("E1")
    Range("E1").CurrentRegion.RemoveDuplicates _
        Columns:=1, Header:=xlYes
End Sub
```

```
Sub BackUp()    '清除
    Range("E1").CurrentRegion.Clear
End Sub
```

第 4 章 使用 Range 对象

4.37 判断是否选中整行

```
Private Sub Worksheet_SelectionChange(ByVal Target As Range)
    If Target.Rows.Count = 1 And Target.Areas.Count = 1 Then
        If Target.Cells.Count = Rows(1).Cells.Count Then
            MsgBox "您已选中了整行，当前行为第 " &
Target.Row & " 行"
        End If
    End If
End Sub
```

4.38 工作表中一次插入多行

Sub BatchInsertRows() ‘循环插入行

```
    Dim i As Integer
    For i = 1 To 3
        Rows(3).Insert
    Next i
End Sub
```

Sub InsertRows() ‘引用插入行

```
    Range("A3").EntireRow.Resize(3).Insert
End Sub
```

Sub InsertRows2()

```
    Rows(3).Resize(3).Insert
End Sub
```

Sub BackUp() ‘恢复

```
    Dim lngFirstRow As Long
    Dim lngLastRow As Long
    Dim i As Long
    lngFirstRow = ActiveSheet.UsedRange.Row
    lngLastRow = lngFirstRow +
ActiveSheet.UsedRange.Rows.Count - 1
    For i = lngLastRow To lngFirstRow Step -1
        If Application.WorksheetFunction.CountA(Rows(i)) = 0 Then
            Rows(i).Delete
        End If
    Next i
End Sub
```

4.39 控制插入单元格区域的格式

```
Sub InsertBlankRows()
    Rows(2).Resize(3).Insert _
CopyOrigin:=xlFormatFromRightOrBelow
End Sub
```

```
Sub InsertRowDefault()
    Rows(2).Resize(3).Insert
End Sub
```

```
Sub BackUp()
    Dim lngFirstRow As Long
    Dim lngLastRow As Long
    Dim i As Long
    lngFirstRow = ActiveSheet.UsedRange.Row
```

```
lngLastRow = lngFirstRow +
```

```
ActiveSheet.UsedRange.Rows.Count - 1
```

```
For i = lngLastRow To lngFirstRow Step -1
```

```
If Application.WorksheetFunction.CountA(Rows(i)) = 0 Then
    Rows(i).Delete
End If
Next i
```

```
End Sub
```

4.40 批量删除奇数行

Sub CycleDelete()

```
    Dim lngRow As Long, i As Long
    lngRow = Range("A" & Rows.Count).End(xlUp).Row
    If lngRow Mod 2 = 0 Then lngRow = lngRow - 1
    For i = lngRow To 2 Step -2
        Rows(i).Delete
    Next i
```

```
End Sub
```

Sub UnionDelete()

```
    Dim lngRow As Long, i As Long
    Dim rngRange As Range
    lngRow = Range("A" & Rows.Count).End(xlUp).Row
    Set rngRange = Rows(3)
    For i = 5 To lngRow Step 2
        Set rngRange = Application.Union(rngRange, Rows(i))
    Next i
    rngRange.Delete
    Set rngRange = Nothing
```

```
End Sub
```

Sub BackUp1()

```
    Sheets("Sheet1").Range("A1").CurrentRegion.Copy
```

```
Sheets("方法 1").Range("A1")
```

```
End Sub
```

Sub BackUp2()

```
    Sheets("Sheet1").Range("A1").CurrentRegion.Copy
```

```
Sheets("方法 2").Range("A1")
```

```
End Sub
```

4.41 数据排序

Sub SortDemo() ‘按总成绩排序

```
    Range("A1").Sort key1:="总成绩", order1:=xlDescending, _
Header:=xlYes
```

```
End Sub
```

Sub SortDemoA() ‘多关键字排序

```
    Range("A1").Sort key1:="总成绩", order1:=xlDescending, _
key2:="学科 1", order2:=xlDescending, _
key3:="学科 2", order3:=xlDescending, _
Header:=xlYes
```

```
End Sub
```

Sub BackUp() ‘恢复

```
    Range("L1").CurrentRegion.Copy Range("A1")
```

```
End Sub
```

第 4 章 使用 Range 对象

4.42 多关键字排序

Sub MoreKeySort() **‘Range Sort 方法 1**

```
With ActiveSheet.Sort.SortFields
    .Clear
    .Add Key:=Range("G1"), SortOn:=xlSortOnValues, _
        Order:=xlDescending
    .Add Key:=Range("B1"), SortOn:=xlSortOnValues, _
        Order:=xlDescending
    .Add Key:=Range("C1"), SortOn:=xlSortOnValues, _
        Order:=xlDescending
    .Add Key:=Range("D1"), SortOn:=xlSortOnValues, _
        Order:=xlDescending
End With
```

```
With ActiveSheet.Sort
    .SetRange Range("A1").CurrentRegion
    .Header = xlYes
    .Apply
End With
End Sub
```

Sub SortByKeysA() **‘Range Sort 方法 2**

```
With Range("A1")
    .Sort Key1:="心理学", order1:=xlDescending,
Header:=xlYes
    .Sort Key1:="教育学", order1:=xlDescending,
Header:=xlYes
    .Sort Key1:="基础知识", order1:=xlDescending,
Header:=xlYes
    .Sort Key1:="总成绩", order1:=xlDescending,
Header:=xlYes
End With
End Sub
```

Sub SortByKeysB() **‘Worksheet Sort 对象**

```
With Range("A1")
    .Sort Key1:="教育学", order1:=xlDescending, _
        Key2:="心理学", order2:=xlDescending,
Header:=xlYes
    .Sort Key1:="总成绩", order1:=xlDescending, _
        Key2:="基础知识", order2:=xlDescending,
Header:=xlYes
End With
End Sub
Sub BackUp() ‘恢复
    Range("O1").CurrentRegion.Copy Range("A1")
End Sub
```

4.43 自定义序列排序

Sub SortByLists()

```
Dim avntList As Variant, lngNum As Long
avntList = Range("E2:E6")
Application.AddCustomList avntList
lngNum = Application.GetCustomListNum(avntList)
Range("A1").Sort Key1:=Range("A1"), _
    Order1:=xlAscending, Header:=xlYes, _
    OrderCustom:=lngNum + 1
Application.DeleteCustomList lngNum
End Sub
Sub BackUp()
    Range("L1").CurrentRegion.Copy Range("a1")
End Sub
```

4.44 创建数据透视表

Sub NewPivotTable() **‘创建数据透视表**

```
Dim wksData As Worksheet
Dim objCache As PivotCache
Dim objTable As PivotTable
Dim avntArr() As Variant
Set wksData = Worksheets("Sheet1")
avntArr = wksData.Range("A1:E1")
Set objCache = ThisWorkbook.PivotCaches.Create( _
    xlDatabase, wksData.Range("a1").CurrentRegion. _
    Address(External:=True))
Set objTable = objCache.CreatePivotTable _
    (wksData.Range("H3"))
With objTable
    .AddFields RowFields:=Array(avntArr(1, 3)), _
        ColumnFields:=Array(avntArr(1, 2)), _
        PageFields:=Array(avntArr(1, 1), avntArr(1, 4))
    .AddDataField .PivotFields(avntArr(1, 5)), , xlSum
    .RowAxisLayout xlOutlineRow
End With
Set wksData = Nothing
Set objCache = Nothing
Set objTable = Nothing
```

End Sub

Sub DeleteTable() **‘删除数据透视表**

```
Columns("H:P").Delete
End Sub
```


第 5 章 使用 Chart 对象

5.1 自动创建图表

Option Explicit

Sub AddChartObject() ‘创建嵌入图

Dim chtObj As ChartObject

Set chtObj = ActiveSheet.ChartObjects.Add(100, 100, 400, 300)

With chtObj.Chart

.SetSourceData Source:=Range("A1:E3"),

PlotBy:=xlColumns

.ChartType = xlColumnClustered

.ApplyDataLabels

.HasTitle = True

.ChartTitle.Text = "产品销量"

End With

chtObj.Name = "产品销量图表"

Set chtObj = Nothing

End Sub

Sub AddChart() ‘创建图表工作表

Dim chtChart As Chart

Set chtChart = Charts.Add

With chtChart

.SetSourceData Source:=Sheet1.Range("A1:e3"),

PlotBy:=xlRows

.ChartType = xlColumnClustered

.HasDataTable = True

.ApplyDataLabels

.HasTitle = True

.ChartTitle.Text = "产品销量"

.HasLegend = False

.Name = "产品销量图表"

End With

Set chtChart = Nothing

End Sub

5.2 创建线柱组合图表

Option Explicit

Sub CreateChartObject2()

Dim chtObj As ChartObject

Dim wksSht As Worksheet

Dim serSeries As Series

Dim sngLeft As Single, sngTop As Single

Set wksSht = ActiveSheet

sngLeft = wksSht.Range("D7").Left

sngTop = wksSht.Range("D7").Top

Set chtObj = wksSht.ChartObjects.Add(sngLeft, sngTop,
450, 300)

With chtObj.Chart

.SetSourceData wksSht.Range("A1:B6"), xlColumns

.ChartType = xlColumnClustered

.HasTitle = True

.ChartTitle.Text = "各店销售情况"

With .Axes(xlValue, xlPrimary)

.MaximumScale = 1000

.MinimumScale = 0

.MajorUnit = 100

.HasTitle = True

.AxisTitle.Text = wksSht.Range("B1")

End With

Set serSeries = .SeriesCollection.NewSeries

With serSeries

.Values = wksSht.Range("C2:C6")

.ChartType = xlLineMarkers

.Name = wksSht.Range("C1")

.AxisGroup = xlSecondary

.MarkerStyle = xlMarkerStyleCircle

.MarkerSize = 5

End With

With .Axes(xlValue, xlSecondary)

.MaximumScale = 1

.MinimumScale = 0

.HasTitle = True

.AxisTitle.Text = wksSht.Range("C1")

End With

.SetElement

(msoElementPrimaryValueAxisTitleVertical)

.SetElement

(msoElementSecondaryValueAxisTitleVertical)

.SetElement (msoElementLegendBottom)

End With

Set chtObj = Nothing

Set wksSht = Nothing

Set serSeries = Nothing

End Sub

5.3 获取数据系列的引用区域

Option Explicit

Function rngSeries(ByVal intIndex As Integer, _

ByVal chtMyChart As Chart) As Range

Dim serSeries As Series

Set serSeries = chtMyChart.SeriesCollection(intIndex)

Set rngSeries = Range(VBA.Split(VBA. _
Split(serSeries.Formula, ",")(2), "!")(1))

Set serSeries = Nothing

End Function

Sub GetrngSeries()

Dim chtChart As Chart

Set chtChart = ActiveSheet.ChartObjects(1).Chart

MsgBox rngSeries(2, chtChart).Address

Set chtChart = Nothing

End Sub

第 5 章 使用 Chart 对象

5.4 自动添加平均值参考线

Sub AddNewSeries() '添加数据系列

```
Dim chtChart As Chart, serSeries As Series
Dim intPointsCount As Integer, i As Integer
Dim sngAverage As Single, strSeries As String
Set chtChart = ActiveSheet.ChartObjects(1).Chart
sngAverage = Application.WorksheetFunction. _
    Average(rngSeries(1, chtChart))
intPointsCount = chtChart.SeriesCollection(1).Points.Count
For i = 1 To intPointsCount
    strSeries = strSeries & "," & sngAverage
Next i
strSeries = Mid(strSeries, 2, Len(strSeries))
strSeries = "{=" & strSeries & "}"
Set serSeries = chtChart.SeriesCollection.NewSeries
serSeries.ChartType = xlLine
serSeries.Values = strSeries
Set serSeries = Nothing
Set chtChart = Nothing
```

End Sub

绘制线条

Function rngSeries(ByVal intIndex As Integer, _

```
ByVal cht As Chart) As Range
Dim serSeries As Series
Set serSeries = cht.SeriesCollection(intIndex)
Set rngSeries = Range(VBA.Split(VBA. _
    Split(serSeries.Formula, ",")(2), "!")(1))
Set serSeries = Nothing
```

End Function

5.5 自定义数据标签文本

Sub SetDataLabelText()

```
Dim rngLbl As Range, i As Integer
Set rngLbl = Me.Range("A2:A8")
With Me.ChartObjects(1).Chart.SeriesCollection(1)
    .HasDataLabels = True
    .DataLabels.Position = xlLabelPositionAbove
    For i = 1 To .Points.Count
        .Points(i).DataLabel.Text = rngLbl.Cells(i).Value
    Next i
End With
Set rngLbl = Nothing
```

End Sub

5.6 动态图表

Private Declare Sub Sleep Lib "kernel32.dll" _

(ByVal dwMilliseconds As Long)

Private Sub Worksheet_SelectionChange _

(ByVal Target As Range)

Dim chtChart As Chart

Dim serSeries As Series

Dim rngRange As Range

Dim i As Long

Static lngRow As Long

Set rngRange = Application.Intersect _
(ActiveCell, Me.Range("A2:M9"))

If Not rngRange Is Nothing And lngRow <> ActiveCell.Row

Then

lngRow = ActiveCell.Row

Set chtChart = Me.ChartObjects(1).Chart

Set serSeries = chtChart.SeriesCollection(1)

chtChart.ChartTitle.Text = Me.Range("A" &
lngRow).Value

For i = 1 To 12

serSeries.Values = Me.Range("B" &

lngRow).Resize(1, i)

DoEvents

Sleep 50

Next i

End If

Set rngRange = Nothing

Set serSeries = Nothing

Set chtChart = Nothing

End Sub

5.7 使用嵌入图表事件

Private WithEvents mchtChart As Chart

Private Sub mchtChart_Calculate()

Dim avntValues As Variant, i As Long

With mchtChart.SeriesCollection(1)

avntValues = .Values

For i = 1 To .Points.Count

If avntValues(i) > 80 Then

.Points(i).Format.Fill.ForeColor.RGB = RGB(0, 128, 255)

ElseIf avntValues(i) > 60 Then

.Points(i).Format.Fill.ForeColor.RGB = RGB(0, 128, 128)

Else

.Points(i).Format.Fill.ForeColor.RGB = RGB(255, 0, 0)

End If

Next

End With

End Sub

Private Sub mchtChart_Activate()

Application.StatusBar = "嵌入图表被激活."

End Sub

Private Sub mchtChart_Deactivate()

Application.StatusBar = False

End Sub

Private Sub Worksheet_Activate()

Set mchtChart = Me.ChartObjects(1).Chart

End Sub

Private Sub Worksheet_Deactivate()

Set mchtChart = Nothing

End Sub

第 5 章 使用 Chart 对象

5.8 条件格式化数据标记

Option Explicit

Private WithEvents mchtChart As Chart

Private Sub mchtChart_Calculate()

Dim objPoint As Point, i As Integer, intNum As Integer

For i = 1 To 6

intNum = Me.Cells(ActiveCell.Row, i + 1).Value

Set objPoint = mchtChart.SeriesCollection(1).Points(i)

If intNum >= 25 Then

Me.Shapes("Point25").CopyPicture xlScreen,

xlPicture

objPoint.Paste

ElseIf intNum >= 15 Then

Me.Shapes("Point15").CopyPicture xlScreen,

xlPicture

objPoint.Paste

Else: Me.Shapes("Point<").CopyPicture xlScreen,

xlPicture

objPoint.Paste

End If

Next i

Set objPoint = Nothing

End Sub

Private Sub Worksheet_SelectionChange(ByVal Target As Range)

Dim serSeries As Series

Dim rngRange As Range, lngRow As Long

Set rngRange = Application.Intersect(ActiveCell, [Data])

With Me

If Not rngRange Is Nothing Then

lngRow = ActiveCell.Row

Set mchtChart = .ChartObjects(1).Chart

Set serSeries

= .ChartObjects(1).Chart.SeriesCollection(1)

serSeries.Values = .Range(.Cells(lngRow, 2), _
.Cells(lngRow, 7))

.ChartObjects(1).Chart.HasTitle = True

.ChartObjects(1).Chart.ChartTitle.Text = _
.Cells(lngRow, 1).Value

End If

End With

Set rngRange = Nothing

Set serSeries = Nothing

Set mchtChart = Nothing

End Sub

5.9 将图表保存为图片

Option Explicit

Sub SaveAsPicture() ‘保存为图片

Dim chtObject As ChartObject

With ActiveSheet.ChartObjects("产品销量图表")

.CopyPicture

Set chtObject = ActiveSheet.ChartObjects. _

Add(500, 100, .Width, .Height)

End With

chtObject.Activate

chtObject.Chart.Paste

Set chtObject = Nothing

End Sub

Sub ExportToPicture() ‘输出图片文件

ActiveSheet.ChartObjects("产品销量图表").Chart.Export _
ThisWorkbook.Path & "\Picture.jpg"

End Sub

第 6 章 使用 Shape 对象

6.1 遍历工作表中的 Shape 对象

Sub ForNextAllShapes() ‘演示

Dim intRow As Integer, i As Integer

Dim strShapeTypeConst As String

intRow = 2

With Sheets("Shape 对象").Shapes

For i = 1 To .Count

With .Range(i)

Sheets("统计").Cells(intRow, 1) = i

Sheets("统计").Cells(intRow, 2) = .Name

Sheets("统计").Cells(intRow, 3) = .Type

Sheets("统计").Cells(intRow, 4) = .AutoShapeType

If .Type = 1 Then

Select Case .AutoShapeType

Case 1: strShapeTypeConst = "矩形"

Case 5: strShapeTypeConst = "圆角矩形"

Case 92: strShapeTypeConst = "五角星"

End Select

Else

Select Case .Type

Case 5: strShapeTypeConst = "任意多边形"

Case 8: strShapeTypeConst = "窗体控件"

Case 9: strShapeTypeConst = "线条"

Case 12: strShapeTypeConst = "OLE 控件对象"

Case 13: strShapeTypeConst = "图片"

End Select

End If

End With

Sheets("统计").Cells(intRow, 5) = strShapeTypeConst

intRow = intRow + 1

Next i

End With

End Sub

Sub ClearResults() ‘清除数据

Sheets("统计").Range("a2:e" & Sheets("统计"
").Range("a65536").End(3).Row).ClearContents

End Sub

第 6 章 使用 Shap 对象

6.1 遍历工作表中的 Shape 对象

```
Sub ForEachAllShapes()  
    Dim objShp As Shape  
    For Each objShp In Sheets("Shape 对象").Shapes  
        Debug.Print objShp.Name, Tab(30), objShp.Type, _  
            Tab(60), objShp.AutoShapeType  
    Next objShp  
    Set objShp = Nothing  
End Sub
```

6.2 在工作表中快速添加 Shape 对象

```
Sub InsertShape() '绘制图形  
    Dim intXOffset As Integer, intYOffset As Integer  
    Dim intRow As Integer, objLine As LineFormat  
    Dim objFreeForm As FreeformBuilder  
    intYOffset = 50: intXOffset = 50  
    With Sheets("数据")  
        For intRow = 2 To 11  
            Set objLine = Sheets("绘图区  
").Shapes.AddLine( _  
                .Cells(intRow, 1) + intXOffset, _  
                .Cells(intRow, 2) + intYOffset, _  
                .Cells(intRow + 1, 1) + intXOffset, _  
                .Cells(intRow + 1, 2) + intYOffset).Line  
            objLine.Weight = .Cells(intRow, 3)  
            objLine.ForeColor.RGB = .Cells(intRow, 4)  
        Next intRow  
    End With  
    intXOffset = intXOffset + 300  
    intRow = 2  
    With Sheets("数据")  
        Set objFreeForm = Sheets("绘图区  
").Shapes.BuildFreeform( _  
            msoEditingAuto, _  
            .Cells(intRow, 1) + intXOffset, _  
            .Cells(intRow, 2) + intYOffset)  
        For intRow = 3 To 12  
            objFreeForm.AddNodes msoSegmentLine,  
msoEditingAuto, _  
                .Cells(intRow, 1) + intXOffset, _  
                .Cells(intRow, 2) + intYOffset  
        Next intRow  
        objFreeForm.ConvertToShape  
    End With  
    intXOffset = intXOffset + 300  
    Sheets("绘图区").Shapes.AddShape(msoShape5pointStar,  
_intXOffset, intYOffset, 266.3, 253.26).Select  
    Selection.ShapeRange.Fill.ForeColor.RGB = RGB(255, 0, 0)  
    Set objLine = Nothing  
    Set objFreeForm = Nothing  
End Sub
```

```
Sub ClearShape() '清除图形  
    Dim objShp As Object  
    For Each objShp In Sheets("绘图区").DrawingObjects  
        If objShp.ShapeRange.AutoShapeType <> 5 Then  
objShp.Delete  
        Next objShp  
        Set objShp = Nothing  
    End Sub
```

6.3 组合多个 Shape 对象

```
Sub GroupShapes() '组合  
    Dim i As Integer  
    Dim astrLineName(1 To 10) As String  
    For i = 1 To 10  
        astrLineName(i) = "直接连接符 " & i  
    Next i  
    With Sheet1.Shapes  
        .Range(astrLineName()).Group.Placement = xlFreeFloating  
        .Range(Array("任意多边形 11", "五角星  
12")).Group.Placement = xlFreeFloating  
    End With  
    Sheet1.Shapes.SelectAll  
End Sub
```

```
Sub UngroupShapes() '取消组合  
    Dim objShp As Shape  
    For Each objShp In Sheet1.Shapes  
        If objShp.Name Like "Group *" Then objShp.Ungroup  
    Next objShp  
    Set objShp = Nothing  
    Sheet1.Shapes.SelectAll  
End Sub
```

6.4 将 Shape 对象另存为图片

```
Sub ExportShpToGIF()  
    Dim objShp As Shape  
    Dim objCht As ChartObject  
    For Each objShp In Sheet1.Shapes  
        objShp.Copy  
        Set objCht = Sheet1.ChartObjects.Add(0, 0,  
objShp.Width, _  
            objShp.Height)  
        objCht.Select  
        With objCht.Chart  
            .Paste  
            .Export ThisWorkbook.Path & "\" &  
objShp.Name & ".gif"  
            .Parent.Delete  
        End With  
        Next objShp  
        Set objShp = Nothing  
        Set objCht = Nothing  
        MsgBox "图片导出完毕！", vbOKOnly, "将 Shape 对象另  
存为图片"  
    End Sub
```

第 6 章 使用 Shap 对象

6.5 编辑 Shape 对象显示的文本

Sub EditCaption() ‘演示

Dim objShp As Shape

For Each objShp In Sheet1.Shapes

Select Case objShp.Type

Case 1

If objShp.TextFrame2.TextRange.Text = "圆角矩形" Then

objShp.TextFrame2.TextRange.Text = "图形按钮"

Sheet1.Hyperlinks.Add Anchor:=objShp, Address:= "",

SubAddress:="EditCaption"

End If

Case 8

If objShp.FormControlType = xlCheckBox Then

objShp.OLEFormat.Object.Caption = "表单控件"

objShp.OLEFormat.Object.Value = True

Elseif objShp.FormControlType = xlButtonControl Then

objShp.OLEFormat.Object.Caption = "表单按钮"

objShp.OnAction = "Test"

End If

Case 12

With Sheet1.OLEObjects(objShp.Name)

If .progID Like "Forms.CheckBox*" Then

.Object.Caption = "ActiveX 控件"

.Object.Value = True

End If

End With

End Select

Next objShp

Set objShp = Nothing

End Sub

Sub Restore() ‘恢复

Dim objShp As Shape

For Each objShp In Sheet1.Shapes

Select Case objShp.Type

Case 1

If objShp.TextFrame2.TextRange.Text = "图形按钮" Then

objShp.TextFrame2.TextRange.Text = "圆角矩形"

Sheet1.Hyperlinks.Delete

End If

Case 8

If objShp.FormControlType = xlButtonControl Then

objShp.OLEFormat.Object.Caption = "按钮 1"

objShp.OnAction = ""

Elseif objShp.FormControlType = xlCheckBox Then

objShp.OLEFormat.Object.Caption = "复选框 1"

objShp.OLEFormat.Object.Value = False

End If

Case 12

With Sheet1.OLEObjects(objShp.Name)

If .progID Like "Forms.CheckBox*" Then

.Object.Caption = "CheckBox1"

.Object.Value = False

End If

End With

End Select

Next objShp

End Sub

第 7 章 使用消息框

7.1 显示简单的信息提示

Sub SimpleMsgBox()

MsgBox "Hello VBA ! "

End Sub

7.2 定制个性化的消息框

Sub CustomMsgbox()

MsgBox Prompt:="欢迎光临 Excel Home!", _

Buttons:=vbOKCancel + vbInformation + vbDefaultButton2, _

Title:="Excel Home"

End Sub

7.3 获取消息框的返回值

Sub ReturnValueForMsgbox()

Dim intMsg As Integer

intMsg = MsgBox("文件即将关闭,是否保存?", _

vbYesNoCancel + vbQuestion)

Select Case intMsg

Case vbYes

ThisWorkbook.Save

Case vbNo

ThisWorkbook.Saved = True

Case vbCancel

Exit Sub

End Select

ThisWorkbook.Close

End Sub

7.4 自动延时关闭的消息框

Private Declare Function MessageBoxTimeout Lib "user32"

Alias "MessageBoxTimeoutA" (ByVal hwnd As Long, ByVal

lpText As String, ByVal lpCaption As String, ByVal wType As Long,

ByVal wlange As Long, ByVal dwTimeout As Long) As Long

Sub AutoClose2() ‘自动关闭消息框 2

MessageBoxTimeout 0, "程序执行完毕, 两秒后关闭!", _

"自动关闭的消息框 2", 0, 0, 2000

End Sub

Sub AutoClose1() ‘自动关闭消息框 1

Dim objShell As Object

Set objShell = CreateObject("Wscript.Shell")

objShell.Popup "程序执行完毕, 两秒后关闭!", 2, _

"自动关闭的消息框 1", 64

Set objShell = Nothing

End Sub

第 8 章 简单的数据输入

8.1 简单的输入界面

```
Sub SimpleInput()  
    Dim strInput As String  
    strInput = InputBox("请输入邮政编码:", "邮政编码",  
"100001")  
    If StrPtr(strInput) <> 0 Then  
        If IsNumeric(Trim(strInput)) = True Then  
            If Len(Trim(strInput)) = 6 Then  
                Cells(1, 1) = strInput  
            Else  
                MsgBox "邮政编码必须是 6 位!"  
            End If  
        Else  
            MsgBox "邮政编码必须是数值!"  
        End If  
    Else  
        MsgBox "你已放弃了输入!"  
    End If  
End Sub
```

8.2 更安全的密码输入界面

```
Public Declare Function FindWindow Lib "user32" Alias
```

```
"FindWindowA" (ByVal lpClassName As String, ByVal  
lpWindowName As String) As Long
```

```
Public Declare Function FindWindowEx Lib "user32" Alias  
"FindWindowExA" (ByVal hWnd1 As Long, ByVal hWnd2 As  
Long, ByVal lpsz1 As String, ByVal lpsz2 As String) As Long
```

```
Public Declare Function SendMessage Lib "user32" Alias  
"SendMessageA" (ByVal hwnd As Long, ByVal wParam As Long,  
ByVal lParam As Any) As Long
```

```
Public Declare Function timeSetEvent Lib "winmm.dll" (ByVal  
uDelay As Long, ByVal uResolution As Long, ByVal lpFunction_  
As Long, ByVal dwUser As Long, ByVal uFlags As Long) As Long
```

```
Public Declare Function timeKillEvent Lib "winmm.dll" (ByVal  
uID As Long) As Long  
Public Const EM_SETPASSWORDCHAR = &HCC
```

```
Public lngTimeID As Long  
Sub TimeProc(ByVal uID As Long, ByVal uMsg As Long, ByVal  
dwUser As Long, ByVal dw1 As Long, ByVal dw2 As Long)  
    Dim lngHwd As Long  
    lngHwd = FindWindow("#32770", "密码")  
    If lngHwd <> 0 Then  
        lngHwd = FindWindowEx(lngHwd, 0, "edit",  
vbNullString)
```

```
SendMessage lngHwd, EM_SETPASSWORDCHAR, 42, 0
```

```
timeKillEvent lngTimeID
```

```
End If
```

```
End Sub
```

```
Sub Password()
```

```
    Dim strPassword As String
```

```
    lngTimeID = timeSetEvent(10, 0, AddressOf TimeProc,  
1, 1)
```

```
line:
```

```
    strPassword = InputBox("请输入密码:", "密码")
```

```
    If StrPtr(strPassword) <> 0 Then
```

```
        If strPassword = "123456" Then
```

```
            MsgBox "密码正确!"
```

```
        Else
```

```
            MsgBox "密码不正确!"
```

```
            GoTo line
```

```
        End If
```

```
    End If
```

```
End Sub
```

```
Public Sub ShowFrom()
```

```
    frmPassWord.Show
```

```
End Sub
```

8.3 轻松获取单元格区域地址

```
Sub RngInput()
```

```
    Dim rngRange As Range
```

```
    On Error GoTo line
```

```
    Set rngRange = Application.InputBox(Prompt:="区域选  
择:", Type:=8)
```

```
    rngRange.Interior.Color = 255
```

```
line:
```

```
End Sub
```

8.4 防止用户输入错误数据

```
Sub InputNumber()
```

```
    Dim vntInput As Variant
```

```
    vntInput = Application.InputBox(Prompt:="请输入数字:",  
Type:=1)
```

```
    If TypeName(vntInput) = "Boolean" Then
```

```
        MsgBox "你已取消了输入!"
```

```
    Else
```

```
        Cells(1, 1) = vntInput
```

```
    End If
```

```
End Sub
```


第 9 章 内置对话框

9.1 使用 Excel 的内置对话框

```
Sub SetDisplay()  
    If Application.Dialogs(xlDialogDisplay).Show( _  
        arg1:=True, arg2:=True, arg3:=False, arg4:=False, _  
        arg5:=3, arg7:=False, arg8:=True, arg9:=1) = True Then  
        MsgBox "显示选项修改完毕"  
    End If  
End Sub  
Sub Restore()  
    If Application.Dialogs(xlDialogDisplay).Show( _  
        arg1:=False, arg2:=True, arg3:=True, arg4:=True, _  
        arg5:=0, arg7:=True, arg8:=False, arg9:=1) = True Then  
        MsgBox "显示选项恢复初始状态"  
    End If  
End Sub
```

9.2 获取用户选择的文件名

```
Sub SelectFile()  
    Dim vntFilename As Variant  
    Dim i As Integer  
    vntFilename = Application.GetOpenFilename(Title:="浏览  
文件", _  
        FileFilter:="所有文件 (*.*)*,*.*,Excel 文件  
(*.*),*.xls*", _  
        FilterIndex:=2, MultiSelect:=True)  
    If IsArray(vntFilename) = True Then  
        With Sheet1  
            .Cells.ClearContents  
            .Cells(1, 1) = "文件名"  
            For i = 1 To UBound(vntFilename)  
                .Cells(i + 1, 1) = vntFilename(i)  
            Next i  
        End With  
    End If  
End Sub
```

9.3 获取用户选择的文件夹名

```
Sub SelectFolder()  
    Dim objDialog As FileDialog  
    Set objDialog =  
Application.FileDialog(msoFileDialogFolderPicker)  
    With objDialog  
        .ButtonName = "就它了"  
        .Title = "浏览文件夹"  
        If .Show = True Then  
            MsgBox .SelectedItems(1)  
        End If  
    End With  
    Set objDialog = Nothing  
End Sub
```

第 10 章 菜单和工具栏

10.1 禁用右键快捷菜单

```
Private Sub Workbook_BeforeClose(Cancel As Boolean)  
    Dim objBar As CommandBar  
    For Each objBar In Application.CommandBars  
        If objBar.Type = msoBarTypePopup Then  
            objBar.Enabled = True  
        End If  
    Next objBar  
    Set objBar = Nothing  
End Sub
```

```
Private Sub Workbook_Open()  
    Dim objBar As CommandBar  
    For Each objBar In Application.CommandBars  
        If objBar.Type = msoBarTypePopup Then  
            objBar.Enabled = False  
        End If  
    Next objBar  
    Set objBar = Nothing  
End Sub
```

10.2 列出所有命令栏控件

```
Sub ControlList()  
    Dim objBar As CommandBar  
    Dim objCtrl As CommandBarControl  
    Dim intRow As Integer  
    intRow = 2  
    With Sheet1  
        .Range("a2:e"  
& .Range("a65536").End(3).Row).ClearContents  
        For Each objBar In Application.CommandBars  
            If objBar.BuiltIn = True Then  
                objBar.Reset  
                For Each objCtrl In objBar.Controls  
                    If objCtrl.BuiltIn = True Then  
                        .Cells(intRow, 1) = objCtrl.Caption  
                        .Cells(intRow, 2) = objCtrl.ID  
                        On Error Resume Next  
                        .Cells(intRow, 3) = objCtrl.FaceId  
                        On Error GoTo 0  
                        .Cells(intRow, 4) = objCtrl.Type  
                        .Cells(intRow, 5) = objBar.Name  
                        intRow = intRow + 1  
                    End If  
                Next objCtrl  
            End If  
        Next objBar  
        Set objBar = Nothing  
        Set objCtrl = Nothing  
    End With  
End Sub
```

第 10 章 菜单和工具栏

10.3 自定义菜单命令组

```
Private Sub Workbook_Open()  
    Dim objBarPopup As CommandBarPopup  
    Dim avntName As Variant  
    Dim avntFacId As Variant  
    Dim i As Integer  
    avntName = Array("七年级", "八年级", "九年级")  
    avntFacId = Array(281, 283, 285)  
    With Application.CommandBars.Item("Worksheet menu  
bar")  
        .Reset  
        Set objBarPopup = .Controls.Add(msoControlPopup)  
        With objBarPopup  
            .Caption = "选择年级"  
            For i = 0 To UBound(avntName)  
                With .Controls.Add(msoControlButton)  
                    .Caption = avntName(i)  
                    .FacId = avntFacId(i)  
                    .OnAction = "Test"  
                End With  
            Next i  
        End With  
    End With  
    Set objBarPopup = Nothing  
End Sub
```

10.4 自定义工具栏组

```
Private Sub Workbook_BeforeClose(Cancel As Boolean)  
    On Error Resume Next  
    Application.CommandBars.Item("MyToolbar").Delete  
End Sub  
Private Sub Workbook_Open()  
    Dim objCtrl As CommandBarComboBox  
    Dim objBar As CommandBar  
    Dim avntName As Variant  
    Dim i As Integer  
    On Error Resume Next  
    Application.CommandBars.Item("MyToolbar").Delete  
    On Error GoTo 0  
    avntName = Array("七年级", "八年级", "九年级")  
    Set objBar = Application.CommandBars.Add("MyToolbar")  
    objBar.Visible = True  
    Set objCtrl = objBar.Controls.Add(msoControlDropdown)  
    With objCtrl  
        For i = 0 To UBound(avntName)  
            .AddItem avntName(i)  
        Next i  
        .Caption = "选择年级"  
        .OnAction = "Test"  
        .Style = msoComboLabel  
    End With  
    Set objCtrl = Nothing  
    Set objBar = Nothing  
End Sub
```

```
Private Sub Workbook_BeforeClose(Cancel As Boolean)  
    Application.CommandBars("cell").Reset  
End Sub
```

10.5 自定义单元格右键快捷菜单

```
Private Sub Workbook_Open()  
    Dim objCtrl As CommandBarControl  
    Dim objButton As CommandBarButton  
    Dim objPicture As IPictureDisp  
    Dim objMask As IPictureDisp  
    Set objPicture = LoadPicture(ThisWorkbook.Path &  
"\image.bmp")  
    Set objMask = LoadPicture(ThisWorkbook.Path &  
"\mask.bmp")  
    With Application.CommandBars("cell")  
        .Reset  
        Set objButton  
        = .Controls.Add(Type:=msoControlButton, _  
            ID:=2521, Temporary:=True)  
        objButton.BeginGroup = True  
        With .Controls.Add(Type:=msoControlButton, _  
            before:=3, ID:=1, Temporary:=True)  
            .Caption = "自定义菜单项 2"  
            .FacId = 17  
            .OnAction = "Test"  
        End With  
        With .Controls.Add(Type:=msoControlButton, _  
            before:=3, ID:=1, Temporary:=True)  
            .Caption = "自定义菜单项 1"  
            .OnAction = "Test"  
            .Picture = objPicture  
            .Mask = objMask  
        End With  
        For Each objCtrl In .Controls  
            If objCtrl.ID = 19 Then objCtrl.Enabled = False  
            If objCtrl.ID = 21 Then objCtrl.Visible = False  
            If objCtrl.ID = 292 Then objCtrl.OnAction =  
"Delete"  
            Next objCtrl  
        End With  
        Set objCtrl = Nothing  
        Set objButton = Nothing  
        Set objPicture = Nothing  
        Set objMask = Nothing  
    End Sub  
Sub Test()  
    MsgBox "这是一个自定义菜单项"  
End Sub  
Sub Delete()  
    MsgBox "该控件原有功能已经失效！"  
End Sub
```

第 10 章 菜单和工具栏

10.6 使用自定义右键快捷菜单输入数据

```
Private Sub Workbook_BeforeClose(Cancel As Boolean)
    On Error Resume Next
    Application.CommandBars("MyPopupMenu").Delete
End Sub

Private Sub Workbook_Open()
    Dim i As Integer, avntClass As Variant
    Dim objPopupMenu As CommandBar
    On Error Resume Next
    Application.CommandBars("MyPopupMenu").Delete
    On Error GoTo 0
    avntClass = Array("一班", "二班", "三班", "四班", _
        "实验班", "实验一班", "实验二班")
    Set objPopupMenu = Application.CommandBars.Add("MyPopupMenu", _
        msoBarPopup)
    For i = 0 To 3
        With objPopupMenu.Controls.Add(Type:=msoControlButton)
            .Caption = avntClass(i)
            .OnAction = "ActionForMenu"
        End With
    Next i
    With objPopupMenu.Controls.Add(Type:=msoControlPopup)
        .Caption = avntClass(4)
        .BeginGroup = True
        For i = 5 To 6
            With .Controls.Add(Type:=msoControlButton)
                .Caption = avntClass(i)
                .OnAction = "ActionForMenu"
            End With
        Next i
    End With
    Set objPopupMenu = Nothing
End Sub

Sub ActionForMenu()
    ActiveCell =
Application.CommandBars.ActionControl.Caption
End Sub
```

10.7 快速列出操作系统中所有字体

```
Sub ShowInstalledFonts()
    Dim i As Long, objFontList As CommandBarControl
    Set objFontList =
Application.CommandBars.FindControl(ID:=1728)
    Range("A:B").ClearContents
    Range("A1:B1") = Array("字体名称", "示例")
    For i = 1 To objFontList.ListCount
        Cells(i + 1, 1) = objFontList.List(i)
        Cells(i + 1, 2) = objFontList.List(i)
        Cells(i + 1, 2).Font.Name = objFontList.List(i)
    Next i
    Set objFontList = Nothing
End Sub
```

第 11 章 Ribbon 功能区

11.1.1 自定义功能区界面（略）

11.1.2 内置控件（附表）

11.2 编写 VBA 代码处理回调

```
Option Explicit
Public pobjRib As IRibbonUI
'Callback for customUI.onLoad
Sub RibbonOnLoaded(ribbon As IRibbonUI)
    Set pobjRib = ribbon
    pobjRib.ActivateTab "MyTab"
End Sub
'Callback for MyButton onAction
Sub rxMybtn_Click(control As IRibbonControl)
    MsgBox "我是一个大尺寸按钮"
End Sub
'Callback for Button1 onAction
Sub rxbtn_Click(control As IRibbonControl)
    If control.ID = "Button1" Then
        MsgBox "我是【按钮 1】"
    ElseIf control.ID = "Button2" Then
        MsgBox "我是【按钮 2】"
    End If
End Sub
```

11.3 使用自定义图片和内置图标

11.3.1 内置控件图标名称（附表）

11.3.2 使用自定义图片和内置图标

```
Sub rxbtn_Click(control As IRibbonControl)
    If control.ID = "Button1" Then
        MsgBox "我使用了自定义图标"
    Else
        MsgBox "我使用了内置图标"
    End If
End Sub

