

Objectifs du module :

Utiliser de façon rationnelle un tableur et ses fonctions principales. Savoir traiter dans un langage structuré un problème simple. (Nous utiliserons le langage orienté objet : " Visual Basic Application").

Emploi du temps : 5 séances de 4h de TP (contrôle de connaissance inclus)

Mode de contrôle des connaissances : 1 contrôle de TP individuel lors du dernier TP

Cours MOODLE : IUTA > IUT G.M .P > BUT1 -G.M.P > S2 > R2.12 Informatique et Base de données > R2.12 Info, clé : cmhiht.

La programmation structurée

C'est une méthode de réflexion permettant de résoudre un problème par informatique faisant appel à un théorème n'ayant jamais été démenti:

Tout problème à traiter par informatique peut être décrit à l'aide de trois structures nécessaires et suffisantes:

- La Séquence
- La condition
- La répétition

```
Sub macro  
Dim variable1 as integer  
Dim variable2 as string  
Dim variable3 as double  
...  
End sub
```

L'instruction Si se traduit par :

```
If condition Then  
..  
..  
Else  
...  
...  
End If
```

L'instruction Répéter .. Jusqu'à se traduit par :

```
Do  
.....  
.....  
.....  
Loop until condition
```

L'instruction faire pour I allant de 1 à 10 se traduit par :

```
dim I as integer  
for I = 1 to 10  
.....  
.....  
.....  
next I
```

T.P.1 : Mise en place d'une macro

1 / Ouvrir un Classeur dans le tableur "Calc"

2 / Cliquer avec Bouton droit sur Feuille 1 (en bas à gauche) , la renommer " F1 " . C'est dans cette feuille que nous allons travailler

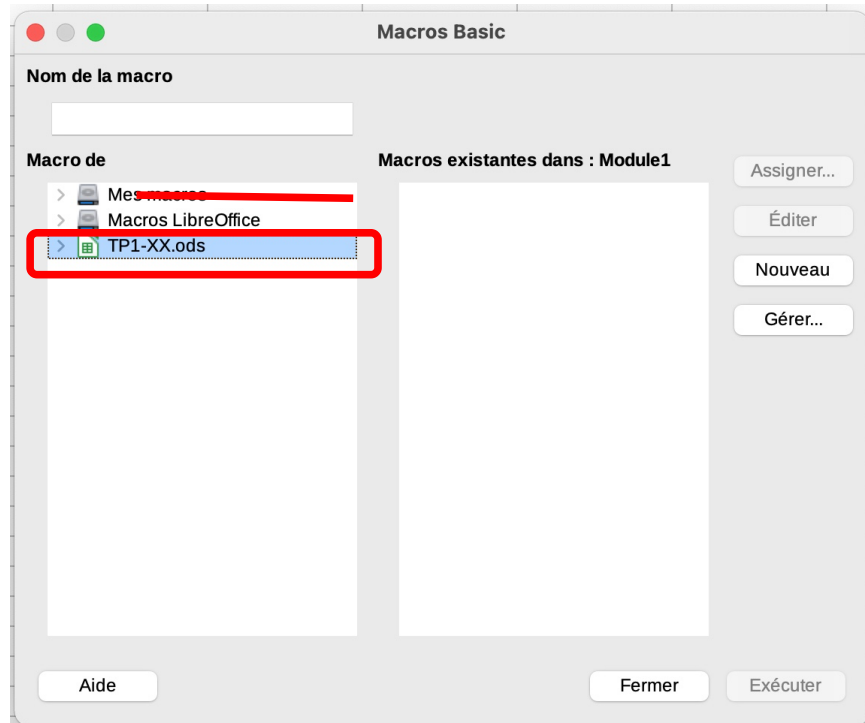
3/ Enregistrer le classeur vide sur votre bureau (*Fichier-Enregistrer-sous*) au format par défaut .ODS sous le nom " TP1 " suivi de vos initiales par exemple : TP1-XX

A : Ecriture d'un programme Basic associé à la feuille F1

ATTENTION : Suivre exactement les instructions suivantes afin d'associer votre programme à votre classeur

1/ Pour ouvrir la fenêtre de programmation en " Basic Objet " , sélectionner **Outils - Macro - Gérer les Macros - Open Office Basic** puis dans le cadre de gauche " Macro de " , développer votre classeur " TP1- XX .ods " , sélectionner " Standard " - Cliquer sur " Nouveau " , il propose **Module1** , cliquer **OK** .

ATTENTION NE PAS SELECTIONNER l'onglet 'mes macros' (barré dans figure ci-dessous).




La fenêtre de programmation en Basic apparaît avec des lignes déjà écrites (Sub Main),

Effacer tout le texte.

2/ Taper alors le premier sous-programme à tester . (sous-programme se traduit " **Sub**Routine en Anglais)

il n'est pas nécessaire de recopier les commentaires en rouge.


```
Option Explicit      ( OBLIGATOIRE - L'interpréteur Basic vérifie la cohérence des variables )  
Sub DisBonjour      ( Nom du sous-programme )  
Dim Nom as string  ( Déclaration de la variable " Nom " en chaîne de caractères )  
Nom=inputBox("Quel est votre nom ? ")  
MsgBox( "Alors , Bonjour " & Nom ) ( & : séparateur de termes à afficher s )  
end sub
```

3/ Faire exécuter le programme en appuyant sur " F5 " ou en cliquant sur  . Corriger les erreurs si nécessaire. Lorsque cela fonctionne , enregistrer la feuille (**Fichier - Enregistrer**) qui comprend également le programme que vous venez de taper.

B: Création d'un Bouton de contrôle

Le programme étant mis au point, nous allons maintenant créer un bouton permettant de faire exécuter ce programme depuis la feuille " F1 "

1/ Dans " **Fenêtre** " cliquer sur "TP1-XX .ods ", la feuille **F1** apparaît.

