Kiến thức lập trình trong Excel

**Tìm hiểu các phần tử tập hợp**

Nếu các bạn chưa có khái niệm về toán cao cấp thì khi đọc những trang này cần liên hệ mật thiết với các vật thể tồn tại quanh ta. Từ đó sẽ giúp chúng ta cùng nhau hiểu các phần tử trong excel dễ hơn.  
  
**1./ Phần tử là gì?**  
  
*a./ Phần tử thuộc về tập hợp:* Trong thế giới quanh ta, thì thành phố, cơ quan, công sở, trường học, phân xưỡng, công nhân viên. . . đều là các phần tử. Phần tử có thể chứa các loại phần tử khác. Tỷ như thành phố có nhiều trường & trong trường có nhều lớp học, trong lớp lại có nhiều học sinh.  
  
Mỗi nhân viên là một phần tử thuộc về một loại phần tử khác: công sở (cơ quan). Để xác định một nhân viên, bạn có thể chỉ ra tên ( anh Trần An), chỉ vị trí (đồng nghiệp ngồi góc phải cuối phòng), chỉ tính chất (bạn mặc áo xanh & đội nón màu cam). Tuy nhiên tập hợp tất cả các nhân viên cũng là một phần tử.  
  
Ta mở chương trình excel 2003, để xuất hiện một bảng tính trắng duy nhất. Nói là duy nhất, vì nếu bạn còn bảng tính nào khác thì nên đóng nó lại (dùng lệnh Unhide trong menu window để áp dụng cho cả bảng tính ẩn)  
  
Trong excel cũng vậy, mỗi bảng tính là một phần tử. Bạn có thể chỉ ra bảng tính bằng tên (mở BaiTap.XLS), bằng vị trí (kích hoạt bảng tính thứ ba trong DS (danh sách)), chỉ trực tiếp (khi lưu bảng tính hiện hành). . . .tập hợp các bảng tính cũng là một phần tử thuộc loại khác, ta tạm gọi là phần tử ‘các bảng tính’.  
  
(ũng đúng như vậy đối với các trang tính, các vùng, các ô trong trang tính. . . .  
  
*b./ Phần tử có tính chất:* Hãy nhớ lại hình ảnh của bạn trước đây một con giáp: một cô/cậu bé cao khoảng 1,55m, bận áo trắng, ngồi giữa dẫy trái của lớp học, tóc không dài & màu đen, mắt không mở. . .  
  
Chiều cao, màu áo, vị trí, màu & độ dài tóc, trạng thái mắt. . . là các tính chất của phần tử. Cô bạn kế bên cũng có các tính chất đó, vì cô ấy cũng là phần tử thuộc loại học sinh. Chiếc ghế cô ta ngồi cũng có tính chất chiều cao, vị trí, cân nặng, nhưng không có tính chất về độ dài tóc, nhưng ngược lại, nó có tính chất nơi sản xuất mà bạn & cô kế bên không có. Như vậy cái ghế là phần tử thuộc loại khác, và các phần tử thuộc các loại khác nhau không có tính chất giống hệt nhau.  
  
Tập hợp ‘các học sinh trong lớp’ học cũng là một phần tử. Nó không có tính chất về chiều cao, cân nặng. . . , nhưng nó có tính chất khác: Tổng số, số nam, số nữ, số có mặt. . . . Như vậy, phần tử ‘các học sinh’ thuộc loại khác với phần tử học sinh vì chúng có tính chất khác nhau.  
  
Một số tính chất có thể thay đổi được. Bạn thay đổi trạng thái tình cảm khi sếp khen/chê trước tập thể, màu tóc của bạn sau một con giáp nữa sẻ không còn đen. . . Nhưng cũng có những tính chất là bất di bất dịch: Ngày sinh (trong phạm vi hoạt động bình thường của tập hợp ‘lớp’ hay ‘trường’ điều này là không khả thi).  
  
(ác phần tử trong excel cũng có tính chất. Mỗi bảng tính đều có ngày sinh, ngày hiệu chỉnh lần cuối, do một vĩ nhân, hay do một tên ngốc nào đó tình cờ lập ra . mỗi hàng, mỗi cột, đều có chiều cao & độ rộng. Mỗi menu đều có tiêu đề, người ta liệt kê số trang cho mỗi trang tính; một số tính chất rất dễ bị tổn thương như tên của trang tính, nhưng một số khác lại khó hơn: thay đổi tổng số ô trong một trang tính.  
  
*c./ Phần tử có phương pháp:* Mỗi nhân viên đều có thể đọc sách & đọc tài liệu, mỗi sinh viên đều có thể gõ & nhập hàm =DSUM() vô ô trang tính. Ở một thời điểm nào đó có thể nhân viên này đang đọc thì dồng sự khác đang viết thư điện tử. . .Nhưng quan trọng là vấn đề có khả năng đọc & có khả năng gởi thư điện tử. Những hành động đọc, viết nầy nọ đó ta gọi là phương pháp của phần tử đó (có nhiều người hay dùng phương thức thay vì phương pháp)  
  
Phương pháp làm thay đổi tính chất. Khi Lê Chanh thực hiện phương pháp đổ nước lên đầu mình, thì tính chất chiếc áo sẽ bị ướt.  
  
Một tập hợp các phần tử có các phương pháp khác với các phương pháp của từng thành phần của nó. Khi một nhân viên mới vô thử việc, thì phương pháp thêm vào là phương pháp của phần tử ‘cơ quan’ không thể bất cứ phần tử ‘nhân viên’ riêng rẽ nào thực hiện được phương pháp này.   
  
Một phương pháp quan trọng của phần tử tập hợp là chỉ ra một phần tử thuộc về nó. Khi giảng viên gọi: “Em Thanh, lên bảng!” Tác động gọi ra một phần tử trong một tập hợp & do đó tạo ra một liên hệ với phần tử đó là một phương pháp của các phần tử, chứ tuyệt nhiên không phải là phương pháp của phần tử ‘Em Thanh’ này.   
  
Ở đây là phải hiểu là giảng viên gọi với cả lớp, chứ không phải chỉ tác động lên phần tử ‘Thanh’. Vì một hiển nhiên là cả lớp đều nghe cô gọi & thấy kết quả của phương pháp gọi này. (Hơn nữa nếu chỉ có giảng viên & ‘Thanh’ thì chưa chắc giảng viên gọi như vậy!)  
  
Mỗi tập hợp các phần tử trong excel đều có phương pháp Add để thêm các phần tử cho mình. Trong trang tính có phương pháp Calculate để tính lại giá trị trong các ô. Đồ thị có phương pháp ChartWizard để thay đổi nhanh chóng các tính nết của đồ thị.  
  
Sự phân biệt giữa phương pháp & tính chất cũng rất mong manh. Khi bị Sếp mắng, nhân viên ‘Hoa’, ‘Hoa’ thực hiện phương thức trừng mắt (một hành động), hay cô ta gán một giá trị mới cho tính chất của mắt (tính dãn đồng tử). Cũng có lúc ta sẽ gặp khi một phương pháp trả về một giá trị.  
Hãy xem phương pháp Intersect sau:

Code:

**Function AntiRange(LRng As Range, NRng As Range) As Range**

Dim NewRng As Range, CurrCell As Range

For Each CurrCell In LRng.Cells

If Intersect(CurrCell, NRng) Is Nothing Then

If NewRng Is Nothing Then

Set NewRng = CurrCell

Else

Set NewRng = Union(NewRng, CurrCell)

End If

End If

Next

Set AntiRange = NewRng

Set NewRng = Nothing : Set CurrCell = Nothing

**End Function**

**2./ Tìm hiểu các bảng tính**  
  
a./ Thêm một bảng tính: Từ bảng tính đang mở, ta sẽ có ít nhất là hai cách để mở CS (cửa sổ) Microsoft VB (Visual Basic), như sau:  
  
• Phải chuột vô vùng trống trên thanh Toolbar & chọn từ DS đỗ xuống mục Visual Basic, khi thanh công cụ này hiện ra ta chọn biểu tượng Visual Basic Editor  
  
• Nhấn tổ hợp phím Atl + {F11}  
  
Từ CS này ta vô menu View -> Immediate để có thêm một CS ‘nghiệp vụ’ mới. Tiến hành thu nhỏ CS MVB (Microsoft Visual Basic) còn phân nữa để thấy được CS workbook bên dưới.  
  
Trong CS Immediate, ta gỏ nhập câu lệnh:

Code:

WorkBooks.Add

Và ấn ENTER. Bạn nên tìm đâu đó trên màn hình kết quả việc VBA thực hiện lệnh của bạn. Phương pháp Add đã thêm một phần tử bảng tính mới vố tập hợp các bảng tính của bạn.  
  
Như vậy là vùng Immediate này hữu dụng để chúng ta tìm hiểu về thành phần, phương thức . . . của VBA nói riêng & Excel nữa.  
  
Nếu thích bạn có thể lặp lại lệnh này, bằng cách giản tiện hơn, đó là ấn trỏ chuột vô dòng lệnh trên & ấn ENTER một lần nữa.   
  
b./ Đếm số bảng tính: Chúng ta vừa áp dụng phương pháp Add lên phần tử ‘các bảng tính’. Phần tử này cũng có những tính chất. Một trong những tính đó là đếm.   
  
Trong CS Immediate ta gỏ nhập lệnh:

Code:

?workbooks.count

Và thực hiện ấn ENTER xuống dòng. Tức thời tại dòng hiện hành xuất hiện con số chỉ ra số bảng tính có trong tập hợp

CS Immediate còn là nới thoải mái vọc của dân tập tễnh đến VBA, này nha:  
  
Bạn vừa tạo ra một kiệt tác UDF vừa ý ư, thì hãy đem đến đây thử các kết quả hàm trả về;  
  
Bạn muốn kiểm nghiệm các hàm trong VBA ư, nơi này cũng là nơi tốt, chẵng hạn:   
  
?1 + Date()

và ấn ENTER xem sao(?)  
  
Như vậy ta thấy, muốn đếm số bảng tính ta dùng phương thúc Count, nhưng phương thức này không làm tăng/giảm số bảng tính; muốn tăng ta dùng phương thức Add. Như vậy, tính chất Count thuộc loại tính chất chỉ đọc (read-only), không khác mấy với tính chất ngày sinh tháng đẻ của phần tử ‘nhân viên’ nào đó.  
  
c./ Đóng bảng tính:   
  
/(/goài phương pháp Add, phần tử WorkBooks có thêm phương pháp Close để đóng bảng tính. Gỏ vô CS Immediate lệnh

Code:

WorkBooks.Close

và nhấn ENTER, trả lời ‘No’ khi được hỏi điều gì đó.  
  
