

## ‘ ソースコード

### ‘ 変数・定数 定義

Dim fName0 As String

Dim fName00 As String

Dim ex0 As String

Dim mm As Integer

Dim pWidth As Long

Dim pHeight As Long

Dim dCount As Integer

Dim maxRow As Long

Dim pScale As Double

Const tName = "DataBase"

Const sDBname = "データベース"

Dim iFlg As Boolean

Const Hozon = "保存データ"

Const Kekka As String = "検索結果"

Const ReadMe As String = "ReadMe"

Dim UpdateFlg As Boolean

Dim updateNo As Long

Dim EditNo As Long

Dim DispNo As Long

Dim errFlgCB1 As Boolean

Dim errFlgCB2 As Boolean

Dim errFlgST As Boolean

Dim errFlgSsn As Boolean

Dim errFlgCmnt As Boolean

Dim errFlgCmnt2 As Boolean

Dim errFlgSrch As Boolean

Const Non = "(なし)"

Const All = "すべて"

Const Deleted = "既に削除済み"

Dim sortFlg As Boolean

Private Declare Sub GetKeyboardState \_

Lib "user32" (pbKeyState As Byte)

### ‘ 開発専用処理

Private Sub ButtonDevelopment\_Click()

Dim msg As String

Dim PW As String

Static flg As Boolean

Dim s(255) As Byte

Dim time As Long

Const passWord = "PwDevelop"

msg = "プログラム開発専用です。パスワードを入力してください。" & vbCrLf & vbCrLf & "開発後はこのシートで [Ctrl]+[s] にてプログラムが起動しま

す。"

```
PW = InputBox(msg, "開発専用 パスワード入力")
```

```
If PW = passWord Then
```

```
    If Sheets(sDBname).Visible = False Then
```

```
        Sheets(sDBname).Visible = True
```

```
    End If
```

```
    If Sheets(Hozon).Visible = False Then
```

```
        Sheets(Hozon).Visible = True
```

```
    End If
```

```
    If Sheets(Kekka).Visible = False Then
```

```
        Sheets(Kekka).Visible = True
```

```
    End If
```

```
    If Sheets(ReadMe).Visible = False Then
```

```
        Sheets(ReadMe).Visible = True
```

```
    End If
```

```
Unload UserForm1
```

```
Application.SendKeys ("{Esc}%lv")
```

```
time = 500
```

```
Application.Wait [Now()] + time / 86400000
```

```
,
```

```
If (s(vbKeyNumlock) And 1) = 1 Then
```

```
Else                                     ' NumLock Off か？
```

```
    Application.SendKeys ("{NUMLOCK}")   ' Numlock On
```

```
End If
```

```
Else
```

```
    rc = MsgBox("パスワードが違います。", vbOKOnly, "パスワードエラー")
```

```
End If
```

```
End Sub
```

## ‘ 初期処理

```
Private Sub UserForm_Initialize()
```

```
Dim i As Long
```

```
If Sheets(sDBname).Visible = True Then
```

```
    Sheets(sDBname).Visible = False
```

```
End If
```

```
If Sheets(Hozon).Visible = True Then
```

```
    Sheets(Hozon).Visible = False
```

```
End If
```

```
If Sheets(Kekka).Visible = True Then
```

```
    Sheets(Kekka).Visible = False
```

```
End If
```

```
If Sheets(ReadMe).Visible = True Then
```

```
    Sheets(ReadMe).Visible = False
```

```
End If
```

```
ComboBox1.clear
```

```
ComboBox2.clear
```

```
ComboBoxSearchTarget.clear
```

```
ComboBoxSeason.clear
```

```
sortFlg = False
```

```
For i = 1 To 1000
```

```
    If ActiveWorkbook.Sheets(Hozon).Cells(i, 1) <> "" Then
```

```
        With ComboBox1
```

```
            .AddItem ActiveWorkbook.Sheets(Hozon).Cells(i, 1)
```

```
        End With
```

```
        With ComboBoxSearchTarget
```

```
            .AddItem ActiveWorkbook.Sheets(Hozon).Cells(i, 1)
```

```
        End With
```

```
    Else
```

```
        i = 1000
```

```
    End If
```

```
Next i
```

```
With ComboBox2
```

```
    .AddItem All
```

```
    .AddItem "01 月"
```

```
    .AddItem "02 月"
```

```
    .AddItem "03 月"
```

```
    .AddItem "04 月"
```

```
    .AddItem "05 月"
```

```
    .AddItem "06 月"
```

```
    .AddItem "07 月"
```

```
    .AddItem "08 月"
```

```
    .AddItem "09 月"
```

```
    .AddItem "10 月"
```

```
    .AddItem "11 月"
```

```
    .AddItem "12 月"
```

```
    .AddItem "季節なし"
```

```
End With
```

```
With ComboBoxSeason
```

```
.AddItem "(元のまま)"  
.AddItem "01 月"  
.AddItem "02 月"  
.AddItem "03 月"  
.AddItem "04 月"  
.AddItem "05 月"  
.AddItem "06 月"  
.AddItem "07 月"  
.AddItem "08 月"  
.AddItem "09 月"  
.AddItem "10 月"  
.AddItem "11 月"  
.AddItem "12 月"  
.AddItem "季節なし"
```

End With

Call init

Call init2

iFlg = False

UpdateFlg = False

dCount = 0

Call clearCell

End Sub

‘ 初期化処理 1

Sub init()

iFlg = True

End Sub

‘ 初期化処理 2

Sub init2()

Dim str As String

ComboBox1.clear

ComboBoxSearchTarget.clear

ComboBox1.Enabled = True

ComboBox1.Locked = False

ComboBoxSearchTarget.Enabled = True

ComboBoxSearchTarget.Locked = False

```

For i = 1 To 1000
    If ActiveWorkbook.Sheets(Hozon).Cells(i, 1) <> "" Then
        With ComboBox1
            .AddItem ActiveWorkbook.Sheets(Hozon).Cells(i, 1)
        End With
        With ComboBoxSearchTarget
            .AddItem ActiveWorkbook.Sheets(Hozon).Cells(i, 1)
        End With
    Else
        i = 1000
    End If
Next i

```

```

ComboBoxComment.clear
ComboBoxComment2.clear
ComboBoxComment.Enabled = True
ComboBoxComment.Locked = False
ComboBoxComment2.Enabled = True
ComboBoxComment2.Locked = False

```

```

For i = 1 To 1000
    If Sheets(Hozon).Cells(i, 5) <> "" Then
        If i <> 2 And i <> 3 Then
            str = Sheets(Hozon).Cells(i, 5).Value
            With ComboBoxComment
                .AddItem str
            End With
        End If
        If i <> 1 And i <> 3 Then
            str = Sheets(Hozon).Cells(i, 5).Value
            With ComboBoxComment2
                .AddItem str
            End With
        End If
    Else
        i = 1000
    End If
Next i

```

End Sub

‘ データ追加処理

Private Sub ButtonDataAdd\_Click()

Dim path As String

Dim Fn As String

Dim i As Long

Call clearCell

If ComboBoxSearchTarget.Text = "" Then

rc = MsgBox("検索対象を選択または入力かつ登録して下さい。 データ追加処理", vbOKOnly, "選択エラー")

Exit Sub

End If

If ComboBoxSearchTarget.Text <> "" Then

For i = 1 To 1000

If Sheets(Hozon).Cells(i, 1) = ComboBoxSearchTarget.Text Then

fName0 = Sheets(Hozon).Cells(i, 2) & "¥"

If fName0 = "¥" Then

Call setPath(ComboBoxSearchTarget.Text, i)

fName0 = Sheets(Hozon).Cells(i, 2) & "¥"

If fName0 = "¥" Then

rc = MsgBox("検索対象のパスが入力されませんでした。 データ追加処理", vbOKOnly, "選択エラー")

Exit Sub

End If

End If

End If

Next i

End If

If ComboBoxSeason.Text = "" Then

rc = MsgBox("季節を選んで下さい。 データ追加処理", vbOKOnly, "選択エラー")

Exit Sub

End If

If TextBoxFileName.Text = "" Then

rc = MsgBox("ファイル名を入力して下さい。 データ追加処理", vbOKOnly, "選択エラー")

Exit Sub

End If

If ComboBoxComment2.Text = "" Then

rc = MsgBox("一ロコメントを選択または入力かつ登録して下さい。 データ追加処理", vbOKOnly, "選択エラー")

Exit Sub

End If

If ComboBoxComment2.Text <> "" Then

For i = 1 To 1000

If Sheets(Hozon).Cells(i, 5) = ComboBoxComment2.Text Then

GoTo addProc

End If

If Sheets(Hozon).Cells(i, 5) = "" Then

rc = MsgBox("一ロコメントを選択または入力かつ登録して下さい。 データ追加処理", vbOKOnly, "選択エラー")