2/ Sélectionner " Affichage " - " Barres d'outils " - et cliquer dans " **Contrôles de Formulaire** " puis activer le mode le mode " Ebauche " en cliquant sur l'icône suivant : 

3/ Cliquer dans l'icône " **Bouton** " et dessiner le bouton sur la feuille " F1" (*il faut faire glisser le pointeur de la souris sur les icônes et attendre une seconde pour que le nom de l'icône apparaisse*)

4/ Cliquer à l'intérieur du bouton avec le bouton droit- Sélectionner " **Contrôles** " puis " **Général** "

5/ Modifier le titre du bouton (étiquette), en le nommant " **BONJOUR** ", agrandir la taille des caractères

6/ Sélectionner " **événements** " , " **exécuter l'action** " - (bouton à droite du texte) 

7/ Cliquer sur le bouton " **Macro** " puis dans *TP1-XX.ods - Standard - Module1*, Le nom de la procédure **DisBonjour** apparaît alors dans la fenêtre de droite . Sélectionner " DisBonjour " puis cliquer sur **OK**

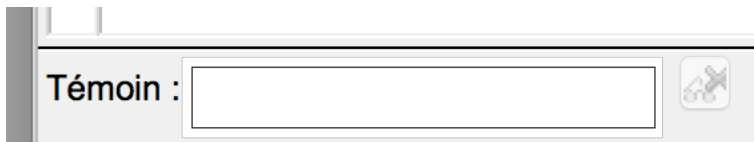
8/ Désactiver le mode conception avec le même icône ayant permis de l'activer et essayer le programme **DisBonjour** qui doit démarrer en cliquant sur votre bouton " **Bonjour** "

9 / Ajouter quelques lignes afin de demander l'âge de l'utilisateur avec un test donnant un message d'erreur si l'âge est négatif ou supérieur à 140 ans.

```
Option Explicit      ( OBLIGATOIRE - L'interpréteur Basic vérifie la cohérence des variables )
Sub DisBonjour      ( Nom du sous-programme )
Dim Nom as string   ( Déclaration de la variable " Nom " en chaîne de caractères )
Dim Age as integer  ( Déclaration de la variable " Age " en entier )
Nom=inputBox("Quel est votre nom ? ")
Age = inputBox("Quel est votre age ? ")
if (Age > 140) then
    MsgBox (" erreur ")
end if
MsgBox( "l'age de " & Nom & " est " & Age)
end sub
```

10 / Tester le programme.

Pour ce faire, ajouter la variable **Age** dans la case Témoin ,



puis exécuter le programme en mode pas à pas en tapant sur la touche F8 ou sur la touche



11/ Modifier le programme avec un test donnant un message d'erreur si l'âge est négatif ou supérieur à 140 ans.

T.P.2 : Algorithmie

Création du classeur TP2-XX et de la feuille de travail

- 1 / Ouvrir un **Classeur** dans le tableur "Calc"
- 2/ Cliquer avec Bouton droit sur Feuille 1 , la renommer " **F1** ". Créer une seconde feuille nommée " **F2** "
- 3/ **Enregistrer le classeur sur votre bureau sous le nom " TP2-XX " avec XX les initiales de vos noms.**

Exercice TP2_1 TP2_1 - **Calcul élémentaire sur entiers**

Ecrire un programme qui demande un nombre à l'utilisateur, puis qui calcule et affiche le cube de ce nombre. Une solution à compléter dans la ou les lignes noires :

Option explicite

Sub TP2_1

REM déclaration des variables

dim nombre as integer

dim cube as integer

REM programme

nombre = inputbox ("taper un nombre")

*cube = nombre * nombre * nombre*

msgbox ("le cube de " & nombre & " est "& cube)

End Sub

Que se passe-t-il quand 'nombre' = 100 ?

Modifier la déclaration de 'cube' par *dim cube as long*

Refaire le teste quand 'nombre' = 100.

Exercice TP2_2 TP2_2 - Affectation

Ecrire un algorithme qui saisit deux entiers A et B et qui échange leurs valeurs.

Option explicite

Sub TP2_2

dim a,b,temp as integer

a = inputbox ("taper la valeur de a")

b = inputbox ("taper la valeur de b")

msgbox ("Avant échange a =" & a & " b=" & b)

temp = a

msgbox ("Après échange a =" & a & " b=" & b)

End Sub

Exercice TP2_3 TP2_3 - Conditions

Ecrire un algorithme qui affiche "glace" si la température saisie est inférieure ou égale à 0, affiche "liquide" si la température est comprise entre 1 et 100 et "vapeur" si la température est supérieure à 100.

```
option explicit
Sub TP2_3
dim temp as integer

temp= inputbox ("taper temperature")

if (temp <=0) then
    msgbox ("glace")
else
    if [REDACTED] then
        [REDACTED]
    else
        [REDACTED]
    end if
end if
msgbox ("fin")

End Sub
```


Exercice TP2_4 TP2_4 - Conditions avec opérateurs logiques

Ecrire un algorithme qui demande deux nombres entiers à l'utilisateur et l'informe ensuite si leur produit est nul, négatif ou positif (on ne laisse pas de côté le cas où le produit est nul). Attention toutefois : on ne doit **pas** calculer le produit des deux nombres.

Une solution à compléter :

```
Option explicit
sub TP2_4
dim a as integer
dim b as integer
a = inputbox ("taper un entier svp")
b = inputbox ("taper un entier svp")

if ((a=0) OR (b=0)) then
    msgbox ("produit nul")
else
    if [REDACTED] then
        msgbox ("produit positif")
    else
        [REDACTED]
    end if
end if
msgbox ("fin")
end sub
```

Exercice TP2_5 TP2_5 - Conditions avec opérateurs logiques et ordre des valeurs

Ecrire un algorithme qui demande l'âge d'un enfant à l'utilisateur. Ensuite, il l'informe de sa catégorie : "Poussin" de 6 à 7 ans, "Pupille" de 8 à 9 ans, et "autre" avant 6 ans et après 9 ans.

```
option explicit
Sub TP2_5
dim age as integer
age= inputbox ("taper age de l'enfant")
if ((age <6) or (age > 9)) then
    msgbox ("autre")
else
    [REDACTED]
        msgbox ("poussin")
    [REDACTED]
        msgbox ("pupille")
    [REDACTED]
end if
msgbox ("fin")
End Sub
```

Exercice TP2_6 TP2_6 - Conditions sur caractères

Ecrire un algorithme qui demande à l'utilisateur de taper un nom d'animal. Puis le programme demande si c'est un Poisson ou un Mammifère (taper "P" ou "M"). Il écrit ensuite sur l'écran " Le xxxx est un joli *mpmpmp*".

