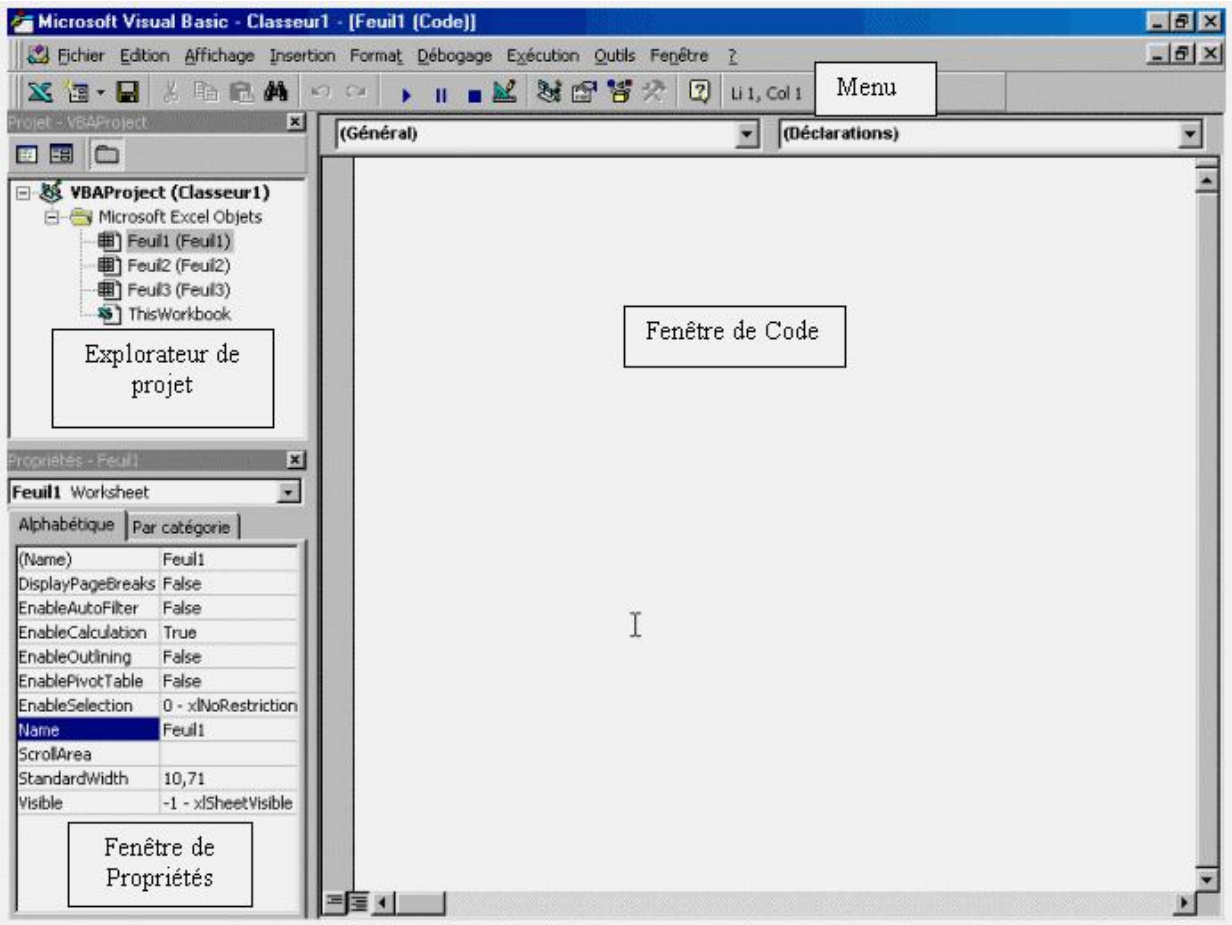
Initiation à la programmation en Visual Basic

1. L'éditeur Visual Basic.

Lorsque l'on désire convivialiser le fonctionnement d'Excel, on est amené à utiliser l'éditeur visual-basic.
C'est en fait un module de programmation, lié à Excel et très proche du langage visual basic 6.0.
On y accède à partir d'Excel avec la commande : *Outils / Macro / Visual Basic Editor* ou *ALT + F 11*

On a alors l'écran suivant :



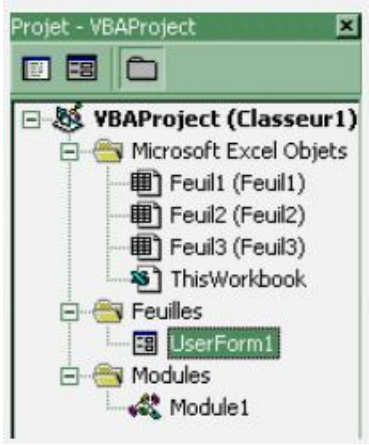
Menu

On retrouve dans la barre de menu les éléments suivants :

1	Bascule Excel	2	Ajout composant	3	Sauvegarde	4	Exécution
5	Arrêt	6	Initialisation	7	Mode création	8	Explorateur de projets
9	Fenêtre de propriétés	10	Explorateur d'objets	11	Boite à outil	12	Aide
13	Position						

Fenêtre de code

C'est dans cette fenêtre que l'on retrouvera le code des programmes (suite des instructions).
La programmation s'effectue en ANGLAIS.



Explorateur de projet

D'apparence identique à l'explorateur de Windows, il permet de visualiser tous les éléments de votre travail, à savoir :

- Les classeurs. (Workbook)
- Les feuilles de calculs. (Worksheet)
- Les feuilles de module (où se situe les programmes - code).
- Les feuilles de dialogue. (UserForm)

Il permet de sélectionner un éléments afin de visualiser le code (voir précédemment) ou les propriétés (voir ci-dessous).

Fenêtre de propriétés

Le langage visual basic est un langage orienté objet, c'est à dire que l'on a des objets (classeur, feuille, cellules,...) qui sont pourvus de propriétés (nom, couleur, police,...).
La programmation se base sur ces objets.

On peut visualiser les propriétés d'un objet (et les modifier) dans la fenêtre de propriétés, pour cela il suffit de sélectionner l'objet.

Nous allons dans la suite progressivement voir comment utiliser cet interface pour réaliser :

- Des fonctions personnalisées.
- Des macros simples.
- Des programmes plus complexes gérants les outils graphiques, les éléments de dialogue, les masques de saisie.

2. Les fonctions personnalisées.

Les fonctions permettent un calcul, elles renvoient donc une valeur et se placent dans une cellule à la suite d'un signe d'égalité. Voyons, à partir d'exemples, comment elles se construisent.

