1- Se lancer en VBA

- 1) Ouvrir Excel
- 2) Utiliser la combinaison de touches ALT+F11. On arrive sur cette interface :

| Microsoft Visual Resic - Classourt | - 🗆 X |
|--|-------------------------------|
| Echer Editor, Affaloge jeweins format (brouge Estavlin Quils Complements Fegete 2) B I I I I I I I I I I I I I I I I I I I | Tapez une question |
| View (Characut) View (Characut) View (Characut) View (View) | |
| - #FixID (FixID) - #FixID (FixID) - 40 Thirdinetook | |
| | |
| | |
| Prepared as - Freedill X | |
| Persit Notavier | |
| Involute/Infance Frien DisolationsCharter | |
| Independence Associations Conductations Contract Model Contract Contract | |
| Visle - 1 - XSHeet/isle | |
| | |
| | |
| | |
| 🖬 Rechercher sur Windows 🔹 🗈 💽 🛕 🕋 🗠 | ■ * 40 ₽ = 10.29 06/01/201 |
| | |
| 3) Double-cliquer sur Feuil I : | |
| Amicrosoft Visual Basic - Classeur1 | |
| Eichier Edition Affichage Insertion Format Débegage Evécution Outile Com | |
| | ι μ ι |
| | _ |
| Projet - VBAProject | |
| | |
| = SVBAProject (Classeur1) | |
| Feuil1 (Feuil1) | |
| — I Feuil2 (Feuil2) | |
| 一町 Feuil3 (Feuil3) | |
| | |
| | |
| | |
| | |
| | |

Une page blanche apparaît alors. C'est sur celle-ci que l'on peut saisir du code.

On peut alors faire un premier programme :

Sub premierProgramme() Msgbox "Ceci vient d'un premier programme"

End Sub

On peut vouloir sauvegarder le programme : il suffit de sauvegarder le classeur qui contient le code. Attention ! Tous les formats proposés ne permettent pas d'enregistrer le code. Le format xlsx (format par défaut), ne conserve pas le code VBA pour des raisons de sécurité. Il faut donc faire l'enregistrement soit :

- Au format .xlsm (classeur Excel supportant les macros)
- Au format .xls (format des classeurs de la version Excel antérieure à 2007)

Ex1

- 1) Saisir ou copier le code de premier programme
- 2) *Exécuter ce code. On peut soit cliquer sur F5 quand le curseur d'écriture est placé sur le code, soit on peut utiliser la flèche verte de lancement de programme :*



3) Que fait ce code ?

Ce premier programme est une première procédure. Une procédure est définie entre Sub et End Sub. Considérons le code ligne à ligne.

Sub premierProgramme()

Cette première ligne indique que l'on va définir une procédure. On indique le nom de la procédure. Le nom ne pas comporter d'espace et ne doit pas être l'un des mots clés utilisés en VBA (Sub, For, Next, End ...)

Msgbox "Ceci vient d'un premier programme"

Cette ligne demande à la machine d'afficher une Msgbox, soit une boîte de dialogue. On indique ensuite le texte à afficher par la Msgbox. Ce texte est entre guillemets : il est traité par la machine comme une chaîne de caractères.

End Sub

Cette dernière ligne du code indique que le code est terminé. A noter que l'on pourrait rajouter d'autres procédures à la suite de cette première procédure.

- 1) Sans éliminer la première procédure créée, écrire une deuxième procédure (un deuxième programme) qui affiche "Hello Word" dans une MsgBox
- 2) Comment peut-on exécuter la seconde procédure ? Comment peut-on exécuter la première ?

2- Manipulation de variables, entrées et sorties.

Variable

Dans un programme, on doit définir des variables pour faire des calculs. Pour chaque variable utilisée, il faut définir son type. Si par exemple on veut travailler avec une variable x réelle, on déclare cette variable :

Dim x as Double

Ici, on crée une variable x réelle. Double correspond au type réel en VBA.

On peut ensuite manipuler cette variable. Soit on lui affecte une valeur :

x=5.6

soit on lui affecte le résultat d'un calcul :

x=5*60

= est le symbole de l'affectation, il affecte la valeur à droite à la variable désignée à gauche. Ainsi, dans la dernière instruction proposée. La machine calcule 5*60 puis affecte la valeur obtenue (300) à la variable x. On peut le tester :

```
Sub premierCalcul()
Dim x as Double
x=5*60
Msgbox x
End Sub
```

Ici, la troisième instruction demande à la machine d'afficher la valeur de la variable x.

Ex3

- 1) Reprendre le code de premier calcul. Que se passe-t-il si on enlève l'instruction d'affichage ?
- 2) Modifier le programme :

Sub premierCalcul()

Dim x as Double x=5*60 x=x+1 Msgbox x End Sub Que donne cette nouvelle version ?

3) Comment pourrait-on connaître la valeur de x avant le premier calcul (x=5*60) ? Quelle est cette valeur ?

Types de données

On dispose de plusieurs types de données en VBA. Dans cette version, on retiendra les types suivants :

- Double : réel
- Integer : entier
- String : chaîne de caractère
- Date : date

Ainsi, on peut déclarer des variables de type divers :

Dim x as Double, n as Integer, chaineC as String

Dim d as Date, y as Double

A noter que les noms de variables ne doivent pas contenir d'espace et ne doivent pas être des mots clés du système.

Ex5

1) On considère le code suivant :

```
Sub calcul2()
```

Dim x as Double, n as Integer

n=5

x=n+10

Msgbox x

End Sub

Qu'affiche t'il ?

