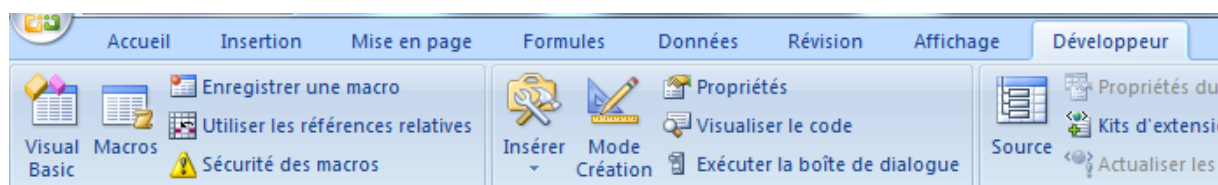
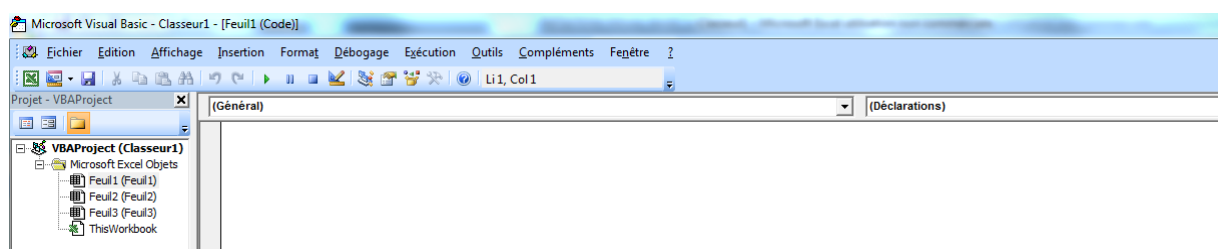

TP3 - VBA - Durée 2h

Visual Basic for Applications (VBA) est une implémentation de Microsoft Visual Basic qui est intégrée dans toutes les applications de Microsoft Office, dans quelques autres applications Microsoft comme Visio et au moins partiellement dans quelques autres applications comme AutoCAD, WordPerfect, MicroStation, Solidworks ou encore ArcGIS. Il remplace et étend les capacités des langages macro spécifiques aux plus anciennes applications comme le langage WordBasic intégré à une ancienne version du logiciel Word

Avec Excel 2007 les macros sont créées dans le menu « Développeur », puis « Visual basic »



Une fenêtre apparaît, dans laquelle on peut écrire des « **macros** » ou des « sous routines » (SUB ROUTINES) qui pourront être ensuite exécutées.



Principe : une « macro » comporte une suite d'instructions qui vont s'exécuter les unes après les autres

La première ligne indique le nom de la macro et (éventuellement) les variables qu'elle doit traiter

La dernière ligne est toujours : END SUB

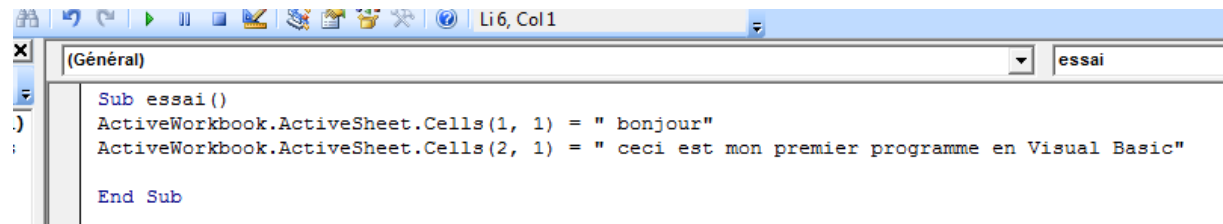
Exemple :

SUB Essai()

Suite d'instructions

'commentaires éventuels

END SUB



```
Sub essai ()  
ActiveWorkbook.ActiveSheet.Cells(1, 1) = " bonjour"  
ActiveWorkbook.ActiveSheet.Cells(2, 1) = " ceci est mon premier programme en Visual Basic"  
  
End Sub
```

Lorsque vous taper une apostrophe ' le texte qui suit n'est pas considéré comme une instruction exécutable, mais comme un commentaire. Cela permet de préciser certaines informations utiles au programmeur

Lorsque vous quittez le module de programmation, cette macro sera incluse dans la liste des macros, et vous pourrez en demander l'exécution. Cette macro sera liée au classeur en cours, mais son exécution pourra concerner des données contenues dans d'autres classeurs.