Exit Sub

End If

```
Next i
```

```
End If
```

```
addProc:
```

```
If ComboBoxComment2.Text = Non Then
```

```
    ComboBoxComment2.Text = ""
```

```
End If
```

```
.....
```

```
Dim fnExt As String
```

```
fnExt = getExt(TextBoxFileName.Text)
```

```
TextBoxFileName.Text = fnExt
```

```
If fileExistCheck(fnExt) = False Then
```

```
    Exit Sub
```

```
End If
```

```
.....
```

```
path = Sheets(Hozon).Cells(ComboBoxSearchTarget.ListIndex + 1, 2) & "¥"
```

```
Fn = path & TextBoxFileName.Text
```

```
On Error GoTo ERR2
```

```
If Dir(Fn) = "" Then
```

```
    rc = MsgBox(Fn & vbCrLf & "が存在しません", vbOKOnly, "ファイル名エラー ButtonDataAdd_Click")
```

```
    Exit Sub
```

```
End If
```

```
.....
```

```
If sDBname = "" Or tName = "" Then
```

```
    Call init
```

```
End If
```

```
rc = MsgBox("シート名 [" & sDBname & "] テーブル名 [" & tName & "] の最後にデータが追加されます。" & vbCrLf & _
```

```
    "本当にこのデータを追加しますか。", vbOKCancel, "メッセージ")
```

```
If rc = vbCancel Then
```

```
    Exit Sub
```

```
End If
```

```
GoTo nextstp
```

```
ERR2:
```

```
rc = MsgBox(Fn & vbCrLf & "が存在しません", vbOKOnly, "ファイル名エラー")
```

Exit Sub

nextstp:

For i = 2 To 100000

If Sheets(sDBname).Cells(i, 1) = "" Then

    Sheets(sDBname).Cells(i, 1) = ComboBoxSearchTarget.Text

    Sheets(sDBname).Cells(i, 2) = ComboBoxSeason.Text

    Sheets(sDBname).Cells(i, 3) = TextBoxFileName.Text

    Sheets(sDBname).Cells(i, 4) = getFileCreated(path & TextBoxFileName.Text)

    Sheets(sDBname).Cells(i, 5) = ""

    Sheets(sDBname).Cells(i, 6) = ""

    Sheets(sDBname).Cells(i, 7) = ""

    Dim Cmnt As String

    If ComboBoxComment2.Text = Non Then

        Cmnt = ""

    Else

        Cmnt = ComboBoxComment2.Text

    End If

    Sheets(sDBname).Cells(i, 5) = Cmnt

    ComboBox1.Text = ComboBoxSearchTarget.Text

    ComboBox2.Text = ComboBoxSeason.Text

    ComboBoxComment.Text = ComboBoxComment2.Text

    Call clearInputArea

    Call ButtonSearch\_Click

    rc = MsgBox("データが追加されました。検索されます。", vbOKOnly, "メッセージ")

    Exit Sub

End If

Next i

End Sub

### ‘ 入力領域クリア処理

Sub clearInputArea()

    TextBoxFrom.Text = ""

    TextBoxTo.Text = ""

    ComboBoxSearchTarget.ListIndex = -1

    ComboBoxSeason.ListIndex = -1

    TextBoxFileName.Text = ""

    TextBoxFileCreated.Text = ""

    ComboBoxComment2.ListIndex = -1

End Sub



### ‘ ファイルの存在チェック

Function fileExistCheck(Fn As String) As Boolean

```
For i = 1 To 100000
    If Sheets(sDBname).Cells(i, 3) = "" Then
        fileExistCheck = True
        Exit Function
    End If
    If Sheets(sDBname).Cells(i, 3) = Fn Then
        rc = MsgBox(Fn & vbCrLf & "は既に登録済みです。", vbOKOnly, "ファイル名エラー fileExistCheck")
        fileExistCheck = False
        Exit Function
    End If
Next i

End Function
```

### ‘ ファイルパスセット処理

Sub setPath(s As String, i As Long)

```
Dim msg As String

msg = s & " のフォルダパス名を入力して下さい。 " & vbCrLf & "なければ入力しなくて良いです。"
Sheets(Hozon).Cells(i, 1).Value = s
Sheets(Hozon).Cells(i, 2).Value = InputBox(msg)

End Sub
```

### ‘ 拡張子取得処理

Function getExt(f As String) As String

```
Dim path As String

path = Sheets(Hozon).Cells(ComboBoxSearchTarget.ListIndex + 1, 2) & "¥"

Dim ext As String

If Dir(path & f & ".jpg") <> "" Then
    getExt = f & ".jpg"
    Exit Function
ElseIf Dir(path & f & ".png") <> "" Then
    getExt = f & ".png"
    Exit Function
ElseIf Dir(path & f & ".bmp") <> "" Then
    getExt = f & ".bmp"
```

```
Exit Function
```

```
ElseIf Dir(path & f & ".gif") <> "" Then
```

```
    getExt = f & ".gif"
```

```
Exit Function
```

```
Else
```

```
    getExt = f
```

```
End If
```

```
End Function
```

## ‘ すべて削除処理

```
Private Sub ButtonDelAll_Click()
```

```
    Dim NoFrom As Long
```

```
    Dim NoTo As Long
```

```
    Dim i As Long
```

```
    Dim delCnt As Long
```

```
    Dim notDelCnt As Long
```

```
    delCnt = 0
```

```
    notDelCnt = 0
```

```
    If dCount = 0 Then
```

```
        rc = MsgBox("検索してからにしてください。", vbOKOnly, "関連エラー")
```

```
        Exit Sub
```

```
    End If
```

```
    rc = MsgBox("本当に指定範囲のすべてのデータを削除しますか。", vbOKCancel, "メッセージ")
```

```
    If rc = vbCancel Then
```

```
        Exit Sub
```

```
    End If
```

```
    NoFrom = Val(TextBoxFrom.Text)
```

```
    NoTo = Val(TextBoxTo.Text)
```

```
    If NoFrom < 2 Then
```

```
        rc = MsgBox("行番号の From が範囲外です。", vbOKOnly, "数値関連エラー")
```

```
        Exit Sub
```

```
    End If
```

```
    If NoTo > dCount + 1 Then
```

```
        rc = MsgBox("行番号の To が範囲外です。", vbOKOnly, "数値関連エラー")
```

```
        Exit Sub
```

```
    End If
```

```
    If NoFrom > NoTo Then
```

```
        rc = MsgBox("行番号の From > To です。", vbOKOnly, "数値関連エラー")
```

```
        Exit Sub
```

```
    End If
```

```
For i = NoFrom To NoTo
```

```
    If Sheets(Kekka).Cells(i, 6) <> Deleted Then
```

```
        delCnt = delCnt + 1
```

```
        Call Datadelete2(ComboBox1.Text, Sheets(Kekka).Cells(i, 4), i)
```

```
    Else
```

```
        notDelCnt = notDelCnt + 1
```

```
    End If
```

```
Next i
```

```
Call clearInputArea
```

```
rc = MsgBox("指定範囲のデータがすべて削除されました。検索の前にソートされます。[" & CStr(delCnt) & "]件 削除、[" & CStr(notDelCnt) & "]件 以前に削除済み。", vbOKOnly, "メッセージ")
```

```
End Sub
```

## ‘ デリート処理 2

```
Sub Datadelete2(Kbn As String, Fn As String, No As Long)
```

```
    Dim i As Long
```

```
    Dim path As String
```

```
    Dim rws As String
```

```
    path = getPath(Kbn)
```

```
    Sheets(sDBname).ListObjects(tName).Range.AutoFilter Field:=1
```

```
    Sheets(sDBname).ListObjects(tName).Range.AutoFilter Field:=2
```

```
    Sheets(sDBname).ListObjects(tName).Range.AutoFilter Field:=5
```

```
    For i = 2 To 100000
```

```
        If Sheets(sDBname).Cells(i, 4) = "" Then
```

```
            Exit Sub
```

```
        ElseIf Fn = path & Sheets(sDBname).Cells(i, 3) Then
```

```
            rws = CStr(i) & ":" & CStr(i)
```

```
            Sheets(sDBname).Activate
```

```
            Sheets(sDBname).Rows(rws).Select
```

```
            Selection.Delete Shift:=xlUp
```

```
            Sheets(Kekka).Activate
```

```
            Sheets(Kekka).Cells(No, 6).Value = Deleted
```

```
            sortFlg = True
```

```
            Exit Sub
```

```
        End If
```

```
    Next i
```

End Sub

## 削除処理

Private Sub ButtonDelete\_Click()

    If UpdateFlg = True Then

        EditNo = Val(TextBoxFrom.Text)

        If EditNo < 2 Then

            rc = MsgBox("行番号が範囲外です。", vbOKOnly, "数値関連エラー")