11.4.1 动态禁用和启用控件
Public pobjRib As IRibbonUI
Public pblnPressed As Boolean
Sub RibbonOnLoaded(ribbon As IRibbonUI)
    Set pobjRib = ribbon
    pblnPressed = True
End Sub
Sub rxtglbtn_Click(control As IRibbonControl, pressed As Boolean)
    pblnPressed = Not pblnPressed
    pressed = pblnPressed
    pobjRib.InvalidateControl "tglbtn"
    pobjRib.InvalidateControl "MyButton"
End Sub
Sub rxtglbtn_GetLabel(control As IRibbonControl, ByRef returnedVal)
    If pblnPressed = True Then
        returnedVal = "禁用 【Button】"
    Else
        returnedVal = "启用 【Button】"
    End If
End Sub
Sub rxtglbtn_getPressed(control As IRibbonControl, ByRef returnedVal)
    returnedVal = pblnPressed
End Sub
Sub rxbtn_getEnabled(control As IRibbonControl, ByRef returnedVal)
    returnedVal = pblnPressed
End Sub
```

第 11 章 Ribbon 功能

11.4.2 隐藏指定的选项卡

Option Explicit

Public pobjRib As IRibbonUI

Public pblnHome As Boolean

Public pblnCustom As Boolean

'Callback for customUI.onLoad

Sub RibbonOnLoad(ribbon As IRibbonUI)

 pblnHome = True

 pblnCustom = True

 Set pobjRib = ribbon

End Sub

'Callback for TabHome getVisible

Sub rxvisible(control As IRibbonControl, ByRef returnedVal)

 If control.ID = "TabHome" Then

 returnedVal = pblnHome

 Else

 returnedVal = pblnCustom

 End If

End Sub

'Callback for chkhome getPressed

Sub rxpressed(control As IRibbonControl, ByRef returnedVal)

 If control.ID = "chkhome" Then

 returnedVal = pblnHome

 Else

 returnedVal = pblnCustom

 End If

End Sub

'Callback for chkhome onAction

Sub rxchk_Click(control As IRibbonControl, pressed As Boolean)

 If control.ID = "chkhome" Then

 pblnHome = Not pblnHome

 pobjRib.InvalidateControlMso "TabHome"

 Else

 pblnCustom = Not pblnCustom

 pobjRib.InvalidateControl "MyTab"

 End If

End Sub

Sub ShowTab()

 pblnHome = True

 pobjRib.InvalidateControlMso "TabHome"

End Sub

Sub HideTab() **隐藏**

 pblnHome = False

 pobjRib.InvalidateControlMso "TabHome"

End Sub

11.4.3 隐藏指定的选项卡组(附加示例)

Option Explicit

Public pobjRib As IRibbonUI

Public pblnClipboard As Boolean

Public pblnCustom As Boolean

Sub RibbonOnLoad(Ribbon As IRibbonUI)

 pblnClipboard = False

 pblnCustom = False

 Set pobjRib = Ribbon

End Sub

Sub rxvisible(control As IRibbonControl, ByRef returnedVal)

 If control.ID = "GroupClipboard" Then

 returnedVal = pblnClipboard

 Else

 returnedVal = pblnCustom

 End If

End Sub

Sub rxpressed(control As IRibbonControl, ByRef returnedVal)

 If control.ID = "chkclipboard" Then

 returnedVal = pblnClipboard

 Else

 returnedVal = pblnCustom

 End If

End Sub

Sub rxchk_Click(control As IRibbonControl, pressed As Boolean)

 If control.ID = "chkclipboard" Then

 pblnClipboard = Not pblnClipboard

 pobjRib.InvalidateControlMso "GroupClipboard"

 Else

 pblnCustom = Not pblnCustom

 pobjRib.InvalidateControl "Group"

 End If

End Sub

Sub rxbtn_Click(control As IRibbonControl)

 MsgBox "你单击了我！"

End Sub

11.5 自定义功能区提示信息加速键

Option Explicit

Sub rxbtn_Click(control As IRibbonControl)

 MsgBox "我是一个菜单项"

End Sub

'Callback for MyButton getSupertip

Sub rxgetSupertip(control As IRibbonControl, ByRef returnedVal)

End Sub

'Callback for MyButton getScreentip

Sub rxgetScreentip(control As IRibbonControl, ByRef returnedVal)

End Sub

第 11 章 Ribbon 功能

11.7 使用 VBA 操作功能区

11.7.1 使用 CommandBars 对象

```
Sub GetEnabled() '命令是否禁用
    If
        Application.CommandBars.GetEnabledMso("MergeCenterMenu") = True Then
            MsgBox "【合并单元格】启用"
        Else
            MsgBox "【合并单元格】禁用"
        End If
    End Sub

Sub FullScreen() '显示模式
    Application.CommandBars.ExecuteMso
    ("ViewFullScreenView")
End Sub
```

11.7.2 隐藏功能区

```
Sub ShowTab()
    Application.ExecuteExcel4Macro
    "show.toolbar("""ribbon""",true)"
End Sub

Sub HideTab()
    Application.ExecuteExcel4Macro
    "show.toolbar("""ribbon""",false)"
End Sub
```

11.8 重置功能区内置命令

```
Sub rxCut_Click(control As IRibbonControl, ByRef cancelDefault)
    With ActiveSheet.[a1].Font
        If MsgBox("增大字号后继续执行剪切吗?", _
            vbYesNo + vbInformation) = vbYes Then
            cancelDefault = False
            .Size = .Size + 3
        Else
            cancelDefault = True
            .Size = .Size + 3
        End If
    End With
End Sub
```

11.10 自定义【文件】选项卡

```
Sub rxbtn_Click(control As IRibbonControl)
    MsgBox "我是一个测试按钮"
End Sub
```

11.12 使用对话框启动器

'Callback for boxbutton onAction

```
Sub rxboxbtn_Click(control As IRibbonControl)
    TestForm.Show
End Sub
```

11.13 使用组合框

```
Option Explicit
Public pobjRib As IRibbonUI
Sub RibbonOnLoaded(ribbon As IRibbonUI)
    Set pobjRib = ribbon
End Sub
Sub rxcmbCount(control As IRibbonControl, ByRef returnedVal)
    returnedVal = ActiveSheet.Comments.Count
    Application.OnTime DateAdd("s", 1, Now), "Refresh"
End Sub
Sub rxcmbID(control As IRibbonControl, index As Integer, ByRef returnedVal)
    returnedVal = "批注" & index
End Sub
```

```
Sub rxcmbLabel(control As IRibbonControl, index As Integer,
ByRef returnedVal)
```

```
    returnedVal = ActiveSheet.Comments(index +
1).Parent.Address(0, 0) & "的批注"
```

```
End Sub
```

```
Sub rxcmbText(control As IRibbonControl, ByRef returnedVal)
```

```
    On Error Resume Next
```

```
    returnedVal = ActiveCell.Comment.text
```

```
    returnedVal = ActiveCell.Address(0, 0) & "的批注"
```

```
    If Err.Number <> 0 Then returnedVal = ""
```

```
End Sub
```

```
Sub rxcmbSupertip(control As IRibbonControl, index As Integer,
ByRef returnedVal)
```

```
    returnedVal = ActiveSheet.Comments(index + 1).text
```

```
End Sub
```

```
Sub rxcmb_Click(control As IRibbonControl, text As String)
```

```
    On Error Resume Next
```

```
    Range(Split(text, "的")(0)).Select
```

```
    If Err.Number <> 0 Then pobjRib.InvalidateControl "cmb2"
```

```
End Sub
```

```
Sub Refresh()
```

```
    pobjRib.InvalidateControl "cmb2"
```

```
End Sub
```

11.14 使用动态菜单

Option Explicit

Public pobjRib As IRibbonUI

Public plngNum As Long

Sub RibbonOnLoaded(ribbon As IRibbonUI)

```
    Set pobjRib = ribbon
```

```
End Sub
```

```
Sub rxdynmenuContent(control As IRibbonControl, ByRef
returnedVal)
```

```
    Dim pobjShp As Shape
```

```
    Dim strXML As String
```

```
    strXML = "<menu
```

```
xmlns=""http://schemas.microsoft.com/" & _
```

```
    "office/2009/07/customui"" itemSize=""large"">"
```

```
    If ActiveSheet.Shapes.Count > 0 Then
```

```
        For Each pobjShp In ActiveSheet.Shapes
```

```
            strXML = strXML & "<button id=""shape" & _
```

```
                plngNum & """" & "label=""& _
```

```
pobjShp.Name & _
```

```
                """" & "onAction=""rxbtn_Click""/>"
```

```
                plngNum = plngNum + 1
```

```
        Next pobjShp
```

```
    Else
```

```
        strXML = strXML & "<button id=""no"" label=""没有图形""/>"
```

```
    End If
```

```
    returnedVal = strXML & "</menu>"
```

```
End Sub
```

```
Sub rxbtn_Click(control As IRibbonControl)
```

```
    ActiveSheet.Shapes(Replace(control.ID, "shape", "")) +
```

```
1).Select
```

```
End Sub
```

第 12 章 控件的应用

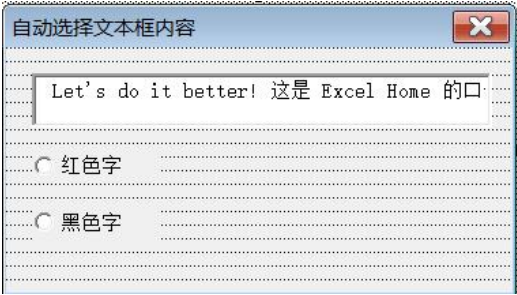
12.1 限制文本框的输入

```
Option Explicit
Public pobjRib As IRibbonUI
Public plngNum As Long
Sub RibbonOnLoaded(ribbon As IRibbonUI)
    Set pobjRib = ribbon
End Sub
Sub rxdynmenuContent(control As IRibbonControl, ByRef
returnedVal)
    Dim pobjShp As Shape
    Dim strXML As String
    strXML = "<menu
xmlns=""http://schemas.microsoft.com/" & _
        "office/2009/07/customui"" itemSize=""large"">"
    If ActiveSheet.Shapes.Count > 0 Then
        For Each pobjShp In ActiveSheet.Shapes
            strXML = strXML & "<button id=""shape" & _
                plngNum & """" & "label=""& _
pobjShp.Name & _
                """" & "onAction=""rxbtn_Click""/>"
            plngNum = plngNum + 1
        Next pobjShp
    Else
        strXML = strXML & "<button id=""no"" label=""没有
图形""/>"
    End If
    returnedVal = strXML & "</menu>"
End Sub
Sub rxbtn_Click(control As IRibbonControl)
    ActiveSheet.Shapes(Replace(control.ID, "shape", "") +
1).Select
End Sub
Sub FormShow() '显示文本框
    frmLimitText.Show
End Sub
```

12.2 自动换行的文本框

```
Option Explicit
Sub FormShow()
    frmNewLine.Show
End Sub
```

12.3 自动选择文本框内容



```
Option Explicit
Sub FormShow()
    frmAutoChoose.Show
End Sub
```

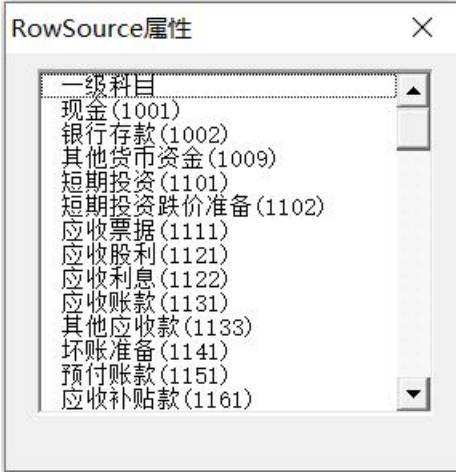
End Sub

12.4 制作游走字幕



```
Option Explicit
Sub FormShow()
    frmWander.Show
End Sub
```

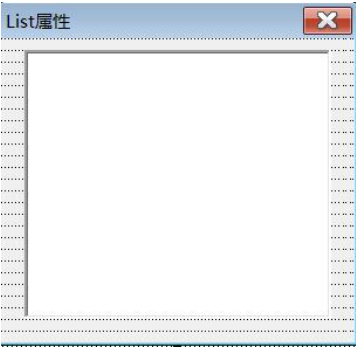
12.5.1 使用 RowSource 属性添加列表项



12.5.2 使用 ListFillRange 属性添加列表项

```
Option Explicit
Private Sub Worksheet_Activate()
    Dim lngLastRow As Long
    lngLastRow = Sheets("数据源").Cells(Rows.Count,
1).End(xlUp).Row
    Me.ListBox1.ListFillRange = "数据源!a1:a" & lngLastRow
End Sub
```

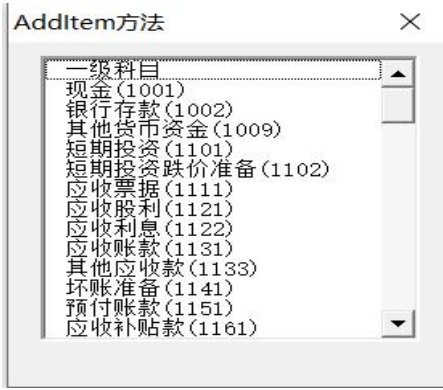
12.5.3 使用 List 属性添加列表项



```
Option Explicit
Sub FormShow()
    frmListDemo.Show
End Sub
```

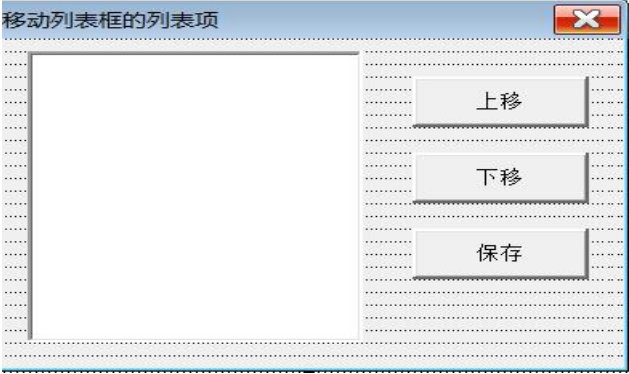

第 12 章 控件的应用

12.5.4 使用 AddItem 方法添加列表项



```
Sub FormShow()  
    frmAddItem.Show  
End Sub
```

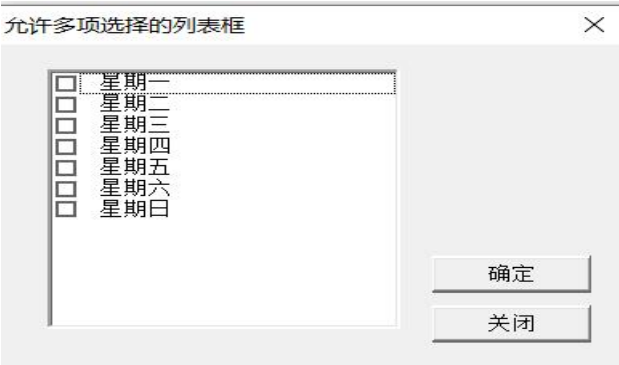
12.6 移动列表框的列表项



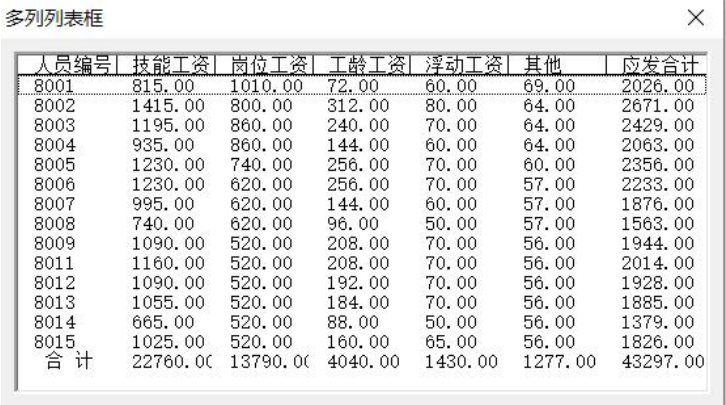
```
Option Explicit  
Sub FormShow()  
    frmMoveListItem.Show  
End Sub
```

12.7 允许多项选择的列表框

```
Option Explicit  
Sub FormShow()  
    frmMultiterm.Show  
End Sub
```

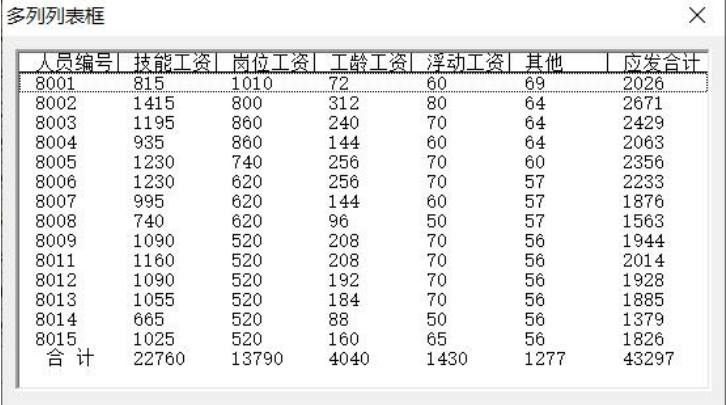


12.8.1 为多列组合框和列表框添加列表项



```
Option Explicit  
Sub FormShow()  
    frmMultipleColumns.Show  
End Sub
```

12.8.2 将多列列表框的数据写入工作表



```
Sub FormShow()  
    frmListToSheet.Show  
End Sub
```

12.9 二级组合框



```
Sub FormShow()  
    frmSecondary.Show  
End Sub
```

第 12 章 控件的应用

12.10 输入时逐步提示信息

Option Explicit

Private Sub Worksheet_SelectionChange(ByVal Target As Range)

‘示例

```
Dim i As Long
If Target.Count = 1 Then
    If Not Application.Intersect(Target, Range("A2:A12"))
Is Nothing Then
        With Me.txtValue
            .Activate
            .Visible = True
            .Top = Target.Top
            .Left = Target.Left
            .Width = Target.Width
            .Height = Target.Height
        End With
        With Me.lstList
            .Visible = True
            .Top = Target.Top
            .Left = Target.Left + Target.Width
            .Width = Target.Width
            .Height = Target.Height * 5
            For i = 2 To Sheet2.Cells(Rows.Count,
1).End(xlUp).Row
                .AddItem Sheet2.Cells(i, 1).Value
            Next
        End With
    Else
        Me.lstList.Clear
        Me.txtValue = ""
        Me.lstList.Visible = False
        Me.txtValue.Visible = False
    End If
End If
End Sub

Private Sub txtValue_KeyUp(ByVal KeyCode As
MSForms.ReturnInteger, ByVal Shift As Integer)

    Dim i As Long

    Dim blnLanguage As Boolean

    Dim strTemp As String

    Me.lstList.Clear

    With Me.txtValue

        For i = 1 To Len(.Value)

            If Asc(Mid$(.Value, i, 1)) > 255 Or
Asc(Mid$(.Value, i, 1)) < 0 Then

                blnLanguage = True
                strTemp = strTemp & Mid$(.Value, i, 1)

            Else

                strTemp = strTemp & LCase(Mid$(.Value, i,
```

```
1))

                End If

            Next

        End With
    End Sub

    Private Sub lstList_DblClick(ByVal Cancel As
MSForms.ReturnBoolean)

        ActiveCell.Value = lstList.Value
        Me.lstList.Clear
        Me.txtValue = ""
        Me.lstList.Visible = False
        Me.txtValue.Visible = False

    End Sub

    Option Explicit

    Function strLChin(ByVal strUser As String) As String

        Dim strTemp As String

        strTemp = StrConv(strUser, vbNarrow)

        If Asc(strTemp) > 0 Or Err.Number = 1004 Then strLChin =
""

        strLChin = WorksheetFunction.VLookup(strTemp, [{"𐄀",
"a", "八", "b", "嚟", "c", "𐄁", "d", "鷄", "e", "发", "f", "
猴", "g", "𐄂", "h", "𐄃", "j", "𐄄", "k", "𐄅", "l", "𐄆", "m", "
𐄇", "n", "𐄈", "o", "𐄉", "p", "七", "q", "𐄊", "r", "𐄋", "s", "
他", "t", "𐄌", "w", "夕", "x", "𐄍", "y", "
𐄎", "z"}], 2)

    End Function
```

12.13 使用多页控件



```
Option Explicit
Sub FormShow()
    frmMorePagesDemo.Show
End Sub
```

第 12 章 控件的应用

12.11.1 使用 DTP 控件输入日期

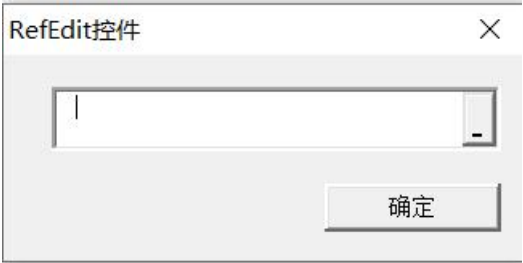
```
Option Explicit
Private Sub Worksheet_SelectionChange(ByVal Target As Range)
    With Me.DTPicker1
        If Target.Count = 1 And Target.Column = 1 And _
            Not Target.Row = 1 Or Target.MergeCells = True
Then
            .Visible = True
            .Top = Selection.Top
            .Left = Selection.Left
            .Height = Selection.Height
            .Width = Selection.Width
            If Target.Cells(1, 1) <> "" Then
                .Value = Target.Cells(1, 1).Value
            Else
                .Value = Date
            End If
        Else
            .Visible = False
        End If
    End With
End Sub
Private Sub Worksheet_Change(ByVal Target As Range)
    If Target.Count = 1 And Target.Column = 1 Or
Target.MergeCells = True Then
        If Target.Cells(1, 1).Value = "" Then
            DTPicker1.Visible = False
        End If
    End If
End Sub
Private Sub DTPicker1_CloseUp()
    ActiveCell.Value = Me.DTPicker1.Value
    Me.DTPicker1.Visible = False
End Sub
```

12.11.2 使用 MonthView 控件查看日期



```
Option Explicit
Sub FormShow()
    frmMonthViewDemo.Show
End Sub
```

12.12 使用 RefEdit 控件获取单元格区域



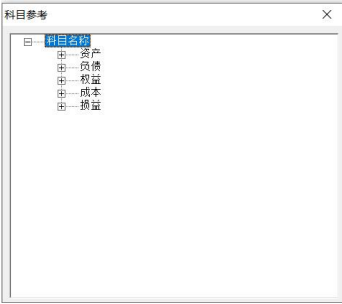
```
Sub FormShow()
    frmRefEditDemo.Show
End Sub
```

12.14 使用 TabStrip 控件



```
Sub FormShow()
    frmTabStripDemo.Show
End Sub
```

12.16 使用 TreeView 控件显示层次



```
Sub FormShow()
    frmTreeViweDemo.Show
End Sub
```

12.15 使用 ListView 控件

12.15.1 使用 Listview 控件显示数据

使用ListView控件显示数据						
人员编号	技能工资	岗位工资	工龄工资	浮动工资	其他	应发合计
8001	1,815.00	810.00	272.00	60.00	69.00	2,026.00
8002	1,415.00	800.00	312.00	80.00	64.00	2,671.00
8003	1,195.00	860.00	240.00	70.00	64.00	2,429.00
8004	1,935.00	860.00	144.00	60.00	64.00	2,063.00
8005	1,230.00	740.00	256.00	70.00	60.00	2,356.00
8006	1,230.00	620.00	256.00	70.00	57.00	2,233.00
8007	1,995.00	620.00	144.00	60.00	57.00	1,876.00
8008	1,740.00	620.00	196.00	50.00	57.00	1,563.00
8009	1,090.00	520.00	208.00	70.00	56.00	1,944.00
8011	1,160.00	520.00	208.00	70.00	56.00	2,014.00
8012	1,090.00	520.00	192.00	70.00	56.00	1,928.00
8013	1,055.00	520.00	184.00	70.00	56.00	1,885.00
8014	2,665.00	520.00	188.00	50.00	56.00	1,379.00
8015	1,025.00	520.00	160.00	65.00	56.00	1,826.00
合 计	20,64...	9,050.00	2,960.00	915.00	828.00	28,19...

```
Option Explicit
Sub FormShow()
    frmListViewDemo.Show
End Sub
```

12.15.2 在 Listview 控件中使用复选框

```
Option Explicit
Sub FormShow()
    frmListViewDemo.Show
End Sub
```

(示例图片转下页)

第 12 章 控件的应用

在ListView控件中使用复选框

人员编号	技能工资	岗位工资	工龄工资	浮动工资	其他	应发合计
<input type="checkbox"/> 8001	1, 815. 00	810. 00	272. 00	60. 00	69. 00	2, 026. 00
<input type="checkbox"/> 8002	1, 415. 00	800. 00	312. 00	80. 00	64. 00	2, 671. 00
<input type="checkbox"/> 8003	1, 195. 00	860. 00	240. 00	70. 00	64. 00	2, 429. 00
<input type="checkbox"/> 8004	1, 935. 00	860. 00	144. 00	60. 00	64. 00	2, 063. 00
<input type="checkbox"/> 8005	1, 230. 00	740. 00	256. 00	70. 00	60. 00	2, 356. 00
<input type="checkbox"/> 8006	1, 230. 00	620. 00	256. 00	70. 00	57. 00	2, 233. 00
<input type="checkbox"/> 8007	1, 995. 00	620. 00	144. 00	60. 00	57. 00	1, 876. 00
<input type="checkbox"/> 8008	1, 740. 00	620. 00	196. 00	50. 00	57. 00	1, 563. 00
<input type="checkbox"/> 8009	1, 090. 00	520. 00	208. 00	70. 00	56. 00	1, 944. 00
<input type="checkbox"/> 8011	1, 160. 00	520. 00	208. 00	70. 00	56. 00	2, 014. 00
<input type="checkbox"/> 8012	1, 090. 00	520. 00	192. 00	70. 00	56. 00	1, 928. 00

保存

12.15.3 调整 ListView 控件的行距

调整ListView控件的行距

人员编号	技能工资	岗位工资	工龄工资	浮动工资	其他	应发合计
8001	1, 815. 00	810. 00	272. 00	60. 00	69. 00	2, 026. 00
8002	1, 415. 00	800. 00	312. 00	80. 00	64. 00	2, 671. 00
8003	1, 195. 00	860. 00	240. 00	70. 00	64. 00	2, 429. 00
8004	1, 935. 00	860. 00	144. 00	60. 00	64. 00	2, 063. 00
8005	1, 230. 00	740. 00	256. 00	70. 00	60. 00	2, 356. 00
8006	1, 230. 00	620. 00	256. 00	70. 00	57. 00	2, 233. 00
8007	1, 995. 00	620. 00	144. 00	60. 00	57. 00	1, 876. 00
8008	1, 740. 00	620. 00	196. 00	50. 00	57. 00	1, 563. 00
8009	1, 090. 00	520. 00	208. 00	70. 00	56. 00	1, 944. 00

```
Option Explicit
Sub FormShow()
    frmSpacing.Show
End Sub
```

12.15.4 在 ListView 控件中排序

在ListView控件中排序

人员编号	技能工资	岗位工资	工龄工资	浮动工资	其他	应发合计
8001	1, 815. 00	810. 00	272. 00	60. 00	69. 00	2, 026. 00
8002	1, 415. 00	800. 00	312. 00	80. 00	64. 00	2, 671. 00
8003	1, 195. 00	860. 00	240. 00	70. 00	64. 00	2, 429. 00
8004	1, 935. 00	860. 00	144. 00	60. 00	64. 00	2, 063. 00
8005	1, 230. 00	740. 00	256. 00	70. 00	60. 00	2, 356. 00
8006	1, 230. 00	620. 00	256. 00	70. 00	57. 00	2, 233. 00
8007	1, 995. 00	620. 00	144. 00	60. 00	57. 00	1, 876. 00
8008	1, 740. 00	620. 00	196. 00	50. 00	57. 00	1, 563. 00
8009	1, 090. 00	520. 00	208. 00	70. 00	56. 00	1, 944. 00
8011	1, 160. 00	520. 00	208. 00	70. 00	56. 00	2, 014. 00
8012	1, 090. 00	520. 00	192. 00	70. 00	56. 00	1, 928. 00
8013	1, 055. 00	520. 00	184. 00	70. 00	56. 00	1, 885. 00
8014	2, 665. 00	520. 00	188. 00	50. 00	56. 00	1, 379. 00
8015	1, 025. 00	520. 00	160. 00	65. 00	56. 00	1, 826. 00

```
Option Explicit
Sub FormShow()
    frmSortDemo.Show
End Sub
```

12.15.5 ListView 控件的图标设置

大图标视图模式

8001

8002

8003

8004

8005

8006

8007

8008

8009

8011

8012

8013

8014

8015

```
Sub FormShow()
    frmBig.Show
End Sub
```

小图标视图模式

8001

8002

8003

8004

8005

8006

8007

8008

8009

8011

8012

8013

8014

8015

```
Sub FormShow2()
    frmSmall.Show
End Sub
```

列表视图模式

8001

8002

8003

8004

8005

8006

8007

8008

8009

8011

8012

8013

8014

8015

```
Sub FormShow3()
    frmList.Show
End Sub
```

报表视图模式

人员编号	技能工资	岗位工资	工龄工资	浮动工资	其他	应发合计
8001	1, 815. 00	810. 00	272. 00	60. 00	69. 00	2, 026. 00
8002	1, 415. 00	800. 00	312. 00	80. 00	64. 00	2, 671. 00
8003	1, 195. 00	860. 00	240. 00	70. 00	64. 00	2, 429. 00
8004	1, 935. 00	860. 00	144. 00	60. 00	64. 00	2, 063. 00
8005	1, 230. 00	740. 00	256. 00	70. 00	60. 00	2, 356. 00
8006	1, 230. 00	620. 00	256. 00	70. 00	57. 00	2, 233. 00
8007	1, 995. 00	620. 00	144. 00	60. 00	57. 00	1, 876. 00
8008	1, 740. 00	620. 00	196. 00	50. 00	57. 00	1, 563. 00
8009	1, 090. 00	520. 00	208. 00	70. 00	56. 00	1, 944. 00

```
Sub FormShow4()
    frmView.Show
End Sub
```

12.16 使用 TreeView 控件显示层次

科目参考

科目名称

资产

负债

权益

成本

损益

```
Option Explicit
Sub FormShow()
    frmTreeViweDemo.Show
End Sub
```

第 12 章 控件的应用

12.17 使用 WebBrowser 显示动态图



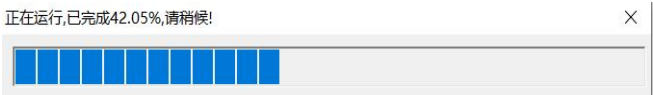
```
Option Explicit
Sub FormShow()
    frmWebBrowserDemo.Show
End Sub
```

12.18 使用 ShockwaveFlash 控件播放 Flash 文件



```
Option Explicit
Sub FormShow()
    frmLogin.Show
End Sub
```

12.19.1 使用 ProgressBar 控件制作进度条



```
Option Explicit
Private Sub CommandButton1_Click()
    Dim i As Integer
    UserForm1.Show 0
    With UserForm1.ProgressBar1
        .Min = 1
        .Max = 10000
        .Scrolling = 0
        For i = 1 To 10000
            Cells(i, 1) = i
            .Value = i
            UserForm1.Caption = "正在运行,已完成" & i /
100 & "%,请稍候!"
        Next
    End With
    Unload UserForm1
    Columns(1).ClearContents
```

End Sub 12.19.2 使用标签控件制作进度条



```
Sub ClearCells() '清除数据
    Range("a:a").ClearContents
End Sub

Sub lblProgress() '显示进度条
    Dim lngMax As Long
    Dim lngTemp As Integer
    lngMax = 10000
    With frmLableProgress
        .Show 0
        For lngTemp = 1 To lngMax
            Cells(lngTemp, 1) = lngTemp
            .lblComplete.Width = lngTemp / lngMax
* .fraDemo.Width
            .lblComplete.Caption = Round(lngTemp / lngMax
* 100, 0) & "%"
        DoEvents
        Next lngTemp
    End With
    Unload frmLableProgress
End Sub
```

12.20 不打印工作表中的控件



12.21.1 使用名称中的变量遍历控件



```
Private Sub Worksheet_SelectionChange(ByVal Target As Range)
    Sheet1.TextBox1.Object.Text = "Microsoft Office Excel 2016"
    Sheet1.TextBox2.Object.Text = "Microsoft Office Word 2016"
    Sheet1.TextBox3.Object.Text = "Microsoft Office Access 2016"
End Sub

Option Explicit
Sub FormShow() '显示窗体
    frmTraverseActiveX.Show
End Sub

Sub ClearText() '清除数据
    Dim i As Integer
    For i = 1 To 3
        Sheet1.OLEObjects("TextBox" & i).Object.Text = ""
    Next
End Sub
```

第 12 章 控件的应用

12.21.2 使用对象类型遍历控件

```
Option Explicit
Sub FormShow() '显示窗体
    frmTraverseActiveX.Show
End Sub
Sub ClearText() '清除内容
    Dim objOLE_Temp As OLEObject
    For Each objOLE_Temp In Sheet1.OLEObjects
        If TypeName(objOLE_Temp.Object) = "TextBox" Then
            objOLE_Temp.Object.Text = ""
        End If
    Next objOLE_Temp
    Set objOLE_Temp = Nothing
End Sub
```

12.21.3 使用程序标识符遍历控件

```
Option Explicit
Sub ClearText() '清除内容
    Dim objOLE_Temp As OLEObject
    For Each objOLE_Temp In Sheet1.OLEObjects
        If objOLE_Temp.progID = "Forms.TextBox.1" Then
            objOLE_Temp.Object.Text = ""
        End If
    Next objOLE_Temp
    Set objOLE_Temp = Nothing
End Sub
```

12.21.4 使用 FormControlType 属性遍历控件

```
Sub ControlType()
    Dim shpTemp As Shape
    For Each shpTemp In Sheet1.Shapes
        With shpTemp
            If .Type = msoFormControl Then
                If .FormControlType = xlCheckBox Then
                    If .ControlFormat.Value = 1 Then
                        .ControlFormat.Value = -4146
                    Else
                        .ControlFormat.Value = 1
                    End If
                End If
            End If
        End With
    Next
    Set shpTemp = Nothing
End Sub
```



- ☐ Microsoft Office Excel
- ☐ Microsoft Office Word 2016
- ☐ Microsoft Office Access

12.22.1 使用 AddFormControl 方法添加表单控件

```
Sub AddButton() '创建控件
    Dim shpTemp As Shape
    On Error Resume Next
    Sheet1.Shapes("MyButton").Delete
    On Error GoTo 0
    Set shpTemp = Sheet1.Shapes.AddFormControl(0, 60, 20, 120, 30)
    With shpTemp
        .Name = "MyButton"
        With .TextFrame.Characters
            .Font.ColorIndex = 0
            .Font.Size = 12
            .Text = "新建的按钮"
        End With
        .OnAction = "MyButton"
    End With
    Set shpTemp = Nothing
End Sub
Sub MyButton()
    MsgBox "这是使用 AddFormControl 方法新建的按钮!"
End Sub
Sub DelButton() '删除新建的按钮
    Sheet1.Shapes("MyButton").Delete
End Sub
```

12.22.2 使用 Add 方法添加表单控件

```
Sub AddButton() '创建控件
    Dim cmdTemp As Button
    On Error Resume Next
    Sheet1.Shapes("MyButton").Delete
    On Error GoTo 0
    Set cmdTemp = Sheet1.Buttons.Add(60, 20, 120, 30)
    With cmdTemp
        .Name = "MyButton"
        .Font.Size = 12
        .Font.ColorIndex = 0
        .Characters.Text = "新建的按钮"
        .OnAction = "MyButton"
    End With
    Set cmdTemp = Nothing
End Sub
Sub MyButton()
    MsgBox "这是使用 Add 方法新建的按钮!"
End Sub
Sub DelButton() '删除新建的按钮
    Sheet1.Shapes("MyButton").Delete
End Sub
```


第 12 章 控件的应用

12.22.3 使用 Add 方法添加 ActiveX 控件

Sub AddButton() '创建控件

```
Dim objOLE_Temp As New OLEObject
On Error Resume Next
Sheet1.OLEObjects("MyButton").Delete
On Error GoTo 0
Set objOLE_Temp = Sheet1.OLEObjects.Add( _
    ClassType:="Forms.CommandButton.1", _
    Left:=60, Top:=20, Width:=120, Height:=30)
With objOLE_Temp
    .Name = "MyButton"
    .Object.Caption = "新建的按钮"
    .Object.Font.Size = 12
    .Object.ForeColor = vbBlack
End With
```

```
With ActiveWorkbook.VBProject.VBComponents( _
    Sheet1.CodeName).CodeModule
    If .Lines(1, 1) <> "Option Explicit" Then
        .InsertLines 1, "Option Explicit"
    End If
    If .Lines(2, 1) = "Private Sub MyButton_Click()" Then
```

Exit Sub

```
.InsertLines 2, "Private Sub MyButton_Click()"
.InsertLines 3, vbTab & "MsgBox ""这是使用 Add 方
法新建的按钮!""""
.InsertLines 4, "End Sub"
End With
Set objOLE_Temp = Nothing
```

End Sub

Sub DelButton() '删除新建的按钮

```
Sheet1.OLEObjects("MyButton").Delete
```

End Sub

12.22.4 使用 AddOLEObject 方法添加 ActiveX 控件

Option Explicit

Sub AddButton() '创建控件

```
Dim shpTemp As Shape
On Error Resume Next
Sheet1.Shapes("MyButton").Delete
On Error GoTo 0
Set shpTemp = Sheet1.Shapes.AddOLEObject( _
    ClassType:="Forms.CommandButton.1", _
    Left:=60, Top:=20, Width:=120, Height:=30)
shpTemp.Name = "MyButton"
With ActiveWorkbook.VBProject.VBComponents( _
    Sheet1.CodeName).CodeModule
    If .Lines(1, 1) <> "Option Explicit" Then
        .InsertLines 1, "Option Explicit"
    End If
    If .Lines(2, 1) = "Private Sub MyButton_Click()" Then
```

Exit Sub

```
End Sub
```

```
.InsertLines 2, "Private Sub MyButton_Click()"
```

```
.InsertLines 3, vbTab & "MsgBox ""这是使用
```

AddOLEObject 方法新建的按钮!""""