2.1 Périmètre Cercle.

Allez dans le module visual basic et créez une feuille de module. *Insertion / Module*
Saisissez la fonction suivante :

```
Function Périmètre (Rayon)
    Périmètre = 2 * 3.14159 * Rayon
End function
```

Retournez dans Excel, et testez votre fonction dans une cellule, en saisissant :

```
= périmètre (10)
```

et le périmètre du cercle de rayon 10 s'affiche, à savoir 62,8318.

2.2 Conversion Francs / Euro.

De la même manière, saisissez deux fonctions qui permet de convertir les Francs en Euro et réciproquement.

La solution est la suivante :

```
Function Euro (Francs)
      Euro = Francs / 6.55957
End function

Function Francs (Euro)
      Francs = Euro * 6.55957
End function
```

Retournez dans Excel, et testez vos fonctions de conversion.

2.3 Heure décimale.

Plusieurs paramètres peuvent intervenir dans une fonction, il suffit de les saisir sous la forme :

```
Function ToTo (param1 ; param2 ; param3 ;...)
```

Saisissez une fonction permettant de convertir les heures en heures décimales.

La solution est la suivante :

```
Function Heure_décimale (H , M , S)
      Heure_décimale = H + ( M + ( S / 60 ) ) / 60
End function
```

Retournez dans Excel, et testez votre fonction en convertissant 1 h 30' 45" La valeur décimale doit être 1,5125.

Rem : le séparateur de l'éditeur VB est la virgule, celui des fonctions excel est le point virgule.

2.4 Remise.

On peut améliorer les fonctions en insérant des choix, par exemple, saisissez une fonction Remise qui :

- si le prix est supérieur à 5000 F, attribut une remise de 20 %
- si le prix est supérieur à 2000 F, attribut une remise de 10 %
- sinon n'attribut aucune remise.

La solution est la suivante :

```
Function Remise (Prix)
      if Prix > 5000 then
            remise=prix*0.2
      else if Prix > 2000 then
            remise = prix*0.1
      else
            remise = 0
      end if
End function
```

Retournez dans excel, testez votre fonction remise.

3. Les macros simples.

Les macros sont en fait la mémorisation d'une suite d'actions dans Excel.
On utilise pour cela un enregistreur de macro qui fonctionne comme un magnétophone : toutes les actions sont enregistrées et lorsqu'on exécute la macro toutes ces actions sont reproduites dans le même ordre.
Voyons, à partir d'exemples, comment elles se construisent.

3.1 Format monnaie sans décimale.

Chargez le fichier : BUDGET.XLS

Sélectionnez les cellules D3:F4 ,puis appliquez le style monétaire.

Ce style comprend 2 décimales, nous allons créer une macro pour réaliser l'affichage sans décimale.

Sélectionnez les cellules D7:F8
On va alors lancer l'enregistrement de la macro, à savoir : *Outils / Macro / Nouvelle macro*
Il faut alors saisir son nom : ***FormatMonnaie***, puis après validation, la macro commence à s'enregistrer...
Il suffit de changer le format du style monétaire, à savoir : *Format / Cellule / Nombre / Monétaire / nb décimales* , de valider, puis d'arrêter l'enregistrement de la macro.

Essayez alors votre macro sur les cellules D11:F12 avec la commande : *Outils / Macro / FormatMonnaie / Exécuter*

Vous pouvez associer à votre macro un raccourci clavier, avec : *Outils / Macro / FormatMonnaie / Options*

Passez dans l'éditeur visual basic, pour voir la procédure (code) associée à votre macro. C'est un petit programme.

```
Sub Macro1()
    Selection.NumberFormat = "#,##0 $"
End Sub
```

3.2 Centrer du texte.

Sur la feuille de calcul BUDGET96, doublez la hauteur de la première ligne.
Sélectionnez la cellule D1 et créez une nouvelle macro : ***CentrerHV***, qui permettra de centrer horizontalement et verticalement une cellule. Essayez alors votre macro sur les cellules E1:F1, puis passez dans l'éditeur, pour voir la procédure associée à votre macro.

```
Sub CentrerHV()
    with Selection
        . HorizontalAlignment = xlCenter
        .VerticalAlignment = xlCenter
        . WrapText = False              (renvoi à la ligne)
        . Orientation = 0               (orientation)
        . ShrinkToFit = False           (ajustement automatique à la largeur)
        . MergeCells = False            (cellule fusionnée)
    End with
End Sub
```

On constate que les 4 dernières propriétés sont à l'état FAUX, elles ne nous servent donc pas directement et l'on peut supprimer les 4 lignes correspondantes. Faites le et testez votre macro en G1.

On va maintenant associer notre macro à un bouton de la barre d'outil.
Utilisez la commande : *Affichage / Barre d'outils / Personnaliser / Commandes / Macros*
Sélectionnez un bouton et tirez le vers la barre d'outil supérieure. Associez lui la macro ***CentrerHV***
(clic droit sur le bouton /affecter macro), fermez et testez votre bouton en H1.

On peut aussi créer un bouton sur la feuille de calcul. Il suffit d'activer la barre d'outil dessin, de créer un objet (rectangle, forme quelconque, wordart,...), puis de lui affecter lui la macro ***CentrerHV*** (clic droit / affecter macro). Validez et testez votre bouton. On peut alors modifier ce bouton en le sélectionnant avec *CTRL + clic*

4. Les objets, méthodes et propriétés.

Nous allons dans cette partie, à l'aide de petites manipulations, comprendre le fonctionnement du langage visual basic et son orientation objet.

Dans ce langage, nous avons des familles (collections) d'objet ou des objets auxquels on affecte des propriétés, on agit sur eux avec des méthodes et leurs évolutions sont des événements.

Pour illustrer cela, considérons une collection d'objet (les voitures), le nombre de voiture est une propriété de la collection.
On peut envisager d'ajouter une voiture à la collection; l'ajout est une méthode.
(la propriété est un attribut de l'objet, la méthode agit sur celui-ci...)
Dans notre collection, on peut extraire des objets (la voiture X, la voiture Y). A chacun de ses objets, on peut affecter des propriétés, par exemple, la couleur (rouge, bleu, vert,...) ou si c'est un break (vrai, faux)...
Enfin ces voitures (objets), on peut par exemple les déplacer, le déplacement est alors une méthode.
Si la voiture X (objet) qui est rouge (propriété) démarre c'est alors un événement. Les événements permettent de

rendre compte d'une évolution...

Cela paraît peut-être compliqué, alors pratiquons à l'aide de manipulations élémentaires.

Pour comprendre ces notions, nous allons "piloter" Excel **uniquement** avec des instructions, sans agir sur les menus, ni les boutons.