            Exit Sub

        End If

        If EditNo > dCount + 1 Then

            rc = MsgBox("行番号が範囲外です。", vbOKOnly, "数値関連エラー")

            Exit Sub

        End If

    If Sheets(Kekka).Cells(EditNo, 6) = Deleted Then

        rc = MsgBox("これは既に削除されています。 No. =" & EditNo & " ファイル名=" & Sheets(Kekka).Cells(EditNo, 4), vbOKOnly, "削除済みエラー")

        Call clearInputArea

        Exit Sub

    End If

    rc = MsgBox("本当にこのデータを削除しますか。", vbOKCancel, "メッセージ")

    If rc = vbCancel Then

        Exit Sub

    End If

    Call dataDelete(TextBoxFileName.Text)

    ComboBoxSearchTarget.Enabled = True

    ComboBoxSearchTarget.Locked = False

    Call textBoxInput(TextBoxFileName)

    Call clearInputArea

    rc = MsgBox("データが削除されました。検索の前にソートされます。", vbOKOnly, "メッセージ")

Else

    rc = MsgBox("データ読み込みが未だです。", vbOKOnly, "関連エラー")

End If

End Sub

## デリート処理

Sub dataDelete(Fn As String)

    Dim i As Long

    Dim path As String

    path = getPath(ComboBox1.Text)

```

Sheets(sDBname).ListObjects(tName).Range.AutoFilter Field:=1
Sheets(sDBname).ListObjects(tName).Range.AutoFilter Field:=2
Sheets(sDBname).ListObjects(tName).Range.AutoFilter Field:=5
For i = 2 To 100000
    If Sheets(sDBname).Cells(i, 4) = "" Then
        Exit Sub
    ElseIf TextBoxFileName.Text = path & Sheets(sDBname).Cells(i, 3) Then
        Dim rws As String
        rws = CStr(i) & ":" & CStr(i)
        Sheets(sDBname).Activate
        Sheets(sDBname).Rows(rws).Select
        Selection.Delete Shift:=xlUp
        Sheets(Kekka).Activate
        Sheets(Kekka).Cells(EditNo, 6).Value = Deleted
        sortFlg = True
        Exit Sub
    End If
Next i

```

End Sub

#### ‘パス名取得関数

Function getPath(Kbn As String) As String

```

Dim i As Long

For i = 1 To 1000
    If Kbn = Sheets(Hozon).Cells(i, 1) Then
        getPath = Sheets(Hozon).Cells(i, 2) & "▼"
        Exit Function
    End If
Next i

```

End Function

#### ‘すべて更新処理

Private Sub ButtonUpdAll\_Click()

```

Dim NoFrom As Long
Dim NoTo As Long
Dim SaveSeason As String
Dim SaveComment As String
Dim i As Long
Dim updateCnt As Long
Dim deletedCnt As Long

```

updateCnt = 0

deletedCnt = 0

If dCount = 0 Then

rc = MsgBox("検索してからにしてください。", vbOKOnly, "関連エラー")

Exit Sub

End If

rc = MsgBox("行番号の From と To を入力し、変更したい [季節]、[一口コメント] を選択してください。", vbOKOnly, "メッセージ")

NoFrom = Val(TextBoxFrom.Text)

NoTo = Val(TextBoxTo.Text)

If NoFrom < 2 Then

rc = MsgBox("行番号の From が範囲外です。", vbOKOnly, "数値関連エラー")

Exit Sub

End If

If NoTo > dCount + 1 Then

rc = MsgBox("行番号の To が範囲外です。", vbOKOnly, "数値関連エラー")

Exit Sub

End If

If NoFrom > NoTo Then

rc = MsgBox("行番号の From > To です。", vbOKOnly, "数値関連エラー")

Exit Sub

End If

If ComboBoxSeason.ListIndex = -1 Then

rc = MsgBox("季節を選択してください。", vbOKOnly, "選択エラー")

Exit Sub

End If

If ComboBoxComment2.ListIndex = -1 Then

rc = MsgBox("一口コメントを選択してください。", vbOKOnly, "選択エラー")

Exit Sub

End If

If ComboBoxSeason.Text = "(元のまま)" And ComboBoxComment2.Text = "(元のまま)" Then

rc = MsgBox("季節も一口コメントも (元のまま) では更新できません。", vbOKOnly, "選択エラー")

Exit Sub

End If

SaveSeason = ComboBoxSeason.Text

SaveComment = ComboBoxComment2.Text

For i = NoFrom To NoTo

If Sheets(Kekka).Cells(i, 6) <> Deleted Then

updateCnt = updateCnt + 1

Call DataBaseEdit2(ComboBox1.Text, Sheets(Kekka).Cells(i, 4), SaveSeason, SaveComment, i)

Else

deletedCnt = deletedCnt + 1

End If

Next i

Call clearInputArea

rc = MsgBox("指定範囲のデータがすべて同一の値で更新されました。「季節を変更した場合」検索の前にソートされます。[" & CStr(updateCnt) & "]件更新、[" & CStr(deletedCnt) & "]件 以前に削除済み。", vbOKOnly, "メッセージ")

End Sub

## ‘ データベース編集 2

Sub DataBaseEdit2(Kbn As String, Fn As String, Season As String, Cmt As String, No As Long)

Dim i As Long

Dim path As String

Dim Cmmt As String

path = getPath(Kbn)

Sheets(sDBname).ListObjects(tName).Range.AutoFilter Field:=1

Sheets(sDBname).ListObjects(tName).Range.AutoFilter Field:=2

Sheets(sDBname).ListObjects(tName).Range.AutoFilter Field:=5

For i = 2 To 100000

If Sheets(sDBname).Cells(i, 4) = "" Then

Exit Sub

End If

If Fn = path & Sheets(sDBname).Cells(i, 3) Then

If Season <> "(元のまま)" Then

Sheets(sDBname).Cells(i, 2) = Season

Sheets(Kekka).Cells(No, 3) = Season

sortFlg = True

End If

If Cmt <> "(元のまま)" Then

If Cmt = Non Then

Cmmt = ""

Else

Cmmt = Cmt

End If

Sheets(sDBname).Cells(i, 5).Value = Cmmt

Sheets(Kekka).Cells(No, 6).Value = Cmmt

End If

Exit Sub

End If

Next i

End Sub

#### ‘ データベース取得処理

Private Sub ButtonGetDataBase\_Click()

rc = MsgBox("新規データベースを取得します。現在あるデータは置き換えられません。", vbOKCancel, "メッセージ")

If rc = vbCancel Then

Exit Sub

End If

Call getDataBase0

End Sub

#### ‘ データベース取得 実処理

Sub getDataBase0()

Call init

'データベース内容を取得する

Dim cnt As Long

cnt = 0

Dim buf As String, msg As String

Dim cb1 As String

rc = MsgBox("画像ファイルで対応しているのは [jpg] [png] [bmp] [gif] です。" & vbCrLf & vbCrLf & \_  
"ファイル名の最初の文字が '\_' のものは取得されません。", vbOKOnly, "メッセージ")

cnt = getTableRowsCount(cnt)

Dim j As Long

For j = 1 To 1000

cb1 = Sheets(Hozon).Cells(j, 1)

If cb1 <> "" Then

If Sheets(Hozon).Cells(j, 2) = "" Then

msg = "[" & cb1 & "]" のフォルダパス名を入力してください。" & vbCrLf & "なければ入力しなくて良いです。"

buf = InputBox(msg)

Sheets(Hozon).Cells(j, 2) = buf

Else

buf = Sheets(Hozon).Cells(j, 2)

End If

If buf <> "" Then



```
Call getDatabase(cb1, buf & "▼", cnt)
```

```
End If
```

```
Else
```

```
    j = 1000
```

```
End If
```

```
Next j
```

```
If cnt > 0 Then
```

```
    Call makeTable1(cnt)
```

```
    Call sortTable
```

```
Else
```

```
    rc = MsgBox("データ件数が0件でした。テーブルは作られていません。", vbOKOnly, "テーブル作成エラー")
```

```
    Sheets(Kekka).Activate
```

```
    Exit Sub
```

```
End If
```

```
Sheets(Kekka).Activate
```

```
rc = MsgBox("シート名 [" & sDBname & "] にテーブルが [" & tName & "] として作られました。", vbOKOnly, "メッセージ")
```

```
End Sub
```

‘ データベースの件数取得関数

```
Function getTableRowsCount(cnt As Long) As Long
```

```
On error GoTo ex
```

```
‘ Sheets(sDBname).Select
```

```
Sheets(sDBname).Activate
```

```
getTableRowsCount = cnt
```

```
If Sheets(sDBname).ListObjects.Count > 0 Then
```

```
    Sheets(sDBname).ListObjects(tName).Name = tName
```

```
    cnt = Sheets(sDBname).ListObjects(tName).ListRows.Count
```

```
    getTableRowsCount = cnt
```

```
    Sheets(sDBname).ListObjects(tName).TableStyle = ""
```

```
    Sheets(sDBname).ListObjects(tName).Unlist
```

```
    Rows("1:1").Select
```

```
    Selection.Delete Shift:=xlUp
```

```
End If
```

```
ex:
```

```
End Function
```