```
.InsertLines 4, "End Sub"
```

```
End With
```

```
Set shpTemp = Nothing
```

End Sub

Sub DelButton() '删除新建的按钮

```
Sheet1.Shapes("MyButton").Delete
```

```
ActiveWorkbook.VBProject.VBComponents("Sheet1").CodeMod
ule.DeleteLines 1, 4
```

End Sub

第 13 章 用户窗体的应用

13.1 调用用户窗体

Sub FormModeless() '显示模态用户窗体

```
Dim i As Integer
frmLogin.Show 0
For i = 1 To 1000
    Sheet1.Cells(i, 1) = i
Next i
```

End Sub

Sub FormModel() '显示非模态用户窗体

```
Dim i As Integer
frmLogin.Show 1
For i = 1 To 1000
    Sheet1.Cells(i, 1) = i
Next i
```

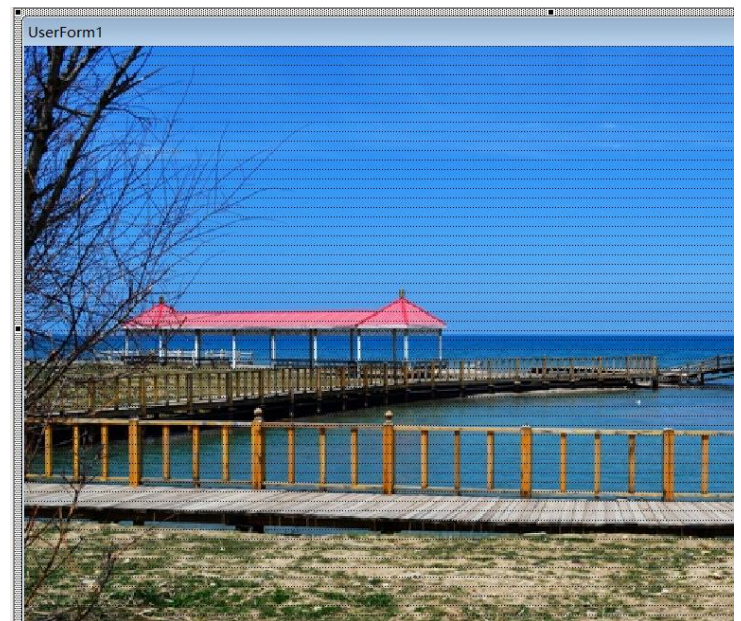
End Sub

Sub ClearCell() '清除数据

```
Range("a:a").ClearContents
```

End Sub

13.2 制作欢迎界面窗体



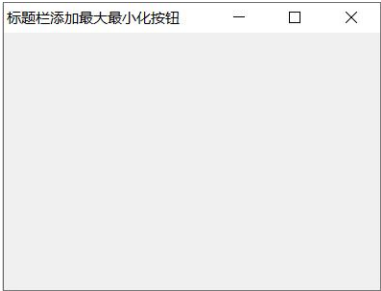
Sub CloseForm()

```
Unload frmLogin
```

End Sub

第 13 章 用户窗体的应用

13.3 在用户窗体标题栏上添加最大化和最小化按钮



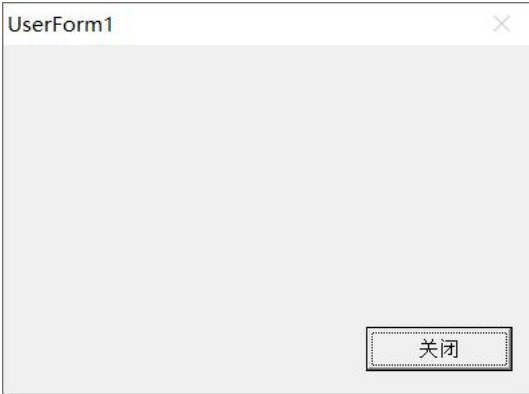
```
Sub FormShow()  
    frmLogin.Show  
End Sub
```

13.4.1 利用 QueryClose 事件禁止关闭窗口体



```
Sub FormShow()  
    frmLogin.Show  
End Sub
```

13.4.2 利用 API 函数禁用用户窗体标题栏的关闭按钮



```
Sub FormShow()  
    frmBanClose.Show  
End Sub
```

13.5 在用户窗体上添加菜单



Option Explicit

```
Private Declare Function CallWindowProc Lib "user32" _  
    Alias "CallWindowProcA" ( _  
        ByVal lpPrevWndFunc As Long, _  
        ByVal Hwnd As Long, _  
        ByVal Msg As Long, _  
        ByVal wParam As Long, _  
        ByVal lParam As Long) As Long  
  
Public glngPreWinProc As Long, glngHwnd As Long  
Function lngMsgProcess(ByVal lngHwnd As Long, _  
    ByVal lngMsg As Long, _  
    ByVal lngwParam As Long, _  
    ByVal lnglParam As Long) As Long  
  
    Select Case lngwParam  
        Case 100  
            MsgBox "你选择的是""保存""菜单项!"  
        Case 101  
            MsgBox "你选择的是""备份""菜单项!"  
        Case 102  
            Unload frmMenu  
        Case 110  
            MsgBox "你选择的是""账户管理""菜单项!"  
        Case 111  
            MsgBox "你选择的是""账户转账""菜单项!"  
        Case 112  
            MsgBox "你选择的是""支出录入""菜单项!"  
        Case 113  
            MsgBox "你选择的是""收入录入""菜单项!"  
        Case 114  
            MsgBox "你选择的是""支出查询""菜单项!"  
        Case 115  
            MsgBox "你选择的是""收入查询""菜单项!"  
        Case Else  
            lngMsgProcess =  
                CallWindowProc(glngPreWinProc, _  
                    lngHwnd, lngMsg, lngwParam, lnglParam)  
    End Select  
End Function  
  
Sub FormShow()  
    frmMenu.Show  
End Sub
```

13.6 在用户窗体上添加工具栏



```
Option Explicit  
Sub FormShow()  
    frmToolbarDemo.Show  
End Sub
```

第 13 章 用户窗体的应用

13.7 在用户窗体上添加状态栏



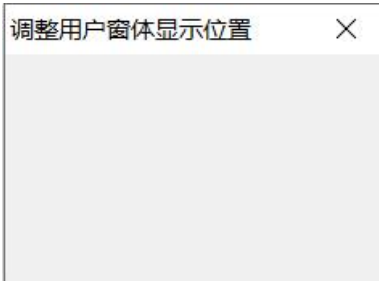
```
Sub FormShow()  
    frmStatusBar.Show  
End Sub
```

13.8 透明的用户窗体



```
Sub FormShow()  
    frmTransparent.Show  
End Sub
```

13.9.1 设置用户窗体的显示位置



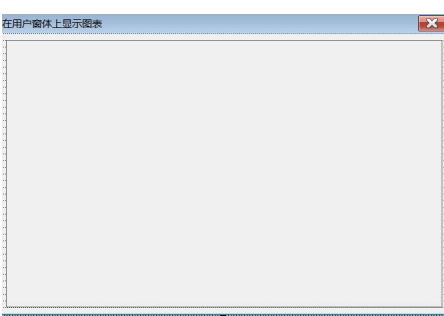
```
Sub FormShow()  
    frmLocation.Show 0  
End Sub
```

13.9.2 由活动单元格确定显示位置

```
Option Explicit  
Private Declare Function FindWindow Lib "user32" (_  
    Alias "FindWindowA" ( _  
        ByVal lpClassName As String, _  
        ByVal lpWindowName As String) As Long  
Private Declare Function GetDC Lib "user32" ( _  
    ByVal hwnd As Long) As Long  
Private Declare Function GetDeviceCaps Lib "gdi32" ( _  
    ByVal hdc As Long, _  
    ByVal nIndex As Long) As Long  
Private Declare Function MoveWindow Lib "user32.dll" ( _  
    ByVal hwnd As Long, _  
    ByVal x As Long, _  
    ByVal y As Long, _
```

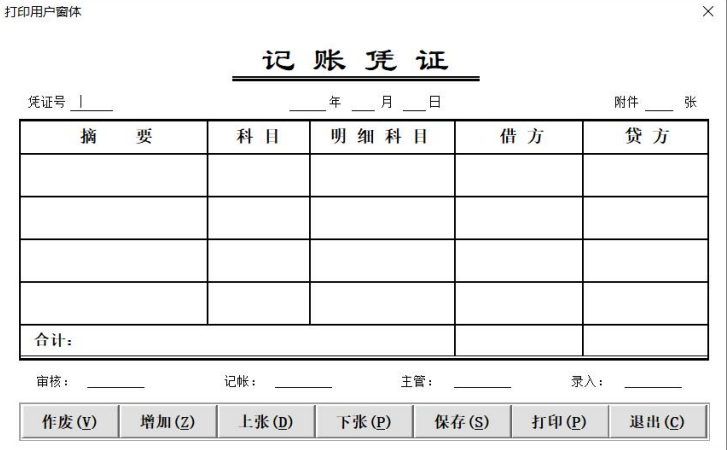
```
    ByVal nWidth As Long, _  
    ByVal nHeight As Long, _  
    ByVal bRepaint As Long) As Long  
Private Sub Worksheet_SelectionChange(ByVal Target As  
Range)  
    Dim lngHwnd As Long, lngDC As Long, lngCaps As Long  
    Dim lngLeft As Long, lngTop As Long  
    Dim sngPiexlToPiont As Single  
    Const LOG_PIXELS_X As Integer = 88  
    lngDC = GetDC(0)  
    lngCaps = GetDeviceCaps(lngDC, LOG_PIXELS_X)  
    sngPiexlToPiont = 72 / lngCaps * (100 /  
ActiveWindow.Zoom)  
    lngLeft = CLng(ActiveWindow.PointsToScreenPixelsX(-7) +  
-  
        (Target.Offset(1, 0).Left / sngPiexlToPiont))  
    lngTop = CLng(ActiveWindow.PointsToScreenPixelsY(0) + _  
        (Target.Offset(1, 0).Top / sngPiexlToPiont))  
    frmLocation.StartUpPosition = 0  
    lngHwnd = FindWindow(vbNullString,  
frmLocation.Caption)  
    MoveWindow lngHwnd, lngLeft, lngTop, 240, 180, True  
    frmLocation.Show 0  
End Sub
```

13.10 在用户窗体上显示图表



```
Sub FormShow()  
    frmChart.Show  
End Sub
```

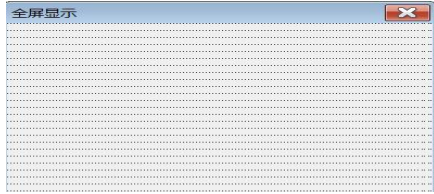
13.11 打印用户窗体



```
Sub FormShow()  
    frmPrint.Show  
End Sub
```

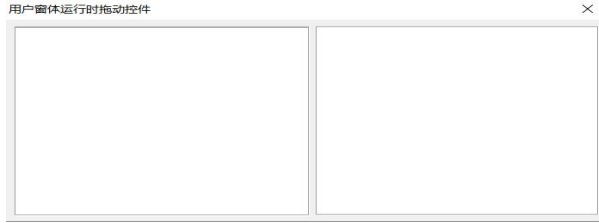
第 13 章 用户窗体的应用

13.12.1 设置用户窗体为应用程序的大小



```
Sub FormShow()  
    frmFullScreen.Show  
End Sub
```

13.13 用户窗体运行时拖动控件



```
Sub FormShow()  
    frmVariable.Show  
End Sub
```

13.14 使用自定义颜色设置用户窗体颜色



```
Sub formshow()  
    frmBackColor.Show  
End Sub
```

13.16 使用代码添加用户窗体及控件

```
Sub CreatingForms()    '创建窗体  
    Dim objMyForm As VBComponent  
    Dim objMyTextBox As Control  
    Dim objMyButton As Control  
    Dim i As Integer  
    On Error Resume Next  
    Set objMyForm =  
ThisWorkbook.VBProject.VBComponents.Add(vbext_ct_MSForm)  
  
    With objMyForm  
        .Properties("Name") = "frmForm"  
        .Properties("Caption") = "演示窗体"  
        .Properties("Height") = "180"  
        .Properties("Width") = "240"  
        Set objMyTextBox  
= .Designer.Controls.Add("Forms.CommandButton.1")  
        With objMyTextBox  
            .Name = "cmdMyTextBox"  
            .Caption = "新建文本框"  
            .Top = 40  
            .Left = 138  
            .Height = 20  
            .Width = 70
```

```
End With  
Set objMyButton  
= .Designer.Controls.Add("Forms.CommandButton.1")  
    With objMyButton  
        .Name = "cmdMyButton"  
        .Caption = "删除文本框"  
        .Top = 70  
        .Left = 138  
        .Height = 20  
        .Width = 70  
    End With  
    With .CodeModule  
        i = .CreateEventProc("Click", "cmdMyTextBox")  
        .ReplaceLine i + 1, Space(4) & _  
            "Dim objMyTextBox As Control" & vbCrLf &  
Space(4) & _  
            "Dim i As Integer" & vbCrLf & Space(4) &  
            "Dim intTop As Integer" & vbCrLf & Space(4) &  
& _  
            "intTop = 10" & vbCrLf & Space(4) & "For i =  
1 To 5" & _  
            vbCrLf & Space(8) & "Set objMyTextBox = "  
& _  
            "Me.Controls.Add(bstrprogid:=""Forms.TextBox.1"")" _  
                & vbCrLf & Space(8) & "With  
objMyTextBox" & vbCrLf &  
                Space(12) & ".Name = ""txtMyTextBox"" &  
i" & vbCrLf & _  
                Space(12) & ".Left = 20" & vbCrLf &  
Space(12) & _  
                ".Top = intTop" & vbCrLf & Space(12) &  
".Height = 18" & _  
                vbCrLf & Space(12) & ".Width = 80" &  
vbCrLf & _  
                Space(12) & "intTop = .Top + 28" & vbCrLf &  
Space(8) & _  
                "End With" & vbCrLf & Space(4) & "Next i"  
& vbCrLf _  
                & Space(4) & "Set objMyTextBox = Nothing"  
        i = .CreateEventProc("Click", "cmdMyButton")  
        .ReplaceLine i + 1, Space(4) & "Dim i As Integer"  
& _  
                vbCrLf & Space(4) & "On Error Resume  
Next" & vbCrLf & _  
                Space(4) & "For i = 1 To 5" & vbCrLf &  
Space(8) & _  
                "Me.Controls.Remove ""txtMyTextBox"" &  
i" & vbCrLf & _  
                Space(4) & "Next i"  
        End With  
    End With  
    Set objMyForm = Nothing  
    Set objMyTextBox = Nothing  
    Set objMyButton = Nothing  
End Sub  
Sub RemoveForm()    '删除窗体并保存  
    Dim objRemoveFrm As VBComponent  
    On Error Resume Next  
    Set objRemoveFrm =  
Application.ActiveWorkbook.VBProject.VBComponents.Item("frmForm")  
    Application.VBE.ActiveVBProject.VBComponents.Remove  
objRemoveFrm  
    ThisWorkbook.Save  
End Sub  
Sub frmShow()    '显示窗体  
    frmForm.show  
End Sub
```

第 14 章 目录和文件操作

14.1 判断文件或文件夹是否存在

Sub FileExist()

```
Dim objFso As Object
Dim strPath As String
strPath = ThisWorkbook.Path & Application.PathSeparator
Debug.Print "文件或文件夹名"; Tab(50); "是否存在"
Debug.Print "使用 Dir 函数"
Debug.Print strPath & "Test"; Tab(50); _
    Dir(strPath & "Test", vbDirectory + vbHidden) <> ""
Debug.Print strPath & "Test\Test.txt"; Tab(50); _
    Dir(strPath & "Test\Test.txt", _
        vbNormal + vbHidden + vbReadOnly) <> ""
Set objFso = CreateObject("Scripting.FileSystemObject")
Debug.Print "使用 FSO 对象"
Debug.Print strPath & "Test"; Tab(50); _
    objFso.FolderExists(strPath & "Test")
Debug.Print strPath & "Test\Test.txt"; Tab(50); _
    objFso.FileExists(strPath & "test\test.txt")
```

End Sub

14.2 重命名文件、文件夹

Sub RenameFile()

```
Dim strPath As String
strPath = ThisWorkbook.Path & Application.PathSeparator
On Error GoTo errHandle
If Dir(strPath & "Test\Test.txt") <> "" Then
    Name strPath & "Test\Test.txt" As strPath &
        "Test\Test1.txt"
    Name strPath & "Test\Test1.txt" As strPath & _
        "Test\Test1\Test1.txt"
    Name strPath & "Test\Test1\Test1.txt" As strPath & _
        "Test\Test.txt"
End If
Exit Sub
errHandle:
MsgBox Err.Description
End Sub
```

14.3 获取文件信息和属性

Sub FileInfo()

```
Dim objFso As Object
Dim objFile As Object
Dim strFile As String
Dim strInfo As String
Dim strPath As String
strPath = ThisWorkbook.Path & Application.PathSeparator
strFile = strPath & "Test\Test.txt"
Set objFso = CreateObject("Scripting.FileSystemObject")
If objFso.FileExists(strFile) Then
    strInfo = strInfo & "文件大小: " & FileLen(strFile) &
        " 字节" & vbCrLf
    strInfo = strInfo & "文件最后修改的时间: " & _
        FileDateTime(strFile) & vbCrLf
```

```
strInfo = strInfo & "文件属性: " & GetAttr(strFile)
Debug.Print "使用内置函数和语句"
Debug.Print strInfo
SetAttr strFile, vbReadOnly + vbHidden
Debug.Print "文件 “" & strFile & ” 已设置了只读和
隐藏属性"
```

```
Set objFile = objFso.GetFile(strFile)
strInfo = strInfo & "属性:" & objFile.Attributes & vbCrLf
strInfo = strInfo & "文件大小(Bytes):" & objFile.Size & vbCrLf
strInfo = strInfo & "创建时间:" & objFile.DateCreated & vbCrLf
strInfo = strInfo & "最后修改时间:" & objFile.DateLastModified
Debug.Print "使用 FSO 对象"
Debug.Print strInfo
objFile.Attributes = objFile.Attributes + 4
Debug.Print "文件 “" & strFile & ” 已设置了系统属性"
objFile.Attributes = objFile.Attributes - 3
Debug.Print "文件 “" & strFile & ” 已去除了只读和隐藏属
性"
objFile.Attributes = objFile.Attributes - 4
End If
Set objFile = Nothing
Set objFso = Nothing
```

End Sub

14.4 获取驱动器信息

Sub GetDriveInfo()

```
Dim objFso As Object
Dim objDrive As Object
Dim strInfo As String
Set objFso = CreateObject("Scripting.FileSystemObject")
Set objDrive = objFso.GetDrive("c")
If objDrive.IsReady = False Then
    strInfo = strInfo & "是否可用: 不可用" & vbCrLf
Else: strInfo = strInfo & "是否可用: 可用" & vbCrLf
    If objDrive.DriveType = 3 Then
        strInfo = strInfo & "名称: " &
            objDrive.ShareName & vbCrLf
    Else: strInfo = strInfo & "卷标: " & _
        objDrive.VolumeName & vbCrLf
    End If
    strInfo = strInfo & "文件系统: " & objDrive.FileSystem
        & vbCrLf
    strInfo = strInfo & "总容量: " & objDrive.TotalSize &
        vbCrLf
    strInfo = strInfo & "可用容量: " & _
        objDrive.AvailableSpace & vbCrLf
    strInfo = strInfo & "序列号: " & _
        Hex(objDrive.SerialNumber) & vbCrLf
End If
MsgBox strInfo
Set objDrive = Nothing
Set objFso = Nothing
End Sub
```


第 14 章 目录和文件操作

14.5 使用 FSO 对象操作文件夹

Sub OperateFolder()

```
Dim objFso As Object, objFolder As Object, strPath As String
    strPath = ThisWorkbook.Path & Application.PathSeparator
    On Error GoTo errHandle
    Set objFso = CreateObject("Scripting.FileSystemObject")
    If objFso.FolderExists(strPath & "Test") = True Then
        objFso.CreateFolder strPath & "Test\TestA"
        Set objFolder = objFso.GetFolder(strPath &
"Test\Test1")
        objFolder.SubFolders.Add "Test2"
        objFolder.Copy strPath & "Test\TestA\TestB"
        objFso.CopyFolder strPath & "Test\Test1\Test2", _
            strPath & "Test\Test2"
        objFso.MoveFolder strPath & "Test\TestA\TestB", _
            strPath & "Test\Test3"
        objFolder.Delete
        objFso.DeleteFolder strPath & "Test\T*"
        objFso.CreateFolder strPath & "Test\Test1"
    End If
    Set objFolder = Nothing
    Set objFso = Nothing
    Exit Sub
```

errHandle:

MsgBox Err.Description

End Sub

14.6 使用 FSO 对象操作文件

Sub OperateFile()

```
Dim objFso As Object, objFile As Object, strPath As String
    strPath = ThisWorkbook.Path & Application.PathSeparator
    Set objFso = CreateObject("Scripting.FileSystemObject")
    If objFso.FileExists(strPath & "Test\Test.txt") Then
        Set objFile = objFso.GetFile(strPath & "Test\Test.txt")
        objFile.Copy strPath & "Test\Test1.txt"
        objFso.CopyFile strPath & "Test\Test.txt", strPath & _
            "Test\File\Test2.txt"
        objFile.Move strPath & "Test\File\Test.txt"
        objFso.MoveFile strPath & "Test\File\Test2.txt", _
            strPath & "Test\Test.txt"
        objFile.Delete
        objFso.DeleteFile strPath & "Test\Test1.txt"
    End If
    Set objFile = Nothing
    Set objFso = Nothing
```

End Sub

14.7 使用 FSO 对象查找文件

Sub FirstList()

```
Dim objFso As Object, objFolder As Object, strInfo As String
    Set objFso = CreateObject("Scripting.FileSystemObject")
    Set objFolder =
```

```
objFso.GetFolder(ThisWorkbook.Path).ParentFolder
```

```
    strInfo = strEnumFile(objFolder)
```

```
    Debug.Print strInfo
```

```
    Set objFso = Nothing
```

```
    Set objFolder = Nothing
```

End Sub

Function strEnumFile(ByVal objFolder As Object) As String

```
    Dim objSubFolder As Object
```

```
    Dim objFile As Object
```

```
    Dim strList As String
```

```
    strList = "文件夹:" & vbTab & objFolder.Path & vbCrLf
```

```
    If objFolder.Files.Count <> 0 Then
```

```
        For Each objFile In objFolder.Files
```

```
            strList = strList & vbTab & "文件:" & objFile.Path
```

```
        & vbCrLf
```

```
        Next objFile
```

```
    End If
```

```
    If objFolder.SubFolders.Count <> 0 Then
```

```
        For Each objSubFolder In objFolder.SubFolders
```

```
            strList = strList & strEnumFile(objSubFolder)
```

```
        Next objSubFolder
```

```
    End If
```

```
    strEnumFile = strList
```

```
    Set objSubFolder = Nothing
```

```
    Set objFile = Nothing
```

End Function

Option Explicit

Private Declare Function ShellExecute Lib "shell32.dll" Alias

"ShellExecuteA" (ByVal hWnd As Long, ByVal lpOperation As

String, ByVal lpFile As String, ByVal lpParameters As String,

ByVal lpDirectory As String, ByVal nShowCmd As Long) As Long

Private Declare Function FindWindow Lib "user32" Alias

"FindWindowA" (ByVal lpClassName As String, ByVal

lpWindowName As String) As Long

Private Declare Function PostMessage Lib "user32" Alias

"PostMessageA" (ByVal hWnd As Long, ByVal wParam As Long,

ByVal lParam As Long) As Long

Private Const WM_CLOSE As Long = &H10

Public Sub OpenF()

```
    ShellExecute 0, "", ThisWorkbook.Path & _
```

```
        Application.PathSeparator & "Test\Test1", "", "", 1
```

```
    ShellExecute 0, "", ThisWorkbook.Path & _
```

```
        Application.PathSeparator & "Test\Test.txt", "", "", 1
```

End Sub

Public Sub CloseF()

```
    Dim lngHwnd As Long
```

```
    lngHwnd = FindWindow(vbNullString, "Test1")
```

```
    PostMessage lngHwnd, WM_CLOSE, 0, 0
```

```
    lngHwnd = FindWindow("Notepad", "Test.txt - 记事本")
```

```
    PostMessage lngHwnd, WM_CLOSE, 0, 0
```

End Sub

第 14 章 目录和文件操作

14.9 获取常用路径

Option Explicit '获取信息

Private Declare Function SHGetSpecialFolderLocation Lib

"Shell32" (ByVal hwndOwner As Long, ByVal nFolder As Integer,
ppidl As Long) As Long

Private Declare Function SHGetPathFromIDList Lib "Shell32"

Alias "SHGetPathFromIDListA" (ByVal pidl As Long, ByVal szPath
As String) As Long

Const PAGETMP = &H20&

Const COOKIES = &H21&

Const HISTORY = &H22&

Sub GetPath()

Dim strTemp As String

Dim lngPpidl As Long

Dim objShell As Object

Debug.Print "Excel 启动文件夹: "; Tab(20);

Application.StartupPath

Debug.Print "Excel 模板文件夹: "; _

Tab(20); Application.TemplatesPath

Debug.Print "COM 加载宏文件夹: "; Tab(20); _

Application.UserLibraryPath

Set objShell = CreateObject("Wscript.Shell")

Debug.Print "桌面: "; Tab(20); _

objShell.SpecialFolders("AllUsersStartup")

Debug.Print "收藏夹: "; Tab(20); _

objShell.SpecialFolders("Favorites")

Debug.Print "系统盘: "; Tab(20); Environ("SystemDrive")

Debug.Print "临时文件夹: "; Tab(20); Environ("TEMP")

Debug.Print "Windows 目录: "; Tab(20); Environ("windir")

Debug.Print "程序文件夹: "; Tab(20);

Environ("ProgramFiles")

strTemp = Space(512)

SHGetSpecialFolderLocation 0, PAGETMP, lngPpidl

SHGetPathFromIDList lngPpidl, strTemp

Debug.Print "网页临时文件: "; Tab(20); Split(strTemp,

Chr(0))(0)

SHGetSpecialFolderLocation 0, COOKIES, lngPpidl

SHGetPathFromIDList lngPpidl, strTemp

Debug.Print "Cookies: "; Tab(20); Split(strTemp, Chr(0))(0)

SHGetSpecialFolderLocation 0, HISTORY, lngPpidl

SHGetPathFromIDList lngPpidl, strTemp

Debug.Print "History: "; Tab(20); Split(strTemp, Chr(0))(0)

Set objShell = Nothing

End Sub

第 15 章 文件的输入输出

15.1 读写文本文件

Option Explicit

Sub WriteFile() '写入数据

Dim intNum As Integer

intNum = FreeFile

Open ThisWorkbook.Path & "\test.txt" For Output As

#intNum

Write #intNum, "Hello "; "ExcelHome"

Write #intNum, "这是一个测试文件"

Write #intNum,

Write #intNum, "以上是由 Write #语句写入的数据"

Print #intNum, "这是一个测试文件"

Print #intNum, "Hello "; "ExcelHome"

Print #intNum,

Print #intNum, Spc(5); "VBA 程序开发版"

Print #intNum, Tab(10); "VSTO 程序开发版"

Close intNum

End Sub

Sub AppendFile() '追加数据

Dim intNum As Integer

intNum = FreeFile

Open ThisWorkbook.Path & "\test.txt" For Append As

#intNum

Print #intNum, "*****"

Print #intNum, "这是添加的内容"

Close intNum

End Sub

Sub ReadFile() '读出数据

Dim strTemp As String

Dim strContent As String

Dim strMsg1 As String

Dim strMsg2 As String

Dim intNum As Integer

intNum = FreeFile

Open ThisWorkbook.Path & "\test.txt" For Input As

#intNum

Debug.Print "文件的第一个字符是: " & Input(1, #intNum)

Close #intNum

Open ThisWorkbook.Path & "\test.txt" For Input As

#intNum

Input #intNum, strMsg1, strMsg2

Debug.Print "第一行第一个区域的数据是: " & strMsg1

Debug.Print "第一行第二个区域的数据是: " & strMsg2

Close #intNum

Open ThisWorkbook.Path & "\test.txt" For Input As

#intNum

strContent = strContent & "文件长度为: " & LOF(intNum)
& "字节" & vbCrLf

strContent = strContent & "文件内容如下: " & vbCrLf

Do While EOF(intNum) = False

Line Input #intNum, strTemp

strContent = strContent & strTemp & vbCrLf

Loop

Debug.Print strContent

Close #intNum

End Sub

第 15 章 文件的输入输出

15.2 FSO 读写文本文件

Option Explicit

Sub WriteFile()

Dim objFso As Object

Dim objFile As Object

Dim objStream As Object

Set objFso = CreateObject("Scripting.FileSystemObject")

Set objStream = objFso.CreateTextFile(ThisWorkbook.Path & "\fsotest1.txt", True)

With objStream

.WriteLine ("这是第一个测试文件")

.WriteBlankLines (1)

.Write ("Hello ")

.Write ("ExcelHome")

.WriteLine

.WriteLine (Space(5) & "VBA 程序开发版")

.Close

End With

Set objStream = objFso.OpenTextFile(ThisWorkbook.Path & "\fsotest2.txt", 2, True)

objStream.WriteLine ("这是第二个测试文件")

objStream.Close

objFso.CreateTextFile (ThisWorkbook.Path & "\fsotest3.txt")

Set objFile = objFso.GetFile(ThisWorkbook.Path &

"\fsotest3.txt")

Set objStream = objFile.OpenAsTextStream(2, 0)

objStream.WriteLine ("这是第三个测试文件")

objStream.Close

Set objFile = Nothing

Set objStream = Nothing

Set objFso = Nothing

End Sub

Sub AppendFile()

Dim objFso As Object

Dim objStream As Object

Set objFso = CreateObject("Scripting.FileSystemObject")

Set objStream = objFso.OpenTextFile(ThisWorkbook.Path & "\fsotest2.txt", 8, True)

objStream.WriteLine ("*****")

objStream.WriteLine ("本行是追加写入的数据")

objStream.Close

Set objStream = Nothing

Set objFso = Nothing

End Sub

Sub ReadFile()

Dim objFso As Object

Dim objStream As Object

Dim strTemp As String

Set objFso = CreateObject("Scripting.FileSystemObject")

Set objStream = objFso.OpenTextFile(ThisWorkbook.Path & "\fsotest1.txt", 1)

Do Until objStream.AtEndOfStream

strTemp = strTemp & objStream.ReadLine() & vbCrLf

Loop

Debug.Print "第一个文件的内容： " & vbCrLf & strTemp

objStream.Close

Set objStream = objFso.OpenTextFile(ThisWorkbook.Path & "\fsotest2.txt", 1)

strTemp = objStream.ReadAll

Debug.Print "第二个文件的内容： " & vbCrLf & strTemp

objStream.Close

Set objStream = objFso.OpenTextFile(ThisWorkbook.Path & "\fsotest3.txt", 1)

strTemp = objStream.Read(1)

Debug.Print "第三个文件的第一个字符是： " & strTemp

strTemp = strTemp &

objStream.Read(FileLen(ThisWorkbook.Path & "\fsotest3.txt"))

Debug.Print "第三个文件的内容： " & vbCrLf & strTemp

objStream.Close

Set objStream = Nothing

Set objFso = Nothing

End Sub

15.3 保存指定区域内容到文本文件

Option Explicit

Sub SaveToText() **保存指定区域内容到一个文本文件**

Dim objFSO As Object

Dim objStream As Object

Dim intRow As Integer

Dim intCol As Integer

Dim avntArr As Variant

Dim strTemp As String

Set objFSO = CreateObject("Scripting.FileSystemObject")

Set objStream = objFSO.CreateTextFile(ThisWorkbook.Path & "\RangeValue.txt", True)

avntArr = Sheet1.Range("A1:C9")

For intRow = 1 To 9

strTemp = ""

For intCol = 1 To 3

strTemp = strTemp & avntArr(intRow, intCol) &

vbTab

Next intCol

objStream.WriteLine strTemp

Next intRow

objStream.Close

Set objStream = Nothing

Set objFSO = Nothing

End Sub

第 15 章 文件的输入输出

15.4 读写文本文件的指定行

Option Explicit

Sub SetAppointedLine() '操作注册表

Dim objFSO As Object

Dim objStream As Object

Dim i As Integer

Dim avntArr As Variant

Dim strAll As String

Dim strTemp As String

Set objFSO = CreateObject("Scripting.FileSystemObject")

Set objStream = objFSO.OpenTextFile(ThisWorkbook.Path
& "\RangeValue.txt", 1)

strAll = objStream.ReadAll

objStream.Close

avntArr = Split(strAll, vbCrLf)

MsgBox "文件的总行数为: " & (UBound(avntArr) + 1)

MsgBox "文件的第 3 行为: " & vbCrLf & avntArr(2)

For i = 0 To UBound(avntArr)

If Left(avntArr(i), 2) = "郑四" Then

avntArr(i) = "郑四" & vbTab & "女" & vbTab & "100"

End If

If Left(avntArr(i), 2) <> "李二" Then

strTemp = strTemp & avntArr(i) & vbCrLf

End If

Next i

Set objStream = objFSO.OpenTextFile(ThisWorkbook.Path
& "\RangeValue.txt", 2)

objStream.Write strTemp

objStream.Close

Set objStream = Nothing

Set objFSO = Nothing

End Sub

15.5 操作注册表

Sub ModifyReg()

Dim i As Integer

Dim avntKey As Variant

Dim strTemp As String

SaveSetting "MyTest", "测试", "测试 1", "这条将被删除"

SaveSetting "MyTest", "测试", "测试 2", 2

SaveSetting "MyTest", "测试", "测试 3", "3"

DeleteSetting "MyTest", "测试", "测试 1"

Debug.Print "注册表项“测试 2”的值是: " & _

GetSetting("MyTest", "测试", "测试 2")

avntKey = GetAllSettings("MyTest", "测试")

strTemp = "子键“测试”下共有" & _

UBound(avntKey) + 1 & "个项: " & vbCrLf

strTemp = strTemp & "名称" & vbTab & "值" & vbCrLf

For i = 0 To UBound(avntKey)

strTemp = strTemp & avntKey(i, 0) & _

vbTab & avntKey(i, 1) & vbCrLf

Next i

Debug.Print strTemp

DeleteSetting "MyTest"

End Sub

Sub WMIReg() 'WMI 操作注册表

Dim avntValue As Variant

Dim avntName As Variant

Dim avntType As Variant

Dim i As Integer

Dim strTemp As String

Dim objWMI As Object

Const HKEY_CURRENT_USER = &H80000001

Set objWMI =

GetObject("winmgmts:\\.\\root\\default:StdRegProv")

objWMI.CreateKey HKEY_CURRENT_USER, "MyTest\测试"

objWMI.SetBinaryValue HKEY_CURRENT_USER, "MyTest\

测试", _

"测试 1", Array(&H0, &H0, &H1)

objWMI.SetStringValue HKEY_CURRENT_USER, _

"MyTest\测试", "测试 2", "2"

objWMI.SetDwordValue HKEY_CURRENT_USER, _

"MyTest\测试", "测试 3", 3

objWMI.GetBinaryValue HKEY_CURRENT_USER, "MyTest\

测试", _

"测试 1", avntValue

For i = 0 To UBound(avntValue)

strTemp = strTemp & avntValue(i)

Next i

Debug.Print "注册表项“测试 1”的值是: " & strTemp

objWMI.DeleteValue HKEY_CURRENT_USER, "MyTest\测

试", "测试 1"

objWMI.EnumValues HKEY_CURRENT_USER, _

"MyTest\测试", avntName, avntType

strTemp = "子键“测试”下共有" & _

UBound(avntName) + 1 & "个项: " & vbCrLf

strTemp = strTemp & "名称" & vbTab & "值" & vbCrLf

For i = 0 To UBound(avntName)

If avntType(i) = 1 Then

objWMI.GetStringValue HKEY_CURRENT_USER, _

"MyTest\测试", avntName(i), avntValue

Elseif avntType(i) = 4 Then

objWMI.GetDwordValue HKEY_CURRENT_USER, _

"MyTest\测试", avntName(i), avntValue

End If

strTemp = strTemp & avntName(i) & vbTab &

avntValue & vbCrLf

Next i

Debug.Print strTemp

objWMI.DeleteKey HKEY_CURRENT_USER, "MyTest\测试"

objWMI.DeleteKey HKEY_CURRENT_USER, "MyTest"

Set objWMI = Nothing

End Sub

第 16 章 数据库应用

16.1 创建数据库连接

Option Explicit

```
Sub CreateConnection() '前期绑定 Connection
    Dim cnADO As New ADODB.Connection
    Dim strPath As String
    strPath = ThisWorkbook.Path & "\员工管理.accdb"
    On Error GoTo ErrMsg
    cnADO.Open "Provider=Microsoft.ACE.OLEDB.12.0;Data
Source=" & strPath
    If cnADO.State = adStateOpen Then
        MsgBox "数据库连接成功！" & vbCrLf & _
            vbCrLf & "ADO 版本为：" & cnADO.Version & _
            vbCrLf & _
            vbCrLf & "Connection 对象提供者名称：" & _
cnADO.Provider
        cnADO.Close
        Set cnADO = Nothing
    Else
        MsgBox "数据库连接失败！"
    End If
    Exit Sub
ErrMsg:
    MsgBox Err.Description, , "错误报告"
End Sub

Sub CreateConnectionQuery() '后期绑定 Connection
    Dim cnADO As Object, strPath As String
    strPath = ThisWorkbook.Path & "\员工管理.accdb"
    Set cnADO = CreateObject("ADODB.Connection")
    On Error GoTo ErrMsg
    With cnADO
        .Provider = "Microsoft.ACE.OLEDB.12.0"
        .ConnectionTimeout = 100
        .Open strPath
    End With
    If cnADO.State = 1 Then
        MsgBox "数据库连接成功！" & vbCrLf & _
            vbCrLf & "ADO 版本为：" & cnADO.Version & vbCrLf & _
            vbCrLf & "Connection 对象提供者名称：" & _
cnADO.Provider
        cnADO.Close
        Set cnADO = Nothing
    Else
        MsgBox "数据库连接失败！", vbInformation, "连接数据库"
    End If
    Exit Sub
ErrMsg:
    MsgBox Err.Description, , "错误报告"
End Sub
```

16.2 创建查询记录集

Option Explicit

Sub CreateQueryRS() 'Recordset 创建记录集

```
Dim cnADO As Object, rsADO As Object, strPath As String
Dim strSQL As String, i As Integer
    Set cnADO = CreateObject("ADODB.Connection")
    Set rsADO = CreateObject("ADODB.RecordSet")
    strPath = ThisWorkbook.Path & "\员工管理.accdb"
    On Error GoTo ErrMsg
    cnADO.Open "Provider=Microsoft.ACE.OLEDB.12.0;Data
Source=" & strPath
    strSQL = "SELECT * FROM 员工档案 WHERE 部门='办公室'"
    rsADO.Open strSQL, cnADO, 1, 3
    Cells.ClearContents
    For i = 0 To rsADO.Fields.Count - 1
        Cells(1, i + 1) = rsADO.Fields(i).Name
    Next i
    Range("A2").CopyFromRecordset rsADO
    rsADO.Close
    cnADO.Close
    Set rsADO = Nothing
    Set cnADO = Nothing
    Exit Sub
ErrMsg:
    MsgBox Err.Description, , "错误报告"
End Sub

Sub CreateQueryExecute() 'Execute 创建记录集
    Dim cnADO As Object, rsADO As Object, strPath As String
    Dim strSQL As String, i As Integer
    strPath = ThisWorkbook.Path & "\员工管理.accdb"
    Set cnADO = CreateObject("ADODB.Connection")
    On Error GoTo ErrMsg
    With cnADO
        .Provider = "Microsoft.ACE.OLEDB.12.0"
        .ConnectionTimeout = 100
        .Open strPath
    End With
    strSQL = "SELECT * FROM 员工档案 WHERE 部门='办公室'"
    Set rsADO = cnADO.Execute(strSQL)
    Cells.ClearContents
    For i = 0 To rsADO.Fields.Count - 1
        Cells(1, i + 1) = rsADO.Fields(i).Name
    Next i
    Range("A2").CopyFromRecordset rsADO
    rsADO.Close
    cnADO.Close
    Set rsADO = Nothing
    Set cnADO = Nothing
    Exit Sub
ErrMsg:
    MsgBox Err.Description, , "错误报告"
End Sub
```

第 16 章 数据库应用

16.3 动态创建 Access 数据库文件

Option Explicit

Sub SetDatabase() '创建 Access 数据库文件

Dim catADO As Object

Dim strPath As String

Dim strTable As String

Dim strSQL As String

Set catADO = CreateObject("ADOX.Catalog")

strPath = ThisWorkbook.Path & "\测试数据库.accdb"

strTable = "员工档案"

If Dir(strPath) <> "" Then Kill strPath

On Error GoTo ErrMsg

catADO.Create "Provider=Microsoft.ACE.OLEDB.12.0;Data

Source=" & strPath

strSQL = "CREATE TABLE " & strTable _

& "(员工编号 long not null primary key," _

& "姓名 text(20) not null," _

& "性别 text(1) not null," _

& "民族 text(20) not null," _

& "部门 text(20) not null," _

& "职务 text(20),电话 text(20)," _

& "学历 text(20),出生日期 date not null," _

& "籍贯 text(20),简历 longtext,照片 longbinary)"

catADO.ActiveConnection.Execute strSQL

MsgBox "创建数据库成功！" & vbCrLf _

& "数据库文件名为：" & strPath & vbCrLf _

& "数据表名称为：" & strTable & vbCrLf _

& "保存位置：" & ThisWorkbook.Path, _

vbOKOnly + vbInformation, "创建数据库"

Set catADO = Nothing

Exit Sub

ErrMsg:

MsgBox Err.Description, "错误报告"

End Sub

Sub SetDatabasePassWord() '创建 Access 数据库文件(含密码)

Dim catADO As Object

Dim strPath As String

Dim strTable As String

Dim strSQL As String

Set catADO = CreateObject("ADOX.Catalog")

strPath = ThisWorkbook.Path & "\测试数据库.accdb"

strTable = "员工档案"

If Dir(strPath) <> "" Then Kill strPath

On Error GoTo ErrMsg

catADO.Create "Provider=Microsoft.ACE.OLEDB.12.0;Data

Source=" & strPath _

& ";Jet OLEDB:Database Password=123456;"

strSQL = "CREATE TABLE " & strTable _

& "(员工编号 long not null primary key," _

& "姓名 text(20) not null," _

& "性别 text(1) not null," _

& "民族 text(20) not null," _

& "部门 text(20) not null," _

& "职务 text(20),电话 text(20)," _

& "学历 text(20),出生日期 date not null," _

& "籍贯 text(20),简历 longtext,照片 longbinary)"

catADO.ActiveConnection.Execute strSQL

MsgBox "创建数据库成功！" & vbCrLf _

& "数据库文件名为：" & strPath & vbCrLf _

& "数据表名称为：" & strTable & vbCrLf _

& "保存位置：" & ThisWorkbook.Path, _

vbOKOnly + vbInformation, "创建数据库"

Set catADO = Nothing

Exit Sub

ErrMsg:

MsgBox Err.Description, "错误报告"

End Sub

16.4 获取数据库所有表的信息

Option Explicit

Sub CatGetTables() 'Catalog 对象获取表的名称和类型

Dim catADO As Object

Dim objTable As Object

Dim strPath As String

Dim i As Integer

On Error GoTo ErrMsg

Set catADO = CreateObject("ADOX.Catalog")

strPath = ThisWorkbook.Path & "\员工管理.accdb"

catADO.ActiveConnection =

"Provider=Microsoft.ACE.OLEDB.12.0;Data Source=" & strPath

Cells.ClearContents

Cells(1, 1) = "数据表名"

Cells(1, 2) = "类型"

i = 1

For Each objTable In catADO.Tables

i = i + 1

Cells(i, 1) = objTable.Name

Cells(i, 2) = objTable.Type

Next objTable

Set objTable = Nothing

Set catADO = Nothing

Exit Sub

ErrMsg:

MsgBox Err.Description, "错误报告"

End Sub

Sub SchemGetTables() 'OpenSchema 方法 1

Dim cnADO As Object

Dim rsADO As Object

Dim strPath As String

Dim i As Integer

On Error GoTo ErrMsg

Set cnADO = CreateObject("ADODB.Connection")

第 16 章 数据库应用

接上页面 16.4

```
Set rsADO = CreateObject("ADODB.Recordset")
strPath = ThisWorkbook.Path & "\员工管理.accdb"
cnADO.Open "Provider=Microsoft.ACE.OLEDB.12.0;Data
Source=" & strPath
Set rsADO = cnADO.OpenSchema(20)
Cells.ClearContents
For i = 0 To rsADO.Fields.Count - 1
    Cells(1, i + 1) = rsADO.Fields(i).Name
Next i
Cells(2, 1).CopyFromRecordset rsADO
rsADO.Close
cnADO.Close
Set rsADO = Nothing
Set cnADO = Nothing
Exit Sub

ErrMsg:
MsgBox Err.Description, , "错误报告"
End Sub

Sub GetTablesSchemaLikeCat() 'OpenSchema 方法 2
Dim cnADO As Object,rsADO As Object,strPath As String
Dim i As Integer
On Error GoTo ErrMsg
Set cnADO = CreateObject("ADODB.Connection")
Set rsADO = CreateObject("ADODB.Recordset")
strPath = ThisWorkbook.Path & "\员工管理.accdb"
cnADO.Open "Provider=Microsoft.ACE.OLEDB.12.0;Data
Source=" & strPath
Set rsADO = cnADO.OpenSchema(20)
Cells.ClearContents
Cells(1, 1) = "数据表名"
Cells(1, 2) = "类型"
i = 1
Do Until rsADO.EOF
    i = i + 1
    Cells(i, 1) = rsADO.Fields("TABLE_NAME")
    Cells(i, 2) = rsADO.Fields("TABLE_TYPE")
    rsADO.MoveNext
Loop
rsADO.Close
cnADO.Close
Set rsADO = Nothing
Set cnADO = Nothing
Exit Sub

ErrMsg:
MsgBox Err.Description, , "错误报告"
End Sub

获取 Excel 工作表信息

Sub SchemaGetWorkSheets() 'OpenSchema 方法
Dim cnADO As Object,rsADO As Object,strMsg As String
```

```
Dim strShtName As String,i As Integer,j As Integer
Set cnADO = CreateObject("ADODB.Connection")
Set rsADO = CreateObject("ADODB.Recordset")
With cnADO
    .Provider = "Microsoft.ACE.OLEDB.12.0"
    .ConnectionString = "Extended Properties=Excel
12.0;Data Source=" & ThisWorkbook.FullName
    .CursorLocation = 3
    .Open
End With
Set rsADO = cnADO.OpenSchema(20)
For i = 1 To rsADO.RecordCount
strShtName = Replace(rsADO("TABLE_NAME").Value, "", "")
If Right(strShtName, 1) = "$" Then
    j = j + 1
    strMsg = strMsg & vbCrLf & Left(strShtName,
Len(strShtName) - 1)
End If
rsADO.MoveNext
Next i
MsgBox "本工作簿中有" & j & "个工作表，如下：" &
strMsg
rsADO.Close
cnADO.Close
Set rsADO = Nothing
Set cnADO = Nothing
End Sub

Sub CatGetWorkSheets() 'Cat 方法
Dim catADO As Object,objTable As Object,strMsg As String
Dim strPath As String,strShtName As String, i As Integer
Set catADO = CreateObject("ADOX.Catalog")
strPath = ThisWorkbook.FullName
catADO.ActiveConnection =
"Provider=Microsoft.ACE.OLEDB.12.0;" & _
"Extended Properties=Excel 12.0;Data Source=" &
strPath
For Each objTable In catADO.Tables
If objTable.Type = "TABLE" Then
    strShtName = Replace(objTable.Name, "", "")
    If Right(strShtName, 1) = "$" Then
        strMsg = strMsg & vbCrLf & Left(strShtName,
Len(strShtName) - 1)
        i = i + 1
    End If
End If
Next objTable
MsgBox "本工作簿中有" & i & "个工作表，如下：" &
strMsg
Set catADO = Nothing
Set objTable = Nothing
End Sub
```


第 16 章 数据库应用

判断表是否存在