/(hi đó mọi bảng tính đều biến mất. Như vậy phương pháp Close tác động lên phần tử WorkBooks (tập hợp các bảng tính); Có nghĩa là nó đóng toàn bộ tập hợp này.  
  
Bạn thử thực hiện lại lệnh ?WorkBooks.Count xem sao (?)  
  
/(/hư vậy rõ một điều, tính chất Count đã bị phương pháp Add & Close thay đổi không thương tiếc.  
  
d./ Chỉ định một bảng tính: Phương pháp Close như trên đôi khi bất lợi, nếu khi ta chỉ cần đóng một hay một vài bảng tính cụ thể nào đó. Giống như giảng viên muốn gọi em Chi lên bảng, thì phải chỉ mặt đặt tên đó là phần tử nào trong tập hợp đang có.  
  
Đến đây bạn nên thực hiện không dưới 7 lần lệnh WorkBooks.Add để tạo nhiều bảng mới.  
  
Là CS soạn thảo nên trong Immediate ta có thể nhấn cùng lúc CTRL+{ENTER} để làm gì bạn thấy ngay đấy; Gỏ nhập vô dòng trống câu lệnh sau:

Code:

?WorkBooks.Item(1).Name

và ấn ENTER. Tại dòng đối thoại sẽ xuất hiện một cái tên mà bạn tự suy ngẫm.  
  
Theo như quyễn ghi chép “Chập chững đến VBA” ta đọc câu lệnh trên từ phải sang trái như sau:  
  
Tên của phần tử thứ nhất của ‘các phần tử’ bảng tính là gì(?)  
  
Từ ‘Name’ là tính chất của một phần tử bảng tính (WorkBook)  
  
Bạn xem thử phần tử WorkBooks có tính chất này không (?)  
  
Tiếp theo bạn gỏ

Code:

Workbooks.Item(2).close

và ấn ENTER.. Sẽ có một bảng tính nào đó bị biến khỏi tầm mắt.  
  
/(/ếu giờ bạn nhập lệnh

Code:

?WorkBooks.Item(2).Name

để xem hai tên được gọi có liền kề hay cách rời nhau không?(Khai báo cho cac muc lien ke nhau. VD: WorkBooks.Item(i).Name ‘nhu i chay tu 1-10)  
  
Item là một phương pháp mà mọi phần tử tập hợp đều chấp nhận. Phương pháp này đặt liên hệ đến một phần tử được xác định rõ của tập hợp đó. Có thể chỉ ra vị trí trong tập hợp đó (như các ví dụ trên), có thể chỉ bằng tên cụ thể ( sẽ chờ một chút tiếp sau)  
  
Trong các câu lệnh VB, bạn không thể dùng cách bỏ chữ ‘s’ để chỉ phần tử trong tập hợp (chẳng hạn bỏ chữ ‘s’ trong chữ WorkBooks để chì phần tử riêng rẽ nào của ‘các phần tử’ workbooks). Muốn chỉ một phần tử cụ thể nào ta phải dùng phương pháp Item; Như để đóng workbooks thứ 5 ta dùng lệnh WorkBooks.Item(5).Close (không thể dùng Workbook(5).Close)  
  
e./ Xác định một bảng tính bằng tên:   
  
Chúng ta vừa mới xác định một bảng tính bằng vị trí của nó trong một tập hợp (Là vị trí tuần tự phát sinh của nó). Một phần tử nhất thiết phải được xác định theo tên của nó trong một tập hợp. Cho rằng để tiếp tục bạn cần phải tạo vài ba bảng tính đang hiện hữu.   
  
Dùng thanh cuốn dời đến cuối CS Immediate, lựa tên 1 bảng tính ở giữa dãy bảng tính bạn đang có (VD là Book9). Gỏ dòng lệnh

Code:

WorkBooks.Item(“Book9”).Activate

và ấn ENTER. Dưới tác động của lệnh này Book9 sẽ được điều động lên nằm đầu dãy các bảng tính.  
  
Activate là một trong những phương pháp của một phần tử bảng tính. Và quan trọng ở đây là ta đã chỉ ra phần tử đó bằng tên của nó. Chúng ta thấy, khi dùng tên, phải để tên trong dấu ngoặt kép. Bạn hãy tự mình dùng câu lệnh cho biến phần tử này khỏi tầm mắt của chúng ta.  
  
Sau đó thi hành lệnh Activate nêu trên, thì Excel sẽ dận dỗi & báo lỗi có còn bảng tính ‘Book9’ nữa đâu mà kích hoạt nó!  
  
f./ Làm việc với bảng tính đang được kích hoạt:  
  
Giả sử cần thực hiện việc gì đó lên bảng tính đang kích hoạt (hiện hành) nên ta có thể tác động lên nó bằng những lệnh hơi khác hơn, ví dụ

Code:

ActiveWorkBook.Close

sẽ làm cho bảng tính này sẽ đi vô dĩ vãng.  
  
g./ Thay đổi giá trị tính chất của bảng tính:  
  
(ác tính chất Count & Name của phần tử workBooks & WorkBook tương ứng là các tính chất chỉ đọc. Các tính chất này bạn có thể xem được nhưng sẽ không cho bạn trực tiếp thay đổi giá trị của chúng.   
  
/(/hưng cũng có những tính chất thuộc loại đọc-ghi cho phép bạn thay đổi giá trị của chúng. Tính chất Saved của phần tử workbook là một tính chất như vậy. Trong CS Immediate ta tạo thêm tối thiểu 3 workbooks bằng cách ấn 3 lần dòng lệnh WorkBooks.Add  
  
Sau đó ta thêm các dòng lệnh sau:

Code:

WorkBooks.Item(1).Activate

Range(“B2: C4”).Value=”N2”

ActiveWorkBook.Close

Dòng lệnh thứ hai ta đã gán giá trị vô một số ô của bảng tính hiện hành;  
Sau khi ấn ENTER của dòng lệnh thứ ba, thì Excel sẽ hỏi ta có lưu những thay đổi vừa qua hay không? Câu trả lời chúng ta là không! (Nhưng nhớ rằng là chúng ta đã được hỏi thăm!)  
  
Nhưng để Excel không hỏi gì cho tốn hao thời gian & tiền của, thay vì câu lệnh 3 ngắn như trên ta thêm cho dài ra một chút. Và để đối chứng ta nhập & thực hiện lại các dòng lệnh như sau:

Code:

WorkBooks.Item(1).Activate

Range(“B1: B2”).Value=”L.T. Thơm, Kiến An N0 -17”

ActiveWorkBook.Close SaveChanges := False

Sau khi ấn ENTER, chúng ta có bị hỏi han nữa không vậy?!  
  
Một điều nên nhớ là nếu chúng ta đã cho macro làm gì đó trên bảng tính. Sau đó muốn đóng lại thì nên ấn định giá trị thuộc tính lưu cho excel hiểu việc mình phải làm, trách để Excel hỏi han lại lôi thôi. Vì trong nhiều trường hợp sau này chúng ta không biết mình đã làm gì để bị hỏi han!

**Bài 2 Phần tử Range**

Code:

**KHÁI NIỆM VỀ PHẦN TỬ RANGE**

Code:

Phần này dựa chủ yếu vô bài viết của chị HANDUNG107

tại :http://www.giaiphapexcel.com/forum/showthread.php?t=52

có chuyển đổi trình tự để theo mạch của chủ đề tập hợp & phần tử của tập hợp

Là một ô, một hàng, một cột, hoặc có thể là một vùng chọn của các ô, chứa một hay nhiều vùng ô kề nhau, hay 3-D range  
Cách thức sử dụng phần tử range  
Các tính chất (thuộc tính) và phương thức (phương pháp) sẽ trả về phần tử range được liệt kê như sau đây:  
• Thuộc tính Range (Range property )  
• Thuộc tính Cells (Cells property )  
• Range and Cells   
• Thuộc tính Offset (Offset property )  
• Phương thức Union (Union method )  
Chúng ta sẽ xét đến từng thuộc tính & phương thức nêu trên cũng như một số thuộc tính có họ hàng với chúng.  
  
***Thuộc tính Range***  
  
Chúng ta sử dụng Range(arg), ở đây arg là tên của range, sẽ được trả về một đối tượng Range là một ô đơn (cell) hay vùng các ô.  
Ví dụ sau đây sẽ gán giá trị của ô B1 vô ô C5

Code:

Worksheets("Sheet1").Range("C5").Value = \_

Worksheets("Sheet1").Range("B1").Value

Ví dụ tiếp theo sẽ gán các số ngẫu nhiên từ 1 đến 9 vô vùng ô từ D1:F8.

Code:

Worksheets("Sheet1").Activate

Range("A1:H8").Formula = "= 1 + 8 \* Rand()" ‘Range is on the active sheet

Còn ví dụ thứ ba sau, sẽ xóa dữ liệu trong vùng có tên là Criteria.

Code:

Worksheets(1).Range("Criteria").ClearContents

Nếu ta sử dụng đối số dạng text trong địa chỉ vùng, chúng ta chỉ được dùng loại kí hiệu A1 (chúng ta không thể dùng loại R1C1)  
  
***Thuộc tính Cells***  
  
Ta sử dụng thuộc tính Cells(row, column), ở đây row là chỉ số hàng & column là chỉ số cột, sẽ được trả về một ô đơn. Ví dụ ta gán giá trị 34 vô ô B2.

Code:

Worksheets(1).Cells(2, 2).Value = 34

Hay chứa tổng 1 vùng vô ô A2, như sau:.

Code:

ActiveSheet.Cells(2, 1).Formula = "=Sum(B1:B5)"

Ví dụ dưới đây sẽ lập bảng tiêu đề cột & tiêu đề hàng cho một trang tính. Qua ví dụ này, chúng ta cũng thấy được sự khác biệt nho nhỏ trong cách dùng thuộc tính Range & thuộc tính Cells  
.

PHP Code:

 Option Explicit   
Sub SetUpTable()   
 Dim TheYear As Byte, TheQuarter As Byte   
 Worksheets("Sheet1").Activate   
 For TheYear = 1 To 5   
    Cells(1, 1 + 2 \* TheYear).Value = 2000 + TheYear   
 Next TheYear   
 For TheQuarter = 1 To 4   
    Range("B" & TheQuarter + 2).Value = "Q" & TheQuarter   
 Next TheQuarter   
End Sub

Nếu ta sử dụng dạng thức expression.Cells(row, column) , mà ở đâu biến expression là một biến được trả về là một đối tượng Range, và row & column có mối liên hệ với góc trái nhất của Range, thì sẽ trả về một phần của Range. Ví dụ sau đây sẽ áp đặt công thức vô ô D5.