Commencez par créer un nouveau classeur, activez l'éditeur Visual Basic, puis insérez une feuille de module.
Affichez la fenêtre d'exécution. *Affichage / Fenêtre d'exécution*
Redimensionnez cette fenêtre pour voir le classeur à l'arrière-plan et pour conserver essentiellement la partie supérieure de la fenêtre (le volet d'exécution).

Vous voilà prêt pour tester des instructions...

Les Classeurs.

- Ajouter un classeur.
  Dans le volet d'exécution, tapez l'instruction : Workbooks.Add
  Un nouveau classeur s'ouvre. Workbooks (Classeurs) est une collection d'objet et ajouter une méthode.

- Compter les classeurs.
  Dans le volet d'exécution, tapez : ?Workbooks.Count
  Le nombre de classeurs actuellement ouvert apparaît. Ajouter un nouveau classeur et répéter l'instruction.
  Count (Nombre de) est une propriété de la collection d'objet Classeurs.

- Fermer un classeur.
  Dans le volet d'exécution, tapez : Workbooks.Close
  (ne valider pas la sauvegarde).
  Tous les classeurs ouverts disparaissent. Close (Fermer) est une méthode de la collection d'objet Classeurs.
  Vous pouvez demander le nombre de classeurs ouverts.

- Faire référence à un seul classeur.
  Commencez par recréer quelques classeurs, puis tapez : ?Workbooks.Item(1).Name
  Le nom du premier classeur s'affiche. Item (Elément) est une méthode de la collection d'objet Classeurs et Name (Nom) une propriété.. On obtient le même résultat avec la commande : ?Workbooks(1).Name
  Tapez alors : Workbooks.Item(1).Close
  Le premier classeur disparaît. Retapez ensuite : ?Workbooks.Item(1).Name

- Faire référence à un classeur par son nom.
  Si vous avez fermé tous les classeurs, commencez par en recréer quelques-uns.
  Dans le bas de la fenêtre, lisez le nom d'un classeurs au milieu de ceux que vous venez d'ouvrir, par exemple
  "*toto*". Tapez alors : Workbooks.Item("toto").Activate
  Le classeur spécifié s'affiche au premier plan.
  Vous pouvez aussi le fermer en tapant : Workbooks.Item("toto").Close

- Faire référence à un classeur en le pointant.
  Pour sélectionner le classeur situé au premier plan, il suffit de sélectionner le classeur actif, par exemple pour le fermer : ActiveWorkbook.Close

- Changer la valeur d'une propriété pour un classeur.
  Dans le volet d'exécution, tapez : ?ActiveWorkbook.Saved
  Le mot " vrai " s'affiche car le classeur n'a pas été modifié. Dés qu'une modification aura lieu sur ce classeur,
  cette propriété passera à " faux ".
  Mais on peut provoquer ce changement en tapant : ActiveWorkbook.Saved = False
  Fermer maintenant ce classeur et excel vous propose de sauvegarder ce classeur...
  Cliquez sur annuler pour ne pas sauvegarder et ramenez la propriété enregistré à l'état " vrai ".
  Fermer ce classeur, excel ne propose pas de sauvegarder.

Les Feuilles.

Commencez par effacer toutes les feuilles de calcul.

- Ajouter une nouvelle feuille.
  Dans le volet d'exécution, tapez : WorkSheets.Add
  Une nouvelle feuille apparaît. Par cette méthode, ajoutez 3 autres feuilles. Tapez alors : ?WorkSheets(1).Name
  La propriété Name peut s'appliquer aussi aux feuilles. Tapez alors : WorkSheets (1).Name = "Valeurs Saisies"
  Le nom de la feuille est alors modifié.

- Copier et déplacer une feuille.
  Dans le volet d'exécution, tapez : WorkSheets ("Valeurs Saisies").Copy
  Copier est une méthode pour notre feuille de calcul et si l'on n'indique pas la destination la copie a lieu au niveau
  d'un nouveau classeur... Tapez alors : WorkBooks(1).Activate pour ramener le classeur original au premier plan.
  Tapez : WorkSheets("Valeurs Saisies").Copy Before := WorkSheets(2)
  Vous indiquez à la méthode copier à quel endroit placer la copie.
  Tapez : WorkSheets(2).Nom = "Valeurs Variant"
  Tapez : WorkSheets("Valeurs Variant").Move Before := WorkSheets(1)
  Déplacer est aussi une méthode applicable aux feuilles.
  Tapez : WorkBooks(2). WorkSheets(1).Name = "Anciennes Valeurs"
  On constate que l'on peut manipuler des objets inactifs.

- Manipuler plusieurs feuilles.
  Nous venons de manipuler soit un objet, soit toute une collection d'objet or parfois il peut être intéressant de ne manipuler qu'une partie des objets. Tapez : WorkSheets(3).Select pour activer la feuille des valeurs saisies.
  Sélectionner est une nouvelle méthode.
  Tapez alors : WorkSheets(Array(1 ,3 ,4)).Select pour sélectionner les feuilles 1, 3 et 4.
  Tapez : WorkSheets(3).Activate pour activer la 3ème feuille sans désélectionner les autres.
  Tapez : ?WorkSheets(Array(1 ,3 ,4).Count

Les Cellules.

Nous allons voir maintenant comment atteindre une cellule, une plage de cellules, se déplacer, redimensionner une plage...

- Sélection de Cellules.
  Dans le volet d'exécution, tapez : Cells.Select Vous sélectionnez l'ensemble des cellules de la feuille de calcul.
  Testez alors les instructions : Cells(2,1).Select ,puis Cells(258).Select
  On peut obtenir des informations sur les cellules avec des instructions du genre : ?Cells.Count ou ?Cells(260).Address

- Sélection de Lignes, Colonnes.
  Dans le volet d'exécution, tapez : Columns(3).Select ,puis Rows(4).Select

- Sélection de Plage de Cellules.
  Testez les instructions :

```
Range("B1").Select
Range("A1","C1").Select
Range("A1,C1").Select
Range("B3:C8").Select
Range("B2,C7,D3,A5").Select
Range("B2:C4").Name = "Tata"
Range("Tata").Select
```

- Se déplacer.
  Testez les instructions :

```
Cells(3,2).Select
ActiveCell.Offset(3,4).Select
ActiveCell.Offset(-2,-3).Select
Range("Tata").Offset(1,0).Select
```

- Combiner les sélections.
  Nous allons commencer par colorer une plage de cellules, pour avoir une référence, en utilisant l'instruction : Range("Tata").Interior.ColorIndex = 15
  Testez alors les instructions :

```
Columns(5).Select
Range("Tata").Columns(5).Select
Cells(4,2).Select
Range("Tata").Cells(4,2).Select
?Rows.Count
?Range("Tata").Rows.Count
Range("Tata").EntireRow.Select
Range("Tata").Resize(1,5).Select
Range("Tata").Offset(-1,-1).Resize(Range("Tata").Rows.Count+2 ,
Range("Tata").Columns.Count+2).Select
```

  Quel est le résultat ?