‘ データベース編集処理

```
Sub getDatabase(EorO As String, pathName As String, cnt As Long)
```

```
    Dim buf As String
```

```
Dim path As String
```

```
Dim i As Long
```

```
Dim setFlg As Boolean
```

```
setflg = False
```

```
path = pathName
```

```
buf = Dir(path & " *.*")
```

```
' cnt = 0
```

```
Do While buf <> ""
```

```
    setFlg = False
```

```
    If Right(buf, 3) = ".jpg" Or Right(buf, 3) = ".png" Or Right(buf, 3) = ".bmp" Or Right(buf, 3) = ".gif" Then
```

```
        If Left(buf, 1) <> "_" Then
```

```
            For i = 1 To cnt
```

```
                If Sheets(sDBname).Cells(i, 3) = buf Then
```

```
                    setFlg = True
```

```
                    i = cnt
```

```
                End If
```

```
            Next i
```

```
            If setFlg = False Then
```

```
                cnt = cnt + 1
```

```
                Sheets(sDBname).Cells(cnt, 1) = EorO
```

```
                Sheets(sDBname).Cells(cnt, 3) = buf
```

```
                Sheets(sDBname).Cells(cnt, 4) = getFileCreated(path & buf)
```

```
                Sheets(sDBname).Cells(cnt, 2) = Format(Month(Sheets(sDBname).Cells(cnt, 4)), "00 月")
```

```
                Sheets(sDBname).Cells(cnt, 5) = ""
```

```
                Sheets(sDBname).Cells(cnt, 6) = ""
```

```
                Sheets(sDBname).Cells(cnt, 7) = ""
```

```
            End If
```

```
        End If
```

```
    End If
```

```
    buf = Dir()
```

```
Loop
```

```
End Sub
```

‘ ファイル作成日取得関数

```
Function getFileCreated(fName) As String
```

```
Dim fso As FileSystemObject
```

```
Set fso = New FileSystemObject ' インスタンス化
```

```
On Error Resume Next
```

```
Dim f As File
```

```
Set f = fso.GetFile(fName) ' ファイルを取得
```

Dim d As Date

getFileCreated = f.DateCreated ' 撮影日時を取得

' 後始末

Set fso = Nothing

End Function

' テーブル作成処理

Sub makeTable1(i As Long)

' DataBase テーブル を作る

Dim rng1 As String

Dim rng2 As String

rng1 = "A1:G" & CStr(i)

rng2 = "\$A\$1:\$G\$" & CStr(i)

Range(rng1).Select

ActiveSheet.ListObjects.Add(xlSrcRange, Range(rng2), , xlNo).Name = \_

tName

End Sub

' テーブルソート処理

Sub sortTable()

Dim s As String

Sheets(sDBname).Activate

Sheets(sDBname).Range(tName).Select

With ActiveWorkbook.Sheets(sDBname).ListObjects(tName)

.Sort.SortFields.clear

s = tName & "[[#Headers],[列 1]]"

.Sort.SortFields.Add \_

Key:=Range(s), SortOn:=xlSortOnValues, Order:= \_

xlAscending, DataOption:=xlSortNormal

s = tName & "[[#Headers],[列 2]]"

.Sort.SortFields.Add \_

Key:=Range(s), SortOn:=xlSortOnValues, Order:= \_

xlAscending, DataOption:=xlSortNormal

s = tName & "[[#Headers],[列 4]]"

.Sort.SortFields.Add \_

```
Key:=Range(s), SortOn:=xlSortOnValues, Order:= _
```

```
xlAscending, DataOption:=xlSortNormal
```

```
End With
```

```
With ActiveWorkbook.Sheets(sDBName).ListObjects(tName).Sort
```

```
.Header = xlYes
```

```
.MatchCase = False
```

```
.Orientation = xlTopToBottom
```

```
.SortMethod = xlPinYin
```

```
.Apply
```

```
End With
```

```
Sheets(Kekka).Activate
```

```
' rc = MsgBox("テーブル [" & tName & "] を「検索対象」「月」「ファイル作成日」の順でソートしました。", vbOKOnly, "メッセージ")
```

```
End Sub
```

#### ‘ 終了処理

```
Private Sub ButtonExit_Click()
```

```
Call clearCell
```

```
Application.Quit
```

```
Workbooks("絵手紙・己書 選択画面.xlsm").Save
```

```
Workbooks("絵手紙・己書 選択画面.xlsm").Close
```

```
End
```

```
End Sub
```

#### ‘ 検索処理

```
Private Sub ButtonSearch_Click()
```

```
' 検索ボタンの処理
```

```
Dim i As Long
```

```
Dim s As String
```

```
Dim startTime As Date
```

```
Dim endtime As Date
```

```
Dim Err As Boolean
```

```
startTime = Now
```

```
errFlgSrch = False
```

```
If sortFlg = True Then
```

```
Call sortTable
```

```
sortFlg = False
```

```
End If
```

```
If Sheets(Kekka).Visible = False Then
```

```
Sheets(Kekka).Visible = True
```

```
End If
```

```
Call clearCell
```

```
If errFlgCB1 = True Then
```

```
    ComboBox1.SetFocus
```

```
    rc = MsgBox("検索対象(検索用)の内容を選択し直すか再入力してください。", vbOKOnly, "メッセージ")
```

```
    Exit Sub
```

```
ElseIf errFlgCB2 = True Then
```

```
    ComboBox2.SetFocus
```

```
    rc = MsgBox("季節(検索用)の内容を選択してください。", vbOKOnly, "メッセージ")
```

```
    Exit Sub
```

```
ElseIf errFlgCmnt = True Then
```

```
    ComboBoxComment.SetFocus
```

```
    rc = MsgBox("一ロコメント(検索用)の内容を選択し直すか再入力してください。", vbOKOnly, "メッセージ")
```

```
    Exit Sub
```

```
End If
```

```
TextBoxFrom.Text = ""
```

```
Call clearInputArea
```

```
updateNo = 0
```

```
EditNo = 0
```

```
DispNo = 0
```

```
If ComboBox1.Text = "" Then
```

```
    rc = MsgBox("検索対象を選んでください。", vbOKOnly, "選択エラー")
```

```
    GoTo exx
```

```
End If
```

```
For i = 1 To 1000
```

```
    If Sheets(Hozon).Cells(i, 1) = ComboBox1.Text Then
```

```
        s = Sheets(Hozon).Cells(i, 2)
```

```
        If s = "" Then
```

```
            Call setPath(s, i)
```

```
            If fName0 <> "" Then
```

```
                fName0 = Sheets(Hozon).Cells(i, 2)
```

```
                fName0 = fName0 & "▼"
```

```
                rc = MsgBox(ComboBox1.Text & "のデータがないので [" & fName0 & "] に追加します。よろしいですか?", vbOKCancel, "メッセージ")
```

```
            If rc = vbOK Then
```

```
                Call getDataBase0
```

```
                GoTo nextstep
```

```

Else
    GoTo nextstep
End If
End If
Else
    fName0 = Sheets(Hozon).Cells(i, 2) & "¥"
    If Sheets(Hozon).Cells(i, 3) <> "" Then
        fName00 = Sheets(Hozon).Cells(i, 3) & "¥"
    Else
        fName00 = fName0 & "¥"
    End If
    GoTo nextstep
End If
End If
Next i

```

nextstep:

```

If fName0 = "¥" Then
    rc = MsgBox("ファイルパス名がありません。検索対象を選択し直すか追加して下さい。", vbOKOnly, "選択エラー")
    GoTo exx
End If

```

```

Sheets(Kekka).Activate

```

```

Columns("A:A").Select
Selection.ColumnWidth = 11.88
Columns("B:B").Select
Selection.ColumnWidth = 11.88
Columns("C:C").Select
Selection.ColumnWidth = 16.5
Columns("D:D").Select
Selection.ColumnWidth = 67.25
Columns("E:E").Select
Selection.ColumnWidth = 17.28
Columns("F:F").Select
Selection.ColumnWidth = 17.28
Columns("G:G").Select
Selection.ColumnWidth = 14#
Columns("H:H").Select
Selection.ColumnWidth = 14#
Columns("I:I").Select
Selection.ColumnWidth = 12.38