Code:

Worksheets(1).Range("C5:F9").Cells(1, 2).Formula = "=Rand()"

Phuong thuc nay dung de chon so thu tu o trong vung da chon

***Range and Cells***  
  
Nếu sử dụng Range(cell1, cell2), mà cell1 & cell2 là đối tượng Range, đặt trưng của các ô đầu & ô cuối, thì sẽ trả về là 1 đối tượng Range. Ví dụ sau đây sẽ tạo đường viền cho các ô B2:I9.

Code:

With Worksheets(1)

.Range(.Cells(2, 2), .Cells(9, 9)) \_

.Borders.LineStyle = xlThick

End With

Tiếp theo, chúng ta khảo sát đoạn mã lệnh sau. Yêu cầu sau khi khảo sát, bạn hãy cho biết biến Rng chấp nhận chưỗi nào trong các chuỗi sau:  
‘A1B5’; ‘A:A’; ‘1:1’; ‘B:X’; ‘Ax1:A1’; ‘5:9’; ‘$A4:b$7’; ‘4 4’

PHP Code:

Sub RangeFromInputbox()   
    Dim Rng As Range       
    Set Rng = Application.InputBox \_   
         (Prompt:="Select any range", Title:="Demo", Type:=8)   
    MsgBox Rng.Address   
End Sub

Thêm một ví dụ nữa, để thấy việc dùng các thuộc tính đang nêu:

Code:

Sub LastRowAndColumn()

Dim lRow As Long, iCol As Integer

lRow = Range("A65432").End(xlUp).Row

iCol = Cells(2, 255).End(xlToLeft).Column

MsgBox Cells(lRow, 1).Address, , Range(Chr(64 + iCol) & 2).Address

End Sub

(Chúng ta cho macro chạy trên một trang tính có dữ liệu để xác định tính đúng đắn của macro này!  
Có thể không thích xái thuộc tính range ta xài thuộc tính Cells, và ngược lại? Ta xét đến ví dụ sau:

PHP Code:

Sub AllLoop()

1 Dim Clls As Range   
 2   For Each Clls In Cells   
 3      If Clls.Value = "@" Then   
 4          Clls.Activate:                      Exit For   
 5       End If   
  6  Next Clls   
End Sub

Vậy bạn thử đổi chữ ‘Range’ trong dòng lệnh 1 thành ‘cells’ xem sao. Sau khi ta nhập xong, VBE cũng chuyễn chữ này thành ‘Cells’, Nhưng khi bấm chạy macro thì nhận được thông báo lỗi!   
Tương tự, nếu ta thay chữ ‘Cells’ của dòng lệnh 2 bằng chữ ‘range’. Thì cũng nhận được kết quả tồi tệ không kém,   
  
  
***Thuộc tính Offset***  
  
Sử dụng thuộc tính Offset(row, column), mà khi đó row & column là số hàng & số cột khác lệch của range được thuộc tính trả về so với địa chỉ nguyên thủy trước đó. Để rõ hơn ta xét ví dụ sau:

PHP Code:

Worksheets("Sheet1").Activate   
  'Can't select unless the sheet is active’   
Selection.Offset(3, 1).Range("A1").Select

Selection.Offset(3, 1).Range("A1").Select ‘cai nay chi chon tiep tu o da chon, enter lan thu 2 thay kqua

range("A1").Offset(3,1).Select ‘cai nay chon tu o a1

Chúng ta xét thêm một ví dụ nữa, sau đây:

Code:

Sub OffsetRanges()

MsgBox Range("A2:B3").Offset(2, 4).Address, , Range("A2:B3").Address

End Sub

Các bạn cho macro chạy & tự rút cho mình kết luận cần thiết.  
Cũng giống trong excel, chúng ta có thể không ghi 1 trong 2 biến của thuộc tính này. Trong VBA chúng ta có thể thấy các dòng lệnh sau

Code:

Set Rng = Range(“B” & Range(“B65432”).End(xlUp).Row).Offset(1)

Temp = Range(“C1”).Offset(,Range(“B65432”).End(xlUp).Row)

***Phương thức Union***  
  
Sử dụng phương thức Union(range1, range2, ...) để được trả về một hỗn hợp vùng — là vùng gộp chung giữa hai hay nhiều hơn những khối ô. Thí dụ sau đây sẽ tạo ra đối tượng xác định bỡi phương thức Union hai vùng A2:B3 and C3:D4, và sau đó chọn nó.

PHP Code:

Sub UnionMethod   
Dim r1 As Range, r2 As Range, myMultiAreaRange As Range   
Worksheets("sheet1").Activate   
Set r1 = Range("A1:B2"):        Set r2 = Range("C3:D4") Set myMultiAreaRange = Union(r1, r2) myMultiAreaRange.Select   
 End Sub

Nếu chúng ta cần làm việc với vùng chọn, mà vùng này bao gồm hơn 1 vùng riêng rẽ, tốt hơn ta dùng thuộc tính Areas. Về thuộc tính Areas xin chưa đi sâu trong bài viết này!  
Để hiểu rõ hơn về phương thức Union, xin các bạn tham khảo thêm bài viết tại:  
http://www.giaiphapexcel.com/forum/showthread.php?t=480 ;  
Có bạn trên diễn đàn chúng ta đặt câu hỏi:  
Có 2 vùng: A=Range("A1:G20"); B=Range("C3:D4")  
Làm sao xác định được vùng C=A-B (là vùng A mà loại bỏ các ô trong vùng B) bằng VBA ?  
Tôi xin giới thiệu một trong những cách đó là dùng phương thức UNION(), như sau:

PHP Code:

Option Explicit   
Sub Range11()   
 Dim Rng As Range, RngA As Range, RngB As Range, RngC As Range   
 Set RngA = Range("A1:E16")   
 Set RngB = Range("B7:C10")   
 For Each Rng In RngA   
    If Intersect(Rng, RngB) Is Nothing Then   
        If RngC Is Nothing Then   
            Set RngC = Rng   
        Else   
            Set RngC = Union(RngC, Rng)   
    End If:     End If   
 Next Rng   
 MsgBox RngC.Address   
End Sub

(Đoạn mã trên dùng để xét từng ô trong toàn bộ các ô của vùng A; Khi gặp 1 ô nào đó không thuộc vùng B thì gán vô biến vùng C bằng phương thức union(). Như vậy để hiểu thật kỹ cách thức trong khi thực thi C = A – B ta phải hiểu thêm phương thức INTERSECT(). Mong các bạn tự tham khảo thêm)  
  
***Thuộc tính Resize***  
  
Nếu trong cửa sồ (CS) VBE (bằng cách nhấn các phím Atl+{F11} hay từ CS Immediate ta nhập chữ Resize, bôi chọn toàn bộ và nhấn {F1}, phần trợ giúp của excel về đặt tính này như sau:   
  
Resizes the specified range. Returns a Range object that represents the resized range.  
expression.Resize(RowSize, ColumnSize)  
expression Required. An expression that returns a Range object to be resized.  
RowSize Optional Variant. The number of rows in the new range. If this argument is omitted, the number of rows in the range remains the same.  
ColumnSize Optional Variant. The number of columns in the new range. If this argument is omitted, the number of columns in the range remains the same.  
Example  
This example resizes the selection on Sheet1 to extend it by one row and one column.  
Worksheets("Sheet1").Activate  
numRows = Selection.Rows.Count  
numColumns = Selection.Columns.Count  
Selection.Resize(numRows + 1, numColumns + 1).Select  
  
This example assumes that you have a table on Sheet1 that has a header row. The example selects the table, without selecting the header row. The active cell must be somewhere in the table before you run the example.  
Set tbl = ActiveCell.CurrentRegion  
tbl.Offset(1, 0).Resize(tbl.Rows.Count - 1, \_  
tbl.Columns.Count).Select  
‘Ghi chu: thuoc tính CurrentRegion: chon mien hien hanh co du lieu,,, (chọn nguyên 1 hàng và cột trống để mở rộng giới hạn, có thể là vùng chọn dữ liệu đc phân từ hàng trống và cột trống)  
(Xin phép cho tôi miễn dịch phần vừa rồi, vì có khi dịch không tốt bằng nguyên bảng).   
  
Để hiểu rõ hơn về thuộc tính này, ta xét thêm vài ví dụ sau đây:  
Ta có macro (Mc) dùng để mở rọng vùng chọn, như sau

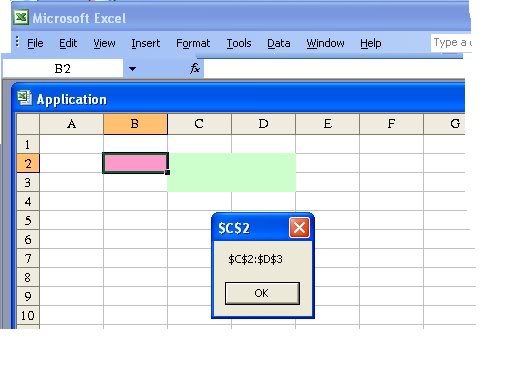
PHP Code:

Sub ExpandingArray()    
 Dim xRng As Range, TrRng As Range   
 Set xRng = Range("c2:d9"):           Set TrRng = Range("b4:c35")   
 Set xRng = xRng.Resize(xRng.Rows.Count, xRng.Columns.Count + 5)   
    MsgBox xRng.Address, , "A"           
 Set TrRng = TrRng.Resize(xRng.Rows.Count + TrRng.Rows. \_   
    Count, TrRng.Columns.Count)   
    MsgBox TrRng.Address, , "B" End Sub

Các bạn cho Mc chạy, khi đó lần lượt xuất hiện 2 hộp thoại, cho ta thông tin địa chỉ vùng đã được mở rộng dưới tác động của đặc tính Resize.  
Đến đây tôi thấy cần phân biệt giữa hai thuộc tính OFFSET() & RESIZE()  
Chúng ta xem xét đến macro sau:

PHP Code:

Sub OffsetAndResize()   
 Dim Rng As Range   
     Set Rng = Range(“B2”).Offset(, 1)   
     MsgBox Rng.Resize(2,2).Address,, Rng.Address       
End Sub

Sau khi ta đọc dữ liệu trong hộp thoại & đối chiếu với các chỉ số trong các thuộc tính, ta sẽ rạch ròi hơn xíu về cách thức của 2 thuộc tính trên.  
  