Objectifs :

- Lire des données dans une feuille
- Traitement de ces données
- Placer les résultats dans une feuille

Avantage : pas besoin de gérer l'interface utilisateur (saisie et présentation des informations)

Les données sont dans des cellules (**Cells**) d'une feuille (**Sheet**) d'un classeur (**Workbook**)

L'adressage d'une cellule doit donc comporter tous ces éléments séparés par un point (.)

Exemple :

X = Monclasseur.Mafeuille.cells(1,2)

: met le contenu de la cellule de la ligne 1 et de la colonne 2 de la feuille Mafeuille du classeur MonClasseur dans la variable X

Une cellule est référencée par son numéro de ligne et son numéro de colonne dans la feuille

Il faut déclarer toutes les variables qui vont être utilisées, par l'instruction **DIM** suivi du Nom de la variable **AS** type de variable

ex :

Dim MaFeuille As Worksheet

Plusieurs façons d'associer la variable Mafeuille à une feuille de données d'un classeur :

Set MaFeuille = ActiveWorkbook.ActiveSheet 'la variable MaFeuille fait référence à la feuille active (celle qui est affichée sur l'écran)

Ou :

Set MaFeuille = ActiveWorkbook.Sheets("Feuil2") 'MaFeuille fait référence à la Feuille nommée « Feuil2 » du classeur actif

Ou :

Set Mafeuille = ActiveWorkbook.Sheets(1) 'Mafeuille fait référence à la feuille numéro 1 du classeur

MaFeuille.Activate

MaFeuille.Select 'MaFeuille devient la feuille active, et s'affiche à l'écran

Je peux maintenant utiliser cette variable pour faire référence à une cellule :

X = MaFeuille.cells(1,2) 'met le contenu de la cellule située ligne 1 et colonne 2 dans la variable nommée X (qu'il faut aussi déclarer, voir plus loin)

Ou

MaFeuille.cells(2,3) = « BONJOUR » 'met la chaîne de caractère placée entre guillemets dans la cellule ligne 2, colonne 3

Ou :

MaFeuille.cells(1,3) = mafeuille.cells(1,1) * mafeuille.cells(1,2) ' : met dans la cellule de la ligne 1, colonne 3, le produit des valeurs contenues dans les cellules des colonnes 1 et 2 de la même ligne

Une cellule a un contenu (numérique ou alphanumérique) et des « propriétés » (couleur, fonte, taille de caractères, etc) ; la valeur (value) d'une cellule fait partie de ses propriétés

Donc, en toute rigueur il faudrait écrire :

X = MaFeuille.cells(1,2).Value

MaFeuille.cells(2,3).Value = « BONJOUR »

Même si ce n'est pas toujours nécessaire

On peut éventuellement définir certaines propriétés de la cellule :

Mafeuille.cells(2,3). Font.Italic = True 'style italique

Mafeuille.cells(2,3). Font.Italic = False 'supprime le style italique

Mafeuille.cells(2,3). Font.bold = True 'met en gras

Mafeuille.cells(2,3). Font.Size = 24 'taille du texte

Mafeuille.cells(2,3). Interior.colorIndex = 5 'colore la cellule avec la couleur correspondante à l'index 5

ou

Mafeuille.cells(2,3). Interior.color = vbBlue 'colore la cellule en bleu

Remarque : le préfixe vb (pour Visual Basic) se retrouve souvent pour définir un caractère (vbtab, vbCR, etc) ou une couleur (vbRed, vbBlack, etc)