Par exemple : "Le cheval est un joli mammifère" si l'utilisateur a saisi "M".

"Le cheval est un joli autre" si l'utilisateur n'a tapé ni "M" ni "P."

Option explicite

Sub TP2_6

dim animal as string

dim categorie as string

dim affiche as ████████

animal= inputbox ("taper le nom d'un animal")

categorie = inputbox (" taper P pour poisson ou M pour mammifère")

affiche = "autre"

if (categorie = "P") then

affiche = "poisson"

else

████████████████████

██

████████████████

end if

msgbox (animal & " est un joli " & affiche)

End Sub

Exercice TP2_8 TP2_8 - Itération sur nombre avec do...loop until ()

Ecrire un algorithme qui demande un nombre de départ, et qui ensuite affiche les dix nombres suivants. Par exemple, si l'utilisateur entre le nombre 17, le programme affichera les nombres de 18 à 27.

```
Sub TP2_8
dim dep as integer
dim nb as integer
dim indice as integer
dep = inputbox ("taper un nombre de départ")
indice =1
do
    [REDACTED]
    [REDACTED]
    [REDACTED]
loop until (indice > 10)
msgbox ("fin")

End Sub
```

Exercice TP2_9 TP2_9 - Itération sur nombre avec for

Ecrire un algorithme qui demande un nombre de départ, et qui ensuite affiche les dix nombres suivants. Par exemple, si l'utilisateur entre le nombre 17, le programme affichera les nombres de 18 à 27.

```
Sub TP2_9
dim dep as integer
dim nb as integer
dim indice as integer
dep = inputbox ("taper un nombre de départ")

for indice = 1 to 10
    [REDACTED]
    msgbox (nb)
next indice

msgbox ("fin")

End Sub
```

Exercice TP2_10 TP2_10 - Test itératif de conformité avec do...loop until ()

Ecrire un algorithme qui demande un nombre compris entre 10 et 20, jusqu'à ce que la réponse convienne. En cas de réponse supérieure à 20, on fera apparaître un message : " Plus petit ! ", et inversement, " Plus grand ! " si le nombre est inférieur à 10.

option explicit

Sub TP2_10

dim nombre as integer

do

nombre = inputbox ("taper un nombre compris entre 10 et 20")

if (nombre <10) then

[REDACTED]

else

if [REDACTED] then

[REDACTED]

end if

end if

loop until [REDACTED]

msgbox ("fin du programme")

End Sub

Exercice TP2_11 TP2_11 - Statistiques en ligne avec do...loop until ()

Ecrire un algorithme qui saisit successivement des entiers. La boucle s'arrête quand l'entier saisi est négatif. L'algorithme doit retourner la valeur max des entiers saisis ainsi que la moyenne (la dernière valeur saisie qui est négative ne doit pas être prise en considération).

Exemple: entiers saisis: 2, 8, 3, 3, -1. Max = 8, Moyenne = 16/4=4.

```
option explicit
Sub TP2_11
dim n as integer
dim max as integer
dim somme as integer
dim nb_saisi as integer
dim moy as double
somme=0
max =-1
nb_saisi =0
do
    n =inputbox ("nouvel entier")
    if (n> max) then
        [REDACTED]
    end if
    if (n >=0) then
        [REDACTED]
    end if
loop until ([REDACTED])
if (nb_saisi >0) then
    [REDACTED]
    [REDACTED] msgbox ("max=" & max & " moyenne=" & moy)
else
    msgbox ("pas d'élément")
end if
End Sub
```


Exercice TP2_12 TP2_12 - Statistiques sur tableau

Ecrire un algorithme qui calcule la valeur moyenne (ou max) des 3 notes saisies dans le tableau. Les notes doivent être comprises entre 0 et 20 lors de la saisie.

```
option explicit
Sub TP2_12
dim notes (1 to 3) as integer
dim i, somme as integer
dim moy as single
REM saisie
for i=1 to 3
    do
        notes (i) = inputbox ("taper la note numero" & i)
    loop until ((notes (i)>= 0) and [REDACTED])
next i

REM moyenne
somme=0
for i=1 to 3
    somme= notes (i) [REDACTED]
[REDACTED]
[REDACTED]
moy= somme / 3
msgbox (" moyenne" & moy)
End Sub
```

Exercice TP2_13 TP2_13 - Indice du tableau

Le salaire horaire d'un employé dépend de sa catégorie d'après le tableau suivant:

Catégorie	Salaire horaire
1	5,5
2	7
3	8,2
4	10

Faire un programme qui calcule le salaire d'un employé après avoir saisi sa catégorie et le nombre d'heures travaillées.

Par exemple, pour un employé de catégorie 1 qui a travaillé 20 heures, le salaire sera égal à 140 euros.

```
option explicit
Sub TP2_13
dim horaire (1 to 4) as single
dim salaire as double
dim heures as integer
dim cat as integer
horaire (1) = 5.5
[REDACTED]
[REDACTED]
horaire (4) = 10
do
    cat = inputbox ("categorie entre 1 et 4")
loop until ((cat >= 1) and [REDACTED])
do
    heures =inputbox ("nombre d'heures")
loop until ([REDACTED])
salaire = [REDACTED] * horaire (cat)
msgbox ("Pour la catégorie" & cat & "et " & heures & "heures" )
msgbox ("Salaire = " & salaire)
End Sub
```

T.P.3 : Nouvelles fonctions

Création du classeur TP3-XX et de la feuille de travail

1 / Ouvrir un **Classeur** dans le tableur "Calc"

2/ Cliquer avec Bouton droit sur Feuille 1 , la renommer " **F1** ". Créer une seconde feuille nommée " **F2** "

3/ Enregistrer le classeur sur votre bureau sous le nom " TP3-XX " avec XX les initiales de vos noms.