Thiết nghĩ cũng cần nêu lại vấn đề danh sách tiêu đề của vùng dữ liệu như ví dụ sau

PHP Code:

Sub TableDataHeaders() Dim rTable As Range   
    Set rTable = Sheet1.Range("A1").CurrentRegion       
    Set rTable = rTable.Resize(rTable.Rows.Count - 1)   
    MsgBox rTable.Address, , "1"   
    Set rTable = rTable.Offset(rTable.ListHeaderRows)   
    MsgBox rTable.Address, , "2" End Sub

Macro này ta đã gặp khi nghiên cứu về thuộc tính CurrentRegion. (bạn nào cần xen lại, có thể nhờ mục tìm kiếm của GPE)  
  
File đính kèm gồm hai bài trong loạt bài khái niệm đơn giản nhất về tập hợp   
 [ExcelVBA.rar](https://sites.google.com/site/chiasecongnghe86/ExcelVBA.rar)

**TÌM HIỂU VỀ PHẦN TỬ APPLICATION**

Trong các macro trước đây, chúng ta đã thấy các câu lệnh đề cập đến phần tử Application, ví dụ :

PHP Code:

1 Application.ScreenUpdating = False   
2 Vloop = Application.VLookup(List, Rng, 4, False)  3 Set myRange = Worksheets("Sheet1").Range("A1:C10")4 answer = Application.WorksheetFunction.Min(myRange) MsgBox answer

Phần từ Application chính là bản thân microsoft excel. Nó là bao trùm, là vũ trụ của mọi phần tử trong excel.  
  
Nếu chúng ta đếm các tính chất & phương pháp của phần tử này thì nhiều vô kể. Để tìm hiểu các tính chất & phương thức của nó, ta có tối thiểu 2 cách sau:  
  
**1./ Vô CS (cửa sổ) Object Browser**.  
  
**2./ Đơn giản hơn**, sau khi vô CS VBE, ta nhập từ “Application.”, sau khi ta nhập dấu chấm, sẽ có danh sách đổ xuống, ghi hầu hết các phương thức & tính chất của vũ trụ excel này.  
  
Bỡi lẽ do nhiều quá những phương thức & tính chất, nên có một số người đã phân loại chúng theo các nhóm sau :  
  
**1\* Các tính chất & phương thức toàn cục (global)** : Đó là trên 40 tinh chất & phương pháp dùng trực tiếp, không cần dùng từ Application ở trước. Chúng ta sẽ nói kỹ nhóm này ở phía dưới ;  
  
**2\* Các hàm của trang tính**; Nếu một ai chưa biết các hàm này, thì chưa thể gọi là đã biết về excel; Chúng ta xem lại các dòng lệnh 2 & 4 nêu trên. Tại dòng lệnh 2, biến Vloop sẽ chứa kết quả trả về của hàm VLOOPKUP(). Dòng lệnh 4, biến answer sẽ chứa giá trị thấp nhất trong mãng giá trị được chúng ta gán tên là myRange   
  
**3\* Các thiết chế cho môi trường làm việc**: Bạn có thể nhấn {F9} để tính toán lại bảng tính hiện hành

PHP Code:

5  Worksheets("Sheet1").UsedRange.Columns("A:C").Calculate   
6  Application.Calculation = xlCalculationManual

(Nếu ta cần diễn dịch dòng lệnh 5, thì nên đọc từ phải qua trái, như sau:   
  
Tính lại cột A đến cột C của vùng sử dụng trên trang tính có tên ‘Sheet1’  
Cách dễ tiếp thu hết các lệnh loại này là ta dùng bộ thu macro để biết một số hành động của chúng ta tác động lên trang tính được viết bỡi ngôn ngữ VBA là như thế nào  
  
**4\* Các thông tin về môi trường làm việc**. Excel có khoảng hơn 30 tính chất như vậy. Chúng cho ta biết các thông tin như Bạn đang làm việc trong windows hay không, bạn còn bao nhiêu bộ nhớ khả dụng. . .  
Ví dụ:

PHP Code:

MsgBox "Welcome to Microsoft Excel version " & \_    
    Application.Version & " running on " &  Application.OperatingSystem & "!"   
ActiveWorkbook.Windows(1).Caption = "Consolidated Balance Sheet" ActiveWorkbook.Windows("Consolidated Balance Sheet") \_   
    .ActiveSheet.Calculate

**5\* Các thiết định về macro**: Ví dụ như câu lệnh trên cùng của bài, dùng để không thay đổi màn hình khi macro chạy. Điều này rất cần thiết & không nên quên khi ta phải duyệt một số lượng lớn các record (Tìm mã mặt hàng, họ tên học sinh trong một cơ sở dữ liệu lớn. . .). Thêm thông báo ở dòng trạng thái trong macro sau

PHP Code:

Sub Statusbar()   
 Dim oldStatusBar, StrC As String   
 Dim iDem As Integer, iZ As Long, jW As Long   
    
 oldStatusBar = Application.DisplayStatusBar   
 Application.DisplayStatusBar = True   
 StrC = "\*\*\*\*\*\*\*\*\*\*\*\*"   
 For iZ = 1 To 60000   
    Cells(iZ, 1) = iZ   
    For jW = 1 To 10 ^ 4   
    Next jW   
    If iZ Mod 6000 = 0 Then   
        iDem = iDem + 1   
        Application.Statusbar = Left(Left(StrC, iDem) & "..........", 10)   
    End If   
 Next iZ   
    Application.Statusbar = False   
 Application.DisplayStatusBar = oldStatusBar   
End Sub

**6\* Những tính chất & phương thức khó xếp vô một nhóm nào**. Như đổi đơn vị đo lường, gởi thư điện từ, gọi hệ trợ giúp . . .  
Ví dụ:

PHP Code:

 Application.Help "OTISAPP.HLP", 65527

Trong cú pháp lệnh này, gồm 2 đối số tùy chọn; Đối số đầu là tên file trợ giúp; đối số sau là chỉ mục của file. Nếu ta không đưa ra chỉ mục thì hộp thoại trợ giúp của excel sẽ hiện ra!

PHP Code:

 Worksheets("Sheet1").PageSetup.LeftMargin = \_   
        Application.InchesToPoints(2.5) Worksheets("Sheet1").PageSetup.LeftMargin = \_   
        Application.CentimetersToPoints(5)

**CÁC TÍNH CHẤT & PHƯƠNG PHÁP TOÀN CỤC.**

Trong các câu lệnh macro, có những từ có thể đứng một mình, trong khi đó cũng có những từ cần phải đứng sau một từ khác chỉ rõ tập hợp bố mẹ của chúng; Ví dụ:

PHP Code:

MsgBox Cells.Item(9).Address, , Range("B1:C9").Item(3).Address

Item ‘ nghia tuần tuan tu

Trong ví dụ trên tập hợp cha mẹ, trong trường hợp thứ nhất là toàn bộ các ô trên trang tính; trong trường hợp thứ hai là 18 ô trên trang tính nào đó đang được kích hoạt  
  
Worksheets("Sheet1").Cells.ClearContents  
Câu lệnh này dùng để xóa toàn bộ nội dung các ô trong trang tính được nêu tên.  
  
Tương tự như trên, các từ Boder hay Interior, . . . cũng phải đi kèm với cha, mẹ cũa chúng; không.thể đứng riêng một mình được.  
Nhưng cũng có một số từ có thể đứng riêng, như Selection, WorkBooks, . . .   
  
Sao lại như vậy?  
  
Chúng ta phải liên tưởng đế các câu sau:  
“ Ông Chủ Nhà Trắng đã tuyên . . . .”  
“Ngày mai tôi sẽ bay từ Hà nội đến Pari. . .”  
Ở câu đầu, ta biết ngay, đó là tổng thống Mĩ; Ở câu hai, người nghe sẽ biết ngay là anh/chị này đã có hộ chiếu đến Pháp.   
Ta xét trường hợp Selection, như sau:

PHP Code:

Sub DongTrongXuongDuoi() 'Assumes the list has a heading'   
 On Error Resume Next   
    With Selection   
        .Sort Key1:=.Cells(2, 3), Order1:=xlAscending, \_   
            Header:=xlYes, OrderCustom:=1, MatchCase:=False, \_   
            Orientation:=xlTopToBottom   
    End With   
End Sub

Với trường hợp workbooks, ta có ví dụ sau:

PHP Code:

Sub workbooks\_() Dim Wh As Workbook   
For Each Wh In Workbooks   
    If Wh.Name <> ThisWorkbook.Name Then   
        Wh.Close savechanges:=True   
    End If Next Wh   
End Sub

Ở trường hợp sau cùng, macro sẽ đóng tất cả các workbooks đang mở, chỉ trừ workbook có chứa macro. Và ta được biết thêm một số điều rằng,   
  
A\*) Phương thức colse có thêm đối số cho nó.   
Và nếu không có đối số, như trong trường hợp  
Workbooks.Close, thì khi excel gặp câu lệnh này, excel sẽ đưa ra cho ta hộp thoại, để hỏi chúng ta có lưu những tác động thay đổi của chúng ta lên các trang tính hay không? Nếu chương trình chúng ta chỉ là một mẫu cỏn con, chúng ta còn có thể nhớ & biết tại sao excel lại làm vậy. Nhưng nếu chúng ta có một macro đồ sộ. Khi đó chúng ta không thể biết tại sao excel lại hỏi. Để khỏi bị bất ngờ trong những trường hợp như vậy, chúng ta nên chọn cách an toàn nhất (cho chúng ta cũng như cho những người sử dụng sau này!)  
  
B\*) Phương thức Close còn tác động lên các cửa sổ. Điều này giành cho các bạn nghiên cứu thông qua phần trợ giúp của CS VBE trong excel.   
Nếu chúng ta coi phần tử Application là quả đất của chúng ta, thì Workbooks như một nước Nga, hay nước Mĩ nào đó; & Selection như nước Viết nam. Ở đây chúng ta phải hiểu ngầm rằng, ngày mai anh bay từ Tokyô đến đến Hoa thịnh đốn; cũng là nói ngày mai anh ấy sẽ bay từ Nhật bản sang Mĩ vậy. Nói đến workbooks là ta phải ngầm định ngay nó là tập hợp con ngay kế với tập hợp Application.  
  