```
Sub DetermineTableExist() '判断数据表是否存在
    Dim cnADO As Object, strPath As String
    Set cnADO = CreateObject("ADODB.Connection")
    strPath = ThisWorkbook.Path & "\员工管理.accdb"
    cnADO.Open "Provider=Microsoft.ACE.OLEDB.12.0;Data
Source=" & strPath

    If cnADO.OpenSchema(20, Array(Empty, Empty, "员工档案
", Empty)).EOF Then
        MsgBox "[员工档案] 表" & "不存在"
    Else
        MsgBox "[员工档案] 表" & "已存在"
    End If
End Sub

Sub DetermineWorkSheetExist() '判断工作表是否存在
    Dim cnADO As Object, strPath As String
    Set cnADO = CreateObject("ADODB.Connection")
    strPath = ThisWorkbook.FullName
    cnADO.Open
"Provider=Microsoft.Ace.OLEDB.12.0;Extended
Properties=Excel 12.0;Data Source=" & strPath

    If cnADO.OpenSchema(20, Array(Empty, Empty, "在册员工
$", Empty)).EOF Then
        MsgBox "“在册员工” 表" & "不存在"
    Else
        MsgBox "“在册员工” 表" & "已存在"
    End If
    cnADO.Close
    Set cnADO = Nothing
End Sub
```

16.5 动态创建数据表

```
Sub SetNewTable() '新建 ACCESS 数据表
    Dim cnADO As Object, rsADO As Object, strPath As String
    Dim strTable As String, strSQL As String

    Set cnADO = CreateObject("ADODB.Connection")
    Set rsADO = CreateObject("ADODB.Recordset")
    strPath = ThisWorkbook.Path & "\员工管理.accdb"
    strTable = "员工档案"

    On Error GoTo ErrMsg
    cnADO.Open "Provider=Microsoft.ACE.OLEDB.12.0;Data
Source=" & strPath

    Set rsADO = cnADO.OpenSchema(20, Array(Empty, Empty,
strTable, Empty))

    If Not rsADO.EOF Then
        strSQL = "DROP TABLE " & strTable
        cnADO.Execute strSQL
    End If

    strSQL = "CREATE TABLE " & strTable _
        & "(员工编号 long not null primary key," _
        & "姓名 text(20) not null,性别 text(1) not null," _
        & "民族 text(20) not null,部门 text(20) not null," _
        & "职务 text(20),电话 text(20),学历 text(20)," _
        & "出生日期 date not null,籍贯 text(20)," _
        & "简历 longtext,照片 longbinary)"
```

```
cnADO.Execute strSQL

MsgBox "创建数据表成功！" & vbCrLf _
    & "数据表名称为：" & strTable, "创建数据表"
rsADO.Close
cnADO.Close
Set rsADO = Nothing
Set cnADO = Nothing
Exit Sub

ErrMsg:
    MsgBox Err.Description, "错误报告"
End Sub

Sub SetNewWorksheet() '新建 Excel 工作表
    Dim cnADO As Object, strPath As String
    Dim strSQL As String, strTable As String
    Set cnADO = CreateObject("ADODB.Connection")
    strPath = ThisWorkbook.Path & "\测试工作簿.xlsx"
    If Dir(strPath) <> "" Then Kill strPath
    cnADO.Open "Provider=Microsoft.Ace.Oledb.12.0;" _
        & "Extended Properties=Excel 12.0 Xml;" _
        & "Data Source=" & strPath
    strTable = "说明"
    strSQL = "CREATE TABLE " & strTable & "(说明 text)"
    cnADO.Execute strSQL
    strTable = "员工档案"
    strSQL = "CREATE TABLE " & strTable
        & "( 员工编号 long,姓名 text,性别 text," _
        & "民族 text,部门 text,职务 text,电话 text," _
        & "学历 text,出生日期 date,籍贯 text,简历 text)"
    cnADO.Execute strSQL
    MsgBox "创建文件成功！" & vbCrLf _
        & "工作簿文件名为：" & strPath & vbCrLf _
        & "保存位置：" & ThisWorkbook.Path
    cnADO.Close
    Set cnADO = Nothing
End Sub

Sub DeleteDatatable() '删除 ACCESS 数据表
    Dim cnADO As Object, rsADO As Object, strPath As String
    Dim strTable As String, strSQL As String

    Set cnADO = CreateObject("ADODB.Connection")
    Set rsADO = CreateObject("ADODB.Recordset")
    strPath = ThisWorkbook.Path & "\员工管理.accdb"
    strTable = "员工档案"

    On Error GoTo ErrMsg
    cnADO.Open "Provider=Microsoft.ACE.OLEDB.12.0;Data
Source=" & strPath

    Set rsADO = cnADO.OpenSchema(20, Array(Empty, Empty,
strTable, Empty))

    If Not rsADO.EOF Then
        strSQL = "DROP TABLE " & strTable
        cnADO.Execute strSQL
        MsgBox "已经删除数据表。", vbInformation, "删除数
据表"
    Else
        MsgBox "数据表不存在。", vbInformation, "删除数据
表"
    End If
    rsADO.Close
    cnADO.Close
    Set rsADO = Nothing
    Set cnADO = Nothing
    Exit Sub

ErrMsg:
    MsgBox Err.Description, "错误报告"
End Sub
```

第 16 章 数据库应用

16.6 动态创建链接表

Sub LinkExcel()

```
Dim cnADO As Object
Dim catADO As Object
Dim tabADO As Object
Dim rsADO As Object
Dim strPath As String
Dim strLinkPath As String
Dim strName As String
Set cnADO = CreateObject("ADODB.Connection")
Set catADO = CreateObject("ADOX.Catalog")
Set tabADO = CreateObject("ADOX.Table")
strPath = ThisWorkbook.Path & "\员工管理.accdb"
strLinkPath = ThisWorkbook.Path & "\临聘人员奖金.xlsx"
strName = "临聘人员奖金 EX"
cnADO.Open "Provider=Microsoft.ACE.OLEDB.12.0;Data
Source=" & strPath
Set rsADO = cnADO.OpenSchema(20, Array(Empty, Empty,
strName, Empty))
If Not rsADO.EOF Then
    cnADO.Execute "DROP TABLE " & strName
End If
Set catADO.ActiveConnection = cnADO
With tabADO
    .Name = strName
    .ParentCatalog = catADO
    .Properties("Jet OLEDB:Link Datasource").Value =
strLinkPath
    .Properties("Jet OLEDB:Remote Table Name").Value =
"临聘人员奖金$"
    .Properties("Jet OLEDB:Link Provider String").Value =
"Excel 12.0;HDR=Yes"
    .Properties("Jet OLEDB:Create Link").Value = True
End With
catADO.Tables.Append tabADO
cnADO.Close
Set cnADO = Nothing
Set rsADO = Nothing
Set catADO = Nothing
Set tabADO = Nothing
```

End Sub

Sub LinkAccess()

```
Dim cnADO As Object, catADO As Object, tabADO As Object
Dim rsADO As Object, strPath As String, strLinkPath As String
Dim strName As String
Set cnADO = CreateObject("ADODB.Connection")
Set catADO = CreateObject("ADOX.Catalog")
Set tabADO = CreateObject("ADOX.Table")
strPath = ThisWorkbook.Path & "\员工管理.accdb"
strLinkPath = ThisWorkbook.Path & "\奖金.accdb"
strName = "奖金 Acc"
cnADO.Open "Provider=Microsoft.ACE.OLEDB.12.0;Data
Source=" & strPath
Set rsADO = cnADO.OpenSchema(20, Array(Empty, Empty,
strName, Empty))
If Not rsADO.EOF Then
```

```
cnADO.Execute "DROP TABLE " & strName
End If
Set catADO.ActiveConnection = cnADO
With tabADO
    .Name = strName
    .ParentCatalog = catADO
    .Properties("Jet OLEDB:Link Provider String").Value =
"MS Access;Pwd=;"
    .Properties("Jet OLEDB:Link Datasource").Value =
strLinkPath
    .Properties("Jet OLEDB:Remote Table Name").Value =
"奖金名单"
    .Properties("Jet OLEDB:Create Link").Value = True
End With
catADO.Tables.Append tabADO
cnADO.Close
Set cnADO = Nothing
Set rsADO = Nothing
Set catADO = Nothing
Set tabADO = Nothing
End Sub
Sub LinkText()
Dim cnADO As Object, catADO As Object, tabADO As Object
Dim rsADO As Object, strPath As String, strLinkPath As String
Dim strName As String
Set cnADO = CreateObject("ADODB.Connection")
Set catADO = CreateObject("ADOX.Catalog")
Set tabADO = CreateObject("ADOX.Table")
strPath = ThisWorkbook.Path & "\员工管理.accdb"
strLinkPath = ThisWorkbook.Path & "\"
strName = "捐款 TEXT"
cnADO.Open "Provider=Microsoft.ACE.OLEDB.12.0;Data
Source=" & strPath
Set rsADO = cnADO.OpenSchema(20, Array(Empty, Empty,
strName, Empty))
If Not rsADO.EOF Then
    cnADO.Execute "DROP TABLE " & strName
End If
Set catADO.ActiveConnection = cnADO
With tabADO
    .Name = strName
    .ParentCatalog = catADO
    .Properties("Jet OLEDB:Link Provider String").Value =
"Text;FMT=Delimited;HDR=Yes"
    .Properties("Jet OLEDB:Link Datasource").Value =
strLinkPath
    .Properties("Jet OLEDB:Remote Table Name").Value =
"捐款.txt"
    .Properties("Jet OLEDB:Create Link").Value = True
End With
catADO.Tables.Append tabADO
cnADO.Close
Set cnADO = Nothing
Set rsADO = Nothing
Set catADO = Nothing
Set tabADO = Nothing
End Sub
```

第 16 章 数据库应用

16.7 在数据库中创建视图

```
Sub CreateView()  
    Dim cnADO As Object  
    Dim rsADO As Object  
    Dim strSQL As String  
    Dim strPath As String  
    Dim strViewName As String  
    Set cnADO = CreateObject("ADODB.Connection")  
    strPath = ThisWorkbook.Path & "\员工管理.accdb"  
    strViewName = "当月生日"  
    On Error GoTo ErrMsg  
    cnADO.Open "Provider=Microsoft.ACE.OLEDB.12.0;Data  
Source=" & strPath  
    Set rsADO = cnADO.OpenSchema(20, Array(Empty, Empty,  
strViewName, Empty))  
    If Not rsADO.EOF Then  
        strSQL = "DROP TABLE " & strViewName  
        cnADO.Execute strSQL  
    End If  
    strSQL = "CREATE VIEW " & strViewName _  
        & " AS SELECT * FROM 员工档案 " _  
        & "Where Month(出生日期)= Month(Now)"  
    cnADO.Execute strSQL  
    MsgBox "视图创建成功!", , "创建视图"  
    rsADO.Close  
    cnADO.Close  
    Set rsADO = Nothing  
    Set cnADO = Nothing  
    Exit Sub  
ErrMsg:  
    MsgBox Err.Description, , "错误报告"  
End Sub  
16.8 向数据表添加、删除、修改字段  
Sub DeleteAddModifyFields() '优先链接数据库  
    Dim cnADO As Object  
    Dim strPath As String  
    Dim strTable As String  
    Dim strSQL As String  
    Dim strMsg As String  
    Set cnADO = CreateObject("ADODB.Connection")  
    strPath = ThisWorkbook.Path & "\测试数据库.accdb"  
    strTable = "员工档案"  
    cnADO.Open "Provider=Microsoft.Ace.OLEDB.12.0;Data  
Source=" & strPath  
    On Error Resume Next  
    strSQL = "ALTER TABLE " & strTable & " DROP 电子邮箱,  
工作时间"  
    cnADO.Execute strSQL  
    On Error GoTo ErrMsg  
    strSQL = "ALTER TABLE " & strTable & " ADD 电子邮箱  
TEXT(10),工作时间 DOUBLE"  
    cnADO.Execute strSQL  
    MsgBox "字段添加成功.", vbInformation, "提示"  
    strSQL = "ALTER TABLE " & strTable & " ALTER 工作时间  
DATETIME"  
    cnADO.Execute strSQL  
    MsgBox "字段数据类型修改成功.", vbInformation, "提示"  
    strSQL = "ALTER TABLE " & strTable & " ALTER 电子邮箱  
TEXT(50)"
```

```
cnADO.Execute strSQL  
MsgBox "字段长度修改成功.", vbInformation, "提示"  
cnADO.Close  
Set cnADO = Nothing  
Exit Sub  
ErrMsg:  
    MsgBox Err.Description, , "错误报告"  
End Sub  
16.9 创建多数据库查询  
Option Explicit  
Sub MultipleDatabaseQuery()  
    Dim cnADO As Object  
    Dim strPath As String  
    Dim strSQL As String  
    Set cnADO = CreateObject("ADODB.Connection")  
    strPath = ThisWorkbook.Path & "\员工管理.accdb"  
    On Error GoTo ErrMsg  
    cnADO.Open "Provider=Microsoft.ACE.OLEDB.12.0;Data  
Source=" & strPath  
    strSQL = "SELECT 人数 FROM [Excel 12.0;Database=" _  
        & ThisWorkbook.FullName & ";].[学历汇总$A1:A" _  
        & Range("A" & Rows.Count).End(xlUp).Row & "]" A" _  
        & " LEFT JOIN (SELECT 学历,COUNT(学历) AS 人数"  
    _  
        & " FROM 员工档案 GROUP BY 学历) B ON A.学历  
=B.学历"  
    Range("B2").CopyFromRecordset cnADO.Execute(strSQL)  
    cnADO.Close  
    Set cnADO = Nothing  
    Exit Sub  
ErrMsg:  
    MsgBox Err.Description, , "错误报告"  
End Sub  
Sub MultipleDatabaseQueryExcel() '优先链接 Excel 工作簿  
    Dim cnADO As Object  
    Dim strPath As String  
    Dim strSQL As String  
    Set cnADO = CreateObject("ADODB.Connection")  
    strPath = ThisWorkbook.Path & "\员工管理.accdb"  
    On Error GoTo ErrMsg  
    cnADO.Open  
    "Provider=Microsoft.ACE.OLEDB.12.0;Extended  
Properties=Excel 12.0;Data Source=" &  
ThisWorkbook.FullName  
    strSQL = "SELECT 人数 FROM [学历汇总$A1:A" &  
    Range("A" & Rows.Count).End(xlUp).Row _  
        & "]" A LEFT JOIN (SELECT 学历,COUNT(学历) AS 人  
    数 FROM [MS Access;pwd=;Database=" & strPath & ";].员工档  
    案 GROUP BY 学历) B ON A.学历=B.学历"  
    Range("B2").CopyFromRecordset cnADO.Execute(strSQL)  
    cnADO.Close  
    Set cnADO = Nothing  
    Exit Sub  
ErrMsg:  
    MsgBox Err.Description, , "错误报告"  
End Sub
```

第 16 章 数据库应用

16.10 将工作表、数据表或查询生成新的数据表

Sub CreateNewDataTable() '将工作表生成新数据表

Dim cnADO As Object

Dim rsADO As Object

Dim strPath As String

Dim strTable As String

Dim strSQL As String

Set cnADO = CreateObject("ADODB.Connection")

Set rsADO = CreateObject("ADODB.Recordset")

strPath = ThisWorkbook.Path & "\员工管理.accdb"

strTable = "测试表"

On Error GoTo ErrMsg

cnADO.Open "Provider=Microsoft.ACE.OLEDB.12.0;Data

Source=" & strPath

Set rsADO = cnADO.OpenSchema(20, Array(Empty, Empty, strTable, Empty))

If Not rsADO.EOF Then

strSQL = "DROP TABLE " & strTable

cnADO.Execute strSQL

End If

strSQL = "SELECT * INTO " & strTable _

& " FROM [Excel 12.0;Database=" _

& ThisWorkbook.FullName & ";].[" &

ActiveSheet.Name & "\$"]"

cnADO.Execute strSQL

MsgBox "已经将工作表数据生成新数据表。",

vbInformation, "生成新数据表"

rsADO.Close

cnADO.Close

Set rsADO = Nothing

Set cnADO = Nothing

Exit Sub

ErrMsg:

MsgBox Err.Description, "错误报告"

End Sub

Sub CreateNewDatatable_1() '将查询生成新数据表

Dim cnADO As Object

Dim rsADO As Object

Dim strPath As String

Dim strTable As String

Dim strSQL As String

Set cnADO = CreateObject("ADODB.Connection")

Set rsADO = CreateObject("ADODB.Recordset")

strPath = ThisWorkbook.Path & "\员工管理.accdb"

strTable = "测试表"

On Error GoTo ErrMsg

cnADO.Open "Provider=Microsoft.ACE.OLEDB.12.0;Data

Source=" & strPath

Set rsADO = cnADO.OpenSchema(20, Array(Empty, Empty, strTable, Empty))

If Not rsADO.EOF Then

strSQL = "DROP TABLE " & strTable

cnADO.Execute strSQL

End If

strSQL = "SELECT * FROM [Excel 12.0;Database=" &

ThisWorkbook.FullName & ";].[" & ActiveSheet.Name & "\$]

WHERE 籍贯 LIKE '%北京%'"

strSQL = "SELECT * INTO " & strTable & " FROM (" & strSQL

& ")"

cnADO.Execute strSQL

MsgBox "已经将查询数据生成新数据表。",

vbInformation, "生成新数据表"

rsADO.Close

cnADO.Close

Set rsADO = Nothing

Set cnADO = Nothing

Exit Sub

ErrMsg:

MsgBox Err.Description, "错误报告"

End Sub

Sub CreateNewDatatable_2() '将工作表生成新数据表，连接工作簿

Dim cnADO As Object

Dim strPath As String

Dim strTable As String

Dim strSQL As String

Set cnADO = CreateObject("ADODB.Connection")

strPath = ThisWorkbook.Path & "\员工管理.accdb"

strTable = "测试表"

On Error GoTo ErrMsg

cnADO.Open

"Provider=Microsoft.ACE.OLEDB.12.0;Extended

Properties=Excel 12.0;Data Source=" &

ThisWorkbook.FullName

strSQL = "SELECT * INTO [MS Access;pwd=;Database=" &

strPath & ";].[" & strTable & " FROM [" & ActiveSheet.Name &

"\$"]"

cnADO.Execute strSQL

MsgBox "已经将工作表数据生成新数据表。",

vbInformation, "生成新数据表"

cnADO.Close

Set cnADO = Nothing

Exit Sub

ErrMsg:

MsgBox Err.Description, "错误报告"

End Sub

Sub CreateNewWorksheet() '将数据表生成新的工作表

Dim cnADO As Object

Dim strPath As String

Dim strTable As String

Dim strSQL As String

Set cnADO = CreateObject("ADODB.Connection")

strPath = ThisWorkbook.Path & "\导出工作簿.xlsx"

strTable = "测试表"

On Error GoTo ErrMsg

cnADO.Open "Provider=Microsoft.ACE.OLEDB.12.0;Data

Source=" _

& ThisWorkbook.Path & "\员工管理.accdb"

If Dir(strPath) <> "" Then Kill strPath

strSQL = "SELECT * INTO [Excel 12.0 Xml;Database=" _

& strPath & ";].Sheet1 FROM " & strTable

cnADO.Execute strSQL

MsgBox "创建文件成功！" & vbCrLf _

& "工作簿文件名为：" & strPath & vbCrLf _

& "保存位置：" & ThisWorkbook.Path

cnADO.Close

Set cnADO = Nothing

Exit Sub

ErrMsg:

MsgBox Err.Description, "错误报告"

End Sub

第 16 章 数据库应用

Sub CreateNewWorksheet_1() '数据表导出工作表，连接本工作簿

```
Dim cnADO As Object
Dim strPath As String
Dim strTable As String
Dim strSQL As String
Set cnADO = CreateObject("ADODB.Connection")
strPath = ThisWorkbook.Path & "\导出工作簿.xlsx"
strTable = "测试表"
On Error GoTo ErrMsg
cnADO.Open
"Provider=Microsoft.ACE.OLEDB.12.0;Extended
Properties=Excel 12.0 Xml;Data Source=" &
ThisWorkbook.FullName
If Dir(strPath) <> "" Then Kill strPath
strSQL = "SELECT * INTO [" & strPath & ";].Sheet1 FROM
[MS Access;pwd=;Database=" & ThisWorkbook.Path & "\员工
管理.accdb;]." & strTable
cnADO.Execute strSQL
MsgBox "创建文件成功！" & vbCrLf & "工作簿文件名为：" & strPath & vbCrLf & "保存位置：" & ThisWorkbook.Path
cnADO.Close
Set cnADO = Nothing
Exit Sub
ErrMsg:
MsgBox Err.Description, , "错误报告"
End Sub
```

Sub DeleteDatatable() '删除数据表

```
Dim cnADO As Object
Dim rsADO As Object
Dim strPath As String
Dim strTable As String
Dim strSQL As String
Set cnADO = CreateObject("ADODB.Connection")
Set rsADO = CreateObject("ADODB.Recordset")
strPath = ThisWorkbook.Path & "\员工管理.accdb"
strTable = "测试表"
On Error GoTo ErrMsg
cnADO.Open "Provider=Microsoft.ACE.OLEDB.12.0;Data
Source=" & strPath
Set rsADO = cnADO.OpenSchema(20, Array(Empty, Empty,
strTable, Empty))
If Not rsADO.EOF Then
    strSQL = "DROP TABLE " & strTable
    cnADO.Execute strSQL
    MsgBox "已经删除数据表。", vbInformation, "删除数
据表"
Else
    MsgBox "数据表不存在。", vbInformation, "删除数据表"
End If
rsADO.Close
cnADO.Close
Set rsADO = Nothing
Set cnADO = Nothing
Exit Sub
ErrMsg:
MsgBox Err.Description, , "错误报告"
End Sub
```

16.11 批量删除数据表中的记录

Option Explicit

```
Sub DeleteExistRecords()
Dim cnADO As Object
Dim rsADO As Object
Dim strPath As String
Dim strTable As String
Dim strWhere As String
Dim strSQL As String
Dim strMsg As String
Set cnADO = CreateObject("ADODB.Connection")
Set rsADO = CreateObject("ADODB.Recordset")
strPath = ThisWorkbook.Path & "\员工管理.accdb"
strTable = "员工档案"
On Error GoTo ErrMsg
cnADO.Open "Provider=Microsoft.Ace.OLEDB.12.0;Data
Source=" & strPath
strWhere = " WHERE EXISTS(" _
    & "SELECT * FROM [Excel 12.0;Database=" _
    & ThisWorkbook.FullName & "].[" &
ActiveSheet.Name & "$" _
    & Range("a1").CurrentRegion.Address(0, 0) & "]" _
    & "WHERE 员工编号=" & strTable & ".员工编号)"
strSQL = "SELECT 员工编号 FROM 员工档案" &
strWhere
rsADO.Open strSQL, cnADO, 1, 3
If rsADO.RecordCount > 0 Then
    strSQL = "DELETE FROM 员工档案" & strWhere
    cnADO.Execute strSQL
    MsgBox rsADO.RecordCount & "条记录被删除。",
vbInformation, "提示"
Else
    MsgBox "没有发现需要删除的记录。",
vbInformation, "提示"
End If
rsADO.Close
cnADO.Close
Set rsADO = Nothing
Set cnADO = Nothing
Exit Sub
ErrMsg:
MsgBox Err.Description, , "错误报告"
End Sub
```

Sub DeleteExistRecordsByIN() 'IN 删除工作表存在的记录

```
Dim cnADO As Object
Dim rsADO As Object
Dim strPath As String
Dim strTable As String
Dim strWhere As String
Dim strSQL As String
Dim strMsg As String
Set cnADO = CreateObject("ADODB.Connection")
Set rsADO = CreateObject("ADODB.Recordset")
strPath = ThisWorkbook.Path & "\员工管理.accdb"
strTable = "员工档案"
On Error GoTo ErrMsg
cnADO.Open "Provider=Microsoft.Ace.OLEDB.12.0;Data
Source=" & strPath '连接数据库
strWhere = " WHERE 员工编号 IN (SELECT 员工编号
FROM [Excel 12.0;Database=" & ThisWorkbook.FullName &
"].[" & ActiveSheet.Name & "$" _
```

第 16 章 数据库应用 接上页 16.11

```
& Range("a1").CurrentRegion.Address(0, 0) & "]"
strSQL = "SELECT 员工编号 FROM 员工档案" &
strWhere
rsADO.Open strSQL, cnADO, 1, 3
If rsADO.RecordCount > 0 Then
    strSQL = "DELETE FROM 员工档案" & strWhere
    [m1] = strSQL
    cnADO.Execute strSQL
    MsgBox rsADO.RecordCount & "条记录被删除。",
vbInformation, "提示"
Else
    MsgBox "没有发现需要删除的记录。",
vbInformation, "提示"
End If
rsADO.Close
cnADO.Close
Set rsADO = Nothing
Set cnADO = Nothing
Exit Sub
ErrMsg:
MsgBox Err.Description, , "错误报告"
End Sub
Sub DeleteExistRecordsByInner() '内连接 删除工作表存在的记录
    Dim cnADO As Object
    Dim rsADO As Object
    Dim strPath As String
    Dim strTable As String
    Dim strWhere As String
    Dim strSQL As String
    Dim strMsg As String
    Set cnADO = CreateObject("ADODB.Connection")
    Set rsADO = CreateObject("ADODB.Recordset")
    strPath = ThisWorkbook.Path & "\员工管理.accdb"
    strTable = "员工档案"
    On Error GoTo ErrMsg
    cnADO.Open "Provider=Microsoft.Ace.OLEDB.12.0;Data
Source=" & strPath '连接数据库
    strWhere = strTable & " A,[Excel 12.0;Database=" &
ThisWorkbook.FullName & ";].[ " & ActiveSheet.Name & "$" _
& Range("A1").CurrentRegion.Address(0, 0) & "]" B
WHERE A.员工编号=B.员工编号"
    strSQL = "SELECT B.员工编号 FROM " & strWhere
    rsADO.Open strSQL, cnADO, 1, 3
    If rsADO.RecordCount > 0 Then
        strSQL = "DELETE A.* FROM " & strWhere
        cnADO.Execute strSQL
        MsgBox rsADO.RecordCount & "条记录被删除。",
vbInformation, "提示"
    Else
        MsgBox "没有发现需要删除的记录。",
vbInformation, "提示"
    End If
    rsADO.Close
    cnADO.Close
    Set rsADO = Nothing
    Set cnADO = Nothing
    Exit Sub
ErrMsg:
MsgBox Err.Description, , "错误报告"
```

End Sub

Sub DeleteExistRecordsByLeftOut() '左外连接删除数据表存在,工作表不存在的记录

```
Dim cnADO As Object
Dim rsADO As Object
Dim strPath As String
Dim strTable As String
Dim strWhere As String
Dim strSQL As String
Dim strMsg As String
Set cnADO = CreateObject("ADODB.Connection")
Set rsADO = CreateObject("ADODB.Recordset")
strPath = ThisWorkbook.Path & "\员工管理.accdb"
strTable = "员工档案"
On Error GoTo ErrMsg
cnADO.Open "Provider=Microsoft.Ace.OLEDB.12.0;Data
Source=" & strPath '连接数据库
    strWhere = strTable & " A LEFT JOIN [Excel
12.0;Database=" & ThisWorkbook.FullName & ";].[ " &
ActiveSheet.Name & "$" &
Range("A1").CurrentRegion.Address(0, 0) _
& "]" B ON A.员工编号=B.员工编号 WHERE B.员工编
号 IS NULL"
    strSQL = "SELECT A.员工编号 FROM " & strWhere
    rsADO.Open strSQL, cnADO, 1, 3
    If rsADO.RecordCount > 0 Then
        strSQL = "DELETE A.* FROM " & strWhere
        cnADO.Execute strSQL
        MsgBox rsADO.RecordCount & "条记录被删除。",
vbInformation, "提示"
    Else
        MsgBox "没有发现需要删除的记录。",
vbInformation, "提示"
    End If
    rsADO.Close
    cnADO.Close
    Set rsADO = Nothing
    Set cnADO = Nothing
    Exit Sub
ErrMsg:
MsgBox Err.Description, , "错误报告"
End Sub
```

Sub DeleteNotexistRecords() 'NOT EXISTS 删除数据表存在,工作表不存在的记录

```
Dim cnADO As Object
Dim rsADO As Object
Dim strPath As String
Dim strTable As String
Dim strWhere As String
Dim strSQL As String
Dim strMsg As String
Set cnADO = CreateObject("ADODB.Connection")
Set rsADO = CreateObject("ADODB.Recordset")
strPath = ThisWorkbook.Path & "\员工管理.accdb"
strTable = "员工档案"
On Error GoTo ErrMsg
cnADO.Open "Provider=Microsoft.Ace.OLEDB.12.0;Data
Source=" & strPath '连接数据库
    strWhere = " WHERE NOT EXISTS(SELECT * FROM [Excel
12.0;Database=" & ThisWorkbook.FullName & ";].[ " &
```


第 16 章 数据库应用

接上页 16.11

ActiveSheet.Name & "\$" _

```
        & Range("a1").CurrentRegion.Address(0, 0) & "]"
WHERE 员工编号=" & strTable & ".员工编号)"
    strSQL = "SELECT 员工编号 FROM 员工档案" &
strWhere
rsADO.Open strSQL, cnADO, 1, 3
If rsADO.RecordCount > 0 Then
    strSQL = "DELETE FROM 员工档案" & strWhere
    cnADO.Execute strSQL
    MsgBox rsADO.RecordCount & "条记录被删除。",
vbInformation, "提示"
Else
    MsgBox "没有发现需要删除的记录。",
vbInformation, "提示"
End If
rsADO.Close
cnADO.Close
Set rsADO = Nothing
Set cnADO = Nothing
Exit Sub

ErrMsg:
MsgBox Err.Description, , "错误报告"
End Sub
```

Sub DeleteRetirementRecords() '删除退休员工(上个月男达到60岁,女达到55岁)

```
Dim cnADO As Object
Dim rsADO As Object
Dim strPath As String
Dim strTable As String
Dim strWhere As String
Dim strSQL As String
Dim strMsg As String
Set cnADO = CreateObject("ADODB.Connection")
Set rsADO = CreateObject("ADODB.Recordset")
strPath = ThisWorkbook.Path & "\员工管理.accdb"
strTable = "员工档案"
On Error GoTo ErrMsg
cnADO.Open "Provider=Microsoft.Ace.OLEDB.12.0;Data
Source=" & strPath '连接数据库
strWhere = " WHERE (性别='男' AND 出生日期<#" &
DateSerial(Year(Date) - 60, Month(Date), 1) & "#) OR (性别='女'
AND 出生日期<=" & DateSerial(Year(Date) - 55, Month(Date),
1) & "#)"
strSQL = "SELECT 员工编号 FROM 员工档案" &
strWhere
rsADO.Open strSQL, cnADO, 1, 3
If rsADO.RecordCount > 0 Then
    strSQL = "DELETE FROM 员工档案" & strWhere
[L1] = strSQL
    cnADO.Execute strSQL
    MsgBox rsADO.RecordCount & "条记录被删除。",
vbInformation, "提示"
Else
    MsgBox "没有发现需要删除的记录。",
vbInformation, "提示"
End If
rsADO.Close
```

```
cnADO.Close
Set rsADO = Nothing
Set cnADO = Nothing
Exit Sub
```

ErrMsg:

MsgBox Err.Description, , "错误报告"

End Sub

.....下面程序上传数据用

Sub UpdateaddRecords() '更新数据库已经存在记录,插入不存在记录

```
Dim cnADO As Object
Dim rsADO As Object
Dim strPath As String
Dim strTable As String
Dim strTemp As String
Dim strSQL As String
Dim strMsg As String
Dim arrFields As Variant
Dim i As Long
Set cnADO = CreateObject("ADODB.Connection")
Set rsADO = CreateObject("ADODB.Recordset")
strPath = ThisWorkbook.Path & "\员工管理.accdb"
strTable = "员工档案"
On Error GoTo ErrMsg
cnADO.Open "Provider=Microsoft.Ace.OLEDB.12.0;Data
Source=" & strPath '连接数据库

' 生成更新 SQL 语句(请注意 Office2007 后需要加 IMEX=0
参数)
strSQL = "SELECT B.员工编号 FROM " & strTable & "
A,[Excel 12.0;IMEX=0;Database=" & ThisWorkbook.FullName &
";].[ " & ActiveSheet.Name & "$" _
& Range("A1").CurrentRegion.Address(0, 0) & "]" B
WHERE A.员工编号=B.员工编号"
rsADO.Open strSQL, cnADO, 1, 3
If rsADO.RecordCount > 0 Then
    arrFields = Range("A1:K1") '工作表中的字段名写入
数组
' 生成更新字符串, 如: a.姓名=b.姓名,a.性别=b.性
别,.....
For i = 2 To UBound(arrFields, 2)
    strTemp = strTemp & ",a." & arrFields(1, i) &
"=b." & arrFields(1, i)
Next
strSQL = "UPDATE " & strTable & " A,[Excel
12.0;IMEX=0;Database=" & ThisWorkbook.FullName & ";].[ " &
ActiveSheet.Name & "$" _
& Range("A1").CurrentRegion.Address(0, 0) & "]"
B SET " & Mid(strTemp, 2) & " WHERE A.员工编号=B.员工编号
"
cnADO.Execute strSQL
strMsg = rsADO.RecordCount & "条记录已更新!"
End If
If rsADO.RecordCount < Range("A" &
Rows.Count).End(xlUp).Row - 1 Then
' 生成数据库不存在记录的 SQL 语句
strSQL = " SELECT A.* FROM [Excel 12.0;Database=" &
ThisWorkbook.FullName & ";].[ " & ActiveSheet.Name & "$" &
Range("A1").CurrentRegion.Address(0, 0) _
& "]" A LEFT JOIN " & strTable & " B ON A.员工编
```

第 16 章 数据库应用 接上页 16.11

号=B.员工编号 WHERE B.员工编号 IS NULL"

Set rsADO = CreateObject("ADODB.Recordset")

rsADO.Open strSQL, cnADO, 1, 3

'插入数据库不存在记录

If rsADO.RecordCount > 0 Then '如果工作表中含有数据库不存在记录

strSQL = "INSERT INTO " & strTable & strSQL

cnADO.Execute strSQL

strMsg = strMsg & vbCrLf & rsADO.RecordCount & "条记录已添加到数据库！"

End If

End If

MsgBox strMsg, vbInformation, "提示"

' 关闭连接释放内存

rsADO.Close

cnADO.Close

Set rsADO = Nothing

Set cnADO = Nothing

Exit Sub

ErrMsg:

MsgBox Err.Description, , "错误报告"

End Sub

16.12 从表或查询中批量向数据表添加记录

Sub AddRecords()

Dim cnADO As Object

Dim strPath As String

Dim strTable As String

Dim strSQL As String

Set cnADO = CreateObject("ADODB.Connection")

strPath = ThisWorkbook.Path & "\员工管理.accdb"

strTable = "员工档案"

On Error GoTo ErrMsg

cnADO.Open "Provider=Microsoft.Ace.OLEDB.12.0;Data

Source=" & strPath

strSQL = "DELETE FROM " & strTable & " A WHERE EXISTS("

— & "SELECT * FROM [Excel 12.0;Database=" & _
ThisWorkbook.FullName & "].[" & ActiveSheet.Name

& "\$" _

& Range("a1").CurrentRegion.Address(0, 0) & "]" _

& "WHERE 员工编号=A.员工编号)"

cnADO.Execute strSQL

strSQL = "INSERT INTO " & strTable _

& "(员工编号,姓名,性别,民族,部门,职务," _

& "电话,学历,出生日期,籍贯,简历)" _

& "SELECT 员工编号,姓名,性别,民族,部门,职务," _

& "电话,学历,出生日期,籍贯,简历 " _

& "FROM [Excel 12.0;Database=" _

& ThisWorkbook.FullName & "].[" &

ActiveSheet.Name & "\$" _

& Range("A1").CurrentRegion.Address(0, 0) & "]"

cnADO.Execute strSQL

MsgBox "纪录添加成功。", vbInformation, "添加纪录"

cnADO.Close

Set cnADO = Nothing

Exit Sub

ErrMsg:

MsgBox Err.Description, , "错误报告"

End Sub

Sub AddRecords_1() '从查询中向数据表添加纪录

Dim cnADO As Object

Dim strPath As String

Dim strTable As String

Dim strSQL As String

Set cnADO = CreateObject("ADODB.Connection")

strPath = ThisWorkbook.Path & "\员工管理.accdb"

strTable = "员工档案"

On Error GoTo ErrMsg

cnADO.Open "Provider=Microsoft.Ace.OLEDB.12.0;Data

Source=" & strPath '连接数据库

strSQL = "DELETE FROM " & strTable & " A WHERE

EXISTS(SELECT * FROM [Excel 12.0;Database=" &

ThisWorkbook.FullName & "].[" & ActiveSheet.Name & "\$]

WHERE 性别='男' AND 员工编号=A.员工编号)"

cnADO.Execute strSQL

strSQL = "INSERT INTO " & strTable & " SELECT * FROM
[Excel 12.0;Database=" & ThisWorkbook.FullName & "].[" &

ActiveSheet.Name & "\$] WHERE 性别='男'"

cnADO.Execute strSQL

MsgBox "纪录添加成功。", vbInformation, "添加纪录"

cnADO.Close

Set cnADO = Nothing

Exit Sub

ErrMsg:

MsgBox Err.Description, , "错误报告"

End Sub

Sub AddRecords_2() '从数据表中向数据表添加纪录

Dim cnADO As Object

Dim strPath As String

Dim strTable As String

Dim strSQL As String

Set cnADO = CreateObject("ADODB.Connection")

strPath = ThisWorkbook.Path & "\员工管理.accdb"

strTable = "员工档案"

On Error GoTo ErrMsg

cnADO.Open "Provider=Microsoft.Ace.OLEDB.12.0;Data

Source=" & strPath '连接数据库

strSQL = "DELETE FROM " & strTable & " A WHERE

EXISTS(SELECT * FROM 测试表 WHERE 性别='女' AND 员工
编号=A.员工编号)"

cnADO.Execute strSQL

strSQL = "INSERT INTO " & strTable & " SELECT * FROM 测
试表 WHERE 性别='女'"

cnADO.Execute strSQL

MsgBox "纪录添加成功。", vbInformation, "添加纪录"

cnADO.Close

Set cnADO = Nothing

Exit Sub

ErrMsg:

MsgBox Err.Description, , "错误报告"

End Sub

Sub UpdateaddRecords() '更新数据库已经存在记录，插入不
存在记录

Dim cnADO As Object

Dim rsADO As Object

Dim strPath As String

Dim strTable As String

Dim strTemp As String

第 16 章 数据库应用 接上页 16. 12

```
Dim strSQL As String
Dim strMsg As String
Dim avntFields As Variant
Dim i As Long
Set cnADO = CreateObject("ADODB.Connection")
Set rsADO = CreateObject("ADODB.Recordset")
strPath = ThisWorkbook.Path & "\员工管理.accdb"
strTable = "员工档案"
On Error GoTo ErrMsg
cnADO.Open "Provider=Microsoft.Ace.OLEDB.12.0;Data
Source=" & strPath '连接数据库

' 生成更新 SQL 语句(请注意 Office2007 后需要加 IMEX=0
参数)
strSQL = "SELECT B.员工编号 FROM " & strTable & "
A,[Excel 12.0;IMEX=0;Database=" & ThisWorkbook.FullName &
";].[ " & ActiveSheet.Name & "$" _
& Range("A1").CurrentRegion.Address(0, 0) & "] B
WHERE A.员工编号=B.员工编号"
rsADO.Open strSQL, cnADO, 1, 3
If rsADO.RecordCount > 0 Then
    avntFields = Range("A1:K1") '工作表中的字段名写入
数组
    ' 生成更新字符串, 如: a.姓名=b.姓名,a.性别=b.性
别,.....
    For i = 2 To UBound(avntFields, 2)
        strTemp = strTemp & ",a." & avntFields(1, i) &
        "=b." & avntFields(1, i)
    Next i
    strSQL = "UPDATE " & strTable & " A,[Excel
12.0;IMEX=0;Database=" & ThisWorkbook.FullName & ";].[ " &
ActiveSheet.Name & "$" _
& Range("A1").CurrentRegion.Address(0, 0) & "]
B SET " & Mid(strTemp, 2) & " WHERE A.员工编号=B.员工编号"
    cnADO.Execute strSQL
    strMsg = rsADO.RecordCount & "条记录已更新！"
End If
If rsADO.RecordCount < Range("A" &
Rows.Count).End(xlUp).Row - 1 Then
    ' 生成数据库不存在记录的 SQL 语句
    strSQL = " SELECT A.* FROM [Excel 12.0;Database=" &
ThisWorkbook.FullName & ";].[ " & ActiveSheet.Name & "$" &
Range("A1").CurrentRegion.Address(0, 0) _
& "] A LEFT JOIN " & strTable & " B ON A.员工编
号=B.员工编号 WHERE B.员工编号 IS NULL"
    Set rsADO = CreateObject("ADODB.Recordset")
    rsADO.Open strSQL, cnADO, 1, 3

    '插入数据库不存在记录
    If rsADO.RecordCount > 0 Then '如果工作表中含有
数据库不存在记录
        strSQL = "INSERT INTO " & strTable & strSQL
        cnADO.Execute strSQL
        strMsg = strMsg & vbCrLf & rsADO.RecordCount
& "条记录已添加到数据库！"
    End If
End If
MsgBox strMsg, vbInformation, "提示"
```

```
' 关闭连接释放内存
rsADO.Close
cnADO.Close
Set rsADO = Nothing
Set cnADO = Nothing
Exit Sub