```

```

If Sheets(Kekka).Visible = True Then

```

```
Sheets(Kekka).Visible = False
```

```
End If
```

```
Call searchData
```

```
If dCount > 0 Then
```

```
    Call editFirstRow
```

```
    Call editDataRow
```

```
Else
```

```
    Sheets(Kekka).Activate
```

```
    rc = MsgBox("該当データはありませんでした。", vbOKOnly, "メッセージ")
```

```
    errFlgSrch = True
```

```
End If
```

```
exx:
```

```
Sheets(sDBname).Activate
```

```
ActiveSheet.ListObjects(tName).Range.AutoFilter Field:=1
```

```
ActiveSheet.ListObjects(tName).Range.AutoFilter Field:=2
```

```
ActiveSheet.ListObjects(tName).Range.AutoFilter Field:=5
```

```
Sheets(Kekka).Activate
```

```
Columns("E:E").Select
```

```
Selection.NumberFormatLocal = "yyyy""""/""""mm""""/"" ""dd"
```

```
Sheets(Kekka).Visible = True
```

```
On Error GoTo noDataBase
```

```
If dCount > 1 And CheckBoxResize.Value = True Then
```

```
    Call smallPicSave
```

```
    CheckBoxResize.Value = False
```

```
End If
```

```
Dim rn As String
```

```
If dCount > 4 Then
```

```
    rn = "A" & CStr(dCount - 4) & ":A" & CStr(dCount - 4)
```

```
Else
```

```
    rn = "A1:A1"
```

```
End If
```

```
Range(rn).Select
```

```
With ActiveWindow
```

```
    .ScrollRow = ActiveCell.Row
```

```
End With
```

```
If errFlgSrch = False Then
```

```
    endtime = Now
```

```
rc = MsgBox("この処理に " & DateDiff("s", startTime, endTime) & " 秒かかりました。", vbOKOnly, "メッセージ")
```

```
End If
```

```
Exit Sub
```

```
noDataBase:
```

```
rc = MsgBox("データベースが作られていません。" & vbCrLf & vbCrLf & "先ずデータベースを作ってください。", vbOKOnly, "エラー")
```

```
End Sub
```

### ‘ 検索結果シートクリア処理

```
Sub clearCell()
```

```
‘ On Error Resume Next
```

```
Sheets(Kekka).Visible = True
```

```
Sheets(Kekka).Activate
```

```
ActiveSheet.Shapes.SelectAll
```

```
Selection.Delete
```

```
Sheets(Kekka).Cells.clear
```

```
dCount = 0
```

```
Range("A1").Select
```

```
End Sub
```

### ‘ データベースフィルター処理

```
Sub searchData()
```

```
Sheets(sDBname).Activate
```

```
On Error GoTo myerror
```

```
ActiveSheet.ListObjects(tName).Range.AutoFilter Field:=1
```

```
ActiveSheet.ListObjects(tName).Range.AutoFilter Field:=2
```

```
ActiveSheet.ListObjects(tName).Range.AutoFilter Field:=5
```

```
Dim jj As Long
```

```
Dim cb1 As String
```

```
For jj = 1 To 1000
```

```
cb1 = Sheets(Hozon).Cells(jj, 1)
```

```
If ComboBox1.Text = cb1 Then
```

```
ActiveSheet.ListObjects(tName).Range.AutoFilter Field:=1, Criteria1:=cb1
```

```
jj = 1000
```

```
End If
```

```
Next jj
```

```
If ComboBox2.Text = "すべて" Then
```

```
ActiveSheet.ListObjects(tName).Range.AutoFilter Field:=2
```

```
ElseIf ComboBox2.Text = Non Then
```

```
ActiveSheet.ListObjects(tName).Range.AutoFilter Field:=2, Criteria1:=""
```

```
Else
```



```
ActiveSheet.ListObjects(tName).Range.AutoFilter Field:=2, Criteria1:=ComboBox2.Text
```

```
End If
```

```
If ComboBoxComment.Text = "すべて" Then
```

```
    ActiveSheet.ListObjects(tName).Range.AutoFilter Field:=5
```

```
ElseIf ComboBoxComment.Text = Non Then
```

```
    ActiveSheet.ListObjects(tName).Range.AutoFilter Field:=5, Criteria1:=""
```

```
Else
```

```
    ActiveSheet.ListObjects(tName).Range.AutoFilter Field:=5, Criteria1:=ComboBoxComment.Text
```

```
End If
```

```
On Error GoTo myerror
```

```
If Cells(Rows.Count, 1).End(xlUp).Row = 1 Then
```

```
    dCount = 0
```

```
Else
```

```
    dCount = Range(Range("A2"), Cells(Rows.Count, 1).End(xlUp)) _  
        .SpecialCells(xlCellTypeVisible).Count
```

```
End If
```

```
If dCount > 0 Then
```

```
    Call Visible_Search(Kekka)
```

```
End If
```

```
myerror:
```

```
End Sub
```

### ‘ フィルター後表示された行のみ取得処理

```
Sub Visible_Search(sName As String)
```

```
    '変数定義
```

```
    Dim p As Long, n As Long                'カウンタ変数
```

```
    Dim r As Range, rr As Range, rs As Range 'Visible セルを取得する変数
```

```
    Dim smallPic As String
```

```
    p = 0
```

```
    n = 2
```

```
    'Range("A1", Range("A" & Rows.Count).End(xlUp)) : A1～データのある最下行
```

```
    'SpecialCells(xlCellTypeVisible) : 表示されているセルのみを対象とする
```

```
    Set r = Range("A1", Range("A" & Rows.Count).End(xlUp)).SpecialCells(xlCellTypeVisible)
```

```
    'r : A1～データの存在する最大行セルまでの範囲 (表示されているセルのみ)
```

```
    'rr : A1～データの存在する最大行セルまで、A1 から 1 つずつ格納される
```

```
    For Each rr In r
```

```
        'rs へ選択セルの値を格納
```

```
        For Each rs In rr.Areas
```

選択セルの行数を取得

```
p = (rs.Row)
```

```
If p > 1 Then
```

```
    'Visible 行のデータを出力
```

```
    Sheets(sName).Rows(n).RowHeight = 75
```

```
    Sheets(sName).Cells(n, 1).Value = CStr(n)
```

```
    Sheets(sName).Cells(n, 3).Value = Sheets(sDBname).Cells(p, 2)
```

```
    Sheets(sName).Cells(n, 4).Value = fName0 & Sheets(sDBname).Cells(p, 3)
```

```
    smallPic = fName00 & Sheets(sDBname).Cells(p, 3)
```

```
    If Dir(smallPic) = "" Then
```

```
        smallPic = fName0 & Sheets(sDBname).Cells(p, 3)
```

```
    End If
```

```
    Call dispImage(n, smallPic, Kekka)
```

```
    Sheets(sName).Cells(n, 5).Value = Sheets(sDBname).Cells(p, 4)
```

```
    Sheets(sName).Cells(n, 6).Value = Sheets(sDBname).Cells(p, 5)
```

```
    n = n + 1
```

```
End If
```

```
Next
```

```
Next
```

```
End Sub
```

## ‘ 画像表示処理

```
Sub dispImage(n As Long, myFileName As String, sName As String)
```

```
    Dim rng As String
```

```
    rng = "B" & CStr(n) & ":B" & CStr(n)
```

```
    Call imageTateYoko(myFileName, n, sName)
```

```
    On Error GoTo fileNotFound1
```

```
    If n = 2 Then
```

```
        Sheets(sName).Activate
```

```
    End If
```

```
    Range(rng).Select
```

--(1) 選択位置に画像ファイルを挿入し、変数 myShape に格納

```
Set myshape = Sheets(sName).Shapes.AddPicture(_
```

```
    fileName:=myFileName, _
```

```
    LinkToFile:=True, _
```

```
    SaveWithDocument:=False, _
```

```
    Left:=Selection.Left, _
```

```
    Top:=Selection.Top, _
```

```
    Width:=0, _
```

```
    Height:=0)
```

--(2) 挿入した画像に対してセルと同じ高さ・幅にする

With myshape

.ScaleHeight pScale, msoTrue

.ScaleWidth pScale, msoTrue

End With

' 画像の縦と横のサイズを調べる

If pWidth > pHeight Then

myshape.LockAspectRatio = msoTrue

myshape.Top = myshape.TopLeftCell.Top

myshape.Left = myshape.TopLeftCell.Left

myshape.Height = myshape.TopLeftCell.Height

myshape.Width = myshape.TopLeftCell.Width

Else

With myshape.TopLeftCell

myshape.LockAspectRatio = msoTrue

myshape.Top = .Top

myshape.Left = .Left

myshape.Width = .Width

myshape.Height = .Height

End With

End If

Exit Sub

fileNotFound1:

rc = MsgBox("ファイル " & myFileName & " がありません。", vbOKOnly, "ファイル名エラー dispImage")

End Sub

' 画像の縦横画素数の取得処理

Sub imageTateYoko(strfile As String, i As Long, sName As String)

Dim pic As Object

Dim sp As Shape

On Error GoTo fileNotFound2

Select Case Mid(strfile, InStrRev(strfile, "."))