Nous allons créer une fonction permettant de calculer le volume d'un cône à partir du rayon de la base (R) et de sa hauteur (H). Dans la fenêtre de programmation Basic (à la suite du programme précédent) taper la fonction suivante :

Function VolCone(R as single, H as single) as single

VolCone=Pi*R^2*H/3

End Function

Dans la feuille " **F1** ", nous allons créer une colonne intitulée Rayon variant de zéro à 3 par pas de 0,1, une seconde colonne avec la hauteur h=2 constante et une troisième colonne qui utilise la fonction VolCone pour afficher le volume

	A	B	C	
1	Rayon	Hauteur	Volume Cône	
2				
3	0	2	0	
4	0,1	2	0,02	
5	0,2	2	0,08	
6	0,3	2	0,19	
7	0,4	2	0,34	
8	0,5	2	0,52	
9	0,6	2	0,75	

Pour parvenir à ce résultat effectuer les opérations suivantes :

1/ Entrer les titres des trois

colonnes et fixer le format des nombres à 4 décimales (**Format-Cellules**)

2/ Dans la colonne Rayon, taper 0 puis 0,1 en dessous , sélectionner ces deux cellules avec la souris, saisir ensuite la poignée en bas à droite et la tirer vers le bas. Cette opération permet de remplir automatiquement la colonne avec des valeurs croissantes suivant l'incrément indiqué.

3/ Effectuer le même traitement dans la colonne Hauteur en tapant au départ deux fois le même nombre (1) .

4/ Dans la colonne Volume Cône se positionner sur la première case à remplir (C3) et taper : = VolCone(

puis cliquer sur la cellule correspondant au rayon , frapper la touche point-virgule cliquer ensuite sur la cellule correspondant à la hauteur . Enfin fermer la parenthèse .

On doit ainsi obtenir : =VOLCONE(A3;B3) et la valeur zéro s'affiche après validation.

5/ Afin de recopier automatiquement cette fonction dans le reste de la colonne, cliquer sur la cellule, saisir la poignée et tirer vers le bas. On doit alors obtenir les valeurs données sur la figure précédente.

Travail à faire par vous même:

Q1 / Dans la même feuille ajouter une colonne (D) intitulée Volume Sphère et écrire une fonction permettant de calculer et afficher dans cette colonne le volume d'une sphère de rayon correspondant aux valeurs indiquées dans la colonne A.

	A	B	C	D
1	Rayon	Hauteur	Volume Cône	Volume Sphère
2				
3	0,0000	2,0000	0,0000	0,0000
4	0,1000	2,0000	0,0209	0,0042
5	0,2000	2,0000	0,0838	0,0335
6	0,3000	2,0000	0,1885	0,1131
7	0,4000	2,0000	0,3351	0,2681
8	0,5000	2,0000	0,5236	0,5236
9	0,6000	2,0000	0,7540	0,9048

Q2 / Renommer la feuille 2 en " F2 " et ouvrir cette feuille . Créer une fonction en basic intitulée " TotHt " acceptant deux données en entrée : le nombre d'article Nart et le prix unitaire Punit et retournant en sortie le prix total hors taxes . Ce qui donne pour en-tête de la fonction :

Function TotHT(Nart As Integer, Punit As Single) As Single

Une réduction de 5% doit être appliquée si le nombre d'articles est > 10 . Essayer la fonction en créant un tableau comme ci-dessous.

Quantité	Prix unitaire HT	Prix Total HT
2	10,00 €	20,00 €
3	15,00 €	45,00 €
12	10,00 €	114,00 €
100	2,00 €	190,00 €
.....

T.P 4 : Notions d'Objet en Visual Basic

DIDACTICIEL

A/ : Création du classeur TP4-XX et de la feuille de travail

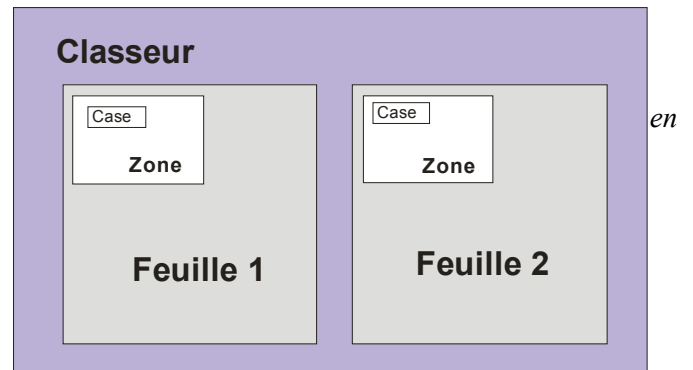
1 / Ouvrir un **Classeur** dans le tableur "Calc"

2/ Cliquer avec Bouton droit sur Feuille 1 , la renommer " **F1** ". Créer une seconde feuille nommée " **F2** "

3/ Enregistrer le classeur sur **votre bureau** sous le nom " **TP4-XX** " avec X X les initiales de vos noms.

B/ : Lecture-Ecriture de données dans la feuille " F1 "

Un **classeur** est un objet qui contient plusieurs **feuilles** identifiées par leur nom (par ex. **F1** , **F2** ..). Chaque feuille contient des **cellules** identifiables par **Zone** (par ex. "A1:D10" diagonale) ou individuellement (par ex 0,0 qui correspond à la case A1)



1/ Ouvrir la fenêtre de programmation en " Basic " (voir TP 1)

2/ Déclarer les variables " Objet " qui seront utilisées avec l'option " explicit " soit :

Option explicit

Dim Classeur As Object

Dim Feuilles As Object *(Un classeur peut contenir plusieurs feuilles)*

Dim FTravail1 As Object *(La feuille de travail N°1)*

Dim Zone1 As Object *(La zone de cases qui pourront être utilisées)*

Dim Cell As Object *(La case qui sera utilisée)*

Dim Cell1 As Object

Dim Cell2 As Object

3/ Taper le sous-programme suivant qui permet d'affecter les variables

! Attention aux majuscules et minuscules des mots entre guillemets ! Exemple " f1 " au lieu de " F1 "

Sub DefEspaceTravail

Classeur = ThisComponent *(Cela signifie le classeur ouvert actuellement)*

Feuilles = Classeur.Sheets

FTravail1 = Feuilles.getByname("F1")