ErrMsg:
    MsgBox Err.Description, , "错误报告"
End Sub

16.13 批量修改数据表中的记录
Option Explicit
Sub UpdateRecords()
Dim cnADO As Object,strPath As String,strTable As String
Dim strField As String,strSQL As String,avntField As Variant
Dim i As Integer
    Set cnADO = CreateObject("ADODB.Connection")
    strPath = ThisWorkbook.Path & "\员工管理.accdb"
    strTable = "员工档案"
    On Error GoTo ErrMsg
    cnADO.Open "Provider=Microsoft.Ace.OLEDB.12.0;Data
Source=" & strPath
    avntField = Range("A1:K1").Value
    For i = 2 To UBound(avntField, 2)
        strField = strField & ",A." & avntField(1, i) & "=B." &
avntField(1, i)
    Next i
    strSQL = "UPDATE " & strTable & " A,[Excel
12.0;Imex=0;Database=" & _
& ThisWorkbook.FullName & ";].[ " &
ActiveSheet.Name & "$" _
& Range("A1").CurrentRegion.Address(0, 0) & "] B " _
& "SET " & Mid(strField, 2) & " WHERE A.员工编号=B.
员工编号"
    cnADO.Execute strSQL
    MsgBox "数据修改成功。", vbInformation, "数据修改"
    cnADO.Close
    Set cnADO = Nothing
    Exit Sub

ErrMsg:
    MsgBox Err.Description, , "错误报告"
End Sub

Sub UpdateRecords_1()
Dim cnADO As New ADODB.Connection,strPath As String
Dim strTable As String,strField As String,strSQL As String
Dim avntField As Variant, i As Integer
    strPath = ThisWorkbook.Path & "\员工管理.accdb"
    strTable = "员工档案"
    On Error GoTo ErrMsg
    cnADO.Open "Provider=Microsoft.Ace.OLEDB.12.0;Data
Source=" & strPath
    strSQL = "UPDATE " & strTable & " SET 学历='大专'
WHERE 学历='高中' OR 学历='中专' OR 学历='初中'"
    cnADO.Execute strSQL
    MsgBox "数据修改成功。", vbInformation, "数据修改"
    cnADO.Close
    Set cnADO = Nothing
    Exit Sub

ErrMsg:
    MsgBox Err.Description, , "错误报告"
End Sub
```

第 16 章 数据库应用

16.14 从 Excel 工作表向数据表添加新记录、更新旧记录

```
Sub UpdateaddRecords()  
Dim cnADO As Object,rsADO As Object,strPath As String  
Dim strTable As String,strField As String,strSQL As String  
Dim strMsg As String, avntField As Variant, i As Integer  
Set cnADO = CreateObject("ADODB.Connection")  
Set rsADO = CreateObject("ADODB.Recordset")  
strPath = ThisWorkbook.Path & "\员工管理.accdb"  
strTable = "员工档案"  
On Error GoTo ErrMsg  
cnADO.Open "Provider=Microsoft.Ace.OLEDB.12.0;Data  
Source=" & strPath  
strSQL = "SELECT B.员工编号 FROM " & strTable & " A," _  
& "[Excel 12.0;Imex=0;Database=" & _  
ThisWorkbook.FullName & ";].[" & ActiveSheet.Name  
& "$" _  
& Range("A1").CurrentRegion.Address(0, 0) & "]" B "  
& "WHERE A.员工编号=B.员工编号"  
rsADO.Open strSQL, cnADO, 1, 3  
If rsADO.RecordCount > 0 Then  
avntField = Range("A1:K1")  
For i = 2 To UBound(avntField, 2)  
strField = strField & ",a." & avntField(1, i) & "=b."  
& avntField(1, i)  
Next  
strSQL = "UPDATE " & strTable & " A," _  
& "[Excel 12.0;Imex=0;Database=" & _  
ThisWorkbook.FullName & ";].[" &  
ActiveSheet.Name & "$" _  
& Range("A1").CurrentRegion.Address(0, 0) & "]"  
B " _  
& "SET " & Mid(strField, 2) & " WHERE A.员工编  
号=B.员工编号"  
cnADO.Execute strSQL  
strMsg = rsADO.RecordCount & "条记录已更新！"  
End If  
If Range("A1").CurrentRegion.Rows.Count - 1 >  
rsADO.RecordCount Then  
strSQL = " SELECT A.* FROM [Excel 12.0;Database=" &  
_  
ThisWorkbook.FullName & ";].[" &  
ActiveSheet.Name & "$" _  
& Range("A1").CurrentRegion.Address(0, 0) & "]"  
A " _  
& "LEFT JOIN " & strTable & " B " _  
& "ON A.员工编号=B.员工编号 WHERE B.员工  
编号 IS NULL"  
Set rsADO = CreateObject("ADODB.Recordset")  
rsADO.Open strSQL, cnADO, 1, 3  
If rsADO.RecordCount > 0 Then  
strSQL = "INSERT INTO " & strTable & strSQL  
cnADO.Execute strSQL  
strMsg = strMsg & vbCrLf & rsADO.RecordCount  
& "条记录已添加到数据库！"  
End If  
End If  
MsgBox strMsg, vbInformation, "提示"  
rsADO.Close  
cnADO.Close  
Set rsADO = Nothing
```

```
Set cnADO = Nothing  
Exit Sub  
ErrMsg:  
MsgBox Err.Description, , "错误报告"  
End Sub  
16.15 在数据库中存储照片  
Sub SavePic()  
Dim abyPic() As Byte,strPicPath As String,strPicName As String  
Dim strFldPath As String,strSQL As String,intFn As Integer  
Dim cnADO As Object, rsADO As Object,strPath As String  
Dim strTable As String, lngPicSum As Long  
Set cnADO = CreateObject("ADODB.Connection")  
Set rsADO = CreateObject("ADODB.Recordset")  
strPath = ThisWorkbook.Path & "\员工管理.accdb"  
strTable = "员工档案"  
With Application.FileDialog(msoFileDialogFolderPicker)  
.AllowMultiSelect = False  
.Title = "选择照片所在文件夹"  
.InitialFileName = ThisWorkbook.Path & "\"  
.AllowMultiSelect = False  
.Show  
If .SelectedItems.Count = 0 Then Exit Sub  
strFldPath = .SelectedItems(1) & "\"  
End With  
On Error GoTo ErrMsg  
cnADO.Open "Provider=Microsoft.ACE.OLEDB.12.0;Data  
Source=" & strPath  
strSQL = "SELECT 员工编号,照片 FROM " & strTable _  
& " WHERE ISNULL(照片)= TRUE"  
rsADO.Open strSQL, cnADO, 1, 3  
Do Until rsADO.EOF  
strPicName = rsADO(0)  
strPicPath = Dir(strFldPath & strPicName & ".*")  
If Len(strPicPath) <> 0 Then  
strPicPath = strFldPath & strPicPath  
intFn = FreeFile  
Open strPicPath For Binary As #intFn  
ReDim abyPic(LOF(intFn) - 1)  
Get #intFn, , abyPic  
Close #intFn  
rsADO("照片") = abyPic  
rsADO.Update  
lngPicSum = lngPicSum + 1  
End If  
rsADO.MoveNext  
Loop  
MsgBox "共有 " & lngPicSum & " 张照片存入数据库" &  
vbCr _  
& "还有 " & rsADO.RecordCount - lngPicSum _  
& " 人未提供照片", , "保存照片"  
rsADO.Close  
cnADO.Close  
Set rsADO = Nothing  
Set cnADO = Nothing  
Exit Sub  
ErrMsg:  
MsgBox Err.Description, , "错误报告"  
End Sub
```

第 16 章 数据库应用

16.16 制作带照片的档案表

Option Explicit

Type mudtGUID

 lngData As Long

 intData1 As Integer

 intData2 As Integer

 abytData(7) As Byte

End Type

Private Declare Function CreateStreamOnHGlobal Lib

"ole32.dll" (ByRef hGlobal As Any, ByVal fDeleteOnResume As Long, ByRef ppstr As Any) As Long

Private Declare Function OleLoadPicture Lib "olepro32.dll"

(ByVal lpStream As IUnknown, ByVal lSize As Long, ByVal fRunmode As Long, ByRef riid As mudtGUID, ByRef lpObj As Any) As Long

Private Declare Function CLSIDFromString Lib "ole32.dll" (ByVal lpsz As Long, ByRef pclsid As mudtGUID) As Long

Private Const SIPICTURE As String =

"{7BF80980-BF32-101A-8BBB-00AA00300CAB}"

Sub ReadRecordPic()

 Dim abytPic() As Byte

 Dim strSQL As String

 Dim cnADO As New ADODB.Connection

 Dim rsADO As New ADODB.Recordset

 Dim strPath As String

 Dim strTable As String

 strPath = ThisWorkbook.Path & "\员工管理.accdb"

 strTable = "员工档案"

 On Error GoTo ErrMsg

 With Sheets("员工档案查询")

 cnADO.Open

 "Provider=Microsoft.ACE.OLEDB.12.0;Data Source=" & strPath
 strSQL = "SELECT * FROM " & strTable & " WHERE 员工编号=" & Val(.Range("g2"))

 rsADO.Open strSQL, cnADO, 1, 3

 If rsADO.EOF Then

 MsgBox .Range("g2") & " 员工编号不存在，请重新输入", , "员工编号错误"

 Else

 .Range("b3") = rsADO("姓名")

 .Range("d3") = rsADO("出生日期")

 .Range("f3") = rsADO("民族")

 .Range("b4") = rsADO("性别")

 .Range("d4") = rsADO("职务")

 .Range("f4") = rsADO("籍贯")

 .Range("b5") = rsADO("学历")

 .Range("d5") = rsADO("部门")

 .Range("f5") = rsADO("电话")

 .Range("b6") = rsADO("简历")

 If IsNull(rsADO("照片")) Then

 .Image1.Visible = False

 .Range("g3") = "暂无照片"

 Else

 abytPic = rsADO("照片")

 .Image1.Visible = True

 .Image1.AutoSize = False

 .Image1.PictureSizeMode =

fmPictureSizeModeStretch

 Set .Image1.Picture =

ByteToPicture(abytPic)

 .Range("g3") = ""

 End If

End If

End With

Set rsADO = Nothing

Set cnADO = Nothing

Exit Sub

ErrMsg:

 MsgBox Err.Description, , "错误报告"

End Sub

Private Function ByteToPicture(ByRef abytData() As Byte) As IPicture

 On Error GoTo errorHandler

 Dim objStrm As IUnknown

 Dim avntGUID As mudtGUID

 If Not

CreateStreamOnHGlobal(abytData(LBound(abytData)), False, objStrm) Then

 CLSIDFromString StrPtr(SIPICTURE), avntGUID

 OleLoadPicture objStrm, UBound(abytData) -

LBound(abytData) + 1, False, avntGUID, ByteToPicture

 End If

 Set objStrm = Nothing

Exit Function

errorHandler:

 Debug.Print "Could not convert to IPicture!"

End Function

16.17 查询不重复的记录

Sub DistinctRecord()

 Dim cnADO As Object

 Dim strSQL As String

 Set cnADO = CreateObject("ADODB.Connection")

 cnADO.Open "Provider=Microsoft.ACE.OLEDB.12.0;" _

 & "Extended Properties=Excel 12.0;" _

 & "Data Source=" & ThisWorkbook.FullName

 strSQL = "SELECT DISTINCT 籍贯 FROM [员工档案\$]"

 Cells.ClearContents

 Range("A1") = "籍贯"

 Range("A2").CopyFromRecordset cnADO.Execute(strSQL)

 cnADO.Close

 Set cnADO = Nothing

End Sub

Sub DistinctRecordFields()

 Dim cnADO As Object

 Dim strSQL As String

 Set cnADO = CreateObject("ADODB.Connection")

 cnADO.Open "Provider=Microsoft.ACE.OLEDB.12.0;" _

 & "Extended Properties=Excel 12.0;" _

 & "Data Source=" & ThisWorkbook.FullName

 strSQL = "SELECT DISTINCT 部门,籍贯 FROM [员工档案\$]"

End Sub

 Cells.ClearContents

 Range("A1:B1") = Array("部门", "籍贯")

 Range("A2").CopyFromRecordset cnADO.Execute(strSQL)

 cnADO.Close

 Set cnADO = Nothing

End Sub

第 16 章 数据库应用

16.18 查询前 n 条最大值记录

Sub TopRecords() '总分前十名

```
Dim cnADO As Object
Dim rsADO As Object
Dim strSQL As String
Dim i As Long
Set cnADO = CreateObject("ADODB.Connection")
Set rsADO = CreateObject("ADODB.Recordset")
cnADO.Open "Provider=Microsoft.ACE.OLEDB.12.0;" _
    & "Extended Properties=Excel 12.0;" _
    & "Data Source=" & ThisWorkbook.FullName
strSQL = "SELECT TOP 10 学号,姓名,班级,总分 FROM [成绩$] " _
    & "ORDER BY 总分 DESC"
Set rsADO = cnADO.Execute(strSQL)
Cells.ClearContents
For i = 0 To rsADO.Fields.Count - 1
    Cells(1, i + 1) = rsADO.Fields(i).Name
Next i
Range("A2").CopyFromRecordset rsADO
rsADO.Close
cnADO.Close
Set cnADO = Nothing
Set rsADO = Nothing
End Sub
```

Sub TopRecords2() '总分前十名并将班级字段降序排列

```
Dim cnADO As Object
Dim rsADO As Object
Dim strSQL As String
Dim i As Long
Set cnADO = CreateObject("ADODB.Connection")
Set rsADO = CreateObject("ADODB.Recordset")
cnADO.Open
"Provider=Microsoft.ACE.OLEDB.12.0;Extended
Properties=Excel 12.0;Data Source=" &
ThisWorkbook.FullName
strSQL = "SELECT TOP 10 学号,姓名,班级,总分 FROM [成绩$] ORDER BY 总分 DESC,班级 DESC"
Set rsADO = cnADO.Execute(strSQL)
Cells.ClearContents
For i = 0 To rsADO.Fields.Count - 1
    Cells(1, i + 1) = rsADO.Fields(i).Name
Next i
Range("A2").CopyFromRecordset rsADO
rsADO.Close
cnADO.Close
Set cnADO = Nothing
Set rsADO = Nothing
End Sub
```

16.19 分组聚合查询

Sub AggregationData() '聚合查询

```
Dim cnADO As Object
Dim strSQL As String
Set cnADO = CreateObject("ADODB.Connection")
cnADO.Open "Provider=Microsoft.ACE.OLEDB.12.0;" _
    & "Extended Properties=Excel 12.0;" _
    & "Data Source=" & ThisWorkbook.FullName
strSQL = "SELECT SUM(总分),COUNT(姓名) FROM [成绩$] " _
```

```
& "WHERE 班级='1 班'"
Cells.ClearContents
Range("A1:B1") = Array("1 班总分", "人数")
Range("A2").CopyFromRecordset cnADO.Execute(strSQL)
cnADO.Close
Set cnADO = Nothing
```

End Sub

Sub AggregationGroupData() '分组聚合 1

```
Dim cnADO As Object
Dim strSQL As String
Dim rsADO As Object
Dim i As Long
Set cnADO = CreateObject("ADODB.Connection")
Set rsADO = CreateObject("ADODB.Connection")
cnADO.Open "Provider=Microsoft.ACE.OLEDB.12.0;" _
    & "Extended Properties=Excel 12.0;" _
    & "Data Source=" & ThisWorkbook.FullName
strSQL = "SELECT 班级,SUM(总分) AS 总分 ,COUNT(姓名) AS 人数 " _
    & "FROM [成绩$] GROUP BY 班级"
Set rsADO = cnADO.Execute(strSQL)
Cells.ClearContents
For i = 0 To rsADO.Fields.Count - 1
    Cells(1, i + 1) = rsADO.Fields(i).Name
Next i
Range("A2").CopyFromRecordset rsADO
rsADO.Close
cnADO.Close
Set cnADO = Nothing
Set rsADO = Nothing
End Sub
```

Sub AggregationData3() '分组聚合 2

```
Dim cnADO As Object
Dim strSQL As String
Dim rsADO As Object
Dim i As Long
Set cnADO = CreateObject("ADODB.Connection")
Set rsADO = CreateObject("ADODB.Connection")
cnADO.Open "Provider=Microsoft.ACE.OLEDB.12.0;" _
    & "Extended Properties=Excel 12.0;" _
    & "Data Source=" & ThisWorkbook.FullName
strSQL = "SELECT 班级,SUM(总分) AS 总分 ,COUNT(姓名) AS 人数 " _
    & "FROM [成绩$] GROUP BY 班级 HAVING COUNT(姓名)>20"
Set rsADO = cnADO.Execute(strSQL)
Cells.ClearContents
For i = 0 To rsADO.Fields.Count - 1
    Cells(1, i + 1) = rsADO.Fields(i).Name
Next i
Range("A2").CopyFromRecordset rsADO
rsADO.Close
cnADO.Close
Set cnADO = Nothing
Set rsADO = Nothing
End Sub
```


第 16 章 数据库应用

16.20 从字段不完全相同的多个工作簿提取数据

```
Sub MultipleWorkbook()  
    Dim cnADO As Object  
    Dim rsADO As Object  
    Dim strSQL1 As String  
    Dim strSQL2 As String  
    Dim strSQL3 As String  
    Dim strSQL As String  
    Dim strPath As String  
    Dim i As Integer  
    On Error GoTo ErrMsg  
    Set cnADO = CreateObject("ADODB.Connection")  
    Set rsADO = CreateObject("ADODB.Recordset")  
    strPath = ThisWorkbook.Path & "\工资表\  
cnADO.Open "Provider=Microsoft.Ace.OLEDB.12.0;" _  
        & "Extended Properties=Excel 12.0;" _  
        & "Data Source=" & strPath & "办公室.xlsx"  
    strSQL1 = "SELECT '办公室' AS 部门,姓名,性别," _  
        & "基础工资,岗位工资,午餐费,0 AS 野外补贴  
FROM [在册员工$]"  
    strSQL2 = "SELECT '办公室' AS 部门,姓名,性别," _  
        & "0 AS 基础工资,岗位工资,午餐费,0 AS 野外补贴  
FROM [临聘人员$]"  
    strSQL3 = "SELECT '工程部' AS 部门,姓名,性别," _  
        & "基础工资,岗位工资,午餐费,野外补贴 " _  
        & "FROM [Excel 12.0;Database=" _  
        & strPath & "\工程部.xlsx;].[工程人员$]"  
    strSQL = strSQL1 & " UNION ALL " & strSQL2 _  
        & " UNION ALL " & strSQL3  
    strSQL = strSQL & " ORDER BY 部门"  
    Set rsADO = cnADO.Execute(strSQL)  
    Cells.ClearContents  
    For i = 0 To rsADO.Fields.Count - 1  
        Cells(1, i + 1) = rsADO.Fields(i).Name  
    Next i  
    Range("A2").CopyFromRecordset rsADO  
    rsADO.Close  
    cnADO.Close  
    Set rsADO = Nothing  
    Set cnADO = Nothing  
    Exit Sub  
ErrMsg:  
    MsgBox Err.Description, , "错误报告"  
End Sub
```

```
Sub MultipleWorkbook_1()  
    Dim cnADO As Object  
    Dim rsADO As Object  
    Dim rsADO2 As Object  
    Dim strSQL As String  
    Dim i As Integer  
    Dim j As Integer  
    Dim intCount As Integer  
    Dim astrPath() As String  
    Dim avntFields As Variant  
    Dim strTemp As String  
    Dim strTmp As String  
    Dim strPath As String  
    Dim strFields As String  
    Dim strFileName As String
```

```
Dim strShtName As String  
On Error GoTo ErrMsg  
strFields = "部门"  
strPath = ThisWorkbook.Path & "\工资表\  
strFileName = Dir(strPath & "*.xlsx")  
Do While strFileName <> ""  
    Set cnADO = CreateObject("ADODB.Connection")  
    cnADO.Open  
    "Provider=Microsoft.Ace.OLEDB.12.0;Extended  
Properties=Excel 12.0;Data Source=" & strPath & strFileName  
    Set rsADO = cnADO.OpenSchema(20)  
    Do Until rsADO.EOF  
        If rsADO.Fields("TABLE_TYPE") = "TABLE" Then  
            strShtName =  
Replace(rsADO("TABLE_NAME").Value, "", "")  
            If Right(strShtName, 1) = "$" Then  
                Set rsADO2 = cnADO.Execute("[ " &  
strShtName & " ]")  
                strTemp = ""  
                For i = 0 To rsADO2.Fields.Count - 1  
                    strTmp = rsADO2.Fields(i).Name  
                    If InStr(", " & strFields & ", ", ", " &  
strTmp & ", ") = 0 Then  
                        strFields = strFields & ", " &  
strTmp  
                    End If  
                    strTemp = strTemp & ", " & strTmp  
                Next i  
                intCount = intCount + 1  
                ReDim Preserve astrPath(1 To 2, 1 To  
intCount)  
                astrPath(1, intCount) = strFileName &  
";].[ " & strShtName & " ]"  
                astrPath(2, intCount) = strTemp & ", "  
                End If  
            End If  
            rsADO.MoveNext  
        Loop  
        strFileName = Dir  
    Loop  
    avntFields = Split(strFields, ",")  
    For i = 1 To intCount  
        strTemp = astrPath(2, i)  
        strTmp = ""  
        For j = 1 To UBound(avntFields)  
            If InStr(strTemp, ", " & avntFields(j) & ", ") Then  
                strTmp = strTmp & ", " & avntFields(j)  
            Else  
                strTmp = strTmp & ",0 as " & avntFields(j)  
            End If  
        Next j  
        strSQL = strSQL & " UNION ALL SELECT " _  
            & Split(astrPath(1, i), ".xlsx")(0) & " AS 部门" &  
strTmp _  
            & " FROM [Excel 12.0;Database=" & strPath &  
astrPath(1, i)  
        Next i  
        strSQL = Mid(strSQL, 12) & " ORDER BY 部门"  
        Cells.ClearContents  
        avntFields = Split(strFields, ",")  
        Range("a1").Resize(, UBound(avntFields) + 1) = avntFields  
        Range("a2").CopyFromRecordset cnADO.Execute(strSQL)  
        rsADO.Close
```

第 16 章 数据库应用

```
rsADO2.Close
cnADO.Close
Set rsADO = Nothing
Set rsADO2 = Nothing
Set cnADO = Nothing
Exit Sub

ErrMsg:
MsgBox Err.Description, , "错误报告"
End Sub

Sub MultipleWorkbook_2()
Dim objDIC As Object
Dim objFso As Object
Dim objFile As Object
Dim cnADO As Object
Dim rsADO As Object
Dim rsADO2 As Object
Dim strSQL As String
Dim i As Integer
Dim j As Integer
Dim intCount As Integer
Dim astrPath() As String
Dim avntKeys As Variant
Dim strTemp As String
Dim strTmp As String
Dim strPath As String
Dim strShtName As String
On Error GoTo ErrMsg
Set objFso = CreateObject("Scripting.FileSystemObject")
Set objDIC = CreateObject("scripting.dictionary")
strPath = ThisWorkbook.Path & "\工资表\"
For Each objFile In objFso.GetFolder(strPath).Files
If objFile.Name Like "*.xlsx" Then
Set cnADO =
CreateObject("ADODB.Connection")
cnADO.Open
"Provider=Microsoft.Ace.OLEDB.12.0;Extended
Properties=Excel 12.0;Data Source=" & objFile
Set rsADO = cnADO.OpenSchema(20)
Do Until rsADO.EOF
If rsADO.Fields("TABLE_TYPE") = "TABLE"
Then
strShtName =
Replace(rsADO("TABLE_NAME").Value, "", "")
If Right(strShtName, 1) = "$" Then
Set rsADO2 = cnADO.Execute("[&strShtName & "]")

strTemp = ""
For i = 0 To rsADO2.Fields.Count - 1
strTmp =
rsADO2.Fields(i).Name
If Not objDIC.Exists(strTmp)
strTemp = strTemp & "," &
strTmp
Next i
intCount = intCount + 1
ReDim Preserve astrPath(1 To 2, 1
To intCount)
astrPath(1, intCount) =
objFile.Name & ";[" & strShtName & "]"
```

```
astrPath(2, intCount) = strTemp &
","
End If
End If
rsADO.MoveNext
Loop
End If
Next
avntKeys = objDIC.Keys
For i = 1 To intCount
strTemp = astrPath(2, i)
strTmp = ""
For j = 0 To objDIC.Count - 1
If InStr(strTemp, "," & avntKeys(j) & ",") Then
strTmp = strTmp & "," & avntKeys(j)
Else
strTmp = strTmp & ",0 as " & avntKeys(j)
End If
Next j
strSQL = strSQL & " UNION ALL SELECT " _
& Split(astrPath(1, i), ".xlsx")(0) & " AS 部门" &
strTmp _
& " FROM [Excel 12.0;Database=" & strPath &
astrPath(1, i)
Next i
strSQL = Mid(strSQL, 12) & " ORDER BY 部门"
Cells.ClearContents
Range("a1") = "部门"
Range("b1").Resize(, objDIC.Count) = objDIC.Keys
Range("a2").CopyFromRecordset cnADO.Execute(strSQL)
Set objFile = Nothing
Set objFso = Nothing
rsADO.Close
rsADO2.Close
cnADO.Close
Set rsADO = Nothing
Set rsADO2 = Nothing
Set cnADO = Nothing
Exit Sub

ErrMsg:
MsgBox Err.Description, , "错误报告"
End Sub

16.21 使用内、外连接实现字段配对
Option Explicit
Sub ADOLeftJoin() '左连接
Dim objCN As Object
Dim objRS As Object
Dim strPath As String
Dim strSQL As String
Dim i As Long
On Error GoTo ErrMsg
Set objCN = CreateObject("ADODB.Connection")
Set objRS = CreateObject("ADODB.Recordset")
strPath = ThisWorkbook.Path & "\员工管理.accdb"
objCN.Open "Provider=Microsoft.ACE.OLEDB.12.0;Data
Source=" & strPath
strSQL = "SELECT a.员工编号, a.姓名 AS 姓名 A, a.性别, " _
& "b.姓名 AS 姓名 B, b.奖金额 From " _
& "员工档案 AS a LEFT JOIN [MS
Access;Pwd=;Database=" _
& ThisWorkbook.Path & "\奖金.accdb;].奖金名
```

第 16 章 数据库应用

```
单 AS b " _
    & "ON a.员工编号 = b.员工编号"
objRS.Open strSQL, objCN, 1, 3
For i = 1 To objRS.Fields.Count
    Cells(1, i) = objRS.Fields(i - 1).Name
Next
Range("a2").CopyFromRecordset objRS
objRS.Close
objCN.Close
Set objRS = Nothing
Set objCN = Nothing
Exit Sub

ErrMsg:
    MsgBox Err.Description, , "错误报告"
End Sub

Sub ADORightJoin() '右连接
    Dim objCN As Object
    Dim objRS As Object
    Dim strPath As String
    Dim strSQL As String
    Dim i As Long
    On Error GoTo ErrMsg
    Set objCN = CreateObject("ADODB.Connection")
    Set objRS = CreateObject("ADODB.Recordset")
    strPath = ThisWorkbook.Path & "\员工管理.accdb"
    objCN.Open "Provider=Microsoft.ACE.OLEDB.12.0;Data
Source=" & strPath
    strSQL = "SELECT a.员工编号, a.姓名 AS 姓名 A, a.性别,
b.姓名 AS 姓名 B, b.奖金额 FROM " _
    & "员工档案 AS a RIGHT JOIN [MS
Access;Pwd=;Database=" & ThisWorkbook.Path _
    & "\奖金.accdb;].奖金名单 AS b ON a.员工编号 = b.
员工编号"
    objRS.Open strSQL, objCN, 1, 3
    For i = 1 To objRS.Fields.Count
        Cells(1, i + 6) = objRS.Fields(i - 1).Name
    Next
    Range("g2").CopyFromRecordset objRS
    objRS.Close
    objCN.Close
    Set objRS = Nothing
    Set objCN = Nothing
    Exit Sub

ErrMsg:
    MsgBox Err.Description, , "错误报告"
End Sub

Sub ADOInnerJoin() '内连接
    Dim objCN As Object
    Dim objRS As Object
    Dim strPath As String
    Dim strSQL As String
    Dim i As Long
    On Error GoTo ErrMsg
    Set objCN = CreateObject("ADODB.Connection")
    Set objRS = CreateObject("ADODB.Recordset")
    strPath = ThisWorkbook.Path & "\员工管理.accdb"
    objCN.Open "Provider=Microsoft.ACE.OLEDB.12.0;Data
Source=" & strPath
```

```
    strSQL = "SELECT a.员工编号, a.姓名 AS 姓名 A, a.性别,
b.姓名 AS 姓名 B, b.奖金额 FROM " _
    & "员工档案 AS a INNER JOIN [MS
Access;Pwd=;Database=" & ThisWorkbook.Path _
    & "\奖金.accdb;].奖金名单 AS b ON a.员工编号 = b.
员工编号"
    objRS.Open strSQL, objCN, 1, 3
    For i = 1 To objRS.Fields.Count
        Cells(1, i + 12) = objRS.Fields(i - 1).Name
    Next
    Range("M2").CopyFromRecordset objRS
    objRS.Close
    objCN.Close
    Set objRS = Nothing
    Set objCN = Nothing
    Exit Sub

ErrMsg:
    MsgBox Err.Description, , "错误报告"
End Sub

16.22 比较两表提取相同项和不同项
Option Explicit
Sub SameTitemsdifferenTitemsd()
    Dim cnADO As Object
    Dim rsADO As Object
    Dim strSQL As String
    Dim i As Integer
    Dim strPath As String
    On Error GoTo ErrMsg
    Set cnADO = CreateObject("ADODB.Connection")
    strPath = ThisWorkbook.FullName
    cnADO.Open "Provider=Microsoft.Ace.OLEDB.12.0;" _
    & "Extended Properties=Excel 12.0;" _
    & "Data Source=" & strPath
    strSQL = "SELECT A.* FROM [Sheet1$] A, [Sheet2$] B " _
    & "WHERE A.员工编号=B.员工编号"
    Set rsADO = cnADO.Execute(strSQL)
    Cells.ClearContents
    Range("A1") = "两表相同项"
    For i = 0 To rsADO.Fields.Count - 1
        Cells(2, i + 1) = rsADO.Fields(i).Name
    Next i
    Range("A3").CopyFromRecordset rsADO
    strSQL = "SELECT A.* FROM [Sheet1$] A LEFT JOIN
[Sheet2$] B " _
    & "ON A.员工编号=B.员工编号 WHERE B.员工
编号 IS NULL"
    Range("G1") = "Sheet1 独有项"
    Set rsADO = cnADO.Execute(strSQL)
    For i = 0 To rsADO.Fields.Count - 1
        Cells(2, i + 7) = rsADO.Fields(i).Name
    Next i
    Range("G3").CopyFromRecordset rsADO
    strSQL = "SELECT A.* FROM [Sheet2$] A LEFT JOIN
[Sheet1$] B " _
    & "ON A.员工编号=B.员工编号 WHERE B.员工
编号 IS NULL"
    Range("M1") = "Sheet2 独有项"
    Set rsADO = cnADO.Execute(strSQL)
    For i = 0 To rsADO.Fields.Count - 1
        Cells(2, i + 13) = rsADO.Fields(i).Name
    Next i
    Range("M3").CopyFromRecordset rsADO
```

第 16 章 数据库应用

```
rsADO.Close
cnADO.Close
Set rsADO = Nothing
Set cnADO = Nothing
Exit Sub

ErrMsg:
    MsgBox Err.Description, , "错误报告"
End Sub

Sub SameTitemsDifferenTitems_1() 'IN,NOT IN
    Dim cnADO As Object
    Dim rsADO As Object
    Dim strSQL As String
    Dim i As Integer
    Dim strPath As String
    On Error GoTo ErrMsg
    Set cnADO = CreateObject("ADODB.Connection")
    strPath = ThisWorkbook.FullName
    cnADO.Open
    "Provider=Microsoft.Ace.OLEDB.12.0;Extended
    Properties=Excel 12.0;Data Source=" & strPath
    strSQL = "SELECT * FROM [Sheet1$] WHERE 员工编号 IN
(SELECT 员工编号 FROM [Sheet2$])"
    Set rsADO = cnADO.Execute(strSQL)
    Cells.ClearContents
    Range("A1") = "两表相同项"
    For i = 0 To rsADO.Fields.Count - 1
        Cells(2, i + 1) = rsADO.Fields(i).Name
    Next i
    Range("A3").CopyFromRecordset rsADO
    strSQL = "SELECT * FROM [Sheet1$] WHERE 员工编号
NOT IN (SELECT 员工编号 FROM [Sheet2$])"
    Range("G1") = "Sheet1 独有项"
    Set rsADO = cnADO.Execute(strSQL)
    For i = 0 To rsADO.Fields.Count - 1
        Cells(2, i + 7) = rsADO.Fields(i).Name
    Next i
    Range("G3").CopyFromRecordset rsADO
    strSQL = "SELECT * FROM [Sheet2$] WHERE 员工编号
NOT IN (SELECT 员工编号 FROM [Sheet1$])"
    Range("M1") = "Sheet2 独有项"
    Set rsADO = cnADO.Execute(strSQL)
    For i = 0 To rsADO.Fields.Count - 1
        Cells(2, i + 13) = rsADO.Fields(i).Name
    Next i
    Range("M3").CopyFromRecordset rsADO
    rsADO.Close
    cnADO.Close
    Set rsADO = Nothing
    Set cnADO = Nothing
    Exit Sub

ErrMsg:
    MsgBox Err.Description, , "错误报告"
End Sub

Sub SameTitemsDifferenTitems_2() 'EXISTS,NOT EXISTS
    Dim cnADO As Object
    Dim rsADO As Object
    Dim strSQL As String
    Dim i As Integer
```

```
Dim strPath As String
On Error GoTo ErrMsg
Set cnADO = CreateObject("ADODB.Connection")
strPath = ThisWorkbook.FullName
cnADO.Open
"Provider=Microsoft.Ace.OLEDB.12.0;Extended
Properties=Excel 12.0;Data Source=" & strPath
    strSQL = "SELECT * FROM [Sheet1$] A WHERE
EXISTS(SELECT * FROM [Sheet2$] WHERE 员工编号=A.员工编
号)"
    Set rsADO = cnADO.Execute(strSQL)
    Cells.ClearContents
    Range("A1") = "两表相同项"
    For i = 0 To rsADO.Fields.Count - 1
        Cells(2, i + 1) = rsADO.Fields(i).Name
    Next i
    Range("A3").CopyFromRecordset rsADO
    strSQL = "SELECT * FROM [Sheet1$] A WHERE NOT
EXISTS(SELECT * FROM [Sheet2$] WHERE 员工编号=A.员工编
号)"
    Range("G1") = "Sheet1 独有项"
    Set rsADO = cnADO.Execute(strSQL)
    For i = 0 To rsADO.Fields.Count - 1
        Cells(2, i + 7) = rsADO.Fields(i).Name
    Next i
    Range("G3").CopyFromRecordset rsADO
    strSQL = "SELECT * FROM [Sheet2$] A WHERE NOT
EXISTS(SELECT * FROM [Sheet1$] WHERE 员工编号=A.员工编
号)"
    Range("M1") = "Sheet2 独有项"
    Set rsADO = cnADO.Execute(strSQL)
    For i = 0 To rsADO.Fields.Count - 1
        Cells(2, i + 13) = rsADO.Fields(i).Name
    Next
    Range("M3").CopyFromRecordset rsADO
    rsADO.Close
    cnADO.Close
    Set rsADO = Nothing
    Set cnADO = Nothing
    Exit Sub

ErrMsg:
    MsgBox Err.Description, , "错误报告"
End Sub

16.23 使用 SQL 查询创建数据透视表
Option Explicit
Sub CreatePivotTable()
    Dim cnADO As Object
    Dim strPath As String
    Dim strTable As String
    Dim strSQL As String
    Dim objPivotCache As PivotCache
    Dim i As Long
    Set cnADO = CreateObject("ADODB.Connection")
    strPath = ThisWorkbook.Path & "\员工管理.accdb"
    strTable = "员工档案"
    On Error GoTo ErrMsg
    Application.ScreenUpdating = False
    cnADO.Open "Provider=Microsoft.ACE.OLEDB.12.0;Data
Source=" & strPath
    strSQL = "SELECT 部门,职务,'年龄段' AS 大类," _
        & "Partition(DATEDIFF('YYYY',出生日
期,DATE()),20,59,10) as 细分,姓名 " _
```

第 16 章 数据库应用

```
& "FROM " & strTable _
& " Union ALL SELECT 部门,职务,'学历',学历,姓名
FROM " & strTable
Cells.Clear
Set objPivotCache =
ActiveWorkbook.PivotCaches.Create(SourceType:=xlExternal)
Set objPivotCache.Recordset = cnADO.Execute(strSQL)
objPivotCache.CreatePivotTable
TableDestination:=Range("A3"), TableName:="数据透视表 1"
With ActiveSheet.PivotTables("数据透视表 1")
.PivotFields("部门").Orientation = xlRowField
.PivotFields("部门").Position = 1
.PivotFields("职务").Orientation = xlRowField
.PivotFields("职务").Position = 2
.PivotFields("大类").Orientation = xlColumnField
.PivotFields("大类").Position = 1
.PivotFields("细分").Orientation = xlColumnField
.PivotFields("细分").Position = 2
.AddDataField ActiveSheet.PivotTables("数据透视表
1").PivotFields("姓名"), "员工人数", xlCount
.RowAxisLayout xlTabularRow
.MergeLabels = True
End With
Range("A1").CurrentRegion.EntireColumn.AutoFit
Set objPivotCache = Nothing
cnADO.Close
Set cnADO = Nothing
Application.ScreenUpdating = True
Exit Sub
```

```
ErrMsg:
MsgBox Err.Description, , "错误报告"
```

End Sub

16.24 查询文本文件中的数据

Sub ADOForTxt() '标准 text 文件导入 Excel

```
Dim objCN As Object
Dim objRS As Object
Dim strSQL As String
Dim i As Integer
On Error GoTo ErrMsg
Set objCN = CreateObject("ADODB.Connection")
Set objRS = CreateObject("ADODB.Recordset")
objCN.Open "Provider=Microsoft.ACE.OLEDB.12.0;" _
& "Extended
Properties='text;HDR=Yes;FMT=Delimited';" _
& "Data Source = " & ThisWorkbook.Path
strSQL = "SELECT * FROM 部分员工资料.txt"
Columns("A:D").ClearContents
objRS.Open strSQL, objCN, 1, 3
For i = 0 To objRS.Fields.Count - 1
Cells(1, i + 1) = objRS.Fields(i).Name
Next i
Range("A2").CopyFromRecordset objRS
objRS.Close
objCN.Close
Set objRS = Nothing
```

Set objCN = Nothing

Exit Sub

ErrMsg:

MsgBox Err.Description, , "错误报告"

End Sub

Sub ADOForTxtWithFixed() '非标准 text 文件导入 Excel

```
Dim objCN As Object
Dim strSQL As String
On Error GoTo ErrMsg
Set objCN = CreateObject("ADODB.Connection")
objCN.Open "Provider=Microsoft.ACE.OLEDB.12.0;" _
& "Extended Properties='text;HDR=No;FMT=Fixed';" _
& "Data Source=" & ThisWorkbook.Path
strSQL = "SELECT * FROM 部分员工资料 B.txt"
Columns("A:D").ClearContents
Range("A1:D1") = Array("员工编号", "姓名", "性别", "出生
日期")
Range("A2").CopyFromRecordset objCN.Execute(strSQL)
objCN.Close
Set objCN = Nothing
Exit Sub
```

ErrMsg:

MsgBox Err.Description, , "错误报告"

End Sub

Sub ADOForTxtWithFixed_1() '非标准 text 文件导入 Excel2

```
Dim objCN As Object
Dim strSQL As String
On Error GoTo ErrMsg
Set objCN = CreateObject("ADODB.Connection")
Open ThisWorkbook.Path & "\schema.ini" For Output As
#1
Print #1, "[部分员工资料 B.txt]" & vbCrLf &
"COLNAMEHEADER = False" _
& vbCrLf & "Format = FixedLength" & vbCrLf _
& "COL1=""员工编号"" long Width 6" _
& vbCrLf & "COL2=""姓名"" Text Width 6" _
& vbCrLf & "COL3=""性别"" Text Width 2" _
& vbCrLf & "COL4=""出生日期"" DateTime Width 10"
Close #1
objCN.Open "Provider=Microsoft.ACE.OLEDB.12.0;" _
& "Extended Properties='text;HDR=No;FMT=Fixed';" _
& "Data Source=" & ThisWorkbook.Path
strSQL = "SELECT * FROM 部分员工资料 B.txt"
Columns("A:D").ClearContents
Range("A1:D1") = Array("员工编号", "姓名", "性别", "出生
日期")
Range("A2").CopyFromRecordset objCN.Execute(strSQL)
objCN.Close
Set objCN = Nothing
Kill ThisWorkbook.Path & "\schema.ini"
Exit Sub
```

ErrMsg:

MsgBox Err.Description, , "错误报告"

End Sub

第 16 章 数据库应用

16.25 多类型表内连接关联查询

Sub InternalLinksQuery() '交叉表查询

```
Dim cnADO As Object,rsADO As Object
Dim strPath As String,strTable As String
Dim strTempTable As String,strBonusTable As String
Dim strBonusTempTable As Strin,strDonationsTable As
```

String

```
Dim strSQL1 As String, strSQL2 As String,strSQL As String
```

```
Dim i As Long
On Error GoTo ErrMsg
Set cnADO = CreateObject("ADODB.Connection")
Set rsADO = CreateObject("ADODB.Recordset")
strPath = ThisWorkbook.Path & "\员工管理.accdb"
cnADO.Open "Provider=Microsoft.ACE.OLEDB.12.0;Data
```

Source=" & strPath

```
strTable = "员工档案"
```

```
strTempTable = "[Excel 12.0;Database=" &
```

ThisWorkbook.Path _

```
& "\临聘人员档案.xlsx;].[临聘人员档案$]"
```

```
strBonusTable = "[MS Access;pwd=;Database=" _
& ThisWorkbook.Path & "\奖金.accdb;].[奖金名单"
```