Types de variables

Les différents types de variables les plus utilisés :

BOOLEAN : ne peut prendre que 2 valeurs : **true** (1 ou -1) ou **false** (0)

INTEGER : nombre entier (entre - 32768 et 32768)

LONG : entier long

DOUBLE : nombre décimal (: double précision)

STRING : chaîne de caractères alphanumériques

Une variable a un nom (X, Y, Taille, Nom), un type et un contenu

Il faut dimensionner (: déclarer) une variable avant de l'utiliser, avec l'instruction DIM

Cette instruction réserve un emplacement en mémoire pour la variable déclarée

On peut imaginer une variable comme un « récipient » qui a un nom et un contenu

Exemple :

DIM X as double, Y as double

X = 10

La variable X de type décimal, contient la valeur 10

Le X fait référence au nom de la variable (le récipient)

Y = X*3.1

La variable Y de type décimal contient 31

Le Y fait référence au nom de la variable (le récipient)

Le X fait référence au contenu du récipient nommé X,

X = X + 1

La variable X contient 11

Ici, le X à gauche du signe = fait référence au « récipient » tandis que le X à droite fait référence à son contenu.

Le signe « = » est une instruction d'affectation (et non de comparaison, comme nous le verrons plus tard)

Cette instruction se lit donc comme : « *mettre dans le récipient X son contenu plus 1* »

Il est évident qu'on ne peut pas mettre dans une variable des valeurs d'un autre type

Par exemple :

DIM X as Integer

X = « Hello »

produit une erreur car on essaie de mettre une chaîne de caractères dans une variable de type numérique de même :

DIM Montexte as String

Montexte = 10

Produit une erreur

Il faut donc faire attention lorsqu'on veut mettre le contenu d'une cellule dans une variable. Par exemple :

DIM X as Double

X = mafeuille.Cells(1,1).value

Produira une erreur si la cellule contient du texte

EXERCICE 1

Dans la feuille 1 du classeur actif, entrer les valeurs suivantes (ou d'autres si vous le souhaitez):

	A	B	C
1	noms	taille	
2	albert	170	
3	raymond	175	
4	georges	162	
5	bruno	183	
6	jean	170	
7	hugo	178	
8	eudes	189	
9	alceste	176	
10			
11			
12			
13			

Nous allons calculer la moyenne et la variance de la série de valeurs de la colonne « taille »

Pour cela, il nous faut une macro qui « lit » les valeurs dans la feuille 1, calcule la moyenne, la variance et l'écart-type, et écrit les résultats dans la feuille 2

Nous avons besoin de définir des variables correspondant à :

La feuille 1

La feuille 2

La somme des X

La somme des X^2

La somme des carrés des écarts

Le nombre de valeurs de la série

La moyenne de la série

La variance de la série

L'écart-type de la série

Remarques :

Le nom d'une variable doit commencer par une lettre, et ne doit ensuite contenir que des lettres ou des chiffres (surtout pas d'espace, ni de lettre accentuées comme é, à, ...)

Les lettres minuscules ou majuscules ne sont pas différenciées
Il est conseillé de donner des noms qui évoquent le contenu de la variable (ex : Moy), mais ce n'est pas obligatoire. Cela facilite simplement la lecture du programme

Notre programme, à ce stade, se présente ainsi :

```
(Général)
Sub CalculMoyVar ()

'déclaration des variables utilisées par la suite
Dim Fdata As Worksheet
Dim Fresult As Worksheet
Dim SomX As Double
Dim SomX2 As Double
Dim N As Integer
Dim SCE As Double
Dim Moy As Double
Dim Variance As Double
Dim EcartType As Double

End Sub
```

Il nous faut maintenant définir les feuilles qui vont être utilisées
Il est souvent intéressant de connaître le nombre de lignes et de colonnes utilisées dans une feuille pour ne parcourir que la plage de cellules réellement utilisées