  Rem :

le caractère "_" permet de passer à la ligne tout en restant dans la même instruction.
le caractère " ' " permet d'insérer un commentaire ou une ligne de commentaire.

- Introduire des valeurs et des formules dans une plage.
  Commencez par changer de feuille de calcul. Sélectionnez une plage : `Range("B2:B6").Select`
  Et testez les instructions :
  ```
  Selection.Formula = 100
  ActiveCell.Formula = 0
  ActiveCell.Offset(-1,0).Formula = 1
  Selection.Formula = "=B1*5"
  ?ActiveCell.FormulaR1C1
  ```
  Changez le format : `Selection.EntireColumn.NumberFormat = "#,##0 F"`
  Et testez les instructions :
  ```
  ?Range("B2").Formula
  ?Range ("B2").Value
  ?Range ("B2").Text
  ?Range ("B2")
  ```
  Quelles sont les différences ?

5. Les macros complexes.

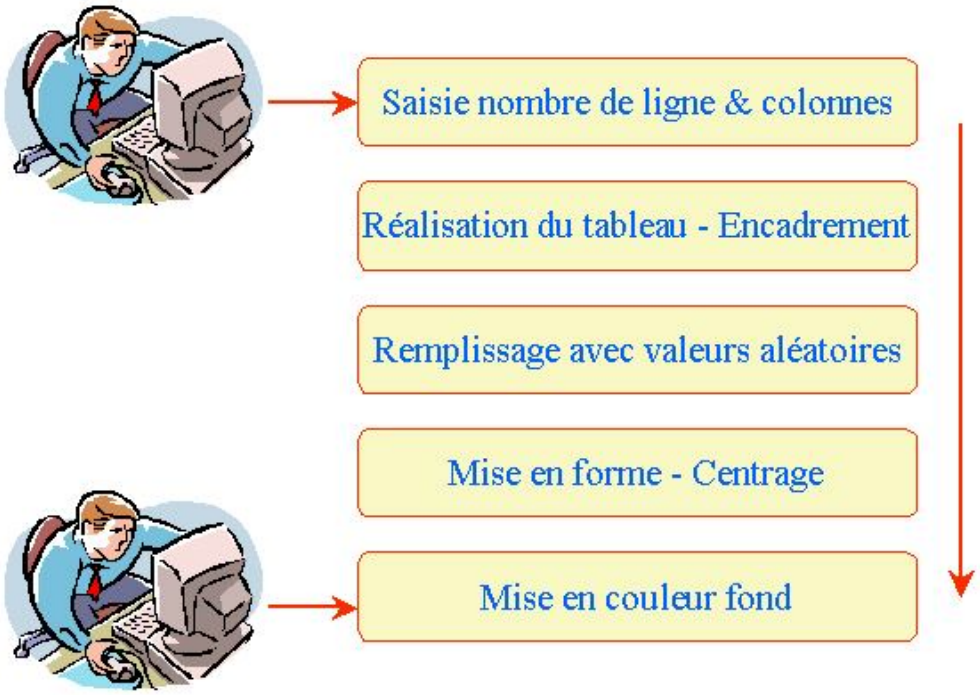
Vous fermerez tous les classeurs et en ouvrirez un nouveau.
Vous allez développer votre première petite application, à savoir la réalisation automatisée d'un tableau (qui pourrait servir de base pour une gestion de notes, par exemple). Le cahier des charges est le suivant :

- Créez un tableau avec un nombre de lignes et de colonnes définies par l'utilisateur.
  On pourra utiliser les boites de saisies automatiques, dont l'instruction correspondante est :
  `variable x = InputBox ("Entrez le nombre de ligne")`
- Dans ce tableau, toutes les cases seront encadrées en trait continus,
  mais de plus la dernière colonne sera encadrée en trait épais.
- Vous remplirez les cases de ce tableau avec des nombres aléatoires (fonction ALEA) compris entre 0 et 10.
- Chaque valeur sera centrée horizontalement et verticalement.
- Les valeurs de la dernière colonnes auront une police en caractères gras et italique.
- On pourra lancer la création d'un tableau à l'aide d'un bouton sur la feuille.
- A la création d'un nouveau tableau, on effacera toutes les traces éventuelles sur la feuille.
- On pourra, à l'aide d'un bouton spécifique, mettre en couleur le fond des cases en fonction de leur valeur, à savoir :

Valeur > 7	couleur verte
Valeur > 4	couleur bleue
Autre valeur	couleur rouge

- On pourra aussi, à l'aide d'un bouton spécifique, annuler l'effet de la commande précédente.

Il est conseillé avant de se mettre à programmer de décomposer votre problème en tâches élémentaires (procédures).
Ensuite, on fera appel à celles-ci dans un programme global (procédure pilote).
Un bon programme est une succession de petites procédures... en voici, par exemple, une décomposition :



Voici alors certaines des procédures associées :

Saisie du nombre de lignes.

```
Option explicit                         cette option oblige à déclarer les variables
Dim Nbl                                 Nbl variable contenant le nombre de lignes
Dim Nbc                                 Nbc variable contenant le nombre de colonnes
Sub Question()
    Nbl = 0
    Nbc = 0
    Nbl = InputBox("Nombre de lignes")
    Nbc = InputBox("Nombre de colonnes")
End Sub
```

Encadrement en trait fin.

```
Sub Encadre_Fin()
    Selection.BorderAround Weight:=xlThin
End Sub
```

Encadrement en trait épais.

```
Sub Encadre_Epais()
    Selection.BorderAround Weight:=xlMedium
End Sub
```

Insertion d'un nombre aléatoire.

```
Sub Aléatoire()
    Selection.Formula = "=ent(alea()*10)"  cette fonction donne une valeur comprise entre 0 et 1
End Sub
```

Alignement horizontal & vertical.

```
Sub Centrer()
    With Selection
        .HorizontalAlignment = xlCenter
        .VerticalAlignment = xlCenter
    End With
End Sub
```