```
strBonusTempTable = "[Excel 12.0;Database=" _
& ThisWorkbook.Path _
```

```
& "\临聘人员奖金.xlsx;].[临聘人员奖金$]"
```

```
strDonationsTable = "[Text;HDR=YES;Database=" _
& ThisWorkbook.Path & "\;].[捐款.txt]"
```

```
strSQL1 = "SELECT 员工编号 AS 编号,姓名,性别," _
```

```
& "'员工' AS 人员类别 FROM " & strTable _
```

```
& " UNION ALL SELECT 编号,姓名,性别," _
```

```
& "'临聘人员' AS 人员类别 FROM " & strTempTable
```

& ""

```
strSQL2 = "SELECT 员工编号 AS 编号,姓名,奖金额,0 AS
捐款 " _
```

```
& "FROM " & strBonusTable _
```

```
& " UNION ALL SELECT 编号,姓名,奖金额,0 AS 捐款
```

FROM " _

```
& strBonusTempTable _
```

```
& " UNION ALL SELECT 编号,姓名,0 AS 奖金额,捐款
```

FROM " _

```
& strDonationsTable
```

```
strSQL = "SELECT A.编号,A.姓名,A.性别,A.人员类别," _
```

```
& "IIF(SUM(B.奖金额)=0,NULL,SUM(B.奖金额)) AS
```

奖金额," _

```
& "IIF(SUM(捐款)=0,NULL,SUM(捐款)) AS 捐款
```

FROM (" _

```
& strSQL1 & ") A Inner Join (" & strSQL2 & ") B " _
```

```
& "ON A.编号=B.编号 GROUP BY A.编号,A.姓名,A.
```

性别,A.人员类别"

```
rsADO.Open strSQL, cnADO, 1, 3
```

```
Cells.ClearContents
```

```
For i = 0 To rsADO.Fields.Count - 1
```

```
Cells(1, i + 1) = rsADO.Fields(i).Name
```

```
Next i
```

```
Range("A2").CopyFromRecordset rsADO
```

```
rsADO.Close
```

```
cnADO.Close
```

```
Set rsADO = Nothing
```

```
Set cnADO = Nothing
```

```
Exit Sub
```

ErrMsg:

```
MsgBox Err.Description, , "错误报告"
```

End Sub

16.26 TRANSFORM 交叉表查询

Sub ADOTransForm() '交叉表查询（指定字段）

```
Dim cnADO As Object,rsADO As Object,strSQL As String
```

```
Dim i As Integer
```

```
On Error GoTo ErrMsg
```

```
Set cnADO = CreateObject("ADODB.Connection")
```

```
Set rsADO = CreateObject("ADODB.Recordset")
```

```
cnADO.Open "Provider=Microsoft.ACE.OLEDB.12.0;" _
```

```
& "Extended Properties=Excel 12.0;" _
```

```
& "Data Source=" & ThisWorkbook.FullName
```

```
strSQL = "TRANSFORM SUM(销售额) " _
```

```
& "SELECT 姓名,SUM(销售额) AS 合计 " _
```

```
& "FROM [数据表$] " _
```

```
& "GROUP BY 姓名 " _
```

```
& "PIVOT FORMAT(日期,'mm 月')"
```

```
rsADO.Open strSQL, cnADO, 1, 3
```

```
Cells.ClearContents
```

```
For i = 0 To rsADO.Fields.Count - 1
```

```
Cells(1, i + 1) = rsADO.Fields(i).Name
```

```
Next i
```

```
Range("A2").CopyFromRecordset rsADO
```

```
rsADO.Close
```

```
cnADO.Close
```

```
Set rsADO = Nothing
```

```
Set cnADO = Nothing
```

```
Exit Sub
```

ErrMsg:

```
MsgBox Err.Description, , "错误报告"
```

End Sub

Sub ADOTransForm2()

```
Dim cnADO As Object,rsADO As Object,strSQL As String
```

```
Dim i As Integer
```

```
On Error GoTo ErrMsg
```

```
Set cnADO = CreateObject("ADODB.Connection")
```

```
Set rsADO = CreateObject("ADODB.Recordset")
```

```
cnADO.Open "Provider=Microsoft.ACE.OLEDB.12.0;" _
```

```
& "Extended Properties=Excel 12.0;" _
```

```
& "Data Source=" & ThisWorkbook.FullName
```

```
strSQL = "TRANSFORM SUM(销售额) " _
```

```
& "SELECT 姓名,SUM(销售额) AS 合计 " _
```

```
& "FROM [数据表$] " _
```

```
& "GROUP BY 姓名 " _
```

```
& "PIVOT FORMAT(日期,'mm 月') IN('04 月','05 月')"
```

```
rsADO.Open strSQL, cnADO, 1, 3
```

```
Cells.ClearContents
```

```
For i = 0 To rsADO.Fields.Count - 1
```

```
Cells(1, i + 1) = rsADO.Fields(i).Name
```

```
Next i
```

```
Range("A2").CopyFromRecordset rsADO
```

```
rsADO.Close
```

```
cnADO.Close
```

```
Set rsADO = Nothing
```

```
Set cnADO = Nothing
```

```
Exit Sub
```

ErrMsg:

```
MsgBox Err.Description, , "错误报告"
```

End Sub

第 16 章 数据库应用

16.27 在数组中存储查询结果

```
Sub RecordsetToArray()  
    Dim cnADO As Object, rsADO As Object  
    Dim strPath As String, strSQL As String  
    Dim avntData() As Variant, avntTitle() As Variant  
    On Error GoTo ErrMsg  
    Set cnADO = CreateObject("ADODB.Connection")  
    Set rsADO = CreateObject("ADODB.Recordset")  
    strPath = ThisWorkbook.Path & "\员工管理.accdb"  
    cnADO.Open "Provider=Microsoft.ACE.OLEDB.12.0;Data  
Source=" & strPath  
    strSQL = "SELECT * FROM 员工档案"  
    rsADO.Open strSQL, cnADO, 1, 3  
    avntTitle = Array("员工编号", "姓名", "性别", "学历")  
    rsADO.Filter = "学历='大专'"  
    avntData = rsADO.GetRows(-1, 1, avntTitle)  
    Cells.ClearContents  
    Range("a1:d1") = avntTitle  
    Range("a2").Resize(UBound(avntData, 2) + 1,  
UBound(avntData, 1) + 1) = Application.Transpose(avntData)  
    rsADO.Close  
    cnADO.Close  
    Set rsADO = Nothing  
    Set cnADO = Nothing  
    Exit Sub  
ErrMsg:  
    MsgBox Err.Description, , "错误报告"  
End Sub
```

```
Sub RecordsetToArray1()  
    Dim cnADO As Object, rsADO As Object  
    Dim strPath As String, strSQL As String  
    Dim avntData() As Variant, avntTitle() As Variant  
    Dim avntResult As Variant, i As Long  
    Dim j As Long  
    On Error GoTo ErrMsg  
    Set cnADO = CreateObject("ADODB.Connection")  
    Set rsADO = CreateObject("ADODB.Recordset")  
    strPath = ThisWorkbook.Path & "\员工管理.accdb"  
    cnADO.Open "Provider=Microsoft.ACE.OLEDB.12.0;Data  
Source=" & strPath  
    strSQL = "SELECT * FROM 员工档案"  
    rsADO.Open strSQL, cnADO, 1, 3  
    avntTitle = Array("员工编号", "姓名", "性别", "学历")  
    rsADO.Filter = "学历='大专'"  
    avntData = rsADO.GetRows(-1, 1, avntTitle)  
    ReDim avntResult(1 To UBound(avntData, 2) + 1, 1 To  
UBound(avntData, 1) + 1)  
    For i = 0 To UBound(avntData, 2)  
        For j = 0 To UBound(avntData)  
            avntResult(i + 1, j + 1) = avntData(j, i)  
        Next j  
    Next i  
    Cells.ClearContents  
    Range("a1:d1") = avntTitle  
    Range("a2").Resize(UBound(avntResult),  
UBound(avntResult, 2)).Value =  
Application.Transpose(avntData)  
    rsADO.Close  
    cnADO.Close
```

```
Set rsADO = Nothing  
Set cnADO = Nothing  
Exit Sub
```

```
ErrMsg:  
    MsgBox Err.Description, , "错误报告"  
End Sub
```

16.28 生成各种统计报表

```
Sub DataGroup()  
    Dim cnADO As Object  
    Dim rsADO As Object  
    Dim strSQL As String  
    Dim i As Integer  
    Dim strPath As String  
    On Error GoTo ErrMsg  
    Set cnADO = CreateObject("ADODB.Connection")  
    strPath = ThisWorkbook.FullName  
    cnADO.Open "Provider=Microsoft.Ace.OLEDB.12.0;" _  
        & "Extended Properties=Excel 12.0;" _  
        & "Data Source=" & strPath  
    strSQL = "SELECT 商品名称,地区," _  
        & "SUM(总价)AS 金额," _  
        & "COUNT(*) AS 单数 " _  
        & "FROM [销售明细$]" _  
        & "WHERE 商品名称 IS NOT NULL " _  
        & "GROUP BY 商品名称,地区"  
    Set rsADO = cnADO.Execute(strSQL)  
    Cells.ClearContents  
    For i = 0 To rsADO.Fields.Count - 1  
        Cells(1, i + 1) = rsADO(i).Name  
    Next i  
    Range("a2").CopyFromRecordset rsADO  
    rsADO.Close  
    cnADO.Close  
    Set rsADO = Nothing  
    Set cnADO = Nothing  
    Exit Sub  
ErrMsg:  
    MsgBox Err.Description, , "错误报告"  
End Sub
```

```
Sub DataGroup2()  
    Dim cnADO As Object  
    Dim rsADO As Object  
    Dim strSQL As String  
    Dim i As Integer  
    Dim strPath As String  
    On Error GoTo ErrMsg  
    Set cnADO = CreateObject("ADODB.Connection")  
    strPath = ThisWorkbook.FullName  
    cnADO.Open "Provider=Microsoft.Ace.OLEDB.12.0;" _  
        & "Extended Properties=Excel 12.0;" _  
        & "Data Source=" & strPath  
    strSQL = "SELECT 商品名称," _  
        & "SUM(总价)AS 金额," _  
        & "COUNT(*) AS 单数 " _  
        & "FROM [销售明细$]" _  
        & "WHERE 商品名称 IS NOT NULL " _  
        & "GROUP BY 商品名称"  
    Set rsADO = cnADO.Execute(strSQL)  
    Cells.ClearContents  
    For i = 0 To rsADO.Fields.Count - 1  
        Cells(1, i + 1) = rsADO(i).Name
```

第 16 章 数据库应用 接上页 16.28

```
Next i
Range("a2").CopyFromRecordset rsADO
rsADO.Close
cnADO.Close
Set rsADO = Nothing
Set cnADO = Nothing
Exit Sub

ErrMsg:
MsgBox Err.Description, , "错误报告"

End Sub

Sub DataGroup3()
Dim cnADO As Object
Dim rsADO As Object
Dim strSQL As String
Dim i As Integer
Dim strPath As String
On Error GoTo ErrMsg
Set cnADO = CreateObject("ADODB.Connection")
strPath = ThisWorkbook.FullName
cnADO.Open "Provider=Microsoft.Ace.OLEDB.12.0;" _
    & "Extended Properties=Excel 12.0;" _
    & "Data Source=" & strPath
strSQL = "SELECT 商品名称,地区," _
    & "SUM(总价)AS 金额 ," _
    & "COUNT(*) AS 单数 " _
    & "FROM [销售明细$]" _
    & "WHERE 数量>10 AND 地区='北京' " _
    & "GROUP BY 商品名称,地区"
Set rsADO = cnADO.Execute(strSQL)
Cells.ClearContents
For i = 0 To rsADO.Fields.Count - 1
    Cells(1, i + 1) = rsADO(i).Name
Next i
Range("a2").CopyFromRecordset rsADO
rsADO.Close
cnADO.Close
Set rsADO = Nothing
Set cnADO = Nothing
Exit Sub

ErrMsg:
MsgBox Err.Description, , "错误报告"

End Sub

Sub DataGroup4()
Dim cnADO As Object
Dim rsADO As Object
Dim strSQL As String
Dim i As Integer
Dim strPath As String
On Error GoTo ErrMsg
Set cnADO = CreateObject("ADODB.Connection")
strPath = ThisWorkbook.FullName
cnADO.Open "Provider=Microsoft.Ace.OLEDB.12.0;" _
    & "Extended Properties=Excel 12.0;" _
    & "Data Source=" & strPath
strSQL = "SELECT 商品名称,地区," _
    & "SUM(总价)AS 金额 ," _
    & "COUNT(*) AS 单数 " _
    & "FROM [销售明细$]" _
    & "WHERE 数量>10 AND 地区='北京' "
```

```
& "GROUP BY 商品名称,地区 " _
& "HAVING SUM(总价)>1000"
Set rsADO = cnADO.Execute(strSQL)
Cells.ClearContents
For i = 0 To rsADO.Fields.Count - 1
    Cells(1, i + 1) = rsADO(i).Name
Next i
Range("a2").CopyFromRecordset rsADO
rsADO.Close
cnADO.Close
Set rsADO = Nothing
Set cnADO = Nothing
Exit Sub

ErrMsg:
MsgBox Err.Description, , "错误报告"

End Sub

16.29 员工管理系统

Public gcnADO As ADODB.Connection
Public grsADO As ADODB.Recordset
Public gstrTable As String

Sub TheStaffManagesSystem()
frmStaffingSystem.Show
End Sub

Sub TheStaffPicManages()
frmStaffingPhotoSystem.Show
End Sub

Sub CreateDatabase()
Dim catADO As New ADOX.Catalog
Dim strPath As String
Dim strSQL As String
strPath = ThisWorkbook.Path & "\员工管理 1.accdb"
gstrTable = "员工档案"
On Error GoTo ErrMsg
catADO.Create "Provider=Microsoft.ACE.OLEDB.12.0;Data
Source=" & strPath
strSQL = "CREATE TABLE " & gstrTable _
    & "(员工编号 int not null primary key," _
    & "姓名 text(20) not null, 性别 text(1) not null," _
    & "民族 text(20) not null,部门 text(20) not null," _
    & "职务 text(20),电话 text(20),学历 text(20)," _
    & "出生日期 date not null,籍贯 text(20),简历 text,
照片 image)"
catADO.ActiveConnection.Execute strSQL
Set catADO = Nothing
Exit Sub

ErrMsg:
MsgBox Err.Description, , "错误报告"

End Sub

Sub UpdateaddRecords()
Dim cnADO As New ADODB.Connection
Dim rsADO As New ADODB.Recordset
Dim strPath As String
Dim strTemp As String
Dim strSQL As String
Dim strMsg As String
Dim avntFields As Variant
Dim i As Long
Worksheets("数据库").Select
strPath = ThisWorkbook.Path & "\员工管理.accdb"
```

第 16 章 数据库应用

```
gstrTable = "员工档案"
On Error GoTo ErrMsg
cnADO.Open "Provider=Microsoft.Ace.OLEDB.12.0;Data
Source=" & strPath
strSQL = "SELECT B.员工编号 FROM " & gstrTable & "
A,[Excel 12.0;IMEX=0;Database=" & ThisWorkbook.FullName &
"].[" & ActiveSheet.Name & "$" _
& Range("A1").CurrentRegion.Address(0, 0) & "]" B
WHERE A.员工编号=B.员工编号"
Set rsADO = New ADODB.Recordset
rsADO.Open strSQL, cnADO, adOpenKeyset,
adLockOptimistic
If rsADO.RecordCount > 0 Then
    avntFields = Range("A1:K1")
    For i = 2 To UBound(avntFields, 2)
        strTemp = strTemp & ",a." & avntFields(1, i) &
"=b." & avntFields(1, i)
    Next i
    strSQL = "UPDATE " & gstrTable & " A,[Excel
12.0;IMEX=0;Database=" & ThisWorkbook.FullName & "].[" &
ActiveSheet.Name & "$" _
& Range("A1").CurrentRegion.Address(0, 0) & "]"
B SET " & Mid(strTemp, 2) & " WHERE A.员工编号=B.员工编号"
    cnADO.Execute strSQL
    strMsg = rsADO.RecordCount & "条记录已更新！"
End If
If rsADO.RecordCount < Range("A" &
Rows.Count).End(xlUp).Row - 1 Then
    strSQL = " SELECT A.* FROM [Excel 12.0;Database=" &
ThisWorkbook.FullName & "].[" & ActiveSheet.Name & "$" &
Range("A1").CurrentRegion.Address(0, 0) _
& "]" A LEFT JOIN " & gstrTable & " B ON A.员工
编号=B.员工编号 WHERE B.员工编号 IS NULL"
    Set rsADO = New ADODB.Recordset
    rsADO.Open strSQL, cnADO, adOpenKeyset,
adLockOptimistic
    If rsADO.RecordCount > 0 Then
        strSQL = "INSERT INTO " & gstrTable & strSQL
        cnADO.Execute strSQL
        strMsg = strMsg & vbCrLf & rsADO.RecordCount
& "条记录已添加到数据库！"
    End If
End If
MsgBox strMsg, vbInformation, "提示"
rsADO.Close
cnADO.Close
Set rsADO = Nothing
Set cnADO = Nothing
Exit Sub
ErrMsg:
MsgBox Err.Description, "错误报告"
End Sub
```

第 17 章 Excel 与 Internet

17.1 批量创建超链接

```
Sub AutoAddLink() '创建超链接
    Dim strFldPath As String
    With Application.FileDialog(msoFileDialogFolderPicker)
        .Title = "请选择指定文件夹。"
        .AllowMultiSelect = False
```

```
        If .Show Then strFldPath = .SelectedItems(1)
        Else Exit Sub
    End With
    Application.ScreenUpdating = False
    Range("a:b").ClearContents
    Range("a1:b1") = Array("文件夹", "文件名")
    Call SearchFileToHyperlinks(strFldPath)
    Range("a:b").EntireColumn.AutoFit
    Application.ScreenUpdating = True
End Sub
Function SearchFileToHyperlinks(ByVal strFldPath As String) As
String '批量创建超链接
    Dim objFld As Object, objFile As Object, objSubFld As Object
    Dim strFilePath As String, lngLastRow As Long, intNum As
Integer
    Set objFld =
CreateObject("Scripting.FileSystemObject").GetFolder(strFldPat
h)
    For Each objFile In objFld.Files
        lngLastRow = Cells(Rows.Count, 1).End(xlUp).Row + 1
        strFilePath = objFile.Path
        intNum = InStrRev(strFilePath, "\")
        Cells(lngLastRow, 1) = Left(strFilePath, intNum - 1)
        Cells(lngLastRow, 2) = Mid(strFilePath, intNum + 1)
        ActiveSheet.Hyperlinks.Add Anchor:=Cells(lngLastRow, 2),
            Address:=strFilePath,
ScreenTip:=strFilePath
    Next objFile
    For Each objSubFld In objFld.SubFolders
        Call SearchFileToHyperlinks(objSubFld.Path)
    Next objSubFld
    Set objFld = Nothing
    Set objFile = Nothing
    Set objSubFld = Nothing
End Function
Sub AddHyperlinkAtShape() '给 Shape 对象创建超链接
    Dim objShape As Shape
    Set objShape =
ActiveSheet.Shapes.AddShape(msoShapeRectangle, _
        20, 20, 100, 50)
    objShape.TextFrame.Characters.Text = "单击打开文件"
    ActiveSheet.Hyperlinks.Add Anchor:=objShape, Address:=
"c:\test\test.xls", SubAddress:="Sheet1!A3"
    Set objShape = Nothing
End Sub
Sub ClearLink() '只删除超链接
    Dim objHlk As Hyperlink
    For Each objHlk In ActiveSheet.Hyperlinks
        objHlk.Range.Clear
    Next objHlk
    Set objHlk = Nothing
End Sub
Sub DoFollowHyperlink() '使用 FollowHyperlink 方法打开超
链接
    ThisWorkbook.FollowHyperlink _
        Address:="http://s.club.excelhome.net/cse/search", _
        method:=msoMethodGet, _
        extrainfo:="s=2089831883715720896&q=excelvba"
End Sub
Sub RemoveLink() '删除超链接和单元格内容
    ActiveSheet.Hyperlinks.Delete
End Sub
```

第 17 章 Excel 与 Internet

17.2 使用 Lotus Notes 发送邮件

Option Explicit

Sub SendEmailbyNotes() '使用 Lotus Notes 发送邮件

Dim objNotes As Object, objDatabase As Object

Dim objDocument As Object, objRTItem As Object

On Error GoTo errHandle

Set objNotes = CreateObject("Notes.NotesSession")

Set objDatabase = objNotes.CURRENTDATABASE

Set objDocument = objDatabase.createDocument

objDocument.Subject = "Try"

objDocument.SendTo = "testname@gmail.com"

objDocument.form = "Memo"

objDocument.SAVEMESSAGEONSEND = True

Set objRTItem =

objDocument.CREATERICHTEXTITEM("Body")

Call objRTItem.AddNewLine(2)

Call objRTItem.AppendText("This is a try to attach the

document")

Call objRTItem.AddNewLine(1)

Call objRTItem.EMBEDOBJECT(1454, "", "c:\test\test.xls")

Call objDocument.send(False)

Set objNotes = Nothing

Set objDatabase = Nothing

Set objDocument = Nothing

Set objRTItem = Nothing

Exit Sub

errHandle:

MsgBox Err.Description

End Sub

17.3 使用 MailEnvelope 发送邮件

Sub SendMailEnvelope()

Dim avntWage As Variant, i As Long, strText As String

Dim objAttach As Object, strAttachPath As String, r As Variant

With Application

.ScreenUpdating = False

.EnableEvents = False

End With

r = Array("test123@163.com", "test124@163.com",

"test124@163.com")

strAttachPath = ThisWorkbook.Path & "\关于企业调整职
工工资的通知.docx"

avntWage = Sheets("工资表").Range("a1").CurrentRegion

For i = 2 To UBound(avntWage)

Range("a2:i2") = Application.Index(avntWage, i)

Range("b1:i2").Select

ActiveWorkbook.EnvelopeVisible = True

With ActiveSheet.MailEnvelope

strText = avntWage(i, 2) & "您好: " & vbCrLf & "

以下是您" & _

avntWage(i, 3) & "月份工资明细, 请查收! "

.Introduction = strText

With .Item

.To = avntWage(i, 1)

.CC =

"test123@163.com;test124@163.com;test124@163.com"

.Subject = avntWage(i, 3) & "月份工资明细"

Set objAttach = .Attachments

Do While objAttach.Count > 0

objAttach.Remove 1

MsgBox objAttach.Count

Loop

.Attachments.Add strAttachPath

.send

End With

End With

Next i

ActiveWorkbook.EnvelopeVisible = False

With Application

.ScreenUpdating = True

.EnableEvents = True

End With

Set objAttach = Nothing

End Sub

17.6 获取百度查询结果

Sub WebQueryBaidu() '抓取百度查询结果的个数

Dim objXMLHTTP As Object, strURL As String

Dim strText As String

Set objXMLHTTP = CreateObject("MSXML2.XMLHTTP")

With objXMLHTTP

strURL = "https://www.baidu.com/s?wd=excelhome"

.Open "GET", strURL, False

.send

strText = .responseText

End With

Range("a1") = "百度: excelhome 结果个数为: "

Range("a2") = Split(Split(strText, "百度为您找到相关结果
")(1), "<")(0)

Set objXMLHTTP = Nothing

End Sub

Sub WebQueryBaiduPN() '抓取百度搜索前 5 页数据

Dim strURL As String, objXMLHTTP As Object, objDOM As Object

Dim objTitle As Object, intPageNum As Integer, k As Integer

Set objXMLHTTP = CreateObject("MSXML2.XMLHTTP")

Set objDOM = CreateObject("htmlfile")

Cells.ClearContents

Range("a1:c1") = Array("序号", "标题", "链接")

k = 1

For intPageNum = 0 To 50 Step 10

strURL = "https://www.baidu.com/s?"

strURL = strURL & "wd=excelhome"

strURL = strURL & "&pn=" & intPageNum

With objXMLHTTP

.Open "GET", strURL, False

.setRequestHeader "If-Modified-Since", "0"

.send

objDOM.body.innerHTML = .responseText

End With

For Each objTitle In

objDOM.getElementsByTagName("h3")

k = k + 1

Cells(k, 1) = k - 1

With objTitle.getElementsByTagName("a")(0)

Cells(k, 2) = .innerText

Cells(k, 3) = .href

End With

Next objTitle

Next intPageNum

Set objXMLHTTP = Nothing

Set objDOM = Nothing

Set objTitle = Nothing

End Sub

第 17 章 Excel 与 Internet

17.7 使用有道实现英汉翻译

```
Sub WebTranslation()  
Dim objXMLHTTP As Object, strURL As String, strText As String  
Dim rngSource As Range, avntSource As Variant, i As Long  
Set objXMLHTTP = CreateObject("MSXML2.XMLHTTP")  
Set rngSource = Range("a2:b" & Cells(Rows.Count,  
1).End(xlUp).Row)  
avntSource = rngSource.Value  
strURL = "http://fanyi.youdao.com/translate"  
With objXMLHTTP  
For i = 1 To UBound(avntSource)  
.Open "POST", strURL, False  
.setRequestHeader "Content-Type",  
"application/x-www-form-urlencoded"  
.send "i=" & avntSource(i, 1) &  
"&from=AUTO&to=AUTO&doctype=json"  
strText = .responseText  
avntSource(i, 2) = Split(Split(strText, "tgt":""")(1), """))(0)  
Next i  
End With  
rngSource.Value = avntSource  
Set objXMLHTTP = Nothing  
Set rngSource = Nothing  
End Sub
```

17.8 获取当当网商品数据

```
Sub WebCrawlerDangD()  
Dim objXMLHTTP As Object, objDOM As Object, objDOMLi As Object  
Dim objShape As Shape, strURL As String, strText As String  
Dim strKey As String, strMsg As String, strMsgYesOrNo As String  
Dim strDOMLi As String, astrResult() As String, vntShapePic As Variant  
Dim intPageNum As Integer, intLiLength As Integer  
Dim lngaResult As Long, i As Long, k As Long  
strKey = Range("a2").Value  
If Len(strKey) = 0 Then  
MsgBox "未在 A2 单元格输入查询关键字。"  
Exit Sub  
End If  
Set objXMLHTTP = CreateObject("MSXML2.XMLHTTP")  
Set objDOM = CreateObject("htmlfile")  
For intPageNum = 1 To 100  
strURL = "http://search.dangdang.com/?"  
strURL = strURL &  
"category_path=01.00.00.00.00.00#J_tab"  
strURL = strURL & "&act=input"  
strURL = strURL & "&key=" & strKey  
strURL = strURL & "&page_index=" & intPageNum  
With objXMLHTTP  
.Open "GET", strURL, False  
.send  
strText = .responseText  
End With  
If InStr(strText, "没有找到") Then Exit For  
objDOM.body.innerHTML = strText  
Set objDOMLi =  
objDOM.getElementById("search_nature_rg").getElementsByTagName("li")  
intLiLength = objDOMLi.Length  
lngaResult = lngaResult + intLiLength  
ReDim Preserve astrResult(1 To 7, 1 To lngaResult)
```

```
For i = 0 To intLiLength - 1  
k = k + 1  
astrResult(1, k) = k  
strDOMLi = objDOMLi(i).innerHTML  
strDOMLi = strDOMLi &  
"now_price>search_pre_price>search_discount>&nbsp;"  
astrResult(4, k) = Val(Mid(Split(strDOMLi,  
"now_price>")(1), 2))  
astrResult(5, k) = Val(Mid(Split(strDOMLi,  
"search_pre_price>")(1), 2))  
If astrResult(5, k) = 0 Then astrResult(5, k) =  
astrResult(4, k)  
astrResult(6, k) = Val(Split(strDOMLi,  
"search_discount>&nbsp;")(1))  
If astrResult(6, k) = 0 Then astrResult(6, k) = ""  
With  
objDOMLi(i).getElementsByTagName("A")(0)  
astrResult(3, k) = .Title  
astrResult(7, k) = .href  
End With  
With  
objDOMLi(i).getElementsByTagName("IMG")(0)  
astrResult(2, k) = .src  
If Left(astrResult(2, k), 4) <> "http" Then  
astrResult(2, k)  
= .getAttribute("data-original")  
End If  
End With  
Next i  
Next intPageNum  
If k = 0 Then  
MsgBox "未找到符合条件的查询结果。"  
Exit Sub  
End If  
ActiveSheet.UsedRange.Offset(3).ClearContents  
Application.ScreenUpdating = False  
For Each objShape In ActiveSheet.Shapes  
If objShape.Type = msoLinkedPicture Then  
objShape.Delete  
Next objShape  
strMsg = "一共有" & k & "张图片需要导入 Excel 工作表。"  
"  
If k > 50 Then strMsg = strMsg & "耗时过长! 不建议导入!"  
"  
strMsgYesOrNo = MsgBox("请选择是否需要导入图书图  
片!" _  
& vbCrLf & strMsg, vbYesNo)  
If strMsgYesOrNo = vbYes Then  
Const PIC_HEIGHT As Integer = 100  
Const RNG_HEIGHT As Integer = 110  
Const RNG_WIDTH As Integer = 16  
Range("B:B").ColumnWidth = RNG_WIDTH  
Range("A5").Resize(k, 1).EntireRow.RowHeight =  
RNG_HEIGHT  
For i = 1 To k  
Set vntShapePic =  
ActiveSheet.Pictures.Insert(astrResult(2, i))  
With Cells(i + 4, 2)  
vntShapePic.Height = PIC_HEIGHT  
vntShapePic.Top = (RNG_HEIGHT -  
PIC_HEIGHT) / 2 + .Top  
vntShapePic.Left = (.Width -
```

第 17 章 Excel 与 Internet 接上页 17.8

```
vntShapePic.Width) / 2 + .Left
    End With
    astrResult(2, i) = ""
    Next i
End If
Range("a4:g4") = Array("序号", "封面", "书名", "现价", "定价", "折扣", "链接")
Range("A5").Resize(k, UBound(astrResult)) = Application.Transpose(astrResult)
Application.ScreenUpdating = True
Set objXMLHTTP = Nothing
Set objDOM = Nothing
Set objDOMLi = Nothing
End Sub
```

17.9 认识 IE 对象

Option Explicit

Sub WebQueryIETable() **'IE 法抓取网页表格数据**

```
Dim objIE As Object
Dim objIEDOM As Object
Dim objTable As Object
Dim objTR As Object
Dim lngRow As Long
Dim intCol As Integer
Set objIE = CreateObject("InternetExplorer.Application")
With objIE
    .Visible = False
    .navigate "http://q.stock.sohu.com/"
    Do Until .readyState = 4
        DoEvents
    Loop
    Set objIEDOM = .document
End With
Cells.ClearContents
Set objTable = objIEDOM.getElementsByTagName("TABLE")(3)
For Each objTR In objTable.Rows
    lngRow = lngRow + 1
    For intCol = 0 To objTR.Cells.Length - 1
        Cells(lngRow, intCol + 1) = objTR.Cells(intCol).innerText
    Next intCol
Next
objIE.Quit
Set objIE = Nothing
Set objIEDOM = Nothing
Set objTable = Nothing
Set objTR = Nothing
End Sub
```

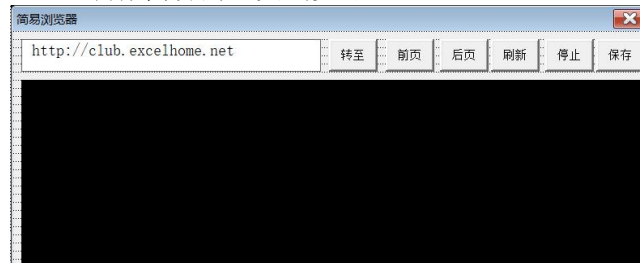
17.10 使用 IE 自动登录网页

Option Explicit

```
Sub LoginWeb()
    Dim strURL As String
    Dim strName As String
    Dim strPassWord As String
    Dim vntTime As Variant
    strName = Range("b1").Value
    strPassWord = Range("b2").Value
    If Len(strName) = 0 Or Len(strPassWord) = 0 Then
        MsgBox "帐户名或密码未输入。"
```

```
Exit Sub
End If
With CreateObject("InternetExplorer.application")
    strURL = "https://xiamen.xueanquan.com/login.html"
    .navigate strURL
    .Visible = True
    Do Until .readyState = 4
        DoEvents
    Loop
    If InStr(1, .LocationURL, "mainpage", vbTextCompare)
Then
        MsgBox "你已登录，无需重复登录。"
    Else
        With .document
            .getElementById("Uname").Value = strName
            .getElementById("PassWord").Value = strPassWord
            .getElementById("LoginButton").Click
        End With
        vntTime = Timer
        Do While Timer < vntTime + 3
            Loop
            If InStr(1, .LocationURL, "mainpage", vbTextCompare) Then
                MsgBox "恭喜，登录成功。"
            Else
                MsgBox "用户名或密码错误，登录失败。"
            End If
        End If
    End With
End Sub
```

17.11 制作简易的网页浏览器



```
Sub CommandButton1_Click()
    frmWebBrowser.Show
End Sub
```

17.12.1 解析 JSON 文档

Sub CreateObjWindow() **'使用 VBA 执行 JavaScript 语句**

```
Dim objDOM As Object
Dim objWindow As Object
Dim strText As String
Set objDOM = CreateObject("htmlfile")
Set objWindow = objDOM.parentWindow
strText = "var obj={name: '张三',sex: '男', age: 20}"
objWindow.execScript strText
MsgBox objWindow.obj.sex
MsgBox objWindow.eval("obj.name")
Set objDOM = Nothing
Set objWindow = Nothing
End Sub
```


第 17 章 Excel 与 Internet

17.12.2 解析 JSON 文档

Sub WebCatchJSONData() '和讯网融资融券交易详情

Dim objXMLHTTP As Object,objDOM As Object

Dim objWindow As Object,strURL As String, strText As String

Dim intPageNum As Integer, i As Integer, j As Integer

Dim lngRow As Long,intCol As Integer,avntTitle As Variant

Set objXMLHTTP = CreateObject("MSXML2.XMLHTTP")

Set objDOM = CreateObject("htmlfile")

Set objWindow = objDOM.parentWindow

Application.ScreenUpdating = False

strURL = "http://stockdata.stock.hexun.com/" _

& "rzrq/data/Stock/DetailCollectDay.ashx"

avntTitle = Array("StockName", "FinancBalance",

"FinancBuy", _

"FinancPay", "FinancAllowance", "FinancSell",

"FinancPayB", _

"FinancSecuritiesBalance", "BalanceDiff",

"ATurnoverMoney", _

"ATurnoverDiff")

ActiveSheet.UsedRange.Offset(1).ClearContents

lngRow = 1

Do While True

intPageNum = intPageNum + 1

With objXMLHTTP

.Open "GET", strURL & "?&page=" &

intPageNum, False

.send

strText = .responseText

End With

strText = Split(Split(strText, "]")(0), "[")(1)

If Len(strText) = 0 Then Exit Do

strText = "<script>data=[" & strText & "]</script>"

objDOM.write strText

For i = 0 To objWindow.eval("data.length") - 1

lngRow = lngRow + 1

intCol = 0

For j = 0 To UBound(avntTitle)

intCol = intCol + 1

Cells(lngRow, intCol) =

objWindow.eval("data[" & i & "]." & avntTitle(j))

Next j

Next i

Loop

Application.ScreenUpdating = True

Set objXMLHTTP = Nothing

Set objDOM = Nothing

Set objWindow = Nothing

End Sub

17.13 获取网页中的表格数据 (获取历史股票价格)

Sub DownloadQueryTables() 'QueryTable

Dim strQuery As String,intYear As Integer

Dim intQtr As Integer, lngStockCode As Long

lngStockCode = Cells(1, 4).Value

intYear = Cells(2, 4).Value

intQtr = Cells(3, 4).Value

strQuery =

"URL;http://quotes.money.163.com/trade/ljsysj_" &

lngStockCode

strQuery = strQuery & ".html?year=" & intYear

strQuery = strQuery & "&season=" & intQtr

ActiveSheet.UsedRange.Offset(4).ClearContents

With ActiveSheet.QueryTables.Add(Connection:=strQuery _

, Destination:=Range("A5"))

.Name = "history"

.RefreshOnFileOpen = False

.BackgroundQuery = True

.SaveData = True

.PreserveFormatting = True

.AdjustColumnWidth = False

.RefreshPeriod = 0

.WebSelectionType = xlSpecifiedTables

.WebFormatting = xlWebFormattingNone

.WebTables = "4"

.Refresh BackgroundQuery:=False

End With

End Sub

Sub DownloadHTMLFile() 'HTMLFILE

Dim objXMLHTTP As Object,objDOM As Objec,objTable As Object

Dim objTR As Object,objCell As Object,strURL As String

Dim intYear As Integer,intQtr As Integer,lngRow As Long

Dim intCol As Integer, lngStockCode As Long

Set objXMLHTTP = CreateObject("MSXML2.XMLHTTP")

Set objDOM = CreateObject("htmlfile")

lngStockCode = Cells(1, 4).Value

intYear = Cells(2, 4).Value

intQtr = Cells(3, 4).Value

strURL = "http://quotes.money.163.com/trade/ljsysj_" &

lngStockCode

strURL = strURL & ".html?year=" & intYear

strURL = strURL & "&season=" & intQtr

With objXMLHTTP

.Open "GET", strURL, False

.send

objDOM.body.innerHTML = .responseText

End With

ActiveSheet.UsedRange.Offset(4).ClearContents

Set objTable =

objDOM.getElementsByTagName("TABLE")(3)

For Each objTR In objTable.Rows

lngRow = lngRow + 1

intCol = 0

For Each objCell In objTR.Cells

intCol = intCol + 1

Cells(lngRow + 4, intCol) = objCell.innerText

Next objCell

Next objTR

Set objXMLHTTP = Nothing

Set objDOM = Nothing

Set objTable = Nothing

Set objTR = Nothing

Set objCell = Nothing

End Sub

Sub DownloadClipboard() 'Clipboard

Dim strURL As String

Dim strText As String

Dim intYear As Integer

Dim intQtr As Integer

Dim lngStockCode As Long

lngStockCode = Cells(1, 4).Value

intYear = Cells(2, 4).Value

intQtr = Cells(3, 4).Value

strURL = "http://quotes.money.163.com/trade/ljsysj_" &

第 17 章 Excel 与 Internet 接上页 17.13

lngStockCode

```
strURL = strURL & ".html?year=" & intYear
strURL = strURL & "&season=" & intQtr
With CreateObject("MSXML2.XMLHTTP")
    .Open "GET", strURL, False
    .send
    strText = .responseText
End With
strText = Split(Split(strText, "<table")(4), "</table>")(0)
strText = "<table" & strText & "</table>"
CopyToClipboard strText
ActiveSheet.UsedRange.Offset(4).ClearContents
Range("a5").Select
ActiveSheet.Paste
Range("a1").Select
```

End Sub

Sub CopyToClipboard(strText As String)

With

CreateObject("new:{1C3B4210-F441-11CE-B9EA-00AA006B1A69}")

.SetText strText

.PutInClipboard

End With

End Sub

17.15 对非 UTF-8 编码的数据进行编码转换

Sub WebCrawlerSina() ' 新浪财经个股上榜统计

Dim objXMLHTTP As Object

Dim objDOM As Object

Dim objTable As Object

Dim objTR As Object

Dim objCell As Object

Dim strURL As String

Dim strText As String

Dim lngRow As Long

Dim intCol As Integer

Set objDOM = CreateObject("htmlfile")

Set objXMLHTTP = CreateObject("MSXML2.XMLHTTP")

Application.ScreenUpdating = False

strURL = "http://vip.stock.finance.sina.com.cn/"

strURL = strURL &

"q/go.php/vLHBData/kind/ggtj/index.phtml"

With objXMLHTTP

.Open "GET", strURL, False

.send

strText = strByteToBstr(.responseBody, "GB2312")

'strText = StrConv(.responseBody, vbUnicode)

End With

objDOM.body.innerHTML = strText

Set objTable = objDOM.getElementById("dataTable")

Cells.ClearContents

For Each objTR In objTable.Rows

lngRow = lngRow + 1

intCol = 0

For Each objCell In objTR.Cells

intCol = intCol + 1

Cells(lngRow, intCol) = objCell.innerText

Next objCell

Next objTR

Application.ScreenUpdating = True

Set objXMLHTTP = Nothing

Set objDOM = Nothing

Set objTable = Nothing

Set objTR = Nothing

Set objCell = Nothing

End Sub

Function strByteToBstr(ByVal aByte As Variant, ByVal CodeBase

As String) As String

Dim objStream As Object

Set objStream = CreateObject("Adodb.Stream")

With objStream

.Type = 1

.Mode = 3

.Open

.write aByte

.Position = 0

.Type = 2

.Charset = CodeBase

strByteToBstr = .ReadText

End With

objStream.Close

Set objStream = Nothing

End Function

17.16 处理防盗链

Sub WebCrawlerSSE()

Dim strURL As String

Dim strText As String

Dim objWINHTTP As Object

Dim objDOM As Object

Dim objWindow As Object

Dim avntTitle As Variant

Dim i As Long

Dim lngRow As Long

Dim intCol As Integer

Set objWINHTTP =

CreateObject("WinHttp.WinHttpRequest.5.1")

Set objDOM = CreateObject("htmlfile")

Set objWindow = objDOM.parentWindow

strURL = "http://query.sse.com.cn/"

strURL = strURL &

"marketdata/tradedata/queryTopMktValByPage.do?"

strURL = strURL & "jsonCallBack=jsonpCallback46746"

strURL = strURL & "&isPagination=true"

strURL = strURL & "&searchDate=2018-05-22"

strURL = strURL & "&_=1527077714965"

With objWINHTTP

.Open "GET", strURL, False

.setRequestHeader "Referer", _

"http://www.sse.com.cn/market/stockdata/marketvalue/"

.send

strText = .responseText

End With

strText = Split(Split(strText, "[")(1), "]")(0)

objDOM.write "<script>data=[" & strText & "]/</script>"

avntTitle = Array("rank", "productA", "productFullName",

—

"productName", "market", "marketPer")

ActiveSheet.UsedRange.Offset(1).ClearContents

lngRow = 1

For i = 0 To objWindow.eval("data.length") - 1

lngRow = lngRow + 1

For intCol = 0 To UBound(avntTitle)

Cells(lngRow, intCol + 1) =

第 17 章 Excel 与 Internet

接上页 17.16