La syntaxe est :

NbreLigne = feuille.UsedRange.Rows.Count

NbreCol = feuille.UsedRange.Columns.Count

dans notre exemple :

```
(General)
Sub CalculMoyVar ()
1) s 'déclaration des variables utilisées par la suite
Dim Fdata As Worksheet
Dim Fresult As Worksheet
Dim SomX As Double
Dim SomX2 As Double
Dim N As Integer
Dim SCE As Double
Dim Moy As Double
Dim Variance As Double
Dim EcartType As Double
Dim derniereLigne As Long, derniereColonne As Long

Set Fdata = ActiveWorkbook.Sheets(1)
Set Fresult = ActiveWorkbook.Sheets(2)

derniereLigne = Fdata.UsedRange.Rows.Count
derniereColonne = Fdata.UsedRange.Columns.Count

End Sub
```

Il nous faut maintenant calculer N (le nombre de valeurs) : c'est le nombre de lignes - 1 (la première contient du texte correspondant aux noms des variables)
 $N = \text{derniereLigne} - 1$

Il nous faut maintenant parcourir la colonne 2 de la feuille Fdata où se trouvent les valeurs dont on veut calculer la moyenne, variance et écart-type

Pour effectuer ce type d'opération, tous les langages utilisent des « boucles » dans lesquelles s'exécutent des instructions répétées.

En VB, les boucles les plus utilisées sont les boucles FOR ... NEXT

For *indice* = *valeurInitiale* to *valeurFinale*

Suite d'instructions

Next *indice*

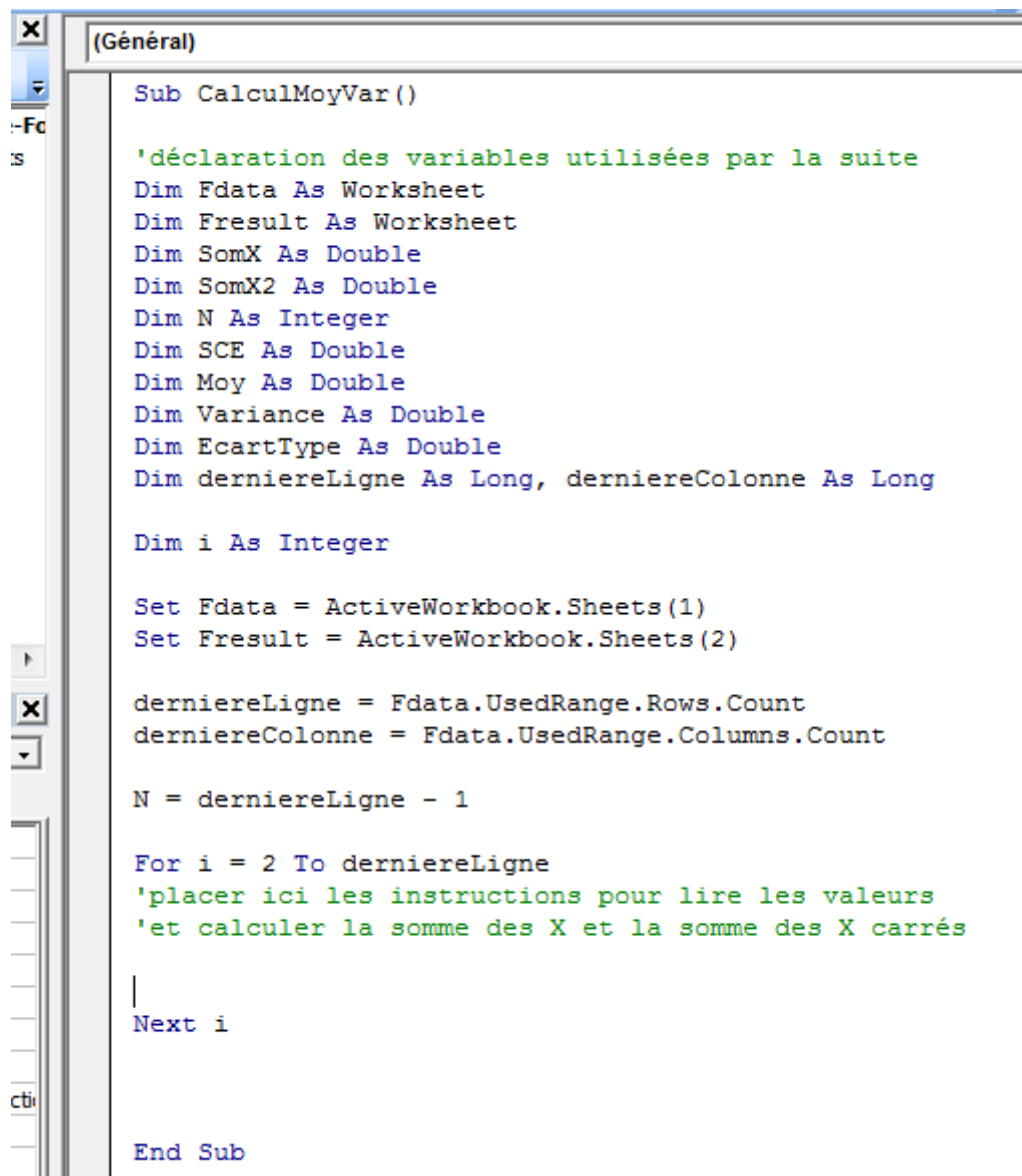
Fonctionnement de la boucle : lorsque le programme rencontre la première instruction For, il met la valeur de la variable *indice* à *valeurInitiale* puis il exécute les instructions jusqu'à la ligne Next *indice*. Il augmente la valeur de *indice* de 1 (par défaut) et vérifie que cette nouvelle valeur n'est pas supérieure à *valeurFinale*. Si *indice* est inférieur ou égal à *valeurFinale*, les instructions sont exécutées jusqu'à l'instruction Next *indice*. Ce cycle se répète jusqu'à ce que *indice* soit supérieur à *valeurFinale*. Le programme sort alors de la boucle et passe à l'instruction qui suit l'instruction Next *indice*.