Création du tableau (encadrement, remplissage, centrage).

```
Sub Créer_Tableau()
Dim I
Dim J
    Question                                        interrogation utilisateur
    Range("B2").Select
```

```
        For I = 1 To Nbl                                boucle sur le nb de lignes
            For J = 1 To Nbc – 1                        boucle sur le nb de colonnes
                Encadre_Fin
                Aléatoire
                Centrer
                ActiveCell.Offset(0, 1).Select          décalage d'une colonne à droite
            Next J
            Encadre_Epais
            Aléatoire
            Centrer
            ActiveCell.Offset(1, 1 - Nbc).Select        retour en 1ere colonne une ligne en dessous
        Next I
    End Sub
```

Mise en couleur des fonds.

```
    Sub Fond_Rouge()
        With Selection.Interior
            .ColorIndex = 3                     couleur : rouge
            .Pattern = xlSolid                  motif : plein
        End With
    End Sub

    Sub Fond_Vert()
        With Selection.Interior
            .ColorIndex = 43                    couleur : verte
            .Pattern = xlSolid                  motif : plein
        End With
    End Sub

    Sub Fond_Bleu()
        With Selection.Interior
            .ColorIndex = 17                    couleur : bleue
            .Pattern = xlSolid                  motif : plein
        End With
    End Sub

    Sub Fond_Rien()
        Selection.Interior.ColorIndex = xlNone      couleur : aucune
    End Sub
```

Choix du fond suivant la valeur.

```
    Sub Choix_Fond()
    Dim Val
        Val = ActiveCell.Value              valeur cellule active
        If Val > 7 Then                     si valeur > 7
            Fond_Vert
        ElseIf Val > 4 Then                 sinon si 7 > valeur > 4
            Fond_Bleu
        Else                                sinon
            Fond_Rouge
        End If
    End Sub
```

Mise en couleur des fonds.

```
    Sub Mise_Couleur_Fond()
    Dim I
    Dim J
        Range("B2").Select
        For I = 1 To Nbl
            For J = 1 To Nbc - 1
                Choix_Fond
                ActiveCell.Offset(0, 1).Select
            Next J
            ActiveCell.Offset(1, 1 - Nbc).Select
        Next I
    End Sub
```

Suppression de la mise en couleur des fonds.

```
    Sub Mise_Couleur_Rien()
```

```
    Dim I
    Dim J
        Range("B2").Select
        For I = 1 To Nbl
            For J = 1 To Nbc - 1
                Fond_Rien
                ActiveCell.Offset(0, 1).Select
            Next J
            ActiveCell.Offset(1, 1 - Nbc).Select
        Next I
    End Sub
```

Fichier Excel 2000 correspondant à cet exercice. (Exercice_Tableau.xls)

6. Les objets graphiques.

Ouvrez un nouveau classeur. Vous allez développer une petite application pour comprendre la gestion des objets graphiques.
Saisissez le tableau de valeurs suivantes :

	Elève A	Elève B	Elève C	Classe
1er Trim	8	11	12	=moyenne( )
2ème Trim	6	10	9	=moyenne( )
3ème Trim	11	15	13	=moyenne( )
	=moyenne( )	=moyenne( )	=moyenne( )	=moyenne( )

Vous allez à partir de ce tableau créer sur la même feuille dans la plage (A6:E19), un graphique de type Histogramme 3D, ayant les séries de données en ligne.
Nous allons alors créer 3 boutons pour changer rapidement le type de graphique (histogramme, aire, cylindre).

Modifier le type d'un graphique.

Désélectionnez le graphique.
Enregistrez une nouvelle macro dont le nom sera : **Aires**.
Cliquez sur le graphique avec le bouton de droite et sélectionnez sur type de graphique, puis sur Aire 3D, puis validez. Appuyez 2 fois sur Echap pour désélectionner le graphique et arrêter l'enregistrement.

En procédant de même créez deux autres macros (Histogrammes, Surfaces) pour avoir un type de graphique soit Histogramme 3D, soit Cylindre 3D.

Vous ajouterez autour de votre graphique 3 boutons (histogramme, aire, cylindre) permettant d'activer les macros précédentes.

Orienter les graphiques 3D.

Nous allons créer des procédures permettant de modifier la rotation, la perspective et l'altitude des graphiques.

Commençons par la rotation, il nous faut deux procédure pour pouvoir augmenter (rotation_plus) ou diminuer (rotation_moins) l'angle. On prendra comme incrément (10°) et les butées seront 0 - 360°.
Dans la fenêtre de module, vous saisirez la procédure suivante :

```
    Sub Rotation_Plus()
        With ActiveSheet.ChartObjects("Graphique 1").Chart
            If .Rotation <= 350 Then
                .Rotation = .Rotation + 10
            Else
                .Rotation = 360
            End If
            Range("C21") = .Rotation
        End With
    End Sub
```

A vous de réaliser l'autre procédure de rotation :

```
Sub Rotation_Moins()
    With ActiveSheet.ChartObjects("Graphique 1").Chart
        If .Rotation >= 10 Then
            .Rotation = .Rotation - 10
        Else
            .Rotation = 0
        End If
        Range("C21") = .Rotation
    End With
End Sub
```

Pour la perspective, on prendra comme incrément (10°) et les butées seront 0 - 100.

```
Sub Perspective_Plus()
    With ActiveSheet.ChartObjects("Graphique 1").Chart
        If .Perspective <= 90 Then
            .Perspective = .Perspective + 10
        Else
            .Perspective = 100
        End If
        Range("D21") = .Perspective
    End With
End Sub
```

A vous de réaliser l'autre procédure de rotation :

```
Sub Perspective_Moins()
    With ActiveSheet.ChartObjects("Graphique 1").Chart
        If .Perspective >= 10 Then
            .Perspective = .Perspective - 10
        Else
            .Perspective = 0
        End If
        Range("D21") = .Perspective
    End With
End Sub
```

Pour l'altitude, on prendra comme incrément (10°) et les butées seront -90° à +90°.

```
Sub Altitude_Plus()
    With ActiveSheet.ChartObjects("Graphique 1").Chart
        If .Elevation <= 80 Then
            . Elevation = . Elevation + 10
        Else
            . Elevation = 90
        End If
        Range("E21") = . Elevation
    End With
End Sub
```

A vous de réaliser l'autre procédure de rotation :

```
Sub Altitude_Moins()
    With ActiveSheet.ChartObjects("Graphique 1").Chart
        If . Elevation >= -80 Then
            . Elevation = . Elevation - 10
        Else
            . Elevation = -90
        End If
        Range("E21") = . Elevation
    End With
End Sub
```

Vous saisirez alors "Rotation", "Perspective" et "Altitude" respectivement dans les cellules C20, D20 et E20.
Et en dessous de la ligne 22, vous placerez deux boutons par caractéristique (-, +) que vous relierez aux macros écrites précédemment.