```
objWindow.eval("data[" & i & "]." & avntTitle(intCol))
    Next intCol
Next i
Set objWINHTTP = Nothing
Set objDOM = Nothing
Set objWindow = Nothing
End Sub

Sub WebCrawlerQzone() '抓取 QQ 空间说说数据
    Dim strURL As String
    Dim strCookie As String
    Dim strText As String
    Dim strGTK As String
    Dim strKey As String
    Dim strUserName As String
    Dim strMsg As String
    Dim lngPageNum As Long
    Dim lngCreateTime As Long
    Dim k As Long
    Dim i As Long
    Dim blnClick As Boolean
    Dim objIE As Object
    Dim objWINHTTP As Object
    Dim objDIC As Object
    Dim objDOM As Object
    Dim objTagA As Object
    Dim objList As Object
    Dim objWindow As Object
    Dim sngTime As Single
    Dim vntQQNum As Variant
    Set objDIC = CreateObject("scripting.dictionary")
    Set objIE = CreateObject("InternetExplorer.Application")
    Set objWINHTTP =
CreateObject("WinHttp.WinHttpRequest.5.1")
    Set objDOM = CreateObject("htmlfile")
    Set objWindow = objDOM.parentWindow
    strURL = "https://xui.ptlogin2.qq.com/cgi-bin/xlogin?"
    strURL = strURL & "proxy_url=https%3A//qzs.qq.com/"
    strURL = strURL & "qzone/v6/portal/proxy.html"
    strURL = strURL & "&appid=549000912"
    strURL = strURL &
"&s_url=https%3A%2F%2Fqzs.qzone.qq.com" _
    &
"%2Fqzone%2Fv5%2Floginsucc.html%3Fpara%3Dizone"
    With objIE
        .navigate strURL
        .Visible = True
        sngTime = Timer
        Do While Timer < sngTime + 4
            Loop
        Do Until .readyState = 4
            DoEvents
        Loop
        For Each objTagA
In .document.getElementsByTagName("a")
            If objTagA.TabIndex = 2 Then
                strUserName = objTagA.innerText
                objTagA.Click
                blnClick = True
                Exit For
```

```
End If
Next objTagA
If Not blnClick Then
    MsgBox strUserName & "您的 QQ 软件未登录
或 QQ 空间未开通。"
    Exit Sub
End If
sngTime = Timer
Do While Timer < sngTime + 4
    Loop
    strCookie = .document.cookie
    .Quit
End With
strKey = Split(Split(strCookie, "p_skey=")(1), ";")(0)
strGTK = strGetGTK(strKey)
vntQQNum = Range("b1").Value
strURL = "https://user.qzone.qq.com/"
strURL = strURL & "proxy/domain/taotao.qq.com/"
strURL = strURL & "cgi-bin/emotion_cgi_msglist_v6?"
strURL = strURL & "num=20"
strURL = strURL & "&callback=_preloadCallback"
strURL = strURL & "&format=jsonp"
strURL = strURL & "&uin=" & vntQQNum
strURL = strURL & "&g_tk=" & strGTK
ActiveSheet.UsedRange.Offset(2).ClearContents
k = 3
On Error Resume Next
Application.ScreenUpdating = False
Do While True
    lngPageNum = lngPageNum + 20
    With objWINHTTP
        .Open "GET", strURL & "&pos=" & lngPageNum -
20, False
        .setRequestHeader "Cookie", strCookie
        .send
        strText = .responseText
    End With
    strText = Split(strText, "_preloadCallback(")(1)
    strText = Left(strText, InStrRev(strText, ")") - 1)
    objDOM.write "<script>var data=" & strText &
"</script>"
    For i = 0 To objWindow.eval("data.msglist.length") - 1
        k = k + 1
        Set objList = objWindow.eval("data.msglist[" & i
& "]"")
        lngCreateTime = CallByName(objList,
"created_time", VbGet)
        If Not objDIC.exists(lngCreateTime) Then
            objDIC(lngCreateTime) = ""
        Else
            Exit Do
        End If
        Cells(k, 1) = CallByName(objList, "createTime",
VbGet)
        Cells(k, 2) = CallByName(objList, "content",
VbGet)
        Cells(k, 3) = CallByName(objList, "cmtnum",
VbGet)
    Next i
    Loop
    Range("A3:C3") = Array("日期", "说说", "评论人数")
    Application.ScreenUpdating = True
    strMsg = "用户: " & strUserName & vbCrLf & "您好!"
```

第 17 章 Excel 与 Internet

接上页 17.16

```
strMsg = strMsg & "目标 QQ" & vntQQNum
strMsg = strMsg & "的说说数据已抓取完成。"
MsgBox strMsg
Set objIE = Nothing
Set objWinHTTP = Nothing
Set objDOM = Nothing
Set objWindow = Nothing
Set objDIC = Nothing
Set objList = Nothing
End Sub

Function strGetGTK(ByVal strKey As String) As String
    Dim objNewDom As Object
    Dim objNewWindow As Object
    Dim strJSON As String
    Set objNewDom = CreateObject("htmlfile")
    Set objNewWindow = objNewDom.parentWindow
    With objNewWindow
        strJSON = "gtk=function(skey)"
        strJSON = strJSON & "{for(var hash=5381,i=0,"
        strJSON = strJSON & "len=skey.length;i<len;++i)"
        strJSON = strJSON & "hash+=(hash<<5)"
        strJSON = strJSON & "+skey.charAt(i).charCodeAt(i);"
        strJSON = strJSON & "return hash&2147483647}"
        strJSON = strJSON & "(" & strKey & ");"
        .execScript strJSON
        strGetGTK = .gtk
    End With
    Set objNewWindow = Nothing
    Set objNewDom = Nothing
End Function

17.14 下载网页中的图片、附件
Option Explicit
Private Declare PtrSafe Function URLDownloadToFile Lib
"urlmon" Alias "URLDownloadToFileA" (ByVal pCaller As Long,
ByVal szURL As String, ByVal szExtName As String, ByVal
dwReserved As Long, ByVal lpfnCB As Long) As Long
Private Declare PtrSafe Function DeleteUrlCacheEntry Lib
"wininet" Alias "DeleteUrlCacheEntryA" (ByVal lpszUrlName As
String) As Long
Sub DownloadPictures() '下载网页中的图片
    Dim objXMLHTTP As Object
    Dim strKey As String
    Dim strURL As String
    Dim strFolderPath As String
    Dim strText As String
    Dim strPicPath As String
    Dim strPicURL As String
    Dim strExtName As String
    Dim avntPicURL As Variant
    Dim avntExtName As Variant
    Dim i As Long
    Dim k As Long
    Set objXMLHTTP = CreateObject("MSXML2.XMLHTTP")
    strFolderPath = ThisWorkbook.Path & "\图片\"
    If Dir(strFolderPath, vbDirectory + vbHidden) > "" Then
        If Dir(strFolderPath & ".*") > "" Then Kill
strFolderPath & ".*"
    Else
        MkDir strFolderPath
```

```
End If
strKey = Range("a2").Value
If Len(strKey) = 0 Then
    MsgBox "未输入查询关键字，程序退出。"
    Exit Sub
End If
strKey = strEncodeURI(strKey)
With objXMLHTTP
    strURL = "http://image.baidu.com/search/index?" _
        & "tn=baiduimage&word=" & strKey
    .Open "GET", strURL, "False"
    .send
    strText = .responseText
End With
avntPicURL = Split(strText, ""pageNum"":")
For i = 1 To UBound(avntPicURL)
    If InStr(1, avntPicURL(i), "objURL", vbTextCompare)
Then
        k = k + 1
        strPicURL = Split(Split(avntPicURL(i),
""objURL"":""")(1), """)(0)
        avntExtName = Split(strPicURL, ".")
        strExtName = "." &
avntExtName(UBound(avntExtName))
        strPicPath = strFolderPath & k & strExtName
        URLDownloadToFile 0, strPicURL, strPicPath, 0, 0
        DeleteUrlCacheEntry strPicURL
    End If
Next i
Set objXMLHTTP = Nothing
End Sub
Function strEncodeURI(ByVal strText As String) As String
    Dim objDOM As Object
    Set objDOM = CreateObject("htmlfile")
    With objDOM.parentWindow
        objDOM.Write "<Script></Script>"
        strEncodeURI = .eval("encodeURIComponent(" &
strText & ")")
    End With
    Set objDOM = Nothing
End Function
Sub DownloadAttachment() '下载网页中的附件
    Dim strURL As String
    Dim strPath As String
    Dim intRetVal As Integer
    strURL = "http://club.excelhome.net/forum.php?"
    strURL = strURL & "mod=attachment"
    strURL = strURL &
"&aid=MTkzNzE5OHwyZTZhNGU1Y3wxNTI2NjE2MjIwfwDE4NzEx
Mjd8MTI4NTIwNg%3D%3D"
    strPath = ThisWorkbook.Path & "\text.rar"
    intRetVal = URLDownloadToFile(0, strURL, strPath, 0, 0)
    If intRetVal = 0 Then
        MsgBox "下载成功！"
    Else
        MsgBox "下载失败！"
    End If
End Sub
```


第 18 章 快速创建 XML 文件

18.1 快速创建 XML 文件

Option Explicit

Sub CreateXML() '创建 XML 文件

Dim objXMLDOM As Object

Dim objNode As Object

Dim objVer As Object

Set objXMLDOM = CreateObject("Microsoft.XMLDOM")

Set objVer =

objXMLDOM.createProcessingInstruction("xml", _

"version=" & Chr(34) & "1.0" & Chr(34))

objXMLDOM.appendChild objVer

Set objNode = objXMLDOM.createElement("Test")

objNode.appendChild

objXMLDOM.createTextNode(vbCrLf)

objXMLDOM.appendChild objNode

CreateNode objNode, "FirstName", "张"

CreateNode objNode, "LastName", "三"

CreateNode objNode, "NikName", "张三"

CreateNode objNode, "Level", "版主"

objXMLDOM.Save ThisWorkbook.Path & "\number.xml"

Set objXMLDOM = Nothing

Set objNode = Nothing

Set objVer = Nothing

End Sub

Sub CreateNode(objNode As Object, _

strName As String, strValue As String)'将活动工作表保存

为 XML 文件

Dim objNewNode As Object

With objNode

.appendChild .OwnerDocument.createTextNode(Spac

e\$(4))

Set objNewNode

= .OwnerDocument.createElement(strName)

objNewNode.Text = strValue

.appendChild objNewNode

.appendChild .OwnerDocument.createTextNode(vbCr

Lf)

End With

Set objNewNode = Nothing

End Sub

Sub SaveExceltoXML() '保存指定范围为 XML 文件

ActiveSheet.SaveAs ThisWorkbook.Path & "\excelxml.xml",

xlXMLSpreadsheet

ActiveWorkbook.Close

End Sub

Sub SaveRangetoXML()

Dim strContent As String

strContent =

ActiveSheet.Range("A1:B3").Value(xlRangeValueXMLSpreadshe

et)

Open ThisWorkbook.Path & "\rangexml.xml" For Binary As

#1

Put #1, , strContent

Close #1

End Sub

18.2 定制自己的 Excel RSS 阅读器

Dim mstrChannelDesc As String, mstrChannelLink As String

Dim mstrChannelTitle As String

Dim mstrItem() As String

Sub ParseFeed()

Dim objXMLHTTP As Object

Dim objXMLDOM As Object

Dim blnRet As Boolean

Dim i As Integer

Dim strText As String

Dim strURL As String

Dim wksSht As Worksheet

strURL = "http://club.excelhome.net/forum.php?" _

& "mod=rss&fid=2&auth=0"

Set objXMLHTTP = CreateObject("MSXML2.XMLHTTP")

objXMLHTTP.Open "GET", strURL, False

objXMLHTTP.setRequestHeader "Content-type",

"text/html"

objXMLHTTP.send

If objXMLHTTP.Status <> 200 Then

MsgBox objXMLHTTP.Status & ":" &

objXMLHTTP.StatusText

GoTo ExitSub

End If

Set objXMLDOM = CreateObject("Microsoft.XMLDOM")

objXMLDOM.async = False

blnRet = objXMLDOM.Load(objXMLHTTP.responseXML)

If Not blnRet Then

MsgBox "不是合法的 XML 格式! 请检查!"

GoTo ExitSub

End If

strText = strParseXML(objXMLDOM)

If strText <> "OK" Then

MsgBox strText

GoTo ExitSub

End If

With Application

.ScreenUpdating = False

.DisplayAlerts = False

End With

For Each wksSht In Worksheets

If wksSht.Name = "RSS" Then wksSht.Delete

Next

Worksheets.Add

ActiveSheet.Name = "RSS"

Range("A1").ColumnWidth = 150

ActiveWindow.DisplayGridlines = False

Range("A1") = mstrChannelTitle

Range("A1").Hyperlinks.Add Anchor:=Selection, _

Address:=mstrChannelLink

Range("A2") = mstrChannelDesc

For i = 0 To UBound(mastrItem)

Cells(i * 4 + 4, 1) = (i + 1) & ". " & mastrItem(i, 1)

Cells(i * 4 + 4, 1).Select

Selection.Hyperlinks.Add Anchor:=Selection, _

Address:=mastrItem(i, 2)

Cells(i * 4 + 5, 1) = mastrItem(i, 3)

Cells(i * 4 + 6, 1) = "Created by: " & mastrItem(i, 4) _

& " at " & mastrItem(i, 5)

Next i

Range("A1").Select

With Application

.ScreenUpdating = True

.DisplayAlerts = True

End With

ExitSub:

Set objXMLDOM = Nothing

Set objXMLHTTP = Nothing

End Sub

第 18 章 快速创建 XML 文件

接上面 18.2

Function strParseXML(ByVal objXMLDOM As Object) As String

Dim objItemNodes As Object

Dim objNode As Object

Dim objChlNode As Object

Dim i As Integer

If objXMLDOM.documentElement.BaseName <> "rss"

Then

strParseXML = "不是合法的 RSS 格式! 请检查!"

Exit Function

End If

Set objChlNode =

objXMLDOM.documentElement.SelectSingleNode("channel")

Set objNode = objChlNode.SelectSingleNode("title")

If Not objNode Is Nothing Then mstrChannelTitle =

objNode.Text

Set objNode =

objChlNode.SelectSingleNode("description")

If Not objNode Is Nothing Then mstrChannelDesc =

objNode.Text

Set objNode = objChlNode.SelectSingleNode("link")

If Not objNode Is Nothing Then mstrChannelLink =

objNode.Text

Set objItemNodes = objChlNode.SelectNodes("item")

If objItemNodes.Length = 0 Then

strParseXML = "RSS 没有项目"

Exit Function

End If

ReDim mstrItem(objItemNodes.Length - 1, 5)

i = 0

For Each objNode In objItemNodes

mstrItem(i, 1) =

objNode.SelectSingleNode("link").Text

mstrItem(i, 2) =

objNode.SelectSingleNode("title").Text

mstrItem(i, 3) =

objNode.SelectSingleNode("description").Text

mstrItem(i, 4) =

objNode.SelectSingleNode("author").Text

mstrItem(i, 5) =

objNode.SelectSingleNode("pubDate").Text

i = i + 1

Next objNode

strParseXML = "OK"

Set objNode = Nothing

Set objItemNodes = Nothing

Set oChannelNode = Nothing

End Function

Sub ListAllElements()

Dim objXMLHTTP As Object

Dim objXMLDOM As Object

Dim objXmlNodeList As Object

Dim objXmlNode As Object

Dim objNode As Object

Dim strURL As String

Dim wksSht As Worksheet

strURL = "http://club.excelhome.net/forum.php?" _

& "mod=rss&fid=2&auth=0"

Set objXMLHTTP = CreateObject("MSXML2.XMLHTTP")

objXMLHTTP.Open "GET", strURL, False

objXMLHTTP.setRequestHeader "Content-type",

"text/html"

objXMLHTTP.send

If objXMLHTTP.Status <> 200 Then

MsgBox objXMLHTTP.Status & ":" &

objXMLHTTP.StatusText

Exit Sub

End If

Set objXMLDOM = CreateObject("Microsoft.XMLDOM")

objXMLDOM.async = False

objXMLDOM.Load (objXMLHTTP.responseXML)

With Application

.ScreenUpdating = False

.DisplayAlerts = False

End With

For Each wksSht In Worksheets

If wksSht.Name = "RSS2" Then wksSht.Delete

Next

Worksheets.Add

ActiveSheet.Name = "RSS2"

ActiveWindow.DisplayGridlines = False

Set objXmlNodeList =

objXMLDOM.getElementsByTagName("*")

For Each objXmlNode In objXmlNodeList

For Each objNode In objXmlNode.ChildNodes

If objNode.nodeType = 3 Then

ActiveCell.Offset(0, 0) =

objXmlNode.nodeName

ActiveCell.Offset(0, 1) = objXmlNode.Text

Elseif objNode.nodeType = 4 Then

ActiveCell.Offset(0, 0) =

objXmlNode.nodeName

End If

Next objNode

ActiveCell.Offset(1, 0).Select

Next objXmlNode

With Application

.ScreenUpdating = True

.DisplayAlerts = True

End With

Set objXMLHTTP = Nothing

Set objXMLDOM = Nothing

Set objXmlNodeList = Nothing

Set objXmlNode = Nothing

Set objNode = Nothing

End Sub

第 19 章 操作其他的 Office 应用程序

19.1 前期绑定与后期绑定

```
Sub CreateOutLook()  
    Dim objOutLook As New Outlook.Application  
    objOutLook  
End Sub  
19.2 将电子表格数据通过 Outlook 邮件发送  
Option Explicit  
Sub SendTest() 'SendTest  
    SendMail "lishi@gmail.com", "wangmazi@21cn.com", "测试", _  
        "这是一个使用 Outlook 自动发送的测试邮件。", _  
        ThisWorkbook.Path & "\try.txt"  
End Sub  
Public Sub SendMail( _  
    ByVal strTo As String, _  
    ByVal strCC As String, _  
    ByVal strSubject As String, _  
    ByVal strBody As String, _  
    Optional ByVal strAttach As String) 'ListUnReadMail  
    Dim objOutLook As Object  
    Dim objMailItem As Object  
    On Error GoTo errHandle  
    Set objOutLook = CreateObject("Outlook.Application")  
    Set objMailItem = objOutLook.CreateItem(0)  
    With objMailItem  
        .To = strTo  
        .CC = strCC  
        .Importance = 2  
        .Subject = strSubject  
        .Body = strBody  
        If Len(strAttach) <> 0 Then .Attachments.Add  
strAttach  
        .Send  
    End With  
    Do Until objMailItem.Sent = True  
        DoEvents  
    Loop  
errHandle:  
    objOutLook.Quit  
    Set objOutLook = Nothing  
    Set objMailItem = Nothing  
End Sub  
Sub ListUnReadMail()  
    Dim objOutLook As Object  
    Dim objNmosp As Object  
    Dim objFolder As Object  
    On Error GoTo errHandle  
    Set objOutLook = CreateObject("Outlook.Application")  
    Set objNmosp = objOutLook.GetNamespace("MAPI")  
    Set objFolder = objNmosp.GetDefaultFolder(6)  
    ListAllFolders objFolder  
errHandle:  
    objOutLook.Quit  
    Set objOutLook = Nothing  
    Set objNmosp = Nothing  
    Set objFolder = Nothing  
End Sub  
Sub ListAllFolders(ByVal objFolder As Object)  
    Dim objItem As Object, str As String  
    Dim objSubFolder As Object  
    Dim strSub As String, strSender As String
```

```
Dim strCC As String, strBody As String  
For Each objItem In objFolder.Items  
    If objItem.UnRead = True Then  
        strSub = objItem.Subject  
        strSender = objItem.SenderEmailAddress  
        strCC = objItem.CC  
        strBody = objItem.HTMLBody  
        MsgBox "发送者: " & strSender & vbCrLf _  
            & "CC: " & strCC & vbCrLf _  
            & "主题: " & strSub & vbCrLf _  
            & "主体: " & strBody  
    End If  
Next objItem  
For Each objSubFolder In objFolder.Folders  
    ListAllFolders objSubFolder  
Next objSubFolder  
Set objItem = Nothing  
Set objSubFolder = Nothing  
End Sub  
19.3 将 Excel 数据输出到 PowerPoint 演示稿  
Sub PPTTest()  
    Dim objPPTApp As Object  
    Dim objPPTPresen As Object  
    Dim objPPTSlide As Object  
    Dim strTemp As String  
    On Error GoTo errHandle  
    strTemp = ThisWorkbook.Path & "\Training.potx"  
    Set objPPTApp = CreateObject("Powerpoint.Application")  
    Set objPPTPresen =  
objPPTApp.Presentations.Add(msoTrue)  
    objPPTApp.Visible = True  
    objPPTPresen.ApplyTemplate Filename:=strTemp  
    Set objPPTSlide = objPPTPresen.Slides.Add(1,  
ppLayoutTitle)  
    With objPPTSlide.Shapes  
        .Placeholders(1).TextFrame.TextRange.Text =  
Range("B1")  
        .Placeholders(2).TextFrame.TextRange.Text = "国内与  
国外浏览器份额比较"  
    End With  
    Set objPPTSlide = objPPTPresen.Slides.Add(2,  
ppLayoutTextAndChart)  
    With objPPTSlide.Shapes  
        .Placeholders(1).TextFrame.TextRange.Text =  
Range("B3")  
        .Placeholders(2).TextFrame.TextRange.Text = "排名前  
三: " & _  
        Range("B5") & "," & _  
        Range("B6") & "," & Range("B7")  
    End With  
    Sheet2.ChartObjects(1).CopyPicture  
objPPTSlide.Shapes.Paste  
    With objPPTSlide.Shapes(4)  
        .left = objPPTSlide.Shapes(3).left  
        .top = objPPTSlide.Shapes(3).top  
        .ScaleWidth 1.2, msoFalse, msoScaleFromTopLeft  
    End With  
    objPPTSlide.Shapes(3).Delete  
    Set objPPTSlide = objPPTPresen.Slides.Add(3,  
ppLayoutTextAndChart)  
    With objPPTSlide.Shapes  
        .Placeholders(1).TextFrame.TextRange.Text =
```

第 19 章 操作其他的 Office 应用程序

接上页 19.3

```
Range("B16")
    .Placeholders(2).TextFrame.TextRange.Text = "排名前三: " & _
        Range("B18") & ", " & _
        Range("B19") & ", " & Range("B20")
End With
Sheet2.ChartObjects(2).CopyPicture
objPPTSlide.Shapes.Paste
With objPPTSlide.Shapes(4)
    .left = objPPTSlide.Shapes(3).left
    .top = objPPTSlide.Shapes(3).top
    .ScaleWidth 1.2, msoFalse, msoScaleFromTopLeft
End With
objPPTSlide.Shapes(3).Delete
objPPTPresen.SaveAs Filename:=ThisWorkbook.Path &
"\PPTTest.pptx"
objPPTPresen.Close
objPPTApp.Quit
errExit:
    Set objPPTSlide = Nothing
    Set objPPTPresen = Nothing
    Set objPPTApp = Nothing
    Exit Sub
errHandle:
    MsgBox Err.Description
    Resume errExit
End Sub
19.4 将 Excel 输出到 Word 文档
19.4.1 在 Excel 中创建 Word 报告
Sub WordOpen()
    Dim objApp As Word.Application
    Dim objDoc As Word.Document
    Set objApp = New Word.Application
    objApp.Visible = True
    Set objDoc =
objApp.Documents.Open(Filename:="D:\WordTest.docx")
    'do something
    objDoc.Close False
    objApp.Quit
    Set objApp = Nothing
    Set objDoc = Nothing
End Sub
Sub WordOpen1()
    Dim objDoc As Word.Document
    'Set objDoc = GetObject("c:\wordtest.docx",
"Word.Document")
    Set objDoc = GetObject("", "Word.Document")
    objDoc.Application.Visible = True
    'do something
    objDoc.Close False
End Sub
Sub GetWord()
    Dim objApp As Object
    Dim objDoc As Object
    On Error Resume Next
    Set objApp = GetObject(, "Word.Application")
    If objApp Is Nothing Then
        MsgBox "当前未检测到打开的 Word 应用程序！"
    Else
        For Each objDoc In objApp.Documents
```

```
        Debug.Print objDoc.Name
        Next objDoc
    End If
    Set objApp = Nothing
    Set objDoc = Nothing
End Sub

Sub WordTest()
    Dim objWordApp As Object
    Dim objWord As Object
    On Error GoTo errHandle
    Set objWordApp = CreateObject("Word.Application")
    Set objWord = objWordApp.Documents.Add
    objWord.Application.Visible = True
    With objWordApp
        .Selection.Style = .ActiveDocument.Styles("标题 1")
        .Selection.TypeText Range("B1").Value
        .Selection.TypeParagraph
        .Selection.TypeText "组件学习份额"
        .Selection.TypeParagraph
        .Selection.Style = .ActiveDocument.Styles("标题 3")
        .Selection.TypeText Range("B3").Value
        .Selection.TypeParagraph
        .Selection.TypeText "排名前三: " & _
            Range("B5") & ", " & _
            Range("B6") & ", " & Range("B7")
        .Selection.TypeParagraph
        Sheets("示例工作表").ChartObjects(1).CopyPicture
        .Selection.Paste
        .Selection.TypeParagraph
        .Selection.Style = .ActiveDocument.Styles("标题 3")
        .Selection.TypeText Range("B16").Value
        .Selection.TypeParagraph
        .Selection.TypeText "排名前三: " & _
            Range("B18") & ", " & _
            Range("B19") & ", " & Range("B20")
        .Selection.TypeParagraph
        Sheets("示例工作表").ChartObjects(2).CopyPicture
        .Selection.Paste
    End With
    objWord.SaveAs Filename:=ThisWorkbook.Path & _
        "\WordTest.docx", _
        FileFormat:=wdFormatXMLDocument
    objWord.Close
    objWordApp.Quit
    Set objWordApp = Nothing
    Set objWord = Nothing
errHandle:
    MsgBox Err.Description
    Resume errExit
End Sub
```

第 19 章 操作其他的 Office 应用程序

19.4.2 在 Excel 中创建邮件合并

Option Explicit

Sub SplitMailMerge()

Dim objWord As Object, objWordNew As Object

Dim objMerge As Object, objFso As Object

Dim avntField As Variant

Dim i As Integer, j As Integer

Dim strMyPath As String, strMyName As String

Dim strNewPath As String, strCopyMyPath As String

strCopyMyPath = ThisWorkbook.Path & "\模板.docx"

strNewPath = ThisWorkbook.Path & "\模板 temp.docx"

Set objFso = CreateObject("Scripting.FileSystemObject")

objFso.CopyFile strCopyMyPath, strNewPath

strMyPath = ThisWorkbook.Path

Set objWord = CreateObject("Word.Application")

With objWord

Set objWordNew = .Documents.Open(strNewPath)

objWord.Visible = False

With .ActiveDocument

.MailMerge.OpenDataSource _

Name:=ThisWorkbook.FullName, _

Connection:="Provider=Microsoft.ACE.OLEDB.12.0;" _

& "Data Source=" &

ThisWorkbook.FullName & ";", _

SQLStatement:="SELECT * FROM `数据源\$`, _

SQLStatement1:="", _

SubType:=1

avntField = Array("编码", "姓名", "部门", "工资

", "卡尾号")

objWord.Selection.HomeKey Unit:=6

For i = 0 To UBound(avntField)

objWord.Selection.Find.Execute

(avntField(i) & "域")

.MailMerge.Fields.Add

Range:=objWord.Selection.Range, Name:=avntField(i)

Next i

Set objMerge = .MailMerge

With objMerge.DataSource

For j = 1 To .RecordCount

.FirstRecord = j

.LastRecord = j

strMyName = .DataFields(1).Value

.ActiveRecord = -2

.Parent.Execute

objWord.ActiveDocument.SaveAs

strMyPath & "\" & strMyName & ".doc"

objWord.ActiveDocument.Close

True

Next j

End With

End With

objWordNew.Close

objWord.Quit

Kill strNewPath

End With

Set objWordNew = Nothing

Set objWord = Nothing

Set objFso = Nothing

Set objMerge = Nothing

MsgBox "生成完毕"

End Sub

第 20 章 使用类模块

20.1 创建和使用自定义对象

1.1 建类模块

Private mstrName As String

Private mstrGrade As String

Public Property Get Name() As String

Name = mstrName

End Property

Public Property Let Name(ByVal strName As String)

mstrName = strName

End Property

Public Property Get Grade() As String

Grade = mstrGrade

End Property

Public Property Let Grade(ByVal strGrade As String)

mstrGrade = strGrade

End Property

Public Sub ShowInfo()

MsgBox "姓名: " & mstrName & vbCrLf & "年级: " &

mstrGrade

End Sub

Private Sub Class_Initialize()

mstrGrade = "一年级"

End Sub

Private Sub Class_Terminate()

MsgBox "objStudent 对象使用的内存及系统资源已经释放"

End Sub

Option Explicit

1.2 运用到普通模块

Sub TestclsStudent()

Dim objStudent As New clsStudent

objStudent.Name = "张三"

objStudent.ShowInfo

Set objStudent = Nothing

End Sub

第 20 章 使用类模块

20.2 设置类的默认属性和为类添加说明

20.2.1 建类模块

```
Private mstrName As String
Private mintScore As Integer
Public Property Get Name() As String
    Name = mstrName
End Property
Public Property Let Name(ByVal strName As String)
    mstrName = strName
End Property
Public Property Get Score() As Integer
    Score = mintScore
End Property
Public Property Let Score(ByVal intScore As Integer)
    mintScore = intScore
End Property
Option Explicit
```

20.2.2 运用到普通模块

```
Sub TestclsSample()
    Dim objSample As New clsSample
    objSample.Name = "这是 Name 属性的值"
    MsgBox objSample
    Set objSample = Nothing
End Sub
```

20.3 捕获应用程序事件和嵌入图表事件

20.3.1 捕获应用程序事件

20.3.1.1 建类模块

```
Public WithEvents xlApp As Application
Private Sub xlApp_WorkbookBeforePrint(ByVal Wb As
Workbook, Cancel As Boolean)
    Dim wksSht As Worksheet
    For Each wksSht In Wb.Worksheets
        With wksSht.PageSetup
            .CenterHeader = "Excel Home"
            .CenterFooter = "第 &P 页 / 共 &N 页"
        End With
    Next wksSht
    Set wksSht = Nothing
End Sub
```

20.3.1.2 运用到普通模块

```
Option Explicit
Public gobjApp As New clsAppEvent
Sub Auto_Open()
    Set gobjApp.xlApp = Application
End Sub
```

20.3.2 捕获嵌入图表事件

20.3.2.1 建类模块

```
Public WithEvents chtChart As Chart
Private Sub chtChart_BeforeDoubleClick(ByVal ElementID As
Long, ByVal Arg1 As Long, ByVal Arg2 As Long, Cancel As
```

Boolean)

Select Case ElementID

Case 4

MsgBox "请不要修改图表标题"

Cancel = True

Case 17

MsgBox "请不要修改坐标轴标题"

Cancel = True

End Select

End Sub

20.3.2.2 运用到普通模块

Option Explicit

Public pObjCht As New clsChart

Sub Auto_Open()

Set pObjCht.chtChart = Sheet1.ChartObjects("图表
1").Chart

End Sub

20.4 设置屏幕分辨率

20.4.1 建类模块

```
Private Declare Function EnumDisplaySettings Lib "User32"
Alias "EnumDisplaySettingsA" (ByVal IpszDeviceName As Long,
ByVal iModeNum As Long, lpDevMode As Any) As Long
Private Declare Function ChangeDisplaySettings Lib "User32"
Alias "ChangeDisplaySettingsA" (lpDevMode As Any, ByVal
dwFlags As Long) As Long
Private Const CCDEVICENAME = 32
Private Const CCFORMNAME = 32
Private Const DM_PELSWIDTH = &H80000
Private Const DM_PELSHEIGHT = &H100000
Private Const ENUM_CURRENT_SETTINGS = -1
Private Type mDEVMODE
    strDeviceName As String * CCDEVICENAME
    intSpecVersion As Integer
    intDriverVersion As Integer
    intSize As Integer
    intDriverExtra As Integer
    lngFields As Long
    intOrientation As Integer
    intPaperSize As Integer
    intPaperLength As Integer
    intPaperWidth As Integer
    intScale As Integer
    intCopies As Integer
    intDefaultSource As Integer
    intPrintQuality As Integer
    intColor As Integer
    intDuplex As Integer
    intYResolution As Integer
    intTTOption As Integer
    intCollate As Integer
    strFormName As String * CCFORMNAME
    intUnusedPadding As Integer
```

第 20 章 使用类模块

接上页 20.4

```
intBitsPerPel As Integer
IngPelsWidth As Long
IngPelsHeight As Long
IngDisplayFlags As Long
IngDisplayFrequency As Long

End Type

Private mDevm As mDEVMODE

Public Property Get Settings() As String
    EnumDisplaySettings 0, ENUM_CURRENT_SETTINGS,
mDevm
    Settings = mDevm.IngPelsWidth & "X" &
mDevm.IngPelsHeight
End Property

Public Sub Change(ByVal intWidth As Integer, ByVal intHeight As
Integer)
MsgBox TypeName(mDevm.strDeviceName)
    Dim lngReturn As Long
    EnumDisplaySettings 0, ENUM_CURRENT_SETTINGS,
mDevm
    mDevm.IngFields = DM_PELSWIDTH Or DM_PELSHEIGHT
    mDevm.IngPelsWidth = intWidth
    mDevm.IngPelsHeight = intHeight
    lngReturn = ChangeDisplaySettings(mDevm, 0)
    If lngReturn <> 0 Then MsgBox "无效的設置"
```

End Sub

20.4.2 运用到普通模块

```
Option Explicit

Sub TestclsChangeDisplay()
    Dim objDisplay As New clsChangeDisplay
    MsgBox objDisplay.Settings
    objDisplay.Change 1280, 960
    MsgBox "分辨率已设置为: 1280X960"
    Set objDisplay = Nothing
```

End Sub

20.5 使用类实现控件数组



类模块

```
Public WithEvents cmdBtn As MSForms.CommandButton

Private Sub cmdBtn_Click()
    With Sheet1.Range("a1").Font
        Select Case cmdBtn.Caption
            Case "小"
                .Size = 8
```

```
Case "标准"
                .Size = 12
            Case "大"
                .Size = 16
            Case "特大"
                .Size = 20
        End Select
    End With
End Sub

20.6 捕获单元格值的改变
类模块

Public WithEvents gwksSht As Worksheet

Public Event Change()

Private mrngRange As Range
Private mvntOriginal As Variant
Private mstrMsg As String

Public Property Get Range() As Range
    Set Range = mrngRange
    Set gwksSht = Nothing
End Property

Public Property Set Range(ByVal rngRange As Range)
    Set mrngRange = rngRange
    mvntOriginal = mrngRange.Value
    Set gwksSht = rngRange.Parent
End Property

Public Sub ShowInfo()
    MsgBox mstrMsg
End Sub

Private Sub Class_Terminate()
    Set mrngRange = Nothing
End Sub

Private Sub gwksSht_Calculate()
    Dim blnChange As Boolean
    Dim vntValue As Variant
    Dim intRow As Integer
    Dim intCol As Integer
    mstrMsg = ""
    vntValue = mrngRange.Value
    For intRow = 1 To UBound(vntValue)
        For intCol = 1 To UBound(vntValue)
            If vntValue(intRow, intCol) <> mvntOriginal(intRow,
intCol) Then
                blnChange = True
                mstrMsg = mstrMsg & _
                    mrngRange.Cells(intRow, intCol).Address(0, 0) & _
                    "单元格变化,原来的值:" & mvntOriginal(intRow,
intCol) & _
                    " 现在的值:" & vntValue(intRow, intCol) & vbCrLf
                mvntOriginal(intRow, intCol) = vntValue(intRow,
intCol)
            End If
        Next intCol
```


第 20 章 使用类模块

接上页 20.6

```
Next intRow
If blnChange = True Then RaiseEvent Change
End Sub
20.7 利用接口实现类的多态
类模块 clsRange
Implements clsSize
Private mrngRange As Object
Private Property Get clsSize_OData() As Object
    Set clsSize_OData = mrngRange
End Property
Private Property Set clsSize_OData(ByVal objOData As Object)
    Set mrngRange = objOData
End Property
Public Sub clsSize_Zoom()
    mrngRange.Font.Size = mrngRange.Font.Size + 1
End Sub
Implements clsSize
Private mobjShape As Object
Public Property Get clsSize_OData() As Object
    Set clsSize_OData = mobjShape
End Property
Public Property Set clsSize_OData(ByVal objOData As Object)
    Set mobjShape = objOData
End Property
类模块 clsShape
Public Sub clsSize_Zoom()
    mobjShape.Height = mobjShape.Height + 5
    mobjShape.Width = mobjShape.Width + 5
End Sub
```

20.7.2 运用到普通模块

```
Option Explicit
Sub TestSize()
    Dim objSize As clsSize
    Dim objRng As New clsRange
    Dim objShp As New clsShape
    Set objSize = objRng
    Set objSize.OData = Sheet1.Range("a1")
    objSize.Zoom
    Set objSize = objShp
    Set objSize.OData = Sheet1.Shapes(1)
    objSize.Zoom
    Set objSize = Nothing
    Set objRng = Nothing
    Set objShp = Nothing
End Sub
```

20.8 创建自定义的集合

20.8.1 建类模块

```
Private mcolRange As New Collection
Public Property Get Count() As Long
```

```
Count = mcolRange.Count
```

```
End Property
```

```
Public Sub add(ByVal rngItem As Range)
```

```
    Dim rngCell As Range
```

```
    For Each rngCell In rngItem
```

```
        mcolRange.add rngCell
```

```
    Next rngCell
```

```
    Set rngCell = Nothing
```

```
End Sub
```

```
Public Sub Remove(ByVal vntItem As Variant)
```

```
    mcolRange.Remove (vntItem)
```

```
End Sub
```

```
Public Function Item(ByVal vntItem As Variant) As Range
```

```
    Set Item = mcolRange(vntItem)
```

```
End Function
```

```
Public Property Get Items() As Collection
```

```
    Set Items = mcolRange
```

```
End Property
```

20.8.2 运用到普通模块

```
Option Explicit
```

```
Sub TestclsCollection()
```

```
    Dim objCol As New clsCollection
```

```
    Dim rngCell As Range
```

```
    Dim strMsg As String
```

```
    objCol.add Sheet1.Range("a1:c6")
```

```
    MsgBox objCol.Item(1)
```

```
    For Each rngCell In objCol.Items
```

```
        If rngCell.HasArray Or rngCell.HasFormula Then
```

```
            strMsg = strMsg & rngCell.Address(0, 0) & _
```

```
                "的公式为: " & rngCell.Formula & vbCrLf
```

```
        End If
```

```
    Next rngCell
```

```
    MsgBox strMsg
```

```
    Set rngCell = Nothing
```

```
    Set objCol = Nothing
```

```
End Sub
```

20.9 跨工程使用类

20.9.1 建类模块

```
Private mstrName As String
```

```
Private mstrGrade As String
```

```
Public Property Get Name() As String
```

```
    Name = mstrName
```

```
End Property
```

```
Public Property Let Name(ByVal strName As String)
```

```
    mstrName = strName
```

```
End Property
```

```
Public Property Let Grade(ByVal strGrade As String)
```

```
    mstrGrade = strGrade
```

```
End Property
```

```
Public Property Get Grade() As String
```

```
    Grade = mstrGrade
```

```
End Property
```


第 20 章 使用类模块 上接 20.9.1

```
Public Sub ShowInfo()  
    MsgBox "姓名: " & mstrName & vbCrLf & "年级: " &  
mstrGrade  
End Sub  
Private Sub Class_Initialize()  
    mstrGrade = "一年级"  
End Sub  
Private Sub Class_Terminate()  
    MsgBox "使用了 TestReference 工程中的 clsStudent 类"  
End Sub
```

20.9.2 运用到普通模块

```
Option Explicit  
Sub MyTest() '动态数组  
    Dim objStudent As TestReference.clsStudent  
    Set objStudent = TestReference.GetClass  
    objStudent.Name = "李四"  
    objStudent.ShowInfo  
    Set objStudent = Nothing  
End Sub
```

20.10 使用 .NET Framework 中的类

```
Option Explicit  
Sub UseNET()  
    Dim i As Integer  
    Dim vntTemp As Variant  
    Dim objArrLst As Object  
    Set objArrLst =  
CreateObject("System.Collections.ArrayList")  
    For i = 1 To 9  
        objArrLst.Add i  
    Next i  
    objArrLst.Remove (5)  
    If objArrLst.Contains(5) = False Then  
        MsgBox "数组元素“5”已经移除"  
    End If  
    objArrLst.Insert 4, 0  
    Sheet1.Range("a2:b" & Sheet1.Range("a" & _  
        Rows.Count).End(3).Row).ClearContents  
    vntTemp = objArrLst.ToArray  
    Sheet1.Range("a2").Resize(9, 1) =  
Application.Transpose(vntTemp)  
    objArrLst.Sort  
    vntTemp = objArrLst.ToArray  
    Sheet1.Range("b2").Resize(9, 1) =  
Application.Transpose(vntTemp)  
    Set objArrLst = Nothing  
End Sub
```

20.11 监控 Shape 对象

20.11.1 建类模块

```
Public WithEvents cbrBars As Office.CommandBars  
Public Event ShapeSelectChange(ByVal objShpRng As
```