Zone1= FTravail1.getCellrangebyname("a1:p30")

end sub

*On choisit la zone de cellules a1-p30,
le programme ne pourra pas travailler
en dehors)*

4/ Taper le sous-programme suivant qui écrit le mot " **Nom** " dans la Cell A1 , vous demande votre nom et l'écrit dans la case B1

Sub EcrireDansCase

Dim x As String

DefEspaceTravail

(Exécution du sous-programme précédent)

Cell = Zone1.getCellByPosition(0,0)

(Défini position de la Case en 0,0 =>A1)

Cell.String = " NOM : "

(Ecris le mot " NOM " dans case A1)

x = InputBox (" Taper votre Nom ")

(Affectation du nom tapé dans la variable X)

Cell = Zone1.getCellByPosition(1,0)

(Défini position de la Case en 1,0 =>B1)

Cell.String = x

(Copie contenu de X dans case B1)

end sub

➡ Pour tester ce sous-programme, créer un bouton (voir TP précédent) nommé " Taper NOM "
et associer ce bouton a ce sous-programme.

➡ Modifier le programme précédent pour qu'il écrive simultanément le nom dans les cases : A3, B1 et B4.

C / Génération automatique d'une colonne de nombres

Ce sous-programme permet de remplir automatiquement la colonne C de la ligne 1 à la ligne 10

Sub RemplirColonneC

Dim i As Long

DefEspaceTravail

For i = 0 To 9

Cell = Zone1.getCellByPosition(2,i)

Cell.Value =2* i

Next i

End sub

➡ Pour tester ce sous-programme, créer un bouton nommé " Remplir "

➡ Modifier le programme précédent pour remplir avec les mêmes nombres, la ligne 3 depuis la colonne A jusqu'à la colonne K .

D / Copie de Cases

Ce sous-programme permet de copier le contenu de la case B5 (Col 1, Lign 4) dans la case B6 (Col 1, Lign 5)

Sub CopierCase

DefEspaceTravail

Cell1 = Zone1.getCellByPosition(1,4) ' Cell B5

Cell2 = Zone1.getCellByPosition(1,5) ' Cell B6

Cell2.Value = Cell1.Value

End sub

*Pour copier une chaîne de caractères on écrit : “ **Cell2.String = Cell1.String** ”*

➡ Recopier ce sous-programme, créer un bouton nommé "Copier Case " et l'exécuter

Travail à faire par vous même :

1/ Copie de la case courante dans une case fixe (a15 par exemple)

La case courante est la case sélectionnée à l'aide de la souris ou des touches de déplacement (flèches) . Dans l'exemple suivante pour que la variable objet nommée CaseCourante contienne la donnée en cours , **il suffit d'ajouter la ligne suivante au bon endroit dans le programme " CopierCase "** .

```
CellCourante = ThisComponent.getCurrentSelection
```

Ne pas oublier de déclarer cette nouvelle variable: **CellCourante en objet (as Object)**

Pour essayer ce sous-programme il suffit de sélectionner une cellule contenant un nombre puis de cliquer sur le bouton " Copier Case " , le contenu de la case courante apparaît alors dans a15 .

➡ Modifier les paramètres de la cellule a15 (police 24 , couleur rouge par ex.) pour améliorer la visibilité.

2 / Copie entre deux feuilles du même classeur

A/ Modifier le programme précédent afin qu'il copie le contenu de la **case courante** de la feuille **F1** dans la cellule " **A1** " de la feuille **F2**.

Pour cela il faudra déclarer deux nouvelles variables objet , **Ftravail2 et Zone2** au même endroit que la déclaration des variables Ftravail1 et Zone1.

Puis il faudra compléter la procédure **DefEspaceTravail** avec les lignes :

```
FTravail2 = Feuilles.getByname("F2")
```

```
Zone2= Ftravail2.getCellRangeByName("a1:p30")
```

B/ Modifier encore ce programme afin qu'il copie une liste de noms se trouvant dans les cases B2 à B11 de la feuille **F1** dans les mêmes cases de la feuille **F2** (on pourra s'inspirer du programme écrivant une suite de nombres)

3 / Table de Pythagore

Ecrire un programme permettant de générer dans la feuille **F2** du tableur une table de pythagore de 10 lignes et 10 colonnes- (table de multiplication) (voir le cours)

4 / Gestion des notes d'un groupe d'étudiants

Ajouter une feuille nommée en "**NOTES**" dans le classeur et travailler dans cette feuille

a / Ecrire un sous-programme activé par un bouton " Saisie Noms " permettant d'entrer au clavier les noms des étudiants et les écrire dans la colonne B . (Utiliser " InputBox "). Le nombre de noms d'étudiants à saisir est demandé auparavant . Dans la colonne A faire écrire un nombre permettant de connaître le numéro d'ordre de l'étudiant (1,2,3,).

Pour ce faire, déclarer la variable contenant le nombre d'étudiants à l'extérieur des procédures :

Global Nbetudiants as integer

b / Ecrire un sous-programme activé par un bouton " Notes " permettant d'entrer au clavier les notes des **Nbetudiants** étudiants . On utilisera une méthode ne permettant d'écrire que des nombres compris entre 0 et 20 . Le sous programme doit accepter la note jusqu'à ce qu'elle soit comprise entre 0 et 20.

c / Ajouter un sous-programme et un bouton " Moyenne " donnant la moyenne du groupe.

5 / Codage binaire

- Ouvrez le fichier 'TP1-mot-binaire-etudiant.ods' et exécutez de la macro exemple 'Complémenter A'.
- Les mots A (cellules A2 à D2), et B (cellules A5 à D5) sont en jaune et vert. Le mot R est en bleu (cellules A8 à D8).

A FAIRE : Complétez le fichier 'TP1-mot-binaire-etudiant.ods' avec les différentes macros suivantes qui seront appelées par des boutons (ne pas oublier de compléter les macros DefMots et Testbinaire)

- Bouton « Ou » : $R = A \text{ or } B$.
- Bouton « ET » : $R = A \text{ and } B$.
- Bouton « Ou Exclusif » : $R = A \text{ xor } B$.
- Conversion de A en base 10 noté A_{10} dans la cellule E2 : $E2 = 8 * A2 + 4 * B2 + 2 * C2 + D2$
- Bouton « Incrémentation » : Incrémentation du mot A ($A = A \text{ plus } 1$) Quand $A_{10} > 15$, A_{10} prend la valeur 0 - Exemple avant appel de la macro $A = (0010)$ – après appel de la macro $A = 0011$.
- Bouton «Addition» : $R = A \text{ plus } B$.