Fichier Excel 2000 correspondant à cet exercice. (Exercice_Graphique.xls)

7. Les boîtes de dialogue.

Nous allons appréhender les interfaces de dialogue (bouton d'option, case à cocher, liste déroulante, compteur,...) à l'aide d'un exemple.

Créer un modèle de remboursement d'emprunt automobile.

Dans un nouveau classeur, saisissez les données suivantes dans la plage B1:C7.

Prix	50 000 F
Apport	20 %
Emprunt	???
Intérêts	8,00 %
Années	3
Mensualités	???

Dans la cellule C4 (à droite d'emprunt) placez la formule = C2*(1-C3) et dans la cellule C7, la formule = VPM (C5/12 ; C6*12 ; C4)
Le montant des mensualités s'affiche : -1253,45 F

Si vous changez le prix (120 000 F), l'emprunt et la mensualité changent automatiquement, par exemple, (96000F et -3008,29F).
Vous venez de créer un formulaire.

Votre formulaire recalcule tout instantanément, mais il ne prend pas en compte des valeurs erronées... (prix, taux, durée,...).
Nous allons remédier à ce défaut.

Construire une liste de voiture.

Saisissez une liste de véhicules et de prix dans la plage K2:L9.

Voiture	Prix
Renault 19	145000
Peugeot 306	120000
Rover 600	130000
Peugeot 406	135000
BMW 316	142000
Citroën ZX	127000
Opel Vectra	122000

Sélectionnez et nommez la plage K2:L9 (*Insertion / Nom / Créer par ligne du haut*).
Activez la barre d'outil : Formulaire.
Insérez une zone liste modifiable sur la plage E2:G2. Cliquez 2 fois sur ce contrôle, pour afficher la boîte de dialogue format de contrôle et dans la zone contrôle tapez **Voiture** pour la plage d'entrée et **H2** pour la cellule liée.
Appuyez sur Echap pour déselectionner ce contrôle et testez-le.

Vous constaterez qu'un chiffre s'affiche en H2 correspondant à la position de la voiture dans la liste.
Reliez alors le formulaire à cette zone en tapant dans la cellule C2 la formule = INDEX (Prix;H2)

Limiter le montant de l'apport à des valeurs plausibles.

Insérez un compteur sur la plage E3:E4. Cliquez 2 fois sur ce contrôle, pour afficher la boîte de dialogue format de contrôle et dans la zone contrôle tapez **100** pour la valeur maximale, **5** pour le changement de pas et **H3** pour la cellule liée.
Appuyez sur Echap pour déselectionner ce contrôle et testez-le.

Reliez alors le formulaire à cette zone en tapant dans la cellule C3 la formule = H3/100

Limiter le taux d'intérêt à des valeurs plausibles.

Insérez une barre de défilement sur la plage E5:G5. Cliquez 2 fois sur ce contrôle, pour afficher la boîte de dialogue format de contrôle et dans la zone contrôle tapez **2000** pour la valeur maximale, **25** pour le changement de pas, **100** pour le changement de page et **H5** pour la cellule liée.
Appuyez sur Echap pour désélectionner ce contrôle et testez-le.

Reliez alors le formulaire à cette zone en tapant dans la cellule C5 la formule = H5/10000

Limiter le nombre d'année à des valeurs raisonnables.

A l'aide du cas précédent, insérez un compteur pour limiter le nombre d'année de 1 à 6.
Faites alors le nécessaire dans le formulaire et testez votre modification.

Choisir des options indépendantes (break, diesel).

Insérez deux boutons d'option en dessous du formulaire. Changez les textes de ceux-ci (Essence, Diesel) et testez-les.
On peut alors modifier le prix de base pour tenir compte de ces options, par exemple Break (+10 000 F) et Diesel (+5 000 F).
Il suffit de stocker l'état des boutons d'option dans 2 cellules (H7 et H8) et de mettre une formule conditionnelle pour le prix, du genre :

```
si Break = Faux alors
        si Diesel = Faux alors
                Prix = Prix_Base
        sinon
                Prix = Prix_Base + 5000
        finsi
sinon
        si Diesel = Faux alors
                Prix = Prix_Base + 10000
        sinon
                Prix = Prix_Base + 15000
        finsi
finsi
```

Choisir des options liées (couleurs).

Insérez 4 cases à cocher en dessous du formulaire, pour saisir la couleur du véhicule
(Rouge, Bleue, Verte, Jaune).
Changez donc les textes des cases et testez-les.

Vous constaterez qu'on peut choisir plusieurs couleurs or en général un véhicule n'en possède qu'une prédominante. Pour palier à cet inconvénient, il suffit d'insérer autour des cases à cocher, une zone de groupe.
En effet, à l'intérieur d'une zone de groupe, il ne peut y avoir qu'une seule case à cocher sélectionnée.
Faites la modification et testez là. Le problème est résolu.

Pour des raisons de temps, on ne prendra pas en compte les modifications de tarifs dues au changement de couleur.

Fichier Excel 2000 correspondant à cet exercice. (Gestion_Emprunt.xls)

8. Les masques de saisie.

Nous allons maintenant aborder le dernier point de la recherche de convivialité avec l'utilisateur, c'est la réalisation de masques de saisie (feuilles de dialogue programmées). Pour cela, nous allons traiter un exemple, à savoir, la gestion de fiches d'intervention d'un service maintenance.

Commencez par créer un nouveau classeur, renommez une feuille de calcul "Base", une autre "Réserve", puis effacez les autres feuilles. La première feuille sera notre base de données (informations sur les interventions) et la seconde servira à stocker les valeurs des paramètres fixes (menus).

Organisez la feuille "Base" comme le montre la figure ci-dessous :

	A	B	C	D	E	F	G	H
1	Date	H_Début	H_Fin	Temps	Machine	Maintenance	Domaine	Intervenant
2	12/11/99	10:00	17:00	07:00	Fraiseuse	Corrective	Régulation	M. Dépan 3
3	14/11/99	08:00	11:00	03:00	Tour //	Préventive	Electricité	M. Dépan 2
4	12/09/99	12:00	18:00	06:00	Tour //	Corrective	Régulation	M. Dépan 3

Organisez la feuille "Réserve" comme le montre la figure ci-dessous :

	A	B
1	Machine	Intervenant
2	Fraiseuse	M. Dépan 1
3	Tour CN	M. Dépan 2
4	Palettic	M. Dépan 3
5	Tour //	M. Dépan 4
6	Ravoux	
7		