Không biết phải vì tầm quan trọng, hay vì cái gì khác, mà một số người lại phân nhóm nhỏ cho các tính chất & phương pháp toàn cục này:  
  
1\*/ Các tính chất để chỉ định: Selection,ThisWorkbook & các tính chất ‘Active’, như ActiveCell, ActiveSheet, ActiveWindow, ActiveWorkBook ActiveChart, ActiveDialog, ActiveMenuBar, ActivePrinter,. . ..(Cần nói rõ rằng, tính chất ThisWorkbook sẽ chỉ đến bảng tính có chứa macro, chứ nó không yêu cầu bảng tính này đang được kích hoạt hay không)  
  
2\*/ Các tập hợp vùng, các phương pháp phổ biến nhất trả về phần tử vùng, như Range, Union, Intersect, Cells, Rows & Columns, . . .  
Nhóm các phương pháp & thuộc tính này chúng ta đã đề cập tại bài đầu.  
  
3\*/ Các tập hợp trang: Các phương pháp hay tính chất trả về các tập hợp trang khác nhau trong bảng tính, như Sheets, Worksheets, Charts, DialogSheets, Modules, Excel4MacroSheets, Excel4IntlMacroSheets, . .   
  
4\*/ CÁc tập hợp khác, như AddInt, MenuBars, Names, ShortcutMenus, Toolbars, Windows & workbooks, . . .  
  
5\*/ Các lệnh, cũng như các phương pháp dùng để thực hiện các việc thông thường, như Calculate, Evaluate, Run & Sndkeys; Cùng với các lệnh dùng cho DDE (liên kết dữ liệu động). Cái này quả là cao cấp & xa xỉ đối với nhiều người, trong đó có tôi!  
Để kết luận, xin được nhắc lại là, những tính chất & phương pháp toàn cục này, tuy không đi kè với Application, nhưng đó là bố đẽ ra chúng. 

**Application Object**

Phần tử Application chính là bản thân excel. Nó là bản chất của mọi phần tử trong excel. Phần tử Application bao gồm:  
  
• Application-wide settings and options (many of the options in the Options dialog box (Tools menu), for example).  
  
• Methods that return top-level objects, such as ActiveCell, ActiveSheet, and so on.  
  
• Các khả năng thiết lập và tùy chọn thông số phong phú cho phần tử (phần lớn các tùy chọn ở trong hộp thoại Options (menu Tools))  
• Các phương thức để chỉ định các đối tượng hiện hành, như ActiveCell, ActiveSheet, v.v.  
  
**Using the Application Object**  
  
**Thao tác với phần tử Application**  
  
Use the Application property to return the Application object. The following example applies the Windows property to the Application object.  
  
Sử dụng các thuộc tính của phần tử Application để thao tác với đối tượng Application. Trong ví dụ sau ta thao tác với thuộc tính Windows của Application.  
  
Application.Windows("book1.xls").Activate  
  
The following example creates a Microsoft Excel workbook object in another application and then opens a workbook in Microsoft Excel.  
  
Còn trong ví dụ sau đây ta tạo ra một đối tượng workbook của Microsoft Excel   
  
Set xl = CreateObject("Excel.Sheet")  
xl.Application.Workbooks.Open "newbook.xls"  
  
**Remarks**  
  
**Lưu ý**  
  
  
Many of the properties and methods that return the most common user-interface objects, such as the active cell (ActiveCell property), can be used without the Application object qualifier. For example, instead of writing Application.ActiveCell.Font.Bold = True, you can write ActiveCell.Font.Bold = True.  
  
Có nhiều thuộc tính và phương thức chỉ định đến các đối tượng thường dùng, như đến cell hiện hành (thuộc tính ActiveCell), các thuộc tính này có thể dùng mà không cần từ khóa Application. Ví dụ: thay vì viết Application.ActiveCell.Font.Bold = True, ta có thể viết ActiveCell.Font.Bold = True.

ĐẶT THỪA SỐ CHUNG CHO CÁC ĐOẠN MÃ TƯƠNG TỰ NHAU  
TRONG MACRO.

Giả sử nhiệm vụ đề ra là ta fải viết macro để theo dõi việc cung cấp hàng hóa từ nhà cung cấp như bảng được trích ra như sau:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Ngay | NhaCC | Ma | Ten | GiaMua | SoLg | Ttien |  |
|  | 1/9/2010 | Thơm | ANL | Áo NLớn | 45.000 | 5 | 225.000 |  |
|  | 1/9/2010 | Thà | ANL | Áo NLớn | 50.000 | 10 | 500.000 |  |
|  | 2/9/2010 | Hoa | ATE | Áo Trẻ em | 30.000 | 5 | 150.000 |  |
|  | 3/9/2010 | My | QNL | Quần NLớn | 60.000 | 5 | 300.000 |  |
|  | 3/9/2010 | Thơm | QNL | Quần NLớn | 70.000 | 5 | 350.000 |  |
|  | 3/9/2010 | Thà | ANL | Áo NLớn | 50.000 | 1 | 50.000 |  |
|  |  | . . . |  | . . |  |  | . . . |  |

Cụ thể nhiệm vụ là cần biết trong 1 ngày nào đó hay cả tháng ông A hay bà C đã cung cấp bao nhiêu mặt hàng, hoặc ngày nào đó bất kỳ có bao nhiêu nhà cung cấp đã nhập hàng.  
Kết quả thống kê này sẽ cho ra trang tính có tên ‘Phieu’ (trong file đính kèm).  
Mà tại trang tính này người ta đã thiết lập 2 combobox tại [C4] & [H4] để lực chọn nhà cung ứng hay ngày tháng cụ thể nào đó cần khảo sát.  
Nếu tại [C4] để trống, bảng kết quả thể hiện toàn thể các nhà cung cấp trong ngày;  
Nếu [H4] để trống, bảng kết quả sẽ là toàn bộ số hàng – tiền của người đó đã cung cấp trong tháng.  
  
Macro đó có nội dung như sau:

PHP Code:

Option Explicit   
 Dim Sh As Worksheet, Rng As Range, sRng As Range   
  Private Sub Worksheet\_Change(ByVal Target As Range)   
 Dim NCC As Boolean, Ngay As Boolean   
 Dim MyAdd As String:                     Dim Offs As Long    '\*\*'   
    
 Set Sh = Sheets("PhatSinh"):             Application.ScreenUpdating = False '\*'   
 If Not Intersect(Target, [c4]) Is Nothing Then  'Nha CC'   
   [B7].Resize(14, 7).ClearContents   
   If [c4].Value = "" Then Exit Sub   
   Set Rng = Sh.Range(Sh.[B5], Sh.[B65500].End(xlUp))   
   If [H4].Value = "" Then Ngay = True   
   Set sRng = Rng.Find([c4].Value, , xlFormulas, xlWhole) 2 Rem  If Not sRng Is Nothing Then GPE\_Copy [H4], Ngay   
  '     \*     \*     \*     \*     \*'   
  [B7].Resize(99).EntireRow.Hidden = False   
   MyAdd = sRng.Address ':                          Application.ScreenUpdating = False'   
   Do   
      With [b21].End(xlUp).Offset(1)   
         If Ngay Then  'All\_ '   
            .Resize(, 5).Value = sRng.Offset(, 0).Resize(, 5).Value        '0=>Add1'   
            .Offset(, 5).Resize(, 2).Value = sRng.Offset(, 7 + 0).Resize(, 2).Value   
         Else   
            If sRng.Offset(, -1).Value = [H4].Value Then                    '-1=> Offs; Cls=> [H4]'   
               .Resize(, 5).Value = sRng.Offset(, 0).Resize(, 5).Value   
               .Offset(, 5).Resize(, 2).Value = sRng.Offset(, 7 + 0).Resize(, 2).Value   
            End If   
         End If   
      End With   
      Set sRng = Rng.FindNext(sRng)   
   Loop While Not sRng Is Nothing And sRng.Address <> MyAdd   
    
   With [B99].End(xlUp)   
      If .Row < 13 Then Offs = 15 Else Offs = .Row + 2   
   End With   
   Range("B99:B" & Offs).EntireRow.Hidden = True   
   Set Sh = Nothing '     \*     \*     \*     \*     \*'   
 ElseIf Not Intersect(Target, [H4]) Is Nothing Then  'Ngày'   
   [B7].Resize(14, 7).ClearContents   
   If [H4].Value = "" Then Exit Sub   
   Set Rng = Sh.Range(Sh.[A5], Sh.[A65500].End(xlUp))   
   If [c4].Value = "" Then NCC = True   
   Set sRng = Rng.Find([H4].Value, , xlFormulas, xlWhole) 3 Rem   If Not sRng Is Nothing Then GPE\_Copy [C4], NCC, 1, 1'     \*     \*     \*     \*     \*'   
   [B7].Resize(99).EntireRow.Hidden = False   
   MyAdd = sRng.Address:                        '     Application.ScreenUpdating = False'   
   Do   
      With [b21].End(xlUp).Offset(1)   
         If NCC Then                'All\_'   
            .Resize(, 5).Value = sRng.Offset(, 1).Resize(, 5).Value '1=>Add1'   
            .Offset(, 5).Resize(, 2).Value = sRng.Offset(, 7 + 1).Resize(, 2).Value   
         Else   
            If sRng.Offset(, 1).Value = [c4].Value Then ' Offs=>1; Cls=> [C4]'   
               .Resize(, 5).Value = sRng.Offset(, 1).Resize(, 5).Value   
               .Offset(, 5).Resize(, 2).Value = sRng.Offset(, 7 + 1).Resize(, 2).Value   
            End If   
         End If   
      End With   
      Set sRng = Rng.FindNext(sRng)   
   Loop While Not sRng Is Nothing And sRng.Address <> MyAdd   
   With [B99].End(xlUp)   
      If .Row < 13 Then Offs = 15 Else Offs = .Row + 2   
   End With   
   Range("B99:B" & Offs).EntireRow.Hidden = True   
   Set Sh = Nothing   
 '     \*     \*     \*     \*     \*'   
 End If End Sub

Macro này đã đáp ứng được yêu cầu nhiệm vụ đề ra. (Các bạn có thể trãi nghiệm qua file đính kèm để khẳng định điều đó).  
Vậy là đòi hỏi đã được đáp ứng 1 cách thỏa đáng.  
Nhưng với chúng ta, những người sản sinh ra những macro đại loại như trên thì khoang vội thỏa mãn.  
Quan sát kỹ lưỡng một chút, ta sẽ thấy macro chạy cho 1 trong 2 ComBo 1 cách độc lập.  
Có nghĩa là ta đụng vào combo nào, thì chỉ những dòng lệnh liên quan đến nó mới lay động mà thôi.  
Hơn nữa, số các dòng lệnh trong mỗi fân nhóm là như nhau.  
Từ hai nhận xét trên, ta tự đề ra nhiệm vụ đặt thừa số chung cho macro này (theo nghĩa đen của toán học đại số)  
Nôm na là ta có biểu thức 2 \* Aa + Bb \* Aa   
Thì ta làm gọn lại như Aa \* (2 + Bb) vậy mà.  
(Có nghĩa là thay vì fải dùng đến 4 fép toán đại số mới ra kết quả thì ta chỉ xài 3 fép thôi  
[GPEf.rar](https://sites.google.com/site/chiasecongnghe86/GPEf.rar)   
  
 Quan sát nội dung macro ở bài 1 bên trên, chúng ta dễ nhận ra 4 dòng không chứa mã lệnh nào; chúng chỉ chứa 5 dấu sao (\*) ở mỗi dòng.  
Nếu xem xét kỹ hơn, ta sẽ thấy các dòng lệnh giữa dòng (\*) 1 & dòng (\*) 2 cũng gần bằng với số dòng lệnh giữa dòng (\*) 3 & 4;  
Không những thế, chúng còn có nội dung rất giống nhau trong từng dòng.  
  