Nous allons utiliser ce type de boucle pour parcourir les cellules de la feuille 1 qui contiennent les valeurs qui nous intéressent et effectuer les calculs préalables au calcul de la moyenne et de la variance.

Pour ces calculs, il nous faut la somme des X (que nous mettrons dans la variable SomX) et la somme des carrés de X (que nous mettrons dans la variable SomX2) Il nous faut aussi une nouvelle variable qui servira d'indice pour parcourir la plage de cellules

Voici comment se présente le programme

Question : comment calculer SomX et SomX2 ?



```
Sub CalculMoyVar()  
    'déclaration des variables utilisées par la suite  
    Dim Fdata As Worksheet  
    Dim Fresult As Worksheet  
    Dim SomX As Double  
    Dim SomX2 As Double  
    Dim N As Integer  
    Dim SCE As Double  
    Dim Moy As Double  
    Dim Variance As Double  
    Dim EcartType As Double  
    Dim derniereLigne As Long, derniereColonne As Long  
  
    Dim i As Integer  
  
    Set Fdata = ActiveWorkbook.Sheets(1)  
    Set Fresult = ActiveWorkbook.Sheets(2)  
  
    derniereLigne = Fdata.UsedRange.Rows.Count  
    derniereColonne = Fdata.UsedRange.Columns.Count  
  
    N = derniereLigne - 1  
  
    For i = 2 To derniereLigne  
        'placer ici les instructions pour lire les valeurs  
        'et calculer la somme des X et la somme des X carrés  
    Next i  
  
End Sub
```


Les tableaux de données

On a souvent besoin de ranger des valeurs dans un « tableau » comportant des lignes et des colonnes.