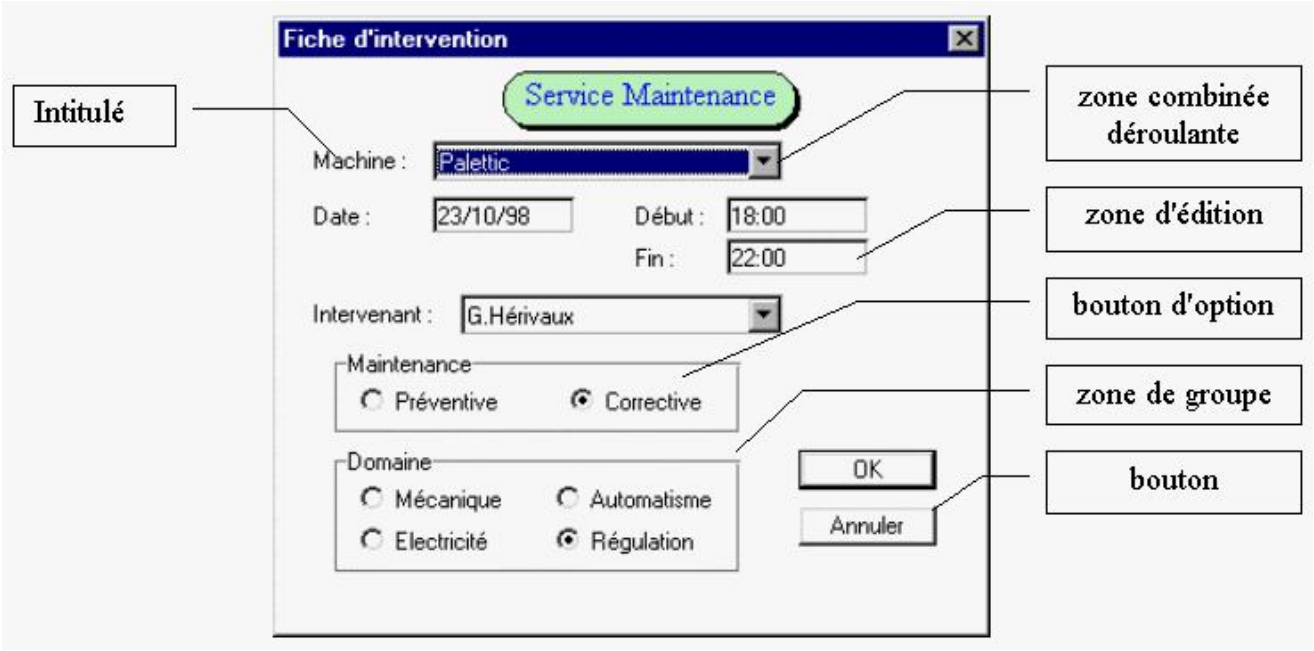
Il existe deux méthodes pour insérer une boite de saisie :

- l'insertion d'une feuille de dialogue dans le classeur.
  c'est la méthode utilisée par excel 5. Les options sont limitées, mais c'est rapide à mettre en œuvre...

- l'insertion d'un UserForm dans le module visual basic.
  c'est la méthode la plus élégante, que l'on abordera en second.

Créer un masque de saisie. (méthode simplifiée)

Nous allons présenter la première méthode, à savoir, insérer une boîte de dialogue, pour cela, sélectionnez un onglet de feuille, cliquez à droite, puis sélectionnez *insérer / boîte de dialogue excel 5*. Une nouvelle feuille de type boîte de dialogue apparaît.
Renommez-là "Mas Int". Vous allez organiser une boîte de saisie comme le montre la figure ci-dessous.



Commencez par changer le nom de la boîte en cliquant dans l'en-tête.
Puis, insérez une zone de texte (menu dessin) pour l'intitulé "Service Maintenance".
Insérez une étiquette (intitulé) "Machine",
puis une zone combinée déroulante faisant référence à la plage machine de la feuille "Réserve".
Insérez ensuite les étiquettes "Date", "Début" et "Fin", puis à coté de ces dernières des zones d'édition pour la saisie respectivement de la date, l'heure de début et l'heure de fin de l'intervention.
Sur le même principe que la partie "Machine", insérez la partie "Intervenant".

Ensuite, insérez les cases d'option "Préventive", "Corrective", "Mécanique", "Électricité", "Automatisme" et "Régulation".
Enfin, créez deux zones de groupe "Maintenance" et "Domaine" autour des cases d'option précédente.

Votre boîte de saisie est presque prête.

Afin de faciliter le suivi de votre programme, nommez les zones qui seront accessibles à l'utilisateur respectivement "Machine", "Date", "Début", "Fin", "Intervenant", "Préventive", "Corrective", "Méca", "Elec", "Auto" et "Régul". Ordonnez ces zones à l'aide du menu contextuel (Ordre de tabulation)

Testez le fonctionnement de votre boîte avec le bouton exécuter la boîte de dialogue.

Programmer l'utilisation d'un masque de saisie.(méthode simplifiée)

Vous allez maintenant réaliser les macros nécessaires à la gestion de votre base de données et au pilotage de votre masque de saisie.

- Déclaration des variables.

  Commencez par déclarer vos variables qui serviront au stockage des informations de la boîte de saisie.
  On peut s'obliger à déclarer les variables à l'aide de la commande : option explicit que l'on insère en début de module.

  Ensuite, on déclare chaque variable avec l'instruction Dim , par exemple :

```
Dim Machine
Dim Date
etc...
```

- Affichage et initialisation de la boîte de saisie.

  Il faut programmer l'affichage de la boîte de saisie lorsque l'on agit sur le bouton "nouvelle intervention".
  De plus, on va l'initialiser à l'affichage. La procédure a la structure suivante :

```
Sub Saisie_int()
    With WorkSheets("Mas Int")
        .EditBoxes.Text = " "                    vide les zones d'édition
        .OptionButtons("Préventive") = xlOn      force l'option "préventive"
        .OptionButtons ("Mécanique") = xlOn      force l'option "mécanique"
        .DropDowns.Value = 1                     initialise les zones combinées au 1er choix
        .Show                                    affiche le masque
    End With
```

```
End Sub
```

- Sortie de la boîte de saisie sans action (Annuler).

  Lorsque l'on actionne le bouton "Annuler", il faut prévoir une macro qui permet de retourner sur la feuille de base, du genre :

```
Sub Annuler_QuandClic()
    Sheets("Base").Activate
    Range("A1").Select
End Sub
```

- Sortie de la boîte de saisie avec action (OK).