Nhiệm vụ bài này sẽ là: Chúng ta sẽ viết 1 macro thay cho 2 đoạn mã lệnh mà chúng ta vừa đề cập trên.  
Công việc này chưa hẵn mang í nghĩa tăng tốc cho toàn bộ chương trình, ý nghĩa của việc làm này là ở chỗ: Tôi cùng các bạn ta sẽ tiến hành thực hiện cẩn trọng từng bước để đạt múc đích làm cho macro của chúng ta dễ sử dụng & bảo trì trong tương lai  
  
Các bước đó là:  
**Bước 1./**  
Chép fân đoạn I ra 1 macro riêng (tạm gọi là macro con) & chỉnh sửa để khi ta thêm dòng lệnh vô macro cha gọi macro con, thì kết quả vẫn như ta thực hiện macro cha khi chưa cho con ở riêng, theo kiểu

PHP Code:

Sub MacroBĐ()  
  ‘ . . . . . . . . . ‘  
‘    \*    \*    \*    \*    \*’  
     (Các câu lệnh fần I)‘    \*    \*    \*    \*    \*’  
‘    \*    \*    \*    \*    \*’  
     (Các câu lệnh fần II)‘    \*    \*    \*    \*    \*’  
‘. . . . . . . . . .’  
End Sub

Trở thành:

PHP Code:

Sub MacroCha()  
  ‘ . . . . . . . . . ‘  
‘    \*    \*    \*    \*    \*’  
‘     (Các câu lệnh fần I) ‘  
‘    \*    \*    \*    \*    \*’  
‘    \*    \*    \*    \*    \*’  
     (Các câu lệnh fần II)‘    \*    \*    \*    \*    \*’  
‘. . . . . . . . . .’  
End Sub

Code:

Sub MacroCon()

(Chứa các câu lệnh fần I) ‘<=|’

End Sub

Như vậy bước I ta thực hiện 2 nhiệm vụ:  
  
**a./ Chép đoạn mã của fần I sang macro mới:**  
  
(Macro con này chúng ta sẽ cho nó cái tên là GPE\_Copy nha các bạn)  
  
Đấu tiên ta đưa con trỏ xuống dòng trống cuối cùng của trang màn hình đang chứa macro cha

PHP Code:

Private Sub Worksheet\_Change(ByVal Target As Range)  
 ‘ .. . . . ‘  
 End Sub

(dưới dòng End Sub)  
  
Ta gỏ nhập “Sub GPE\_Copy” & {ENTER} để ta có được sau sự bổ sung của VBE như sau:

PHP Code:

 Sub GPE\_Copy()  
End Sub

Sau đó ta chỉ việc dùng chuột tô chọn khối các dòng lện fần I của macro cha & chép vô giữa 2 dòng lệnh vừa xuất hiện  
Sau các thao tác đó, chúng ta có kết quả như sau

PHP Code:

 Sub GPE\_Copy()11  [B7].Resize(99).EntireRow.Hidden = False  
   MyAdd = sRng.Address ':                          Application.ScreenUpdating = False'13   Do  
      With [b21].End(xlUp).Offset(1)15         If Ngay Then  'All\_ '  
            .Resize(, 5).Value = sRng.Offset(, 0).Resize(, 5).Value        '0=>Add1'17            .Offset(, 5).Resize(, 2).Value = sRng.Offset(, 7 + 0).Resize(, 2).Value  
         Else19            If sRng.Offset(, -1).Value = [H4].Value Then                    '-1=> Offs; Cls=> [H4]'  
               .Resize(, 5).Value = sRng.Offset(, 0).Resize(, 5).Value  
111               .Offset(, 5).Resize(, 2).Value = sRng.Offset(, 7 + 0).Resize(, 2).Value  
            End If113        End If  
      End With  
115      Set sRng = Rng.FindNext(sRng)  
   Loop While Not sRng Is Nothing And sRng.Address <> MyAdd  
  
117   With [B99].End(xlUp)  
      If .Row < 13 Then Offs = 15 Else Offs = .Row + 2  
119   End With  
   Range("B99:B" & Offs).EntireRow.Hidden = True  
121   Set Sh = Nothing  
End Sub

Đó là xong nhiệm vụ chép thôi; còn nhiệm vụ chỉnh sửa nữa mới là fần quan trọng.  
Để tiện trong bước tiếp theo, tôi đã đánh số các dòng lệnh như các bạn thấy bên trên  
  
**b./ Chỉnh sửa macro con.**  
  
(Nếu đến thời điểm này mà ta :  
- Thêm dòng lệnh gọi macro con này từ ngay trước fần I macro cha  
- Vô hiệu hóa toàn bộ các dòng lệnh fần I của macro cha  
và tiến hành thao tác để chạy macro cha ( Cụ thể thay đổi [C4] ở trang tính ‘Phieu’) ta sẽ bị báo lỗi.)  
  
Chuyện đó là đương nhiện theo các lẽ sau:  
  
Ta chưa khai báo biến nào cho macro con (Vì trên cùng ta có dòng Option Explicit & dòng lệnh 12 ta xài biến MyAdd chưa khai báo chẳng hạn)  
  
Thêm nữa, trong dòng lệnh 16 ta có xài biến kiểu Range sRng, mà trên chương trình cha ta đã khai báo. & gán cho nó là ô chứa kết quả tìm được trong biến Rng.  
Như vậy ta có thể đưa khai báo sRng này làm tài sản chung cho 2 macro cha & con.  
  
Vấn đề nữa, đó là tại dòng 15 ta ra điều kiện chứa trong biến Ngay kiểu Yes/No. Như vậy chương trình con cần fải biết tại thời điểm đó biến Ngay là Yes hay No để mà ứng xử cho đúng. Có nghĩa là tham biến này fải được truyền từ chương trình cha sang cho chương trình con.  
  
Tóm lại: Ta cần rà soát từng dòng lệnh để:  
(\*) Khai báo biến dùng chung trong 1 số trường hợp  
(\*) Truyền tham biến từ chương trình cha cho chương trình con, để anh con này làm được đúng yêu cầu;  
(\*) Khai báo thêm 1 số biến fát sinh, nếu cần   
  
Ta tiến hành kiểm theo từng dòng lệnh một:  
  
D11: Không fát sinh, vì ta đang thao tác trên trang tính hiện hành;  
D12 Biến MyAdd nên là 1 biến cha & con cùng dùng (Có nghĩa là ta sẽ fải khai báo 1 biến dùng chung trong toàn module);  
(Hai dòng 13 & 14) không có vấn đề gì. . . - cho qua)  
D15, Tham biến Ngay cần được truyền từ cha sang con; Nhưng đến chương trình con nó có thể mang tiên khác chút; đó là tôi dùng tiên All\_ trong chương trình con;  
D16 & D17: Tham biến sRng xài chung như MyAdd;  
D19 Ô [H4] hiện chưa có ván đề; Hẹn bài sau sẽ fát sinh vấn đề;  
Từ đây cho đến dòng 117 sẽ là 1 trong những trường hợp ta đã xét; Các bạn tự xuy luận giúp.  
D118: Biến Offs ta có thể khai báo dùng chung, nhưng tiết kiệm tài nguyên 1 chút, ta sẽ khai báo thêm tại chương trình con.   
  
Với tất cả những gì chúng ta vừa lập luận, ta đẻ ra 2 macro cha & con có những nội dung như sau:

PHP Code:

Option Explicit  
 Dim Sh As Worksheet, Rng As Range, sRng As Range  
 Private Sub Worksheet\_Change(ByVal Target As Range)  
 Dim NCC As Boolean, Ngay As Boolean  
 Dim MyAdd As String:                     Dim Offs As Long    '\*\*'  
  
 Set Sh = Sheets("PhatSinh"):             Application.ScreenUpdating = False '\*'  
 If Not Intersect(Target, [c4]) Is Nothing Then  'Nha CC'  
   [B7].Resize(14, 7).ClearContents  
   If [c4].Value = "" Then Exit Sub  
   Set Rng = Sh.Range(Sh.[B5], Sh.[B65500].End(xlUp))  
   If [H4].Value = "" Then Ngay = True  
   Set sRng = Rng.Find([c4].Value, , xlFormulas, xlWhole)2 Rem  If Not sRng Is Nothing Then GPE\_Copy [H4], Ngay  
  '     \*     \*     \*     \*     \*'  
  [B7].Resize(99).EntireRow.Hidden = False  
   MyAdd = sRng.Address ':                          Application.ScreenUpdating = False'  
   Do  
      With [b21].End(xlUp).Offset(1)  
         If Ngay Then  'All\_ '  
            .Resize(, 5).Value = sRng.Offset(, 0).Resize(, 5).Value        '0=>Add1'  
            .Offset(, 5).Resize(, 2).Value = sRng.Offset(, 7 + 0).Resize(, 2).Value  
         Else  
            If sRng.Offset(, -1).Value = [H4].Value Then                    '-1=> Offs; Cls=> [H4]'  
               .Resize(, 5).Value = sRng.Offset(, 0).Resize(, 5).Value  
               .Offset(, 5).Resize(, 2).Value = sRng.Offset(, 7 + 0).Resize(, 2).Value  
            End If  
         End If  
      End With  
      Set sRng = Rng.FindNext(sRng)  
   Loop While Not sRng Is Nothing And sRng.Address <> MyAdd  
  