Case ".bmp", ".ico", ".rle", ".wmf", ".emf", ".gif", ".jpg"

On Error GoTo fileseterr1

Set pic = LoadPicture(strfile)

pWidth = CLng(CDb1(pic.Width) \* 24 / 635)

pHeight = CLng(CDb1(pic.Height) \* 24 / 635)

Case Else

On Error GoTo fileseterr2

```
Set sp = ActiveSheet.Shapes.AddPicture(_  
    fileName:=strfile, _  
    LinkToFile:=False, _  
    SaveWithDocument:=True, _  
    Left:=0, _  
    Top:=0, _  
    Width:=0, _  
    Height:=0 _  
)
```

With sp

```
.LockAspectRatio = msoTrue  
.ScaleHeight 1, msoTrue  
.ScaleWidth 1, msoTrue  
pWidth = CLng(.Width * 4 / 3)  
pHeight = CLng(.Height * 4 / 3)  
.Delete
```

End With

End Select

nextxx:

```
Sheets(sName).Cells(i, 7).Value = str(pWidth)  
Sheets(sName).Cells(i, 8).Value = str(pHeight)
```

If pWidth > pHeight Then

```
Call setPscale(pWidth)
```

Else

```
Call setPscale(pHeight)
```

End If

```
Sheets(sName).Cells(i, 9).Value = str(pScale)
```

Exit Sub

fileNotFound2:

```
rc = MsgBox("ファイル " & strfile & " がありません。", vbOKOnly, "ファイル名エラー imageTateYoko")
```

Exit Sub

fileseterr1:

```
rc = MsgBox("ファイル " & strfile & " JPG、BMP、GIF セットエラー。", vbOKOnly, "画像貼り付けエラー imageTateYoko")
```

Exit Sub

fileseterr2:

```
rc = MsgBox("ファイル " & strfile & " PNG セットエラー。", vbOKOnly, "画像貼り付けエラー imageTateYoko")
```

Exit Sub

End Sub

‘ 画像の縮小率計算処理

Sub setPscale(x As Long)

Dim y As Double

y = 97 / x

pScale = y

End Sub

‘ 検索結果先頭行の編集処理

Sub editFirstRow()

Sheets(Kekka).Visible = True

Sheets(Kekka).Activate

Range("A1").Select

ActiveCell.FormulaR1C1 = "N o . "

ActiveCell.Characters(1, 4).PhoneticCharacters = "なんぼー"

Range("B1").Select

ActiveCell.FormulaR1C1 = "小画像"

ActiveCell.Characters(1, 3).PhoneticCharacters = "ショウガゾウ"

Range("C1").Select

ActiveCell.FormulaR1C1 = "月"

ActiveCell.Characters(1, 1).PhoneticCharacters = "ツキ"

Range("D1").Select

ActiveCell.FormulaR1C1 = "ファイル名"

ActiveCell.Characters(5, 1).PhoneticCharacters = "メイ"

Range("E1").Select

ActiveCell.FormulaR1C1 = "入力日"

ActiveCell.Characters(1, 2).PhoneticCharacters = "ニュウリョク"

ActiveCell.Characters(3, 1).PhoneticCharacters = "ヒ"

Range("F1").Select

ActiveCell.FormulaR1C1 = "一口コメント"

ActiveCell.Characters(1, 6).PhoneticCharacters = "ヒトクチコメント"

Range("G1").Select

ActiveCell.FormulaR1C1 = "画素数 横"

ActiveCell.Characters(1, 5).PhoneticCharacters = "ガソスウ ヨコ"

Range("H1").Select

ActiveCell.FormulaR1C1 = "画素数 縦"

ActiveCell.Characters(1, 5).PhoneticCharacters = "ガソスウ タテ"

Range("I1").Select

```
ActiveCell.FormulaR1C1 = "縮小率"
```

```
ActiveCell.Characters(1, 3).PhoneticCharacters = "シュクショウリツ"
```

```
Range("A1:I1").Select
```

```
Range("A1:I1").Select
```

```
With Selection
```

```
.HorizontalAlignment = xlCenter
```

```
.VerticalAlignment = xlCenter
```

```
.WrapText = False
```

```
.Orientation = 0
```

```
.AddIndent = False
```

```
.IndentLevel = 0
```

```
.ShrinkToFit = False
```

```
.ReadingOrder = xlContext
```

```
.MergeCells = False
```

```
End With
```

```
With Selection.Font
```

```
.Name = "MS Pゴシック"
```

```
.Size = 14
```

```
.Strikethrough = False
```

```
.Superscript = False
```

```
.Subscript = False
```

```
.OutlineFont = False
```

```
.Shadow = False
```

```
.Underline = xlUnderlineStyleNone
```

```
.ThemeColor = xlThemeColorLight1
```

```
.TintAndShade = 0
```

```
.ThemeFont = xlThemeFontMinor
```

```
End With
```

```
Selection.Font.Bold = True
```

```
End Sub
```

## ‘データ業の書式編集処理

```
Sub editDataRow()
```

```
Sheets(Kekka).Activate
```

```
Range("A2:I" & CStr(dCount + 1)).Select
```

```
With Selection
```

```
.HorizontalAlignment = xlCenter
```

```
.VerticalAlignment = xlCenter
```

```
.WrapText = False
```

```
.Orientation = 0
```

```
.AddIndent = False
```

```
.IndentLevel = 0
```

```
.ShrinkToFit = False
```

```
.ReadingOrder = xlContext
```

```
.MergeCells = False
```

```
End With
```

```
With Selection.Font
```

```
.Name = "MS Pゴシック"
```

```
.Size = 14
```

```
.Strikethrough = False
```

```
.Superscript = False
```

```
.Subscript = False
```

```
.OutlineFont = False
```

```
.Shadow = False
```

```
.Underline = xlUnderlineStyleNone
```

```
.ThemeColor = xlThemeColorLight1
```

```
.TintAndShade = 0
```

```
.ThemeFont = xlThemeFontMinor
```

```
End With
```

```
Selection.Font.Bold = True
```

```
End Sub
```

## ‘ 縮小画像保存処理

```
Sub smallPicSave()
```

```
‘ 画像をファイルとして保存する
```

```
Sheets(Kekka).Activate
```

```
Dim i As Long
```

```
Dim ext As String
```

```
Dim temp As Variant
```

```
Dim Fn As String
```

```
Dim tobj As Object
```

```
Dim TCht As Chart
```

```
Dim fileNM As String
```

```
i = 1
```

```
‘ワークシートの全オブジェクトをループ
```

```
For Each tobj In ActiveSheet.Shapes
```

```
    If tobj.Type = 11 Then 'オブジェクトが画像なら Type=11 となる。
```

```
        i = i + 1
```

```
        fName = Cells(i, 4) 'ファイル名を取得
```

```
        ‘ ファイル名変換処理（縮小版用に）
```

```
        temp = Split(fName, "¥")
```

```
        Fn = temp(UBound(temp))
```

```
' 拡張子を作る  
ext = Right(Fn, 3)  
ext = UCase(ext)
```

```
fileNM = getFilePath(ComboBox1.ListIndex) & Fn
```

```
If Dir(fileNM) = "" Then
```

```
    tobj.CopyPicture  
    ACWidth = tobj.Width 'オブジェクトのサイズを取得 (高さ)  
    ACHeight = tobj.Height 'オブジェクトのサイズを取得 (高さ)
```

```
'オブジェクトとほぼ同サイズの空のグラフを一時的に作る
```

```
Set TCht = ActiveSheet.ChartObjects.Add(0, 0, ACWidth, ACHeight).Chart
```

```
On Error GoTo pasteErr
```

```
TCht.Paste 'グラフに画像をペーストする。
```

```
'エクスポート先は縮小版のパス。ファイル名は縮小版ファイル名。
```

```
TCht.Export fileName:=getFilePath(ComboBox1.ListIndex) & Fn, filtername:=ext
```

```
TCht.Parent.Delete 'グラフを削除する。
```

```
GoTo pasteOK
```

```
pasteErr:
```

```
    rc = MsgBox(fName & "は縮小版が作れませんでした。", vbOKOnly, "縮小版作成エラー")
```

```
pasteOK:
```

```
    End If
```

```
End If
```

```
Next
```

```
Exit Sub
```

```
End Sub
```

### ' 縮小版ファイルのパス取得関数

```
Function getFilePath(kbnIndex As Long)
```

```
' サーチ区分ごとのパスの取得
```

```
Dim path As String
```

```
If Sheets(Hozon).Cells(ComboBox1.ListIndex + 1, 3) = "" Then
```

```
    Sheets(Hozon).Cells(ComboBox1.ListIndex + 1, 3) = InputBox(ComboBox1.Text & "の縮小版を入れるフォルダ名を入力して下さい。")
```

```
End If
```

```
If Sheets(Hozon).Cells(ComboBox1.ListIndex + 1, 3) = "" Then
```

```
    fName00 = Sheets(Hozon).Cells(ComboBox1.ListIndex + 1, 2) & "▼縮小版"
```

```
    rc = MsgBox("入力がないので [" & fName00 & "] とします。", vbOKOnly, "メッセージ")
```

```
    Sheets(Hozon).Cells(ComboBox1.ListIndex + 1, 3).Value = fName00
```

```
End If
```



```
If Dir(fName00, vbDirectory) = "" Then
```

```
    makePath (fName00)
```

```
End If
```

```
getFilePath = fName00 & "¥"
```

```
End Function
```