```
ShapeRange, blnCountChange As Boolean)  
Public Event ShapesAddDelete(blnAdd As Boolean)  
Private mobjSelectShpRng As ShapeRange  
Private mstrName As String  
Private mintSelectCount As Integer  
Private mlngCount As Long  
Public Property Get Name() As String  
    Name = mstrName  
End Property  
Public Property Let Name(ByVal strName As String)  
    On Error GoTo line  
    Application.ActiveSheet.Shapes(strName).Select  
    mstrName = strName  
    Exit Property  
line:  
    MsgBox "指定的图形不存在"  
End Property  
Public Property Get Count() As String  
    Count = mlngCount  
End Property  
Public Sub Resize(ByVal intWidth As Integer, intHeight As  
Integer)  
    If mstrName <> "" Then  
Application.ActiveSheet.Shapes(mstrName).LockAspectRatio =  
msoFalse  
        Application.ActiveSheet.Shapes(mstrName).Width =  
intWidth  
        Application.ActiveSheet.Shapes(mstrName).Height =  
intHeight  
    Else  
        MsgBox "请选择要改变大小的图形"  
    End If  
End Sub  
Private Function GetShape() As ShapeRange  
    Dim objShpRng As ShapeRange  
    On Error Resume Next  
    Set objShpRng = Application.Selection.ShapeRange  
    On Error GoTo 0  
    Set GetShape = objShpRng  
    Set objShpRng = Nothing  
End Function  
Private Sub Class_Initialize()  
    Set cbrBars = Application.CommandBars  
    mlngCount = Application.ActiveSheet.Shapes.Count  
    Set mobjSelectShpRng = GetShape()  
    If Not mobjSelectShpRng Is Nothing Then  
        mstrName = mobjSelectShpRng.Item(1).Name  
        mintSelectCount = mobjSelectShpRng.Count  
    Else  
        mstrName = ""  
        mintSelectCount = 0  
    End If  
End Sub
```

第 20 章 使用类模块

```
Private Sub ShapeEvent()  
    With Application.ActiveSheet  
        If .Shapes.Count > mlngCount Then  
            RaiseEvent ShapesAddDelete(True)  
        ElseIf .Shapes.Count < mlngCount Then  
            RaiseEvent ShapesAddDelete(False)  
        End If  
        If Not mobjSelectShpRng Is Nothing Then  
            If .Shapes.Count = mlngCount Then  
                If mobjSelectShpRng.Item(1).Name <>  
mstrName Then  
                    RaiseEvent  
ShapeSelectChange(mobjSelectShpRng, False)  
                End If  
                If mobjSelectShpRng.Count <>  
mintSelectCount Then  
                    RaiseEvent  
ShapeSelectChange(mobjSelectShpRng, True)  
                End If  
            End If  
        Else  
            If mstrName <> "" Then  
                RaiseEvent  
ShapeSelectChange(mobjSelectShpRng, True)  
            End If  
        End With  
    End Sub  
Private Sub cbrBars_OnUpdate()  
    Set mobjSelectShpRng = GetShape()  
    Call ShapeEvent  
    mlngCount = Application.ActiveSheet.Shapes.Count  
    If mobjSelectShpRng Is Nothing Then  
        mstrName = ""  
        mintCount = 0  
    Else  
        mstrName = mobjSelectShpRng.Item(1).Name  
        mintSelectCount = mobjSelectShpRng.Count  
    End If  
End Sub  
20.11.2 运用到普通模块  
Option Explicit  
Dim mobjTestShp As New clsSuperShape  
Sub TestShape()  
    MsgBox mobjTestShp.Count  
    MsgBox mobjTestShp.Name  
    mobjTestShp.Name = "Oval 2"  
    mobjTestShp.Resize 50, 50  
End Sub
```

第 21 章 VBE 相关操作

21.1 设置信任对 VBA 工程模型的访问

Sub TrustVBAProjectSetting() **'自动设置设置信任对 VBA 工程对象模型的访问**

```
    Dim objVBAProject As Object  
    Dim strPrompt As String  
    strPrompt = "已完成代码自动勾选" & vbNewLine & _  
        "【信任对 VBA 工程对象模型的访问】"  
    On Error Resume Next  
    Set objVBAProject = ThisWorkbook.VBProject  
    If Err.Number = 1004 Then  
        CreateObject("WScript.shell").SendKeys "%V{TAB}~"  
        Application.CommandBars.ExecuteMso  
("MacroSecurity")  
        MsgBox strPrompt, vbExclamation, _  
            ThisWorkbook.VBProject.Name  
    End If  
    Set objVBAProject = Nothing  
End Sub  
21.2 引用 VBA 扩展对象库  
Option Explicit  
'显示已使用的引用  
Sub GetRefInfor()  
    Dim i As Integer  
    For i = 1 To ThisWorkbook.VBProject.References.Count  
        With ThisWorkbook.VBProject.References(i)  
            Debug.Print .GUID, .Major, .Minor, _  
                .Name, .Description, .FullPath  
        End With  
    Next i  
End Sub
```

21.2 引用 VBA 扩展对象库

Option Explicit

'显示已使用的引用

Sub GetRefInfor()

Dim i As Integer

For i = 1 To ThisWorkbook.VBProject.References.Count

With ThisWorkbook.VBProject.References(i)

Debug.Print .GUID, .Major, .Minor, _
 .Name, .Description, .FullPath

End With

Next i

End Sub

'检查并自动引用了 VBIDE 对象库

Sub CheckAndRefVBIDE()

Dim i As Long

Dim strGuid As String

Dim intRefCnt As Integer

strGuid = "{0002E157-0000-0000-C000-000000000046}"

With ThisWorkbook.VBProject

For i = 1 To .References.Count

If .References(i).GUID = strGuid Then

MsgBox "本工程已经引用了 VBIDE 库",

vbInformation

Exit Sub

End If

Next i

intRefCnt = .References.Count

.References.AddFromGuid strGuid, 5, 3

If .References.Count - intRefCnt = 1 Then

MsgBox "本工程未引用 VBIDE 库" & _
 "代码已经自动添加该库的引用",

vbInformation

第 21 章 VBE 相关操作

接上页 21.2

```
Else
    MsgBox "本工程未引用 VBIDE 库" & _
        "代码自动添加引用失败", vbCritical
End If
End With
End Sub
'通过文件添加引用
Sub AddReferenceByFile()
    Dim strOlbFile As String
    strOlbFile = "C:\Program Files (x86)\Microsoft Office\" & _
        "root\VFS\ProgramFilesCommonX86\Microsoft
Shared\" & _
        "VBA\VBA6\VBE6EXT.OLB"
    ThisWorkbook.VBProject.References.AddFromFile
strOlbFile
End Sub
'通过 GUID 添加引用
Sub AddReferenceByGUID()
    Dim strGuid As String
    strGuid = "{0002E157-0000-0000-C000-000000000046}"
    ThisWorkbook.VBProject.References.AddFromGuid strGuid,
5, 3
End Sub
'通过 GUID 移除引用
Sub RemoveReferenceByGUID()
    Dim i As Integer
    Dim strGuid As String
    Dim intRefCnt As Integer
    strGuid = "{0002E157-0000-0000-C000-000000000046}"
    With ThisWorkbook.VBProject
        For i = 1 To .References.Count
            If .References(i).GUID = strGuid Then
                intRefCnt = .References.Count

ThisWorkbook.VBProject.References.Remove .References(i)
                If intRefCnt - .References.Count = 1 Then
                    MsgBox "移除引用成功", vbInformation
                Else
                    MsgBox "移除引用失败",
vbInformation
                End If
            End If
        Exit Sub
    End If
    Next
End With
MsgBox "类型库未被引用", vbInformation
End Sub
'查询 VBIDE 引用状态
Sub CheckStatus()
```

```
Dim i As Long
Dim strGuid As String
Application.ScreenUpdating = False
strGuid = "{0002E157-0000-0000-C000-000000000046}"
[Status] = "未引用"
With ThisWorkbook.VBProject
    For i = 1 To .References.Count
        If .References(i).GUID = strGuid Then
            [Status] = "已引用"
        End If
    Next i
End With
Application.ScreenUpdating = True
End Sub
21.3 列出工程中所有部件信息
Sub GetVBComponentsInfo() '列出工程中所有部件信息
    Dim objVBComp As Object
    Dim strVBCompType As String
    Dim i As Long
    i = 1
    Range("A1:C1") = Array("部件名称", "部件类型", "代码行
数")
    For Each objVBComp In
ThisWorkbook.VBProject.VBComponents
        With objVBComp
            Select Case .Type
                Case vbext_ct_Document
                    strVBCompType = "Microsoft Excel 对
象"
                Case vbext_ct_StdModule
                    strVBCompType = "标准模块"
                Case vbext_ct_MSForm
                    strVBCompType = "用户窗体"
                Case vbext_ct_ClassModule
                    strVBCompType = "类模块"
            End Select
            i = i + 1
            Cells(i, 1) = .Name
            Cells(i, 2) = strVBCompType
            Cells(i, 3) = .CodeModule.CountOfLines
        End With
    Next objVBComp
    Set objVBComp = Nothing
End Sub
Sub ClearData() '清空数据
    Range("A:C").ClearContents
End Sub
```

第 21 章 VBE 相关操作

21.4 自动添加模块和代码

```
Sub AddModuleAndCode()  
    Dim objVBComp As VBComponent  
    Dim strMdlName As String  
    strMdlName = "mdlDemo"  
    If blnModuleExist(ActiveWorkbook, strMdlName) Then  
        MsgBox "VBA 工程中已存在模块: " & strMdlName,  
vbInformation  
        Exit Sub  
    Else  
        Set objVBComp = ActiveWorkbook.VBProject. _  
            VBComponents.Add(vbext_ct_StdModule)  
        objVBComp.Name = strMdlName  
    End If  
    With objVBComp.CodeModule  
        If Trim(.Lines(1, 1)) <> "Option Explicit" Then  
            .InsertLines 1, "Option Explicit"  
        End If  
        .InsertLines 2, "Private Const STR_CAPTION As String  
= " & _""Excel Home"""  
        .InsertLines 3, "Sub Auto_Open()  
        .InsertLines 4, vbTab & "MsgBox ""自动添加的  
Auto_Open 事件代码"" & _  
            ", vbInformation, STR_CAPTION"  
        .InsertLines 5, "End Sub"  
        .AddFromString "Sub Auto_Close()" & vbCr & vbTab &  
_  
            "MsgBox ""自动插入的  
Auto_Close 事件代码"" & _  
            ", vbInformation, STR_CAPTION" &  
vbCr & _  
            "End Sub"  
    End With  
    Set objVBComp = Nothing  
End Sub  
Function blnModuleExist(ByVal wkbWork As Workbook, _  
    ByVal strMdlName As String) As Boolean  
    On Error Resume Next  
    blnModuleExist = (Len(wkbWork.VBProject. _  
        VBComponents(strMdlName).Name) > 0)  
End Function  
Sub RemoveModule()  
    With ActiveWorkbook.VBProject.VBComponents  
        .Remove .Item("mdlDemo")  
    End With  
End Sub
```

21.5 快速列出模块中的所有过程

'ProcKindString 根据过程类型值返回一个相应过程类型的字符串

```
Function strProcKind(ByVal ProcKind As vbext_ProcKind) As String  
    Select Case ProcKind
```

```
        Case vbext_pk_Get  
            strProcKind = "Property Get"  
        Case vbext_pk_Let  
            strProcKind = "Property Let"  
        Case vbext_pk_Set  
            strProcKind = "Property Set"  
        Case vbext_pk_Proc  
            strProcKind = "Sub/Function"  
    End Select  
End Function  
'GetProcInModule 获取指定工作簿中指定组件代码模块中的所有过程名称  
Function strGetProcInfor(ByVal wkbWork As Workbook, _  
    ByVal strComponent As String) As String  
    Dim ProcKind As vbext_ProcKind  
    Dim lngLineNum As Long  
    Dim lngProcLine As Long  
    Dim strMsg As String  
    Dim strProcName As String  
    With  
wkbWork.VBProject.VBComponents(strComponent).CodeModule  
        lngLineNum = .CountOfDeclarationLines + 1  
        Do Until lngLineNum >= .CountOfLines  
            strProcName = .ProcOfLine(lngLineNum, ProcKind)  
            lngProcLine = .ProcCountLines(strProcName,  
ProcKind)  
            strMsg = strMsg & strProcName & ": " &  
lngProcLine & ", " & strProcKind(ProcKind) & vbLf  
            lngLineNum = lngProcLine + lngLineNum  
        Loop  
    End With  
    strGetProcInfor = strMsg  
End Function  
Sub PrintProcInfor()  
    Debug.Print strGetProcInfor(ThisWorkbook, "mdlListProc")  
End Sub  
Sub AutoCreateEvents()  
    Dim objCodeMdl As VBIDE.CodeModule  
    Dim strCodeName As String  
    Dim lngStart As Long  
    Dim i As Integer  
    With ThisWorkbook  
        For i = 1 To .Worksheets.Count  
            strCodeName = .Worksheets(i).CodeName  
            Set objCodeMdl = _  
                .VBProject.VBComponents(strCodeName).CodeModule  
            lngStart =  
objCodeMdl.CreateEventProc("BeforeRightClick", _  
                "Worksheet")
```

第 21 章 VBE 相关操作 接上页 21.5

```
objCodeMdl.ReplaceLine lngStart + 1, vbTab & _
    "If Not Application.Intersect(Target,
[A1:D6])) & " Is Nothing Then"
objCodeMdl.InsertLines lngStart + 2, vbTab & _
    vbTab & "Cancel = True"
objCodeMdl.InsertLines lngStart + 3, vbTab & _
    "End If"

Next i
End With
Set objCodeMdl = Nothing
End Sub
Sub ClearCodesInSheets()
    Dim objCodeMdl As VBIDE.CodeModule
    Dim strCodeName As String
    Dim lngStart As Long
    Dim i As Integer
    With ThisWorkbook
        For i = 1 To .Worksheets.Count
            strCodeName = .Worksheets(i).CodeName
            Set objCodeMdl
= .VbProject.VbComponents(strCodeName).CodeModule
            objCodeMdl.DeleteLines 1,
objCodeMdl.CountOfLines
            Next i
        End With
        Set objCodeMdl = Nothing
    End Sub
21.6 自动为对象添加事件过程和代码
Sub AutoCreateEvents() '为每个工作表自动插入事件代码
    Dim objCodeMdl As VBIDE.CodeModule
    Dim strCodeName As String
    Dim lngStart As Long
    Dim i As Integer
    With ThisWorkbook
        For i = 1 To .Worksheets.Count
            strCodeName = .Worksheets(i).CodeName
            Set objCodeMdl = _
                .VbProject.VbComponents(strCodeName).C
odeModule
            lngStart =
objCodeMdl.CreateEventProc("BeforeRightClick", _
                "Worksheet")
            objCodeMdl.ReplaceLine lngStart + 1, vbTab & _
                "If Not Application.Intersect(Target,
[A1:D6])) & _
                " Is Nothing Then"
            objCodeMdl.InsertLines lngStart + 2, vbTab & _
                vbTab & "Cancel = True"
            objCodeMdl.InsertLines lngStart + 3, vbTab & _
                "End If"
```

```
Next i
End With
Set objCodeMdl = Nothing
End Sub
Sub ClearCodesInSheets() '清除所有工作表的所有代码
    Dim objCodeMdl As VBIDE.CodeModule
    Dim strCodeName As String
    Dim lngStart As Long
    Dim i As Integer
    With ThisWorkbook
        For i = 1 To .Worksheets.Count
            strCodeName = .Worksheets(i).CodeName
            Set objCodeMdl
= .VbProject.VbComponents(strCodeName).CodeModule
            objCodeMdl.DeleteLines 1,
objCodeMdl.CountOfLines
            Next i
        End With
        Set objCodeMdl = Nothing
    End Sub
```

第 22 章 数组与字典

22.1 利用数组完成数据交换

```
Sub uRngTomRng()
    Dim avntNum() As Variant, avntDate() As Variant
    avntNum = Range("a1:a11")
    avntDate = Range("b1:b11")
    Range("a1:a11") = avntDate()
    Range("b1:b11") = avntNum()
End Sub
```

```
Sub ArrTest()
    Dim avntTest() As Variant
    Dim avntTestTwo(0 To 2) As Variant
    avntTest() = Array(1, 2, 3)
End Sub
```

```
Option Explicit
Sub BuildArrTwo()
    Dim astrContent(1 To 3, 1 To 3) As String
    astrContent(1, 1) = "实战"
    astrContent(2, 2) = "技巧"
    astrContent(3, 3) = "精粹"
    Range("a1:c3") = astrContent()
End Sub
```

```
Sub ClearData()
    ActiveSheet.Range("A:C").ClearContents
End Sub
```

第 22 章 数组与字典

22.2 罗列符合条件的信息

Option Explicit

Sub ClearRange()

Range("a2:d17").ClearContents

End Sub

Sub LoopArr()

Dim avntData() As Variant, avntResults(1 To 30, 1 To 4) As Variant

Dim intCount As Integer, i As Integer

avntData() = Sheets("数据源").Range("a2:d31").Value

For i = 1 To UBound(avntData(), 1)

If avntData(i, 2) = "EH002" Then

intCount = intCount + 1

avntResults(intCount, 1) = avntData(i, 1)

avntResults(intCount, 2) = avntData(i, 2)

avntResults(intCount, 3) = avntData(i, 3)

avntResults(intCount, 4) = avntData(i, 4)

End If

Next i

Range("A2").Resize(intCount, 4) = avntResults()

End Sub

Sub Lbound_Ubound()

Dim avntTest(2 To 30, 3 To 6) As Variant

MsgBox "数组 avntTest 第一维的下界为: " &

Lbound(avntTest(), 1) & Chr(10) & _

"数组 avntTest 第一维的上界为: " &

UBound(avntTest(), 1) & Chr(10) & _

"数组 avntTest 第二维的下界为: " &

Lbound(avntTest(), 2) & Chr(10) & _

"数组 avntTest 第二维的上界为: " &

UBound(avntTest(), 2)

End Sub

22.3 按指定字符拆分字符串

Option Explicit

Sub SplitDemo() **‘一维数组写入单元格**

Dim astrNum() As String

Dim strMyString As String

strMyString = "EH001,EH002,EH003,EH004,EH005,EH006"

astrNum() = Split(strMyString, ",")

Range("a1").Resize(1, UBound(astrNum) + 1) = astrNum()

End Sub

Sub SplitDemoPortrait() **‘一维数组转置**

Dim astrNum() As String, strMyString As String

strMyString = "EH001,EH002,EH003,EH004,EH005,EH006"

astrNum() = Split(strMyString, ",")

Range("a1").Resize(UBound(astrNum) + 1, 1) =

WorksheetFunction.Transpose(astrNum())

End Sub

Sub ClearRange() **‘清除**

Cells.ClearContents

End Sub

22.4 以指定分隔符连接字符串

Option Explicit

Option Base 1

Sub JoinDemo3()

Dim avntSource() As Variant, avntTranspose() As Variant

Dim avntMine() As Variant, strTemp As String

avntSource() = Range("A1:F1").Value

avntTranspose() =

WorksheetFunction.Transpose(avntSource())

avntMine() =

WorksheetFunction.Transpose(avntTranspose())

strTemp = Join(avntMine(), "-")

MsgBox strTemp

End Sub

Sub JoinDemo()

Dim avntSource() As Variant

Dim strMyStr As String

avntSource() = Array("EH001", "EH002", 65536, "EH003",

Date, "连接")

strMyStr = Join(avntSource(), "-")

MsgBox strMyStr

End Sub

Sub JoinDemo1()

Dim avntSource() As Variant

Dim avntMine() As Variant

Dim MyStr As String

avntSource() = Range("A1:A6")

avntMine() = WorksheetFunction.Transpose(avntSource())

MyStr = Join(avntMine(), "-")

MsgBox MyStr

End Sub

Sub JoinDemo2()

Dim avntSource() As Variant, avntMine() As Variant

Dim strTemp As String

avntSource() = Range("A1:A6").Value

avntMine() = WorksheetFunction.Transpose(avntSource())

strTemp = Join(avntMine(), "-")

MsgBox strTemp

End Sub

22.5 以指定条件连接字符串

Function ConditionJoin(ByVal rngList As Range, ByVal
strCondition As String) **‘指定条件连接字符串**

Dim avntList() As Variant, avarResult() As Variant

Dim i As Integer, j As Integer

avntList() = Intersect(rngList,

rngList.Parent.UsedRange).Value

For i = 1 To UBound(avntList(), 1)

If avntList(i, 2) = strCondition Then

j = j + 1

ReDim Preserve avarResult(1 To j)

avarResult(j) = avntList(i, 3)

End If

Next

ConditionJoin = Join(avarResult(), "、")

End Function

Sub RedimArray()

Dim avntData() As Variant, strRedim As String

avntData() = Array("EH001", "EH002", 65536, "EH003",

Date, "连接")

ReDim avntData(1 To 7)

End Sub

Sub RedimPreArray()

Dim avntRedimPre() As Variant, strRedimPre As String

avntRedimPre() = Array("EH001", "EH002", 65536,

"EH003", Date, "连接")

ReDim Preserve avntRedimPre(0 To 7)

End Sub

Sub StaticArray()

Dim astrStatic(1 To 5, 1 To 4) As String

ReDim Preserve astrStatic(1 To 5, 1 To 3)

End Sub

第 22 章 数组与字典

22.6 多表查询

Option Explicit

Sub ClearRange()

Range("b2:d65536").ClearContents

End Sub

Sub TablesQuery() '取两列重复与独有

Dim wksList As Worksheet

Dim avntList() As Variant, avntResults() As Variant

Dim i As Integer, j As Integer

For Each wksList In Worksheets

If wksList.Name <> "多表查询" Then

avntList() =

wksList.Range("a1").CurrentRegion.Value

For i = 2 To UBound(avntList())

If avntList(i, 4) >= 2000 Then

j = j + 1

ReDim Preserve avntResults(1 To 4, 1 To j)

avntResults(1, j) = wksList.Name

avntResults(2, j) = avntList(i, 2)

avntResults(3, j) = avntList(i, 3)

avntResults(4, j) = avntList(i, 4)

End If

Next i

End If

Next wksList

Range("a2").Resize(j, 4) =

WorksheetFunction.Transpose(avntResults())

End Sub

22.7 两列数据对比重复

Sub getSame()

Dim avntData() As Variant, avntList() As Variant

Dim astrResultsSame() As String, astrResultsDis() As String

Dim avntTemp

Dim intCountSame As Integer, intCountDis As Integer

Dim intTemp As Integer

avntData() =

WorksheetFunction.Transpose(Range("a2:a13").Value)

avntList() =

WorksheetFunction.Transpose(Range("c2:c13").Value)

For intTemp = 1 To UBound(avntData())

avntTemp = Filter(avntList(), avntData(intTemp), True)

If UBound(avntTemp) >= 0 Then

intCountSame = intCountSame + 1

ReDim Preserve astrResultsSame(1 To

intCountSame)

astrResultsSame(intCountSame) =

avntData(intTemp)

Else

intCountDis = intCountDis + 1

ReDim Preserve astrResultsDis(1 To intCountDis)

astrResultsDis(intCountDis) = avntData(intTemp)

End If

Next intTemp

Range("F2").Resize(intCountSame, 1) = _

WorksheetFunction.Transpose(astrResultsSame())

Range("G2").Resize(intCountDis, 1) = _

WorksheetFunction.Transpose(astrResultsDis())

End Sub

Sub FilterDemo()

Dim avntList() As Variant, avntTemp

avntList = Array("A", "AB", "BC", "AC", "ac", "ab", "a")

avntTemp = Filter(avntList(), "A", False, vbTextCompare)

MsgBox Join(avntTemp, ",")

End Sub

Sub ClearRange()

Range("f2:g13").ClearContents

End Sub

22.8 精确查找数组数据

Option Explicit

Sub IndexMatch()

Dim avntData() As Variant, avntResult() As Variant

Dim avntTemp()

Dim intMatch As Integer

avntData() = Sheets("数据源

").Range("a1").CurrentRegion.Value

avntTemp() = Application.Index(avntData(), 0, 1)

intMatch = Application.Match(1122, avntTemp(), 0)

avntResult() = Application.Index(avntData(), intMatch, 0)

Range("A2:F2") = avntResult()

End Sub

Sub ClearRange()

Range("A2:F2").ClearContents

End Sub

22.9 按条件拆分工作表

Sub EraseDemo()

Dim aTest(1 To 3, 1 To 3) As String

aTest(1, 1) = 1

aTest(2, 2) = 2

aTest(3, 3) = 3

Erase aTest

End Sub

Sub SplitTable()

Dim avntList() As Variant, avntTemp()

Dim avntTableName() As Variant

Dim i As Integer, intResult As Integer

Dim wksNew As Worksheet

Dim intCount As Integer, j As Integer

Application.ScreenUpdating = False

avntList() = Range("a1").CurrentRegion.Value

avntTableName = Array("EH001", "EH002", "EH003",

"EH004")

For i = 0 To UBound(avntTableName)

For j = 2 To UBound(avntList())

If avntList(j, 2) = avntTableName(i) Then

intResult = intResult + 1

ReDim Preserve avntTemp(1 To 4, 1 To

intResult)

avntTemp(1, intResult) = avntList(j, 1)

avntTemp(2, intResult) = avntList(j, 2)

avntTemp(3, intResult) = avntList(j, 3)

avntTemp(4, intResult) = avntList(j, 4)

End If

Next j

Set wksNew =

Worksheets.Add(after:=Worksheets(Worksheets.Count))

wksNew.Name = avntTableName(i)

With wksNew

.Range("a1:d1") = Array("序号", "编号", "日期",

"金额")

.Range("a2").Resize(intResult, 4) =

WorksheetFunction.Transpose(avntTemp)

.UsedRange.Borders.LineStyle = xlContinuous

End With

Erase avntTemp()

intResult = 0

Next i

Application.ScreenUpdating = True

End Sub

Sub DeleteSht()

Sheets(Array("EH001", "EH002", "EH003",

"EH004")).Delete

End Sub

第 22 章 数组与字典

22.10 利用数组制作工资条

Option Explicit

Sub ArticleWages()

Dim avntList() As Variant, avntDetail() As Variant

Dim avntResults() As Variant

Dim intRows As Integer, i As Integer, j As Integer

avntList() = Range("a1:f18").Value

ReDim Preserve avntResults(1 To 6, 1 To UBound(avntList()
1) * 3 - 3)

For i = 2 To UBound(avntList(), 1)

avntDetail() = Application.Index(avntList(), i, 0)

intRows = (i - 1) * 3 - 2

For j = 1 To UBound(avntDetail())

avntResults(j, intRows) =

Application.Index(avntList(), 1, 0)(j)

avntResults(j, intRows + 1) = avntDetail(j)

Next j

Next i

With Sheets("工资条

").Range("a1").Resize(UBound(avntResults(), 2), 6)

.Value = Application.Transpose(avntResults())

Range("a1:f3").Copy

Sheets("工资条").Select

.SpecialCells(xlCellTypeConstants, 23).Select

.PasteSpecial xlPasteFormats

Application.CutCopyMode = False

End With

End Sub

Sub ClearRange()

Range("a:f").Clear

End Sub

22.11 冒泡排序

Option Explicit

Sub Bubble_sort()

Dim avntList() As Variant, i As Long, j As Long

Dim lngTemp As Long, strTemp As String

avntList() = Range("a2:b18").Value

For i = UBound(avntList()) To 1 Step -1

For j = 1 To i - 1

If avntList(j, 2) >= avntList(j + 1, 2) Then '比较相

邻数据

strTemp = avntList(j, 1)

lngTemp = avntList(j, 2)

avntList(j, 1) = avntList(j + 1, 1)

avntList(j, 2) = avntList(j + 1, 2)

avntList(j + 1, 1) = strTemp

avntList(j + 1, 2) = lngTemp

End If

Next j

Next i

Range("e2").Resize(UBound(avntList(), 2) = avntList()

End Sub

Sub ClearRange()

Range("e2:f18").ClearContents

End Sub

22.12 字典的前期绑定与后期绑定

Option Explicit

Sub BuildDicBefore()

Dim dicTemp As New Dictionary

dicTemp.Add "Excel", "VBA"

MsgBox "前期绑定方式创建字典, 关键字 Excel 的对应条
目为: " & dicTemp("Excel")

End Sub

Sub buildAfter()

Dim dicTemp As Object

Set dicTemp = CreateObject("Scripting.Dictionary")

dicTemp.Add "Excel", "VBA"

MsgBox "后期绑定方式创建字典, 关键字 Excel 的对应条
目为: " & dicTemp("Excel")

End Sub

22.13 字典对象的常用方法与属性

Option Explicit

Sub BuildDic()

Dim dicTemp As Object

Set dicTemp = CreateObject("Scripting.Dictionary")

dicTemp.Add "Excel", "Home"

MsgBox dicTemp("Excel")

Set dicTemp = Nothing

End Sub

Sub UpdateKey()

Dim dicTemp As Object

Set dicTemp = CreateObject("Scripting.Dictionary")

dicTemp.Add "Excel", "VBA"

dicTemp.Key("Excel") = "Microsoft Excel"

Set dicTemp = Nothing

End Sub

Sub ReadKey()

Dim dicTemp As Object

Set dicTemp = CreateObject("Scripting.Dictionary")

With dicTemp

.Add "Excel", "VBA"

.Key("Excel") = "Microsoft Excel"

.Item("Microsoft Excel") = "Excel Home"

MsgBox .Item("Microsoft Excel")

End With

Set dicTemp = Nothing

End Sub

Sub ItemBuild()

Dim dicTemp As Object

Set dicTemp = CreateObject("Scripting.Dictionary")

dicTemp.Item("Excel") = "Home"

MsgBox dicTemp.Item("Excel")

Set dicTemp = Nothing

End Sub

Sub RemoveDemo()

Dim dicTemp As Object

Dim strExists As String

Set dicTemp = CreateObject("Scripting.Dictionary")

With dicTemp

.Add "Excel", "VBA"

.Add "Word", "域"

.Add "PPT", "动画"

.Remove "Excel"

strExists = If(.Exists("Excel"), "", "不") & "存在"

MsgBox "字典中" & strExists & "关键字 Excel" &

Chr(10) & "字典中含有" & .Count & "条数据"

End With

Set dicTemp = Nothing

End Sub

第 22 章 数组与字典

22.14 利用字典实现条件查询

Option Explicit

Function strLookDic(ByVal rngData As Range, ByVal rngLook As Range) As String

Dim dicData As Object

Dim avntList() As Variant, i As Integer

Set dicData = CreateObject("Scripting.Dictionary")

avntList() = Intersect(rngData,

rngData.Parent.UsedRange).Value

For i = 1 To UBound(avntList())

dicData.Item(avntList(i, 1)) = avntList(i, 2)

Next i

strLookDic = dicData.Item(rngLook.Value)

Set dicData = Nothing

End Function

22.15 字典汇总数据源

Sub dicTotal()

Dim dicData As Object

Dim avntList() As Variant, aNum(), aAmount(), i As Integer

Set dicData = CreateObject("Scripting.Dictionary")

avntList() = Sheets("数据源

").Range("a1").CurrentRegion.Value

With dicData

For i = 2 To UBound(avntList())

.Item(avntList(i, 1)) = avntList(i, 2)

+ .Item(avntList(i, 1))

Next i

Range("a2").Resize(UBound(.Keys) + 1, 1) =

Application.Transpose(.Keys)

Range("b2").Resize(UBound(.Keys) + 1, 1) =

Application.Transpose(.Items)

End With

Set dicData = Nothing

End Sub

Sub ClearRange()

Range("a2:b17").ClearContents

End Sub

22.16 字典创建二级菜单

Sub ShowLogin()

frmLogin.Show

End Sub



22.17 字典实现多条件查询

Private Sub Worksheet_Change(ByVal Target As Range)

If Target.Column = 1 Then

Target.Offset(, 1).ClearContents

End If

End Sub

Private Sub Worksheet_SelectionChange(ByVal Target As Range)

With dicData

If Target.Column = 1 Then

Call mdlDicNest.DicNest

With Target.Validation

.Delete

.Add _

Type:=xlValidateList, _

AlertStyle:=xlValidAlertStop, _

Formula1:=Join(dicData.Keys, ",")

End With

ElseIf Target.Column = 2 Then

If Target.Offset(, -1) = "" Then Exit Sub

Call mdlDicNest.DicNest

With Target.Validation

.Delete

.Add Type:=xlValidateList, _

AlertStyle:=xlValidAlertStop, _

Formula1:=Join(dicData(Target.Offset(, -1).Value & "").Keys, ",")

End With

End If

End With

End Sub

Option Explicit

Function strDicLook(ByVal rngRegion As Range _

, ByVal rngConditionO As Range _

, ByVal rngConditionT As Range _

, ByVal intColumn As Integer) As String

Dim dicData As Object, avntList() As Variant, i As Integer

Set dicData = CreateObject("scripting.dictionary")

avntList = Intersect(rngRegion,

rngRegion.Parent.UsedRange).Value

For i = 1 To UBound(avntList)

dicData(avntList(i, 1) & avntList(i, 2)) = Array(avntList(i, 1), avntList(i, 2), avntList(i, 3), avntList(i, 4), avntList(i, 5))

Next i

strDicLook = dicData(rngConditionO &

rngConditionT)(intColumn - 1)

End Function

Option Explicit

Public dicData As Object

Sub DicNest()

Dim i As Integer

Dim strFirst As String, strSecond As String

Dim avntList() As Variant

avntList() = Sheet2.Range("a1").CurrentRegion.Value

Set dicData = CreateObject("Scripting.Dictionary")

With dicData

For i = 2 To UBound(avntList())

strFirst = avntList(i, 1)

strSecond = avntList(i, 2)

If .Exists(strFirst) = False Then

Set .Item(strFirst) =

CreateObject("Scripting.Dictionary")

End If

.Item(strFirst).Item(strSecond) = ""

Next i

End With

End Sub

第 23 章 代码调试

23.2 使用 Debug 对象

```
Option Explicit
Sub DebugDemo()
    Dim blnVar As Boolean
    Dim dteVar As Date
    Dim vntVar As Variant
    Debug.Print blnVar
    dteVar = Now
    Debug.Print
    Debug.Print dteVar
    vntVar = Null
    Debug.Print vntVar
    vntVar = Empty
    Debug.Print vntVar
    Set vntVar = ActiveCell
    Debug.Print vntVar.Address
End Sub
Function AssertDemo(intNumerator As Integer, _
    intDenominator As Integer) As Double
    Debug.Assert intDenominator <> 0
    If intDenominator <> 0 Then
        AssertDemo = intNumerator / intDenominator
    End If
End Function
Sub DemoMain()
    MsgBox AssertDemo(88, 2)
    MsgBox AssertDemo(88, 0)
End Sub
```

23.3 使用【立即窗口】

```
Sub ImmWinDemo()
    Dim wksSht As Worksheet
    Dim strName As String
    For Each wksSht In Worksheets
        If strName = "" Then
            strName = wksSht.Name
        Else
            strName = strName & vbCrLf & wksSht.Name
        End If
    Next wksSht
    MsgBox strName
End Sub
Sub Demo2()
    Dim intVar As Integer
    For intVar = 1 To 100
        Debug.Print intVar * intVar
    Next intVar
End Sub
```

23.4 使用【本地窗口】

```
Option Explicit
Dim mstrName As String
Sub GetSheetsList()
    Dim wksSht As Worksheet
    For Each wksSht In Worksheets
        If mstrName = "" Then
            mstrName = wksSht.Name
        Else
            mstrName = mstrName & vbCrLf &
wksSht.Name
        End If
    Next wksSht
    MsgBox mstrName
End Sub
```

23.5 使用【监视窗口】

```
Option Explicit
Sub WatchDemo()
```

```
    Dim intRadius As Integer
    Dim blnFlag As Boolean
    For intRadius = 1 To 10
        Debug.Print GetCircleArea(intRadius)
        If intRadius > 7 Then
            blnFlag = True
        End If
    Next intRadius
End Sub
Function GetCircleArea(ByVal intRadius As Integer) As Double
    GetCircleArea = intRadius ^ 2 * 3.14
End Function
```

第 24 章 错误处理

24.1 捕捉错误

```
Sub ErrTrapDemo1()
    On Error GoTo ErrHandler
    ChDrive "Z"
    Exit Sub
ErrHandler:
    MsgBox "程序发生错误，请通知技术支持人员" & vbCrLf
    & "错误编号:" & Err.Number & "," & Err.Description, _
        vbInformation, "错误提示"
    Err.Clear
End Sub
Sub ErrTrapDemo2()
    On Error GoTo 100
    ChDrive "Z"
    Exit Sub
100    MsgBox "程序发生错误，请通知技术支持人员" &
vbCrLf & "错误编号:" & Err.Number & "," & Err.Description, _
        vbInformation, "错误提示"
    Err.Clear
End Sub
Sub ErrTrapDemo3()
    On Error Resume Next
    ChDrive "Z"
    '后续操作
End Sub
Sub ErrTrapDemo4()
    On Error Resume Next
    ChDrive "Z"
    If Err.Number = 68 Then
        MsgBox "程序发生错误：设备不可用，请通知技术
支持人员", vbInformation, "错误提示"
    Exit Sub
    ElseIf Err.Number <> 0 Then
        '其它错误
    Exit Sub
    Else
        '没有错误发生
    End If
End Sub
Sub ErrTrapDemo5()
    On Error Resume Next
    ChDrive "Z"
    On Error GoTo ErrHandler
    i = 5 / 0
    Exit Sub
ErrHandler:
    MsgBox "程序发生错误，请通知技术支持人员" & vbCrLf
    & "错误编号:" & _
        Err.Number & "," & Err.Description, vbInformation, "
错误提示"
    Err.Clear
End Sub
```

第 24 章 错误处理

24.2 处理错误

```
Sub OverWriteFile(strFile As String, strContent As String)
    On Error GoTo ErrHandler
    ChDrive Left(strFile, 1)
    If Dir(strFile) <> "" Then
        Open strFile For Output As #1
        Print #1, strContent
        Close #1
    Else
        MsgBox strFile & "不存在", vbInformation, "提示"
    End If
ErrExit:
    Exit Sub
ErrHandler:
    If Err.Number = 68 Then
        If MsgBox("设备未准备好, YES-重试, NO-退出。", _
            vbYesNo, "错误提示") = vbYes Then
            Resume
        End If
    ElseIf Err.Number = 52 Then
        MsgBox strFile & "非法文件名。", vbInformation, "提示"
    ElseIf Err.Number = 75 Then
        If MsgBox("文件不可写, 更改属性后按 YES 重试, _
            按 NO 退出。", _
            vbYesNo, "错误提示") = vbYes Then
            Resume
        End If
        Close #1
    Else
        Dim intErr As Integer
        Close #1
        intErr = Err.Number
        Err.Clear
        Err.Raise Number:=intErr
    End If
    Resume ErrExit
End Sub
Sub DemoMain()
    Call OverWriteFile("D:\try.txt", "This is a demo")
End Sub
```

24.3 退出错误处理过程

```
Sub ResumeLineDemo()
    Dim sngNumer As Single, sngDenom As Single
    Dim sngResult As Single
ErrRestart:
    On Error GoTo ErrHandler
    sngNumer = Val(InputBox("请输入被除数", "提示"))
    sngDenom = Val(InputBox("请输入除数", "提示"))
    sngResult = sngNumer / sngDenom
    MsgBox sngNumer & " / " & sngDenom & " = " & sngResult
    Exit Sub
ErrHandler:
    If Err.Number = 11 Or Err.Number = 6 Then
        Resume ErrRestart
    Else
        MsgBox "其他错误 " & Err.Number & ":" & _
            Err.Description, vbInformation
    End If
    Resume Next
End Sub
```

24.4 生成错误

```
Sub ErrRaise2(blnTurnOn As Boolean)
    If blnTurnOn = True Then
        Err.Raise Number:=1001, Description:="Raise by ErrRaise2"
    End If
End Sub

Sub ErrRaise1(blnTurnOn1 As Boolean, blnTurnOn2 As Boolean)
    Call ErrRaise2(blnTurnOn2)
    If blnTurnOn1 = True Then
        Err.Raise Number:=1002, Description:="Raise by ErrRaise1"
    End If
End Sub
```

```
End Sub
Sub ErrRaiseDemo()
    On Error GoTo ErrHandler
    Call ErrRaise1(True, True)
    Exit Sub
ErrHandler:
    If Err.Number = 1002 Then
        MsgBox "这是第一层调用里的错误", vbInformation, _
            Err.Description
    ElseIf Err.Number = 1001 Then
        MsgBox "这是第二层调用里的错误", vbInformation, _
            Err.Description
    Else
        MsgBox Err.Description
    End If
    Resume Next
End Sub
```

第 25 章 代码优化与高效编程

25.1 避免使用 Variant 类型

```
Sub VariantDemo()
    Dim i As Long, dteStart As Date
    Dim strTime As String, intX As Integer
    Dim intY As Integer, intRes As Integer
    Dim vntX As Variant, vntY As Variant
    Dim vntRes As Variant
    vntX = 50
    vntY = 25
    dteStart = Timer
    For i = 1 To 8000000
        vntRes = (vntX + vntY) * (vntX - vntY)
        vntRes = (vntX + vntY) / (vntX - vntY)
    Next i
    strTime = Format((Timer - dteStart), "0.00000")
    Debug.Print "Variant: " & strTime & "秒"
    intX = 50
    intY = 25
    dteStart = Timer
    For i = 1 To 8000000
        intRes = (intX + intY) * (intX - intY)
        intRes = (intX + intY) / (intX - intY)
    Next i
    strTime = Format((Timer - dteStart), "0.00000")
    Debug.Print "Integer: " & strTime & "秒"
End Sub
```

25.4 让代码“专注”运行

```
Sub ScreenUpdatingDemo()
    Dim i As Long
    Dim dteStart As Date
    Dim strTime As String
    Application.ScreenUpdating = False
    dteStart = Timer
    For i = 1 To 10000
        Cells(3, 1) = Now
        Cells(3, 1).Interior.Color = vbGreen
    Next i
    strTime = Format((Timer - dteStart), "0.00000")
    Debug.Print "禁用屏幕更新: " & strTime & "秒"
    Application.ScreenUpdating = True
    dteStart = Timer
    For i = 1 To 10000
        Cells(3, 1) = Now
        Cells(3, 1).Interior.Color = vbGreen
    Next i
    strTime = Format((Timer - dteStart), "0.00000")
    Debug.Print "启用屏幕更新: " & strTime & "秒"
End Sub
```