   With [B99].End(xlUp)  
      If .Row < 13 Then Offs = 15 Else Offs = .Row + 2  
   End With  
   Range("B99:B" & Offs).EntireRow.Hidden = True  
   Set Sh = Nothing'     \*     \*     \*     \*     \*'  
 ElseIf Not Intersect(Target, [H4]) Is Nothing Then  'Ngày'  
   [B7].Resize(14, 7).ClearContents  
   If [H4].Value = "" Then Exit Sub  
   Set Rng = Sh.Range(Sh.[A5], Sh.[A65500].End(xlUp))  
   If [c4].Value = "" Then NCC = True  
   Set sRng = Rng.Find([H4].Value, , xlFormulas, xlWhole)3 Rem   If Not sRng Is Nothing Then GPE\_Copy [C4], NCC, 1, 1'     \*     \*     \*     \*     \*'31   [B7].Resize(99).EntireRow.Hidden = False  
   MyAdd = sRng.Address:                        '     Application.ScreenUpdating = False'33   Do  
      With [b21].End(xlUp).Offset(1)35         If NCC Then                'All\_'  
            .Resize(, 5).Value = sRng.Offset(, 1).Resize(, 5).Value '1=>Add1'37            .Offset(, 5).Resize(, 2).Value = sRng.Offset(, 7 + 1).Resize(, 2).Value  
         Else39            If sRng.Offset(, 1).Value = [c4].Value Then ' Offs=>1; Cls=> [C4]'  
               .Resize(, 5).Value = sRng.Offset(, 1).Resize(, 5).Value  
301               .Offset(, 5).Resize(, 2).Value = sRng.Offset(, 7 + 1).Resize(, 2).Value  
            End If303         End If  
      End With  
305      Set sRng = Rng.FindNext(sRng)  
   Loop While Not sRng Is Nothing And sRng.Address <> MyAdd  
307   With [B99].End(xlUp)  
      If .Row < 13 Then Offs = 15 Else Offs = .Row + 2  
309   End With  
   Range("B99:B" & Offs).EntireRow.Hidden = True  
311   Set Sh = Nothing  
 '     \*     \*     \*     \*     \*'  
 End IfEnd Sub

**(Bổ sung số cho các dòng lệnh từ 31. . 311 ngày 19/09/2010)**

Code:

Sub GPE\_Copy(Cls As Range, All\_ As Boolean, Optional Offs As Integer = -1, Optional Add1 As Byte)

Dim MyAdd As String

[B7].Resize(99).EntireRow.Hidden = False

MyAdd = sRng.Address: Application.ScreenUpdating = False

Do

With [b21].End(xlUp).Offset(1)

If All\_ Then

.Resize(, 5).Value = sRng.Offset(, Add1).Resize(, 5).Value

.Offset(, 5).Resize(, 2).Value = sRng.Offset(, 7 + Add1).Resize(, 2).Value

Else

If sRng.Offset(, Offs).Value = Cls.Value Then

.Resize(, 5).Value = sRng.Offset(, Add1).Resize(, 5).Value

.Offset(, 5).Resize(, 2).Value = sRng.Offset(, 7 + Add1).Resize(, 2).Value

End If

End If

End With

Set sRng = Rng.FindNext(sRng)

Loop While Not sRng Is Nothing And sRng.Address <> MyAdd

With [B99].End(xlUp)

If .Row < 13 Then Offs = 15 Else Offs = .Row + 2

End With

Range("B99:B" & Offs).EntireRow.Hidden = True

Set Sh = Nothing

End Sub

Ở đây ta dùng tham số Cls thay cho [H4]; & ta bỏ từ khóa ‘Rem’ tại dòng lệnh mang số 2 là & thêm dòng lệnh

PHP Code:

Exit Sub

kề dưới dòng này nữa (Dòng lệnh ta vừa bỏ chữ Rem) là mọi chuyện trở nên mĩ mãn.  
  
Bây giờ ta có thể chạy macro cha; Đến lúc cần, macro cha sẽ gọi macro con chạy tiếp sức cho đến kết quả khi ta chọn khách hàng trong ô [C4]  
Còn nếu ta chọn ngày nhập hàng tại [H4] Chỉ có macro cha chạy 1 mạcch cho đến kết quả cuối cùng mà thôi.  
  
Ở bài 2, chúng ta ta viết lại macro con, mà thực tế nó được chép từ khối I của macro cha. Lúc đó macro con sẽ được sửa lại để làm nhiệm vụ thay cho khối lệnh I đã bị vô hiệu hóa sau đó.  
  
Nhiệm vụ của ta bài này sẽ là biến macro con này làm fần việc của khối lệnh II (Nằm giữa 2 dòng chứa 5 dấu sao 3 & 4 mà ta đã đề cập ở bài 2 nêu trên.)  
  
Để tiện cùng nhau trao đổ thông tin, tôi xin fép đánh số các dòng lệnh ở khối II của macro cha ( đoạn mã gần cuối bài 2) & (các dòng lệnh của macro con đã được gán số như fần đầu bài 2). Hơn nữa, các dòng lệnh trong macro con đều bắt đầu từ số 1; Còn các dòng lệnh trong khối lệnh II ta đã bắt đầu bỡi số 3.  
  
Bây giờ ta cần đối chiếu từng cặp dòng lệnh giữa 2 khối này.; Nhưng để tiết kiệm thời gian cũng như có dịp đi sâu vô các dòng lệnh khác biệt nhau, cho fép tôi không nói lại những dòng lệnh gần giống hệt nhau cả về cú fáp cũng như kết quả thực hiện chúng.  
  
Ta đi vô dòng 35 có nội dung: If NCC Then (& tương ứng với nó là dòng 14 If Ngay Then)  
  
Như cuối bài 2 ta thấy, khi gọi macro con, chúng ta đã truyền cho nó tham số Ngay  
  
Thì dòng lệnh gọi macro con thay cho khối II cũng sẽ truyền cho macro con tham biến NCC;   
Nhưng khi đến macro con này, nó có thể mang 1 tên khác, chẳng hạn All\_ như bạn sẽ thấy. Tất nhiên, nó vẫn có thể giữ nguyên tên là Ngay, nhưng như vậy ta tự làm khó ta vì dễ làm ta tẩu hỏa nhập ma lúc nào đó & lộn tùng xèn không đáng có.  
  
Trong các dòng lệnh từ sau dòng 35 này cho đến dòng 306, biến sRng của khối cũng xuất hiện trong khối các dòng lệnh I.  
Như vậy, có nghĩa là ta cần khai báo biến sRng dùng chung cho 2 macro cha & con.  
  
Để vậy, chúng ta khai báo biến sRng chỉ ngay sau dòng lệnh Option Explicit của module chứa cả hai macro. (Macro con để luôn trong macro sự kiện của trang tính ‘Phieu’)  
  
Tuy nhiên cũng cần mở ngoặt thêm chổ này để cùng rõ hơn:  
Vì trước khối lệnh I vùng Rng để tìm kiếm được gán cho các ô **Sh.Range(Sh.[B5], Sh.[B65500].End(xlUp))**  
Nhưng trước khối lệnh II, vùng Rng lại là **Sh.Range(Sh.[A5], Sh.[A65500].End(xlUp))**  
  
Nên sRng là ô tìm thấy tương ứng với 2 vùng trên sẽ khác cột nhau.   
Bỡi lẽ đó các giá trị cần gán vô form của trang tính ‘Phieu’ (vùng từ B7: H99) sẽ tương ứng khác nhau vế cột trong 2 đoạn mã giữa khối I & khối II  
  
Để cụ thể hơn, ta xét trường hợp mã hàng cần điền vô cột ‘C’ của trang tính ‘Phieu’  
  
Trường hợp ứng với khối dòng lệnh I, ta cần tìm tên các nhà cung cấp có tại cột ‘B’ của trang tính ‘PhatSinh’, thì mã hàng là cột liền kề bên fải so với ô được tìm thấy trên ‘B’;  
Nhưng trường hợp II thì ta tìm ngày nhập (Ở cột ‘A’ trên trang ‘PhatSinh’), lúc đó mã hàng là ô cách ô sRng tím thấy 1 cột.  
  
Qua đó chúng ta cũng sẽ thấy các trị cần gán vô form sẽ khác cột với nhau 1 cách tương ứng khi macro thực hiện các câu lệnh trong khối I & khối II.  
  
Qua thí dụ điển hình này, chúng ta sẽ rút ra kết luận, là cần cung cấp cho macro con trong trường hợp II các thông số về độ lệnh cột giữa lần chép I & lần chép II;   
Nhưng việc cung cấp này sẽ được mặc định 1 trị từ lệnh gọi (dòng lệnh số 2) bằng từ khóa Optional trong fần đầu của chương trình con. (Còn gọi macro con từ dòng lệnh 3 của macro cha sẽ fải cung cấp tham số đầy đủ).  
  
Mệnh đề **Optional Add1 As Byte** nói lên rằng: Tham số Add1 có thể được truyền, thì sẽ là trị số được truyền, nếu không được truyền thì tham số đó bằng 0 một cách mặc định  
  
Đó là chúng ta mới nói đến tham số có tên Add1 (Xem trong macro GPE\_Copy); Còn 1 tham số được truyền từ cha sang con nữa, đó là Offs. ( Hơn nữa, nếu không được truyền, tham số này mặc định là -1)  
  
Ta sẽ xét kỹ thêm về nó:  
  
Để vậy, chúng ta xét từ xuất fát điểm của ta là đi tìm ngày nhập hàng cụ thể nào đó của 1 NCC (nhà cung cấp) cụ thể nào đó. Mà ta biết rằng, ta có nhiều NCC cung ứng hàng nhiều lần trong tháng, thậm chí cung ứng nhiều lần trong ngày nữa & trong 1 ngày có thể có nhiều NCC nhập hàng. Bên trang CSDL dữ liệu về NCC được lưu ở cột ‘B’ còn ngày nhập hàng ghi ở cột ‘A’  
  
Khi áp dụng fương thức tìm kiếm 1 ngày nào đó, thì macro sẽ lọc ra DS (danh sách) NCC hôm đó; Nhiệm vụ tiếp theo là trong DS này, ta fải lựa ra 1 khách hàng ta cần mà thôi. Tất cả điều này liên quan đến khối lệnh tìm theo ngày;  
  
Còn khối lệnh tìm theo khách hàng cung ứng hàng trong tháng thì ngược lại chút đĩnh: Ta sẽ lọc ra từ DS các ngày trong tháng mà khách hàng này cung cấp hàng để chọn ra ngày cụ thể theo yêu cầu tại ô [H4]  
  
Nói nôm na là trường hợp đầu tứ cột ‘A’ ta tìm sang ‘B’ & ngược lại, trường hợp sau ta fải từ DS của cột ‘B’ sưu truy sang cột ‘A’  
  
Việc này sẽ fải nhờ đền biến Offs (để nó offset(,-1) hay Offset(,1) giúp ta trong trường hợp cụ thể gọi macro con từ dòng lệnh 2 hay từ dòng lệnh 3 từ macro cha)  
  
Nếu các bạn sưu tra thêm vế tham số Offs này, các bạn quan tâm đến nó đang trong vòng lặp Do . . . Loop mà thôi. (Còn dưới vòng lặp đó chẳng qua là tận dùng để làm việc khác mà khỏi khai báo thêm biến.)  
  
