

第4章

データベース的マクロ

実務では何千、何万という大量のデータから検索して処理を行うなど Excel をデータベース的な使い方を使用するケースが多々あります。

データが何千件あっても処理速度が極端に遅くなることはありません。

▶ データベース処理

■ VBA の処理速度

課題

使用ファイル：「db.xlsx」ワークシート：「Sheet1」保存ファイル名：「db.xlsm」
1セルずつデータを検索して、担当者名「佐藤」を MsgBox で表示します。
確認のため、Timer 関数を使用して大まかなプログラムの処理時間を計測します。
(プロシージャ名：test1)

```
Sub test1()  
    Dim i As Long, Start As Single, Finish As Single  
    Start = Timer  
    With Worksheets("Sheet1")  
        For i = 2 To .Cells(Rows.Count,1).End(xlUp).Row  
            If .Cells(i, 3) = "佐藤" Then  
                MsgBox .Cells(i, 3)  
            End If  
        Next  
    End With  
    Finish = Timer  
    MsgBox "今の処理に約" & (Finish - Start) & " 秒かかりました。"  
End Sub
```

課題

使用ファイル：「db.xlsm」ワークシート：「Sheet1」
データを検索して該当するデータを1回ずつ「Sheet2」シートへコピー&
ペーストを行う処理を実行します。
(プロシージャ名：test2)

```
Sub test2()  
    Dim i As Long, Start As Single, Finish As Single  
    Start = Timer  
    With Worksheets("Sheet1")  
        For i = 2 To .Cells(Rows.Count,1).End(xlUp).Row  
            If .Cells(i, 3) = "岡野" Then  
                .Range(.Cells(i,1),.Cells(i,5)).Copy _  
                Worksheets("Sheet2"). _  
                Cells(Rows.Count,1).End(xlUp).Offset(1)  
            End If  
        Next  
    End With  
    Finish = Timer  
    MsgBox "今の処理に約" & (Finish - Start) & " 秒かかりました。"  
End Sub
```

Point

▶ Timer関数

午前0時から何秒経っているかを表わす関数でSingle型の値を返します。

ここでは、プログラム実行前と実行後の差を求めてプログラムにかかった時間を計算しています。

▶ 条件に一致するセルを検索…Find/FindNext/FindPrevious メソッド

■ 条件に一致する最初のセルを返す…Find メソッド

セル範囲の中から指定した条件に一致するセルを検索します。条件に一致するセルが複数ある場合は、見つかったセルの次のセルを検索するのに「FindNext メソッド」、前のセルを検索するのに「FindPrevious メソッド」を使用します。条件に一致するセルが存在しない場合には、「Nothing」を返します。

構文：

`Object.Find(What, After, LookIn, LookAt, SearchOrder, SearchDirection, _
MatchCase, MatchByte, SearchFormat)`

`Object.FindNext(After)`

`Object.FindPrevious(After)`

設定項目	内容
What	検索する文字列を指定
After [省略可能]	単一のセルを指定。この引数に指定したセルの次のセルから検索を開始する。省略すると Object で指定したセル範囲の左上端セルの次のセルから検索する
LookIn [省略可能]	検索の対象を指定。数式 (xlFormulas : 既定値)、値 (xlValues)、コメント (xlComments)
LookAt [省略可能]	完全に一致するセルを検索 (xlWhole)、一部分が一致するセルを検索 (xlPart : 既定値)
SearchOrder [省略可能]	検索方向を指定する。列方向に検索する (xlByColumns)、行方向に検索する (xlByRows : 既定値)
SearchDirection [省略可能]	前方に検索 (xlNext : 既定値)、後方に検索 (xlPrevious)
MatchCase [省略可能]	大文字と小文字を区別する (True)、区別しない (False : 既定値)
MatchByte [省略可能]	半角と全角を区別する (True : 既定値)、区別しない (False)
SearchFormat [省略可能]	書式を検索する (True)、検索しない (False) (※) Excel 2002 以降で使用可

課題

使用ファイル：「db.xlsm」ワークシート：「Sheet1」

InputBox に伝票番号を入力し、検索された伝票番号の文字色を赤にします。
(プロシージャ名：test3)

```
Sub test3()
    Dim buf As String
    Worksheets("Sheet1").Activate
    buf = InputBox(" 伝票番号を入力してください ", " 検索 ")
    Range("A:A").Find(What:=buf, LookAt:=xlWhole) _
        .Font.Color = 255
End Sub
```

▶ 実行結果

	A	B	C	D	E
1	伝票番号	日付	担当者	商品番号	数量
2	20548	1月3日	小倉	B-005	383
3	20567	1月3日	阿部	C-002	314

Point

▶ Findメソッド

セル範囲内で特定の情報を検索する、Rangeオブジェクトのメソッドで、ワークシート操作の「検索と置換」の「検索」の機能をVBAで使うものです。

Point

▶ Findメソッドの引数

多数ありますが、[検索と置換]ダイアログボックスでセルを検索するときに指定する内容とほぼ同じです。

LookIn、LookAt、SearchOrder、MatchByte の設定は、このメソッドを使用するたびに保存されます。つまり、一度実行後 [検索と置換] ダイアログボックスを表示すると指定した引数が設定されています。そのため、このメソッドを使用するときにこれらの引数の指定を省略すると、以前保存された設定が使用されます。このような設定の変更によって生じる問題 evitar には、メソッドを使用するたびに、引数を明示します。