Ex :

DIM VarX(100,10) as Double

Création d'un tableau (en mémoire) de 100 lignes (ex : des sujets) et 10 colonnes (ex : des conditions de mesures ou des essais de pratique) pouvant contenir des nombres décimaux

Autre exemple :

DIM NomSujet(50) as String

Opérations sur les variables numériques

Addition : +

Soustraction : -

Multiplication : *

Division : /

Racine carrée : sqr(X)

Logarithme : Log(X)

Exposant : X^2

Partie entière d'un nombre : INT(X)

Sinus, cosinus, tangente : sin(X) cos(X) Tan(X)

Opérations sur les variables alphanumériques (chaînes de caractères)

Elles doivent être déclarées comme STRING

Ex :

DIM Mot as String

DIM Phrase as String

On peut les spécifier en mettant leur contenu entre guillemets :

Ex :

Mot = "Bonjour"

Phrase = "Comment allez vous ?"

On peut extraire des bouts de chaînes de caractère avec 3 fonctions :

Left, right, mid

Ex :

Mot = left(Phrase, 3) 'Mot contient les 3 caractères à gauche de Phrase ("Com")

Mot = Right(phrase,4) 'Mot contient les 4 caractère les plus à droite de Phrase ("us ?")

Mot = Mid(Phrase,2,5) 'Mot contient les 5 caractères à partir du 2^{ème} ("ommen")

On peut recherche si une chaîne se trouve dans une autre :

Dim Trouve as Long

Trouve = Instr(premCar,ChaineCible, ChaineCherchee)

Cette instruction recherche, en partant du caractère **premCar** si **ChaineCherchee** est inclus dans **ChaineCible**, et renvoie la position (**Trouve**) de sa première occurrence

Ex :

Mot = "vous"

Phrase = "Comment allez vous ?"

Trouve = Instr(1,Phrase, Mot) 'Trouve sera égal à 15

(note : si la recherche doit commencer au 1^{er} caractère, le paramètre peut être omis)

Si la chaîne cherchée ne se trouve pas dans la chaîne cible, la fonction renvoie 0

Autres procédures :

Phrase = UCASE(Phrase) 'met toute la chaîne en MAJUSCULES (Upper Case)

Phrase = LCASE(Phrase) 'met toute la chaîne en minuscules (Lower Case)

Parcourir un ensemble de données dans un tableau

Soit un tableau de données X(100) dont on veut calculer la moyenne

Une possibilité : calculer **Somme = X(1) + X(2) + X(3) + X(99) + x(100)**

Et ensuite : **Moyenne = Somme/100**

Pas terrible..., d'où l'intérêt de faire une boucle pour parcourir le tableau :

DIM i as integer

DIM Somme as Double, Moyenne as Double

Somme = 0

For i = 1 to 100

Somme = Somme + X(i)

Next i

Moyenne = Somme/100

i : indice de la variable dans le tableau

On verra plus loin d'autres types de boucle

Toutes les instructions doivent être incluses dans des procédures (sauf DIM qui est la seule à pouvoir apparaître en dehors d'une procédure)

Pour créer une procédure :

Menu Outils

Macros

Visual Basic Editor

Une feuille s'affiche où on va écrire les procédures

Format général :

SUB NomProcédure(nom_variable AS type_variable,...)

Instructions

END SUB

Si la procédure ne comporte pas de paramètres, elle apparaîtra dans le menu « macro ». Si elle comporte des paramètres, elle ne pourra être appelée que depuis une autre procédure

Exemple de procédure pour calculer la moyenne de N valeurs :

Sub CalculMoyenne (X() as double, N as integer, Moyenne as Double)

DIM i as integer

DIM Somme as Double

Somme = 0

For i = 1 to N

Somme = Somme + X(i)

Next i

Moyenne = Somme/N

End Sub

Exemple : les données sont dans la première colonne d'une feuille de calcul

« Feuil1 »

On veut pouvoir calculer leur moyenne et leur écart-type et placer ces informations dans « Feuil2 »

On crée d'abord une procédure qui pourra être appelée depuis le menu Outils..