Remarque : les étapes d'incrémentation et d'addition nécessitent de convertir le mot de 4 bits en une valeur entière comprise entre 0 et 15, puis de faire l'opération en base 10, et de convertir l'entier ainsi modifié en un mot de 4 bits. Quand l'entier modifié est égal à 16 (ou -1), on le modifie à 0 (15).

Lire l'aide dans les 3 pages suivantes

Rappel de logique et aide

Opérations binaires

a	b	a OR b	a AND b	a XOR b
0	0	0	0	1
0	1	1	0	0
1	0	1	0	0
1	1	1	1	1

Conversion base 10 et base 2

Système décimale	Système octal	Système binaire
0	0	0
1	1	1
2	2	10
3	3	11
4	4	100
5	5	101
6	6	110
7	7	111
8	10	1000
9	11	1001
10	12	1010
11	13	1011
12	14	1100
13	15	1101
14	16	1110
15	17	1111

Incrémentation : $I = A+1$

A3	A2	A1	A0	I3	I2	I1	I0
0	0	0	0	0	0	0	1
0	0	0	1	0	0	1	0
0	0	1	0	0	0	1	1
0	0	1	1	0	1	0	0
0	1	0	0	0	1	0	1
0	1	0	1	0	1	1	0
0	1	1	0	0	1	1	1
0	1	1	1	1	0	0	0
1	0	0	0	1	0	0	1
1	0	0	1	1	0	1	0
1	0	1	0	1	0	1	1
1	0	1	1	1	1	0	0
1	1	0	0	1	1	0	1
1	1	0	1	1	1	1	0
1	1	1	0	1	1	1	1
1	1	1	1	0	0	0	0

Conversion binaire ->décimale

REM A Conversion binaire -> décimal

```
valA10 = 0
puis = 1
i=0
do
    valA10=valA10 + puis*CellA(i).value
    puis = puis *2
    i = i+1
loop until (i >3)
```

Conversion décimale ->binaire

```
i=0
do
    REM modulo 2
        CellA(i).value = val10 mod 2
    REM division entière
        val10=val10\2
        i = i+1
loop until (i >3)
```

T.P 5 : Création et gestion de boites de dialogue

Création du classeur TP5-XX et de la feuille de travail

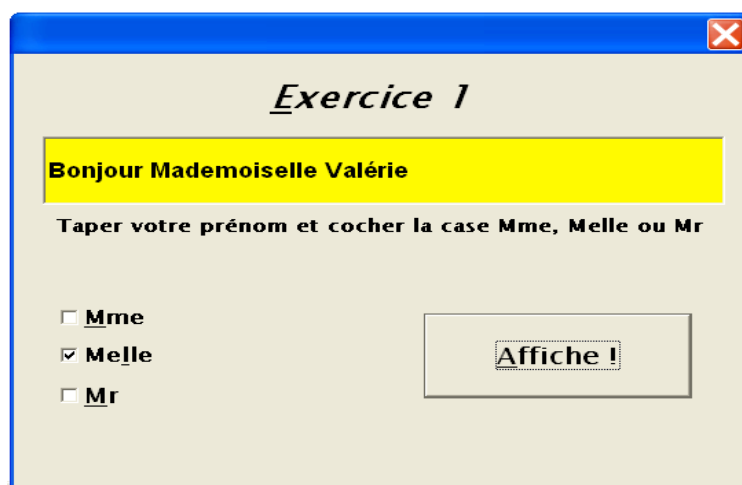
1 / Ouvrir un **Classeur** dans le tableur "Calc"

2/ Cliquer avec Bouton droit sur Feuille 1 , la renommer " **F1** ". Créer une seconde feuille nommée " **F2** "

3/ *Enregistrer le classeur sur votre bureau sous le nom " TP5-XX " avec XX les initiales de vos noms.*

Création d'une boîte de dialogue comme représentée ci-dessous

A/ Dessin de la boite de dialogue



1/ Choisissez Outils - Macros - **Gérer les Boîtes de Dialogue** .

2/ Dans la catégorie "TP5-XX", cliquez sur "Standard, puis sur " **Nouveau** " .

3/ Nommer cette boite " **Boite1** " et cliquez sur **OK** puis **Éditer**. La fenêtre **BASIC** s'ouvre et présente une boîte de dialogue vide ainsi qu'une **boîte d'outils** semblable aux outils déjà utilisés pour créer des boutons. Si la boite n'apparaît pas, cocher cette option dans " Affichage ".

4 / Choisir " **Champ d'étiquette** " et écrire le titre (" Exercice 1 " en police 16 gras-italique)

5/ Cliquez sur l'outil permettant de créer une **Zone de Texte** et dessiner cette zone dans la boîte de dialogue. Ouvrir la fenêtre " propriétés " à l'aide d'un clic-droit, nommer " **ZTEXTE1** " la zone de texte , choisir une police de caractères 14 points et une couleur arrière plan jaune (par ex.) ,

6/ Sous cette boîte insérer un **Champ d'étiquette** avec le texte d'accueil (taper votre)

7/ Dessiner un **Bouton** , le nommer " Bouton1 " avec comme titre : " Affiche ! " en caractères gras

8/ Dessiner une **Case à cocher** , la nommer " **CASE1** " avec comme titre : " Mme" , une autre case nommée " **CASE2** " avec comme titre " Melle " et enfin " **CASE3** " et Mr . (**Attention aux majuscules/minuscules**)

B/ Ecriture du programme de contrôle en Basic

1/ Ouvrir la feuille F1 , choisissez Outils - Macros - Macro et cliquez sur **TP5-XX** - Standard puis Nouveau Module1 (ou Editer Module1 si il existe déjà). Effacer le sous-programme Main si nécessaire.