  
Kết quả của việc làm của chúng ta qua 3 bài là như thế này:  
  
**Macro cha:**

PHP Code:

Option Explicit  
 Dim Sh As Worksheet, Rng As Range, sRng As Range  
 Private Sub Worksheet\_Change(ByVal Target As Range)  
 Dim NCC As Boolean, Ngay As Boolean  
 Dim MyAdd As String:                     Dim Offs As Long    '\*\*'  
  
 Set Sh = Sheets("PhatSinh"):             Application.ScreenUpdating = False '\*'  
 If Not Intersect(Target, [C4]) Is Nothing Then  'Nha CC'  
   [B7].Resize(14, 7).ClearContents  
   If [C4].Value = "" Then Exit Sub  
   Set Rng = Sh.Range(Sh.[B5], Sh.[B65500].End(xlUp))  
   If [H4].Value = "" Then Ngay = True  
   Set sRng = Rng.Find([C4].Value, , xlFormulas, xlWhole)2  If Not sRng Is Nothing Then GPE\_Copy [H4], Ngay  
 ElseIf Not Intersect(Target, [H4]) Is Nothing Then  'Ngày'  
   [B7].Resize(14, 7).ClearContents  
   If [H4].Value = "" Then Exit Sub  
   Set Rng = Sh.Range(Sh.[A5], Sh.[A65500].End(xlUp))  
   If [C4].Value = "" Then NCC = True  
   Set sRng = Rng.Find([H4].Value, , xlFormulas, xlWhole)3  If Not sRng Is Nothing Then GPE\_Copy [C4], NCC, 1, 1  
 End IfEnd Sub

**Còn đây là macro con (nằm luôn trong cửa sổ VBE sự kiện này luôn):**

Code:

**Sub GPE\_Copy(Cls As Range, All\_ As Boolean, Optional Offs As Integer = -1, Optional Add1 As Byte)**

Dim MyAdd As String

[B7].Resize(99).EntireRow.Hidden = False

MyAdd = sRng.Address: Application.ScreenUpdating = False

Do

With [b21].End(xlUp).Offset(1)

If All\_ Then

.Resize(, 5).Value = sRng.Offset(, Add1).Resize(, 5).Value

.Offset(, 5).Resize(, 2).Value = sRng.Offset(, 7 + Add1).Resize(, 2).Value

Else

If sRng.Offset(, Offs).Value = Cls.Value Then

.Resize(, 5).Value = sRng.Offset(, Add1).Resize(, 5).Value

.Offset(, 5).Resize(, 2).Value = sRng.Offset(, 7 + Add1).Resize(, 2).Value

End If

End If

End With

Set sRng = Rng.FindNext(sRng)

Loop While Not sRng Is Nothing And sRng.Address <> MyAdd

With [B99].End(xlUp)

If .Row < 13 Then Offs = 15 Else Offs = .Row + 2

End With

Range("B99:B" & Offs).EntireRow.Hidden = True

Set Sh = Nothing

**End Sub**

**BÀI TẬP ĐẶT THỪA SỐ CHUNG CHO CÁC ĐOẠN MÃ TƯƠNG TỰ NHAU**  
**TRONG MACRO.**

PHP Code:

Option Explicit  
Dim Sh As Worksheet:                        Dim eRw As LongPrivate Sub Worksheet\_Change(ByVal Target As Range)  
 Dim WF, Cls As Range, Rng As Range  
 Dim lRs As Long, Jj As Byte  
  
 Set WF = Application.WorksheetFunction  
 Set Rng = Sh.[B6].Resize(eRw, 10)  
 [b1:B99].EntireRow.Hidden = False  
1 'Xu Lí Vói Tháng Nam:'  
 If Not Intersect(Target, [e3]) Is Nothing Then  
    [b20].CurrentRegion.Offset(3, 1).ClearContents  
11 'Xu Lý Vói Tùng Tháng Cu The Trong Nam:'  
    If Target.Value <> "All" Then  
        With [B99].End(xlUp).Offset(1)  
            .Value = "'" & Right("0" & [e3].Value, 2) & "/" & [G3].Value  
5            .Offset(, 5).Value = WF.DSum(Rng, Sh.[g6], Sh.[AA1].Resize(2, 2))  
            .Offset(, 6).Value = WF.DSum(Rng, Sh.[H6], Sh.[AA1].Resize(2, 2))  
            .Offset(, 7).Value = WF.DSum(Rng, Sh.[I6], Sh.[AA1].Resize(2, 2))  
        End With  
12 'Xu Lí Vói Toàn Bo Các Tháng Trong Nam'  
    Else  
        Sh.[ac1].Value = "Th"  
        For Jj = 1 To 12  
            Sh.[ac2].Value = Jj  
            With [B99].End(xlUp).Offset(1)  
                .Value = "'" & Right("0" & Sh.[ac2].Value, 2) & "/" & [G3].Value  
6                .Offset(, 5).Value = WF.DSum(Rng, Sh.[g6], Sh.[Ab1].Resize(2, 2))  
                .Offset(, 6).Value = WF.DSum(Rng, Sh.[H6], Sh.[Ab1].Resize(2, 2))  
                .Offset(, 7).Value = WF.DSum(Rng, Sh.[I6], Sh.[Ab1].Resize(2, 2))  
            End With  
        Next Jj  
    End If  
    Range([b11].End(xlDown).Offset(2), [B98]).EntireRow.Hidden = True  
2 'Xu Lý Vói Ten Dói Tác:'  
 ElseIf Not Intersect(Target, [e4]) Is Nothing Then  
    [b20].CurrentRegion.Offset(3, 1).ClearContents  
    Sh.[ac1].Value = Sh.[d6].Value  
21 'Xu Lý Voi Tùng Dói Tác:'  
    If Target.Value <> "All" Then  
        Sh.[ac2].Value = [G4].Value  
        If [e3].Value <> "All" Then  
            With [B99].End(xlUp).Offset(1)  
                .Value = "'" & Right("0" & [e3].Value, 2) & "/" & [G3].Value  
                .Offset(, 2).Value = [G4].Value  
                .Offset(, 3).Value = [e4].Value  
7                .Offset(, 5).Value = WF.DSum(Rng, Sh.[g6], Sh.[AA1].Resize(2, 3))  
                .Offset(, 6).Value = WF.DSum(Rng, Sh.[H6], Sh.[AA1].Resize(2, 3))  
                .Offset(, 7).Value = WF.DSum(Rng, Sh.[I6], Sh.[AA1].Resize(2, 3))  
            End With  
        Else  
            Sh.[ad1].Value = "Th"  
            For Jj = 1 To 12  
                Sh.[ad2].Value = Jj  
                With [B99].End(xlUp).Offset(1)  
                    .Value = "'" & Right("0" & Jj, 2) & "/" & [G3].Value  
                    .Offset(, 2).Value = [G4].Value  
                    .Offset(, 3).Value = [e4].Value  
8                    .Offset(, 5).Value = WF.DSum(Rng, Sh.[g6], Sh.[Ab1].Resize(2, 3))  
                    .Offset(, 6).Value = WF.DSum(Rng, Sh.[H6], Sh.[Ab1].Resize(2, 3))  
                    .Offset(, 7).Value = WF.DSum(Rng, Sh.[I6], Sh.[Ab1].Resize(2, 3))  
                End With  
            Next Jj  
        End If22 'Xu Lý Vói Tát Ca Các Dói Tác:'  
    Else  
        Dim Nam As String, Crit As Range  
  
        If [e3].Value <> "All" Then  
            Set Crit = Sh.[AA1].Resize(2, 3)  
            Nam = Right([C3].Value, 5) & " " & [e3].Value & "/" & [G3].Value  
        Else  
            Set Crit = Sh.[Ab1].Resize(2, 2)  
            Nam = Right([b11].Value, 3) & " " & [G3].Value  
        End If  
        For Each Cls In Sheets("DKien").Range("MaDT").SpecialCells(xlCellTypeConstants, 2).Offset(1)  
            If Cls.Value = "All" Then Exit For  
            Sh.[ac2].Value = Cls.Value  
            With [B99].End(xlUp).Offset(1)  
                .Value = Nam  
                .Offset(, 2).Value = Cls.Value  
                .Offset(, 3).Value = Sheets("DKien").Range("MaDT"). \_  
                    Find(Cls.Value, , xlFormulas, xlWhole).Offset(, -1).Value  
9                .Offset(, 5).Value = WF.DSum(Rng, Sh.[g6], Crit)  
                .Offset(, 6).Value = WF.DSum(Rng, Sh.[H6], Crit)  
                .Offset(, 7).Value = WF.DSum(Rng, Sh.[I6], Crit)  
            End With  
        Next Cls  
    End If  
    Range([b11].End(xlDown).Offset(2), [B98]).EntireRow.Hidden = True  
 End IfEnd Sub

Quan sát macro ta thấy 3 dòng lệnh kế tiếp của các dòng lệnh được đánh số từ 5 đến 9 có cấu trúc na ná nhau;  
  
Nhiệm vụ đề ra là ta thử gộp chúng vô 1 macro con như cách thúc mà các bài 1 đến 3 về đặt thừa số chung mà ta đã xét qua.

[Tuấn Đinh Anh](https://plus.google.com/116953857201197706003)at [10:16](http://www.hocviendaotao.com/2013/02/kien-thuc-lap-trinh-trong-excel.html?m=1)