Macros

Pour ce type de procédure, il ne doit pas y avoir de variables (: arguments) qui lui sont passés

Sub Moyenne ()

Dim FeuilDonnees as worksheet

Dim FeuilResultat as worksheet

Dim Nbre as integer

Dim Y(1000) as Double 'on prévoit large

Dim Moy as Double, Ecartype as Double

Dim i as integer

Set FeuilDonnees = ActiveWorkbook.Sheets("Feuil1")

Set FeuilResultat = ActiveWorkbook.Sheets("Feuil2")

Nbre = FeuilDonnees.UsedRange.Rows.Count

```
For i = 1 to Nbre
    Y(i) = FeuilDonnees.cells(i,1)
Next i
```

'on appelle la procédure de calcul de la moyenne déjà définie plus haut :
call CalculMoyenne(Y(), Nbre, Moy)

'on appelle la procédure de calcul de l'écartype qu'on va écrire plus loin
call calculEcartype(Y(), Nbre, Ecartype)

' on écrit les résultats

FeuilResultat.cells(1,1) = "Nombre de Données:"

FeuilResultat.cells(1,2) = Nbre

FeuilResultat.cells(2,1) = "Moyenne: "

FeuilResultat.cells(2,2) = Moy

FeuilResultat.cells(3,1) = "Ecart-Type: "

FeuilResultat.cells(3,2) = Ecartype

End sub

SUB calculEcartype(X() as double, N as integer, s as double)

DIM j as integer

DIM Somme as double

DIM Somme2 as double

For j = 1 to N

Somme = Somme + X(j)

Somme2 = Somme2 + X(j)*X(j)

Next i

S = sqr((somme2 - (Somme*Somme)/N)/(N-1))

end sub

vous remarquerez que les variables n'ont pas le même nom dans la procédure appelante (Y(),Nbre, Moy) et dans la procédure appelée (X(), N, Moyenne)

autres types de boucles :

boucle While...Wend

i = 0

while i < N

'ces instructions seront exécutées tant que i < N

i = i + 1

somme = somme + i

wend

boucle Do...Loop until

i=0

Do

i = i + 1

somme = somme + X(i)

Loop Until i = N 'quand i = N on sort de la boucle

Boucles imbriquées

Ex : une feuille contient les mesures d'une variable faites sur plusieurs sujets (en ligne) dans différentes conditions (en colonnes)

La première ligne contient le nom des mesures

La première colonne contient le nom des sujets

Sub LireDonnees()

Dim Nsujets as integer, Nmesures as integer

Dim X(100,50) as double

Dim Nomsujets(100) as string, NomMesures(50) as String

Dim i as integer, j as integer

Nsujets = ActiveSheet.UsedRange.Rows.Count - 1

Nmesures = ActiveSheet.UsedRange.Columns.Count - 1

For j = 1 to Nmesures

NomMesures(j) = cells(1,j+1)

' pas besoin de préciser la feuille si c'est la feuille active

Next j

For i = 1 to Nsujets

Nomsujets(i) = cells(i+1,1)

For j = 1 to Nmesures

X(i,j) = cells(i+1,j+1)

next j

next i

End sub

Informations sur les classeurs (wokbook) et les feuilles (Sheets)

'combien de feuilles dans un classeur ?