2/ Taper le programme suivant qui permettra d'afficher la boîte de dialogue crée précédemment.

- Les deux premières lignes du programme permettant de décrire le chemin permettant de trouver la boîte et de l'affecter à une variable objet intitulée " Fenetre1 " .

- La troisième ligne permet de relier la variable objet " Texte " à la zone de texte " ZTEXTE1 "

- La quatrième ligne (**Fenetre1.Execute()**) correspond à une **méthode** de l'objet Fenetre1 , c.a.d à une séquence de code exécutable propre à cet objet. Dans ce cas cela fait apparaître la boîte de dialogue.

Option explicite

Global Fenetre1 as object

Global Texte as object

Sub Montre_Fen1



**Attention ! Bien respecter
majuscules et minuscules dans
Boite1 , ZTEXTE1**


```
DialogLibraries.LoadLibrary( "Standard" )
Fenetre1 = CreateUnoDialog( DialogLibraries.Standard.Boite1 )
Texte = Fenetre1.GetControl("ZTEXTE1")
Fenetre1.Execute()
end sub
```

3/ Essayer ce programme , la boite de dialogue intitulée " **Boite1** " , que vous avez crée précédemment doit apparaître.

En cas d'erreur, vérifier l'orthographe des mots : Boite1 , ZTEXTE1 qui doivent absolument être identiques dans les dessins de boites et votre programme

4/ Ecrire le sous-programme suivant permettant d'écrire le prénom dans la Zone de texte et ensuite d'écrire Bonjour , suivi du prénom, lors d'un 'clic' sur le Bouton1 .

```
Sub Affiche
    Prenom = Texte.text
    Texte.text = "Bonjour " & Prenom
End Sub
```

Ne pas oublier de déclarer la variable Prénom comme chaîne de caractère. Ce module doit être appelé lors de l'appui sur le **Bouton** nommé " Bouton1 " avec comme titre : " Affiche ! " .

5/ Pour pouvoir utiliser plusieurs fois le bouton et effacer automatiquement la zone de texte , ajouter le sous-programme suivant et le faire exécuter lorsque le Curseur de la souris est à l'intérieur de la zone de texte (Propriétés – événement) de la zone de texte.

Sub Efface

Texte.text=""

End sub

6 / Afin d'écrire la phrase de bienvenue complète , il faut tester les cases à cocher . Pour la case " Mme " , il faut déclarer une variable objet en Global , par exemple :

Mme as object

puis dans le module Montre_Fen1 ajouter la ligne suivante **(au dessus de la ligne : Fenetre1.Execute())**

Mme= Fenetre1.GetControl("CASE1")

et enfin, ajouter dans le module affiche :

if Mme.State = 1 then : Texte.text = "Bonjour Madame " & Prenom : Endif

Tester ce module et le bon fonctionnement de la case à cocher.

En cas d'erreur, vérifier l'orthographe du mot : CASE1 qui doit absolument être identique dans les dessins de boîtes et votre programme

7 / Ajouter les lignes de programme pour gérer les deux autres cases à cocher .

8 / En utilisant les propriétés des cases à cocher et un sous programme comme ci-dessous , faire en sorte que lorsqu'on coche une case les deux autres deviennent blanches .

Sub Raz_Mme

Mr.State=0 : Melle.State=0

Travail à faire par vous même :

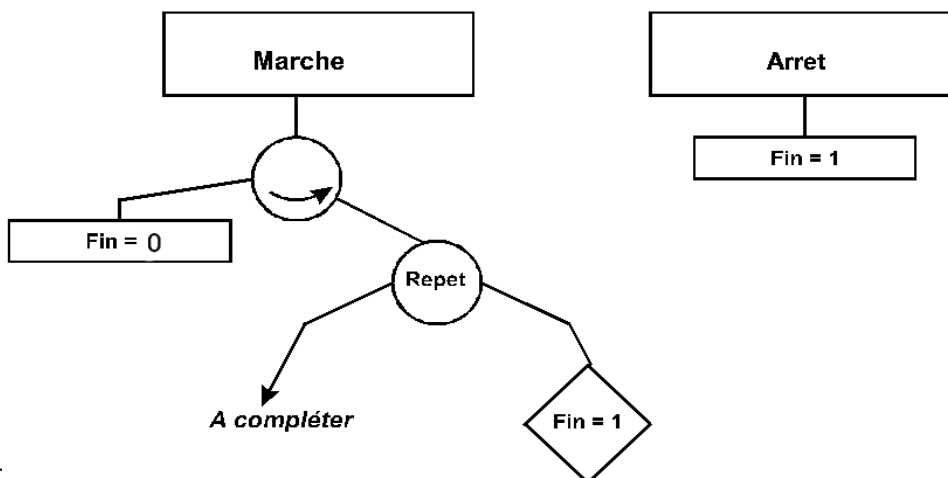
1/ Créer un compteur / décompteur comme celui représenté sur la figure suivante .



Instructions utiles et recommandations

- Le compteur ne doit pas être remis à zéro lorsqu'on passe de comptage à décomptage

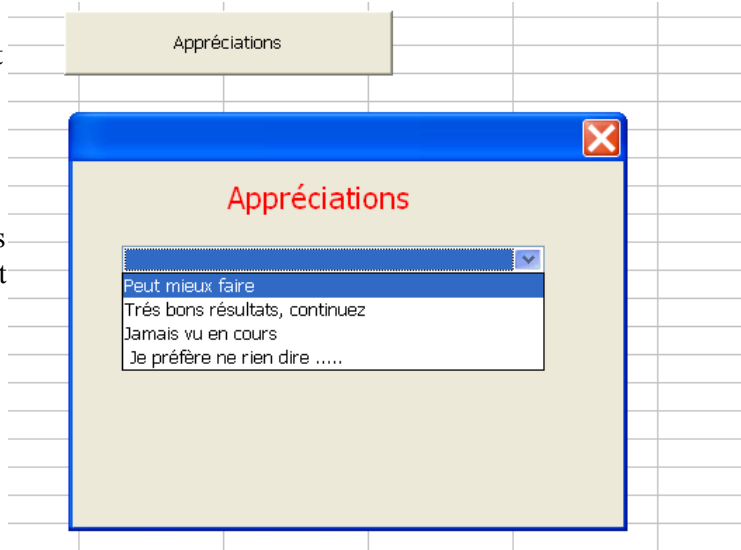
- Nommer cette boîte de dialogue " **COMPTEUR** ", c'est plus explicite que Boite2
- Temporisation pour éviter un comptage trop rapide : introduire l'instruction " **Wait 500** " dans votre programme (attente 500 mS)
- Il est conseillé de compléter l'arbre programmatique ci-après avant d'écrire le programme. En plus du module principal pour faire apparaître la fenêtre, faire deux sous programmes nommés " Marche " et " Arrêt " , chacun d'eux étant actionné par un bouton.
- Le fenêtre représentant la valeur numérique doit être représenté par un objet de type NumericField.