2) On considère le code suivant :

```
Sub calcul3()
Dim x as Double, n as Integer
n=7
x=5
x=(n+10) *x
n=3
Msgbox x
End Sub
Qu'affiche t'il ?
```

Proposer un code qui crée deux variables : i comme un entier et y comme un réel. Dans ce programme, on affecte la valeur 5.4 à y puis on affecte la valeur de y à i (i=y). Afficher la valeur de i après cette opération ? Comment peut-on l'expliquer ?

En VBA, on dispose des opérateurs classiques : +, -, /, *. A noter que l'on dispose également d'un opérateur pour obtenir la puissance d'un nombre : $^{>}$

Ex7

Créer une procédure qui permet de calculer 2⁹

Manipulation de date

Les dates peuvent également être manipulées

On peut créer une première procédure :

```
Sub manipDate1()
```

```
Dim d as Date
```

D=Now()

Msgbox d

End Sub

Ceci affichera la date du jour. Now() est une fonction qui permet d'obtenir la date du système.

Faire une procédure qui prend la date de jour et lui enlève 5. Afficher la date résultat obtenue. Que constate t'on ?

Entrées et sorties

Jusqu'à présent, on a utilisé Msgbox. Cette instruction permet d'afficher un boîte de dialogue. Elle permet de donner un retour à l'utilisateur : on parle d'instruction de sortie. A noter que l'on dispose également d'instructions d'entrée : celles-ci permettent à l'utilisateur de rentrer des valeurs pendant un programme. Ainsi :

- Msgbox -> instruction de sortie
- InputBox -> instruction d'entrée.

On peut par exemple faire un programme qui demande à l'utilisateur un montant HT et lui renvoie le montant TTC (en supposant une TVA à 20%) :

```
Sub calculTVA1()
Dim montantHT as Double, montantTTC as Double
montantHT=InputBox("Veuillez saisir un montant HT")
montantTTC=montantHT*(1,2)
Msgbox montantHT
End Sub
```

Ici, la machine va récupérer la valeur saisie par l'utilisateur, l'augmenter du montant de la TVA, puis afficher le montantHT. A noter que l'on pourrait vouloir une réponse mieux rédigée. Pour cela, on peut rajouter un peu de texte :

```
Sub calculTVA1()
Dim montantHT as Double, montantTTC as Double
montantHT=InputBox("Veuillez saisir un montant HT")
montantTTC=montantHT*(1,2)
Msgbox "Le montant TTC : " & montantTTC
End sub
```

& est un opérateur de concaténation. C'est un opérateur qui joint, en les accolant, différents éléments pour fonder une chaîne de caractères.

Ici la machine va construire une nouvelle chaîne, composée de "Le montant TTC : " et de la valeur contenue dans la variable montantHT. Cette chaîne sera ensuite affichée avec la MsgBox.

On dispose maintenant de quelques moyens pour interagir avec l'utilisateur.

Ex9

On propose de faire un nouveau programme : la machine demande à l'utilisateur un montantHT et un taux de TVA. L'utilisateur rentrera ce taux sous la forme 0,2 ou 0,15 etc. En retour la machine affichera le montant TTC correspondant.

3- Vers les cases de la feuille

Une autre forme d'interaction possible est l'interaction avec les cellules de la feuille. En VBA, on dispose de l'opérateur Cells. Cells(n,c) désigne la cellule de la ligne n et de la colonne c. Ainsi, cells(1,5) désigne la cellule de la ligne 1 et de la colonne 5, soit la cellule E1, cells(4,4) désigne la cellule de la ligne 4 et de la colonne 4, soit D4.

On peut mettre du contenu dans les cellules :

```
Sub interaction1()
Cells(5,4)=1
Cells(4,4)="Test"
Cells(2,2)=Now()
End sub
```

En testant ce code, on voit que l'on peut mettre du contenu dans les cases.

On peut également intégrer le contenu des cases dans des calculs :

```
Sub interaction2()
Cells(5,4)=1
Cells(6,4)=cells(5,4)*10
Cells(7,4)=cells(6,4)^2
Dim x as double
x=7.5
cells(8,4)=x*3
cells(9,4)=InputBox("Donner une valeur SVP")
End Sub
```

4- Créer une forme réactive ou un bouton

Pour l'instant, le code que l'on produit est relativement inaccessible : il faut à chaque fois aller sur l'interface VBA et exécuter le code. On voudrait faire quelque chose de plus réactif. On va créer un code qui affiche en B1 le montant TTC quand le montant HT est en A1.

Ex 10

- 1- Rédiger le code qui permet d'obtenir le montant TTC en B1 quand le montant HT est en A1. On suppose un taux de TVA de 20%. Appeler la procédure calculTVA2
- 2- *Rédiger le code qui permet d'obtenir le montant TTC en B1 quand le montant HT est en A1. Le taux de TVA sera sasi par l'utilisateur (InputBox). Appeler la procédure calculTVA3*

Une fois que l'on a rédigé un programme, on peut aller dans l'interface d'Excel qui permet de dessiner des formes :



On dessine la forme que l'on veut

Ensuite on exécute un clic droit dessus



Et on clic sur "Affecter une macro"

Il ne reste plus qu'à choisir la macro que l'on veut voir exécutée lorsque le bouton est cliqué.

Ex 11

Insérer un bouton et faire en sorte qu'il appelle la procédure calculTVA3 de l'exercice 10.

Ex 12

```
Quelles seront les valeurs des variables A, B et C après exécution des instructions suivantes ?

Sub ex12()

Dim A as Integer, B as Integer, C as Integer

A = 5

B = 3

C = A + B

A = 2

C = B - A

B = B + 1

End Sub
```

Donner un algorithme qui prend un nombre en cellule A1 et donne sa partie entière en cellule A2 et sa partie décimale en cellule A3.

Ex 15

A partir du classeur VBA1.xslm,

- 1) Faire un algorithme qui permet de compter les cellules grisées.
- 2) Faire un bouton qui appelle cet algorithme.