'mettre leurs nom dans les cellules d'une feuille

Dim Wb as workbook 'la variable Wb fera référence à un classeur

Dim f As Worksheet 'la variable f fera référence à une feuille

Dim Nf as integer, i as integer 'ces 2 variables contiendront des nombres entiers

Set Wb = activeWorkbook 'Wb fait maintenant référence au classeur actif

Nf = Wb.Sheets.Count 'Nf contient maintenant le nombre de feuilles de Wb

Set f = Wb.ActiveSheet 'f fait référence à la feuille active de Wb

For i = 1 To Nf

f.Cells(i, 1) = " feuille " & i 'on place dans la colonne 1 de la ligne i le mot Feuille et le numéro i qui lui correspond

f.Cells(i, 2) = Wb.Sheets(i).Name 'on place dans la colonne 2 de la ligne i le nom de la feuille correspondante

Next i

'ajouter une feuille au classeur Wb

Wb.Sheets.Add

Dialogue avec l'utilisateur

Donner une information à l'utilisateur :

Fonction **MsgBox(message)**

Exemple :

MsgBox(« opération terminée »)

Une boîte de dialogue apparaît, et reste visible jusqu'à que l'utilisateur clique « ok »

(on peut créer d'autres formes de dialogue avec MsgBox : voir aide dans VisualBasic)

Demander une information à l'utilisateur :

Fonction **InputBox(message, titre, défaut)**

Sub DemandeNombre(Nbre As Integer)

Dim Message as string, Titre as string, Defaut as string, Reponse as string

Message = "Entrez le nombre voulu :"

Titre = "Démonstration de InputBox" ' Définit le titre.

Defaut = "1" ' Définition la valeur par défaut.

' Affiche le message, le titre et la valeur par défaut.

Reponse = InputBox(Message, Titre, Defaut)

'attente d'un click sur OK

Nbre = Val(Reponse)

End sub

instructions conditionnelles

Certaines instructions ne doivent être exécutées que si une certaine condition est vraie

```
IF condition THEN  
    Instructions à exécuter si la condition est vraie  
Else  
    Instructions à exécuter si la condition est fausse (optionnel)  
END IF
```

Exemple : calculer la moyenne des valeurs de la colonne 1 en ignorant les cellules vides :

```
Sub MoyenneIgnoreVide()  
Dim FeuilDonnees as worksheet  
Dim FeuilResultat as worksheet  
DIM i as integer  
Dim Nbre as integer, N as integer, Moy as double, ecartype as double  
Set FeuilDonnees = ActiveWorkbook.Sheets("Feuil1")  
Set FeuilResultat = ActiveWorkbook.Sheets("Feuil2")  
Nbre = FeuilDonnees.UsedRange.Rows.Count  
N=0  
For i = 1 to Nbre  
    If FeuilDonnees.cells(i,1) <> "" then  
        'ou if IsNumeric(FeuilDonnees.Cells(i, 1)) then  
            'instructions exécutées uniquement si la condition est vraie  
        N=N+1  
        Y(N) = FeuilDonnees.cells(i,1)  
    End if  
    'à la sortie de la boucle, N sera égal au nombre de cellules non vides  
Next i  
call CalculMoyenne(Y(), N, Moy)  
call calculEcartype(Y(), N, Ecartype)  
    ' on écrit les résultats  
FeuilResultat.cells(1,1) = "Nombre de Données:"  
FeuilResultat.cells(1,2) = N  
FeuilResultat.cells(2,1) = "Moyenne: "  
FeuilResultat.cells(2,2)= Moy  
FeuilResultat.cells(3,1) = "Ecart-Type: "  
FeuilResultat.cells(3,2) = Ecartype  
End Sub  
Note : <> signifie « différent » : not equal
```

Note : lorsqu'une variable est dimensionnée dans une procédure, elle n'existe que pour cette procédure

Pour qu'une variable soit « visible » par toutes les procédures, il faut la déclarer (dimensionner) en dehors d'une procédure, en tête de la feuille Visual basic

Coloriage de cellules

Comment connaître les index des couleurs

Sub couleur()

Dim i As Integer

Dim MaFeuille As Worksheet

Set MaFeuille = ActiveWorkbook.Sheets("Feuil1")

For i = 1 To 56

MaFeuille.Cells(i, 1) = i

MaFeuille.Cells(i, 2).Interior.ColorIndex = i

Next i

End Sub