2 / Un enseignant en lycée désire automatiser l'écriture des appréciations dans les bulletins de note . Pour

cela il crée une boîte de dialogue comportant un menu déroulant (" Zone de liste " ou " ListBox ") (voir ci-dessous)

Dans la feuille de calcul du tableau contenant les noms et notes des élèves il ajoute un bouton (appréciations) qui permet de faire apparaître cette boîte de dialogue. Pour ajouter une appréciation dans une case de la feuille, il suffit de positionner le curseur dans cette case , de cliquer sur le bouton et choisir une phrase dans le menu déroulant.



Le menu déroulant est géré de la façon suivante :

Sub Montre_Boite2

```
DialogLibraries.LoadLibrary( "Standard" )
```

```
Boite_a_phrases = CreateUnoDialog( DialogLibraries.Standard.Boite2 )
```

```
ListePhrases = Boite_a_phrases.GetControl("Phrases")
```

' Ci-dessous la liste des phrases du menu déroulant à compléter suivant votre imagination ...)

```
ListePhrases.additem("Peut mieux faire ",0)
```

```
ListePhrases.additem("Très bons résultats, continuez ",1)
```

```
ListePhrases.additem("Jamais vu en cours ",2)
```

```
ListePhrases.additem(" Je préfère ne rien dire ..... ",3)
```

```
Boite_a_phrases.Execute()
```

```
end sub
```

Le menu déroulant dessiné dans la boite2 est nommé " Phrases " et la variable objet correspondante est " ListePhrases " , l'instruction " ListePhrases.additem(" xxxxxx ", i) permettant d'ajouter une phrase à la liste en position i .

Pour affecter la phrase sélectionnée par un clic souris à une variable de type string, il suffit d'écrire l'instruction suivante :

a= ListePhrases.Selecteditem (dans cet exemple la variable " a " contiendra la phrase sélectionnée)

Travail à faire : Dessiner la Boite2 et écrire le programme correspondant dans le module2 du classeur TPINFO_3 , feuille F2 . Essayer le programme.

(Dans les propriétés de la Zone de Liste on trouvera l'option " Déroulante " à activer. Dans " Événements " , on trouvera la possibilité d'exécuter une macro par un clic dans la liste – Attention , ne pas choisir " Boîte combinée " ou " Combo-Box " les commandes sont différentes)

Pour connaître le numéro de la ligne qui a été sélectionné dans la liste déroulante , il suffit d'utiliser l'instruction suivante :

NumLigne = ListePhrases.selectedItemPos

avec **NumLigne** , une variable de type " entier "

- Dessiner une zone de texte dans la boite 2 (ne pas oublier de déclarer les variables correspondantes)
- Ajouter quelques lignes de programme permettant de faire afficher dans cette zone de texte le numéro de la ligne qui a été sélectionné dans la liste déroulante .

TP 6 (optionnel) : Empreinte carbone

Ce dernier TP utilise uniquement des fonctions qui ont été étudiées dans les séances de travaux pratiques précédentes. Si vous avez fait l'effort d'analyser et comprendre tous les exercices, vous avez acquis suffisamment d'autonomie pour écrire le programme proposé dans ce TP sans l'aide de l'enseignant.

Le thème de cette séance de TP est le calcul de l'empreinte carbone générée par le transport de marchandises. Il est basé sur le tableau donné à la page suivante.

Dans le cadre de ce TP, on s'intéressera uniquement à la masse équivalente de carbone produite par le transport d'une certaine quantité de marchandises sur une distance donnée.

Etape 1 Bilan carbone par train

La macro 1 doit permettre à l'utilisateur de bâtir le parcours d'une masse M de marchandises à l'aide du train en spécifiant le nombre de kilomètres parcourus.

Il faudra construire une boîte de dialogue permettant de demander la masse de marchandise à transporter et le nombre de kilomètres à parcourir.

Etape 2 Bilan carbone par camion

La macro 2 doit permettre à l'utilisateur de bâtir le parcours d'une masse M de marchandises à l'aide du camion en spécifiant le nombre de kilomètres parcourus et le type de camion.

Il faudra construire une boîte de dialogue permettant de demander la masse de marchandise à transporter, le nombre de kilomètres à parcourir et le type de camion selon un menu déroulant.

Etape 3 Bilan carbone par avion

La macro 3 doit permettre à l'utilisateur de bâtir le parcours d'une masse M de marchandises à l'aide d'un avion en spécifiant le nombre de kilomètres parcourus et le type d'avion.

Il faudra construire une boîte de dialogue permettant de demander la masse de marchandise à transporter, le nombre de kilomètres à parcourir et le type d'avion selon un menu déroulant.

Etape 4 Synthèse

Le détail du parcours sera récapitulé dans la feuille de calcul sous forme de plusieurs lignes indiquant le moyen de transport utilisé , le nombre de km parcourus et le nombre de kg équivalent carbone produit. La dernière ligne donnera le total de kg équivalent carbone produit. Exemple :

Moyen Transport	Distance (km)	Équivalent carbone (kg)
Camion de 1,5 à 2,5 tonnes essence 105		101.5
Avion AirBus A319	1600	2176.1
Train , Allemagne	250	27.8
Camion de 1,5 à 2,5 tonnes diesel	15	13.7
Total	1970	2119.1

Vous pouvez laisser libre cours à votre imagination pour la présentation de ce programme