‘パスのフォルダを作成する処理

```
Sub makePath(path As String)
```

```
‘フォルダがなければ作る
```

```
‘path フォルダがなければ作成する
```

```
If Dir(path, vbDirectory) = "" Then
```

```
    Mkdir path
```

```
End If
```

```
End Sub
```

‘拡大表示処理

```
Private Sub ButtonDispLarge_Click()
```

```
Dim i As Long
```

```
On Error GoTo numerr
```

```
DispNo = 0
```

```
i = Val(TextBoxFrom.Text)
```

```
If i > 1 And i <= (dCount + 1) Then
```

```
    DispNo = i
```

```
    If Sheets(Kekka).Cells(DispNo, 6) = Deleted Then
```

```
        rc = MsgBox("これは既に削除されています。 No. =" & DispNo & " ファイル名=" & Sheets(Kekka).Cells(EditNo, 4), vbOKOnly, "削除済みエラー")
```

```
        Exit Sub
```

```
    End If
```

```
    Sheets(Kekka).Cells(DispNo, 1).Select
```

```
    Call OpenFile(Cells(DispNo, 4))
```

```
    Exit Sub
```

```
Else
```

```
    rc = MsgBox("番号が範囲外です。", vbOKOnly, "関連エラー")
```

```
    Exit Sub
```

```
End If
```

```
GoTo ok
```

```
numerr:
```

```
rc = MsgBox("番号が正しくありません", vbOKOnly, "数値エラー")
```

ok:

End Sub

### ‘ 画像ファイルを開く処理

Sub OpenFile(Fn As String)

On Error Resume Next

Dim sfn As String

sfn = Chr(34) & Fn & Chr(34)

With CreateObject("Wscript.Shell")

.Run sfn, 5

End With

End Sub

### ‘ 次を拡大表示処理

Private Sub ButtonDispNext\_Click()

If DispNo > 1 And DispNo < dCount + 1 Then

DispNo = DispNo + 1

If Sheets(Kekka).Cells(DispNo, 6) = Deleted Then

rc = MsgBox("これは既に削除されています。 No. =" & DispNo & " ファイル名=" & Sheets(Kekka).Cells(EditNo, 4), vbOKOnly, "削除済みエラー")

Exit Sub

End If

TextBoxFrom.Text = CStr(DispNo)

Sheets(Kekka).Cells(DispNo, 1).Select

Call OpenFile(Cells(DispNo, 4))

Exit Sub

ElseIf DispNo = 0 Then

rc = MsgBox("一度拡大表示してからにしてください。", vbOKOnly, "関連エラー")

Else

rc = MsgBox("番号が最大値を超えました。", vbOKOnly, "関連エラー")

End If

End Sub

### ‘ データ読込処理

Private Sub ButtonEditData\_Click()

Dim n As Long

n = Val(TextBoxFrom.Text)

On Error GoTo numerr1

```
If dCount = 0 Then
```

```
    rc = MsgBox("検索してからにしてください。", vbOKOnly, "関連エラー")
```

```
    Exit Sub
```

```
End If
```

```
If n > 1 And n <= (dCount + 1) Then
```

```
    Sheets(Kekka).Cells(n, 1).Select
```

```
    EditNo = n
```

```
    If Sheets(Kekka).Cells(EditNo, 6) = Deleted Then
```

```
        rc = MsgBox("これは既に削除されています。 No. =" & EditNo & " ファイル名=" & Sheets(Kekka).Cells(EditNo, 4), vbOKOnly, "削除済みエラー")
```

```
        Call clearInputArea
```

```
        Exit Sub
```

```
    End If
```

```
    ComboBoxSearchTarget.Enabled = False
```

```
    ComboBoxSearchTarget.Locked = True
```

```
    Call textBoxNotInput(TextBoxFileName)
```

```
    Call editingArea(n)
```

```
    UpdateFlg = True
```

```
Else
```

```
    rc = MsgBox("番号が範囲外です。", vbOKOnly, "関連エラー")
```

```
    Exit Sub
```

```
End If
```

```
GoTo ok1
```

```
numerr1:
```

```
    rc = MsgBox("番号が正しくありません", vbOKOnly, "数値エラー")
```

```
ok1:
```

```
End Sub
```

‘ テキストボックスをロックする処理

```
Sub textBoxNotInput(tb As MSForms.TextBox)
```

```
    tb.Enabled = False
```

```
    tb.Locked = True
```

```
End Sub
```

## ‘ 入力領域にデータを編集する処理

### Sub editingArea(n As Long)

```
TextBoxNo.Text = CStr(n)
ComboBoxSearchTarget.ListIndex = ComboBox1.ListIndex
ComboBoxSeason = Sheets(Kekka).Cells(n, 3)
TextBoxFileName = Sheets(Kekka).Cells(n, 4)
TextBoxFileCreated.Text = Sheets(Kekka).Cells(n, 5)
ComboBoxComment2 = Sheets(Kekka).Cells(n, 6)
```

End Sub

## ‘ 次を読み込み処理

### Private Sub ButtonEditNext\_Click()

```
If dCount = 0 Then
    rc = MsgBox("検索してからにしてください。", vbOKOnly, "関連エラー")
    Exit Sub
ElseIf EditNo = 0 Then
    rc = MsgBox("データ読込が未だです", vbOKOnly, "関連エラー")
    Exit Sub
ElseIf EditNo > dCount Then
    rc = MsgBox("番号が最大値を超えました。", vbOKOnly, "関連エラー")
    Exit Sub
ElseIf Sheets(Kekka).Cells(EditNo + 1, 6) = Deleted Then
    rc = MsgBox("これは既に削除されています。 No. =" & EditNo & " ファイル名=" & Sheets(Kekka).Cells(EditNo + 1, 4), vbOKOnly, "削除済みエラー")
    Exit Sub
End If
EditNo = EditNo + 1
TextBoxFrom.Text = CStr(EditNo)
Sheets(Kekka).Cells(EditNo, 1).Activate
ComboBoxSearchTarget.Enabled = False
ComboBoxSearchTarget.Locked = True
Call textBoxNotInput(TextBoxFileName)
Call editingArea(EditNo)
UpdateFlg = True
```

End Sub

## ‘ 更新処理

### Private Sub ButtonUpdate\_Click()

```
If UpdateFlg = True Then
    If Sheets(Kekka).Cells(EditNo, 6) = Deleted Then
        rc = MsgBox("これは既に削除されています。 No. =" & EditNo & " ファイル名=" & Sheets(Kekka).Cells(EditNo, 4), vbOKOnly, "削除済みエラー")
        Call clearInputArea
```

Exit Sub

End If

Call DataBaseEdit(TextBoxFileName.Text)

ComboBoxSearchTarget.Enabled = True

ComboBoxSearchTarget.Locked = False

Call textBoxInput(TextBoxFileName)

Call clearInputArea

rc = MsgBox("データが更新されました。「季節を変更した場合」検索の前にソートされます。", vbOKOnly, "メッセージ")

Else

rc = MsgBox("データ読込が未だです。", vbOKOnly, "関連エラー")

End If

End Sub

### ‘ データベースを編集する処理

Sub DataBaseEdit(Fn As String)

Dim i As Long

Dim path As String

path = getPath(ComboBox1.Text)

Sheets(sDBname).ListObjects(tName).Range.AutoFilter Field:=1

Sheets(sDBname).ListObjects(tName).Range.AutoFilter Field:=2

Sheets(sDBname).ListObjects(tName).Range.AutoFilter Field:=5

For i = 2 To 100000

If Sheets(sDBname).Cells(i, 4) = "" Then

Exit Sub

ElseIf TextBoxFileName = path & Sheets(sDBname).Cells(i, 3) Then

updateNo = i

If ComboBoxSeason.Text <> "(元のまま)" Then

Sheets(sDBname).Cells(i, 2) = ComboBoxSeason.Text

Sheets(Kekka).Cells(EditNo, 3) = ComboBoxSeason.Text

sortFlg = True

End If

If ComboBoxComment2.Text <> "(元のまま)" Then

Dim Cmnt As String

If ComboBoxComment2.Text = Non Then

Cmnt = ""

Else

Cmnt = ComboBoxComment2.Text

End If

Sheets(sDBname).Cells(i, 5).Value = Cmnt

Sheets(Kekka).Cells(EditNo, 6).Value = Cmnt

```
End If
```

```
Exit Sub
```

```
End If
```

```
Next i
```

```
End Sub
```

#### ‘ テキストボックスのロックを解く処理

```
Sub textBoxInput(tb As MSForms.TextBox)
```

```
tb.Enabled = True
```

```
tb.Locked = False
```

```
End Sub
```

#### ‘ ヘルプボタンの処理

```
Private Sub ButtonReadMe_Click()
```

```
Sheets(ReadMe).Activate
```

```
Sheets(ReadMe).Visible = True
```

```
rc = MsgBox("これをご覧ください。", vbOKOnly, "メッセージ ここをドラッグすると移動できます。")
```

```
Sheets(ReadMe).Visible = False
```

```
Sheets(Kekka).Activate
```

```
End Sub
```

#### ‘ 検索対象(検索用)追加処理

```
Private Sub ComboBox1_Exit(ByVal Cancel As MSForms.ReturnBoolean)
```

```
If ComboBox1.Text = "" Then
```

```
errFlgCB1 = True
```

```
Else
```

```
For i = 1 To 1000
```

```
If Sheets(Hozon).Cells(i, 1) = ComboBox1.Text Then
```

```
errFlgCB1 = False
```

```
Exit Sub
```

```
ElseIf Sheets(Hozon).Cells(i, 1) = "" Then
```

```
rc = MsgBox("検索対象(検索用)の内容が新しいので追加します。よろしいですか？ [" & ComboBox1.Text & "]", vbOKCancel, "メッセージ")
```

```
If rc = vbOK Then
```

```
Sheets(Hozon).Cells(i, 1) = ComboBox1.Text
```

```
ComboBox1.AddItem ComboBox1.Text
```

```
ComboBox1.ListIndex = ComboBox1.ListCount - 1
```

```
Dim idx As Long
```

```
idx = ComboBoxSearchTarget.ListIndex
```

```
ComboBoxSearchTarget.AddItem ComboBox1.Text
```

```
ComboBoxSearchTarget.ListIndex = idx
```

```
errFlgCB1 = False
```

```

Else
    ComboBox1.ListIndex = -1
    ComboBox1.Text = ""
    errFlgCB1 = True
End If
Exit Sub
End If
Next i
End If
End Sub

```

‘ 季節(検索用)追加処理

```
Private Sub ComboBox2_Exit(ByVal Cancel As MSForms.ReturnBoolean)
```

```

    If ComboBox2.Text = "" Then
        errFlgCB2 = True
    Else
        errFlgCB2 = False
    End If

```

```
End Sub
```

‘ 季節(データ追加・変更用)追加処理

```
Private Sub ComboBoxSeason_Exit(ByVal Cancel As MSForms.ReturnBoolean)
```

```

    If ComboBoxSeason.Text = "" Then
        errFlgSsn = True
    Else
        errFlgSsn = False
    End If

```

```
End Sub
```

‘ 検索対象(データ追加・変更用)追加処理

```
Private Sub ComboBoxSearchTarget_Exit(ByVal Cancel As MSForms.ReturnBoolean)
```

```

    If ComboBoxSearchTarget.Text = "" Then
        errFlgST = True
    Else

```

```
        For i = 1 To 1000
```

```

            If Sheets(Hozon).Cells(i, 1) = ComboBoxSearchTarget.Text Then
                errFlgST = False
                Exit Sub
            End If

```

```
        ElseIf Sheets(Hozon).Cells(i, 1) = "" Then
```

```

            rc = MsgBox("検索対象(データ追加用)の内容が新しいので追加します。よろしいですか？ [" & newComboboxSearchTarge & "]",
vbOKCancel, "メッセージ")

```

```
            If rc = vbOK Then
```

```
                Sheets(Hozon).Cells(i, 1) = ComboBoxSearchTarget.Text
```

```
                ComboBoxSearchTarget.AddItem ComboBoxSearchTarget.Text
```

```
                ComboBoxSearchTarget.ListIndex = ComboBoxSearchTarget.ListCount - 1
            End If
        End For
    End If
End Sub

```

```
Dim idx As Long
idx = ComboBox1.ListIndex
ComboBox1.AddItem ComboBoxSearchTarget.Text
ComboBox1.ListIndex = idx
```

```
errFlgST = False
```

```
Else
```

```
ComboBoxSearchTarget.ListIndex = -1
```

```
ComboBoxSearchTarget.Text = ""
```

```
errFlgST = True
```

```
End If
```

```
Exit Sub
```

```
End If
```

```
Next i
```

```
End If
```

```
End Sub
```

‘ 一ロコメント(検索用)追加処理

Private Sub ComboBoxComment\_Exit(ByVal Cancel As MSForms.ReturnBoolean)

```
If ComboBoxComment.Text = "" Then
```

```
errFlgCmnt = True
```

```
Else
```

```
For i = 1 To 1000
```

```
If ComboBoxComment.Text = Sheets(Hozon).Cells(1, 5) Or ComboBoxComment.Text = Sheets(Hozon).Cells(4, 5) Then
```

```
Exit Sub
```

```
End If
```

```
If ComboBoxComment.Text = Sheets(Hozon).Cells(2, 5) Or ComboBoxComment.Text = Sheets(Hozon).Cells(3, 5) Then
```

```
rc = MsgBox("一ロコメント(検索用)の内容にこれは使用できません [" & ComboBoxComment.Text & "]", vbOKOnly, "メッセージ")
```

```
ComboBoxComment.ListIndex = -1
```

```
ComboBoxComment.Text = ""
```

```
errFlgCmnt = False
```

```
Exit Sub
```

```
End If
```

```
If ComboBoxComment.Text = Sheets(Hozon).Cells(i, 5) Then
```

```
Exit Sub
```

```
ElseIf Sheets(Hozon).Cells(i, 5) = "" Then
```

```
rc = MsgBox("一ロコメント(検索用)の内容が新しいので追加します。よろしいですか? [" & ComboBoxComment.Text & "]", vbOKCancel,
```

```
"メッセージ")
```

```
If rc = vbOK Then
```

```
Sheets(Hozon).Cells(i, 5) = ComboBoxComment.Text
```

```
ComboBoxComment.AddItem ComboBoxComment.Text
```

```
ComboBoxComment.ListIndex = ComboBoxComment.ListCount - 1
```

```
errFlgCmnt = False
```

```
Else
```

```
ComboBoxComment.ListIndex = -1
```



```

        ComboBoxComment.Text = ""
        errFlgCmnt = True
    End If
    Exit Sub
End If
Next i
End If
End Sub

```

#### ‘ 一ロコメント(データ追加・変更用)追加処理

```
Private Sub ComboBoxComment2_Exit(ByVal Cancel As MSForms.ReturnBoolean)
```

```

    If ComboBoxComment2.Text = "" Then
        errFlgCmnt2 = True
    Else
        For i = 1 To 1000
            If ComboBoxComment2.Text = Sheets(Hozon).Cells(2, 5) Or ComboBoxComment2.Text = Sheets(Hozon).Cells(4, 5) Then
                Exit Sub
            End If
            If ComboBoxComment2.Text = Sheets(Hozon).Cells(1, 5) Or ComboBoxComment2.Text = Sheets(Hozon).Cells(3, 5) Then
                rc = MsgBox("一ロコメント(検索用)の内容にこれは使用できません [" & ComboBoxComment2.Text & "]", vbOKOnly, "メッセージ")
                ComboBoxComment2.ListIndex = -1
                ComboBoxComment2.Text = ""
                errFlgCmnt2 = False
                Exit Sub
            End If
            If ComboBoxComment2.Text = Sheets(Hozon).Cells(i, 5) Then
                Exit Sub
            ElseIf Sheets(Hozon).Cells(i, 5) = "" Then
                rc = MsgBox("一ロコメント(データ追加・変更用)の内容が新しいので追加します。よろしいですか? [" & ComboBoxComment2.Text & "]",
vbOKCancel, "メッセージ")
                If rc = vbOK Then
                    Sheets(Hozon).Cells(i, 5) = ComboBoxComment2.Text
                    ComboBoxComment2.AddItem ComboBoxComment2.Text
                    ComboBoxComment2.ListIndex = ComboBoxComment2.ListCount - 1
                    errFlgCmnt2 = False
                Else
                    ComboBoxComment2.ListIndex = -1
                    ComboBoxComment2.Text = ""
                    errFlgCmnt2 = True
                End If
            End If
        Next i
    End If
End Sub

```

## ブックの閉じるボタン処理

```
Private Sub UserForm_QueryClose(Cancel As Integer, CloseMode As Integer)
```

```
    If CloseMode = vbFormControlMenu Then
```

```
        rc = MsgBox(" [終 了] ボタンを使用してください", vbOKOnly, "メッセージ")
```

```
        Cancel = True
```

```
    End If
```

```
End Sub
```