  De même, lorsque l'on agit sur le bouton "OK", il faut prévoir par programmation :
  - la récupération (transfert dans des variables) des données de la boite de saisie.
  - le traitement (calcul, si nécessaire).
  - le remplissage de la base de données.

```
Sub Ok_Quand_Clic()
    With ActiveDialog
        Machine = .DropDowns("Machine").Value
        Date = .EditBoxes("Date").Text
        H_Début = . EditBoxes ("Début").Text
        etc...
        Maintenance = .OptionButtons("Préventive").Value
        etc...
    End With
    Mise_A_Jour          traitement & remplissage de la base
End Sub
```

- Traitement et remplissage.

  Pour programmer la mise à jour de la base de données, vous allez devoir créer des procédures pour :

  - Ajouter une ligne en bas du tableau.
  - Faire les calculs (durée).
  - Rechercher la machine et l'intervenant.
    (la zone combinée déroulante d'une boîte de saisie retourne la position dans la liste et non la valeur de la zone...)
  - Transférer les données des variables vers la feuille.
  - Encadrer les cellules.

- Terminer le programme.

  Pour terminer votre programme, il vous reste à prévoir :

  - La suppression d'une ligne avec confirmation.

```
Sub Supprimer_int()
Dim Réponse; Question; Titre
    Question = "Voulez-vous supprimer cette intervention ?"
    Titre = "Suppression intervention"
    Réponse = MsgBox(Question; 276; Titre)
    If Réponse = 6 Then
        Supprimer_ligne
    Else
        Range("A1").Select
    End If
End Sub
```

  - Le tri des données (par date ,durée, maintenance ou intervenant).

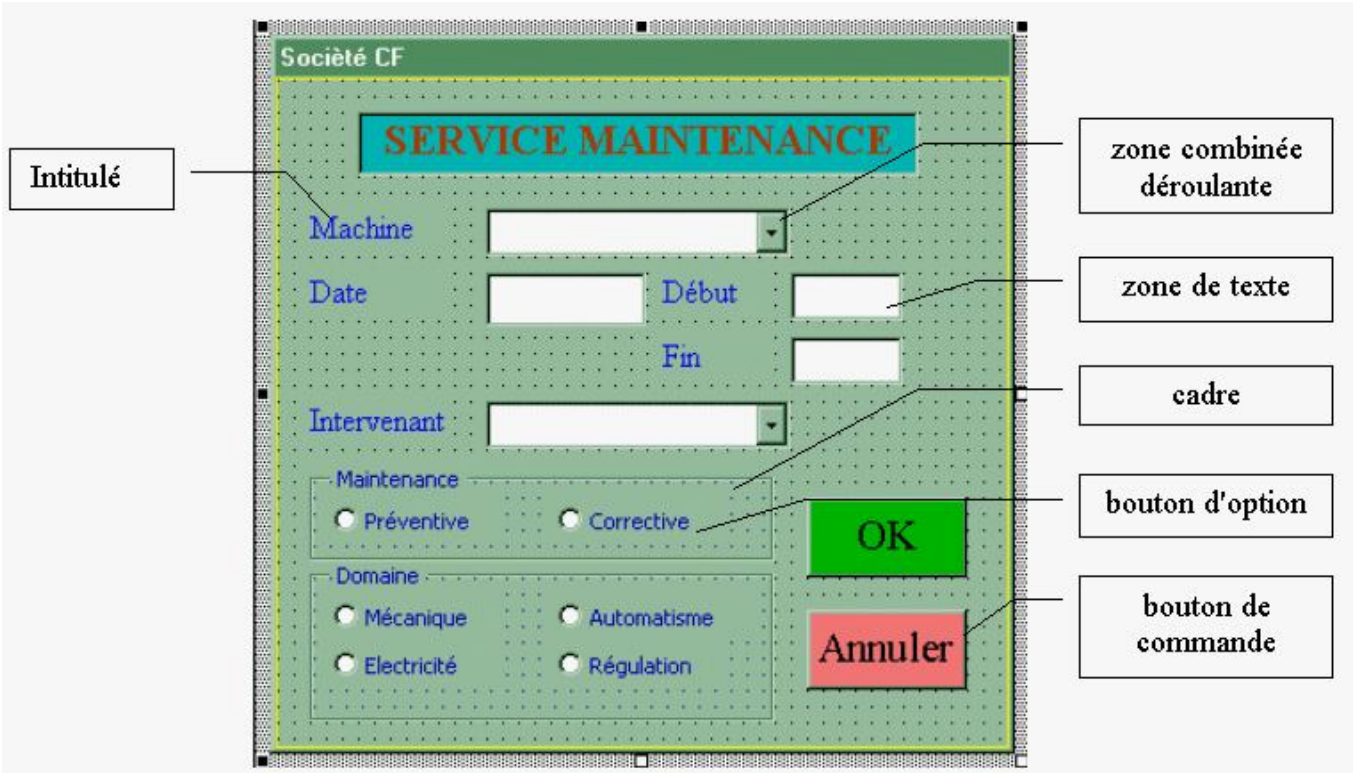
Il ne vous reste plus qu'à associer vos macros aux boutons de la base et de tester votre application.
Si tout fonctionne bien, ... Bravo!!!
Sinon faites les modifications nécessaires....

Télécharger Fichier Excel 2000 correspondant à cet exercice.

(Planning_Intervention.xls)

Créer un masque de saisie.(méthode programmeur)

La seconde méthode consiste à créer un objet utilisateur (UserForm) dans le module de programmation visual basic, à savoir, *insérer / UserForm*. Un nouvel objet apparaît.
Vous pouvez le renommer avec la propriété "Caption".
Vous allez organiser votre objet comme le montre la figure ci-dessous.



Le principe est un peu analogue aux boîtes de saisie, sauf que l'on est orienté objet et par conséquent on modifie les propriétés de ces objets à volonté, ce qui donne plus de possibilités.
Les principales propriétés sont :

- Caption : nom de l'étiquette
- Name : nom de l'objet
- Font : choix de la police de caractère
- ForeColor : couleur de la police.
- BackColor : couleur du fond
- TextAlign : alignement texte (2-centré)

Programmer l'utilisation d'un masque de saisie.(méthode programmeur)

Vous allez maintenant réaliser les macros nécessaires à la gestion de votre base de données et au pilotage de votre masque de saisie.

- Déclaration des variables.

  La démarche est identique à la méthode simplifiée , cependant il vaut mieux déclarer les variables comme publiques
  (cad accessible de partout), avec l'instruction Public , par exemple :

```
Public Machine
Public Date
etc...
```

- Affichage et initialisation de la boîte de saisie.

  Il faut programmer l'affichage de la boîte de saisie lorsque l'on agit sur le bouton "nouvelle intervention".
  De plus, on va l'initialiser à l'affichage de l'objet. La procédure a la structure suivante :

```
Sub Saisie_int()
    Load USF_Inter
```

```
        With USF_Inter
            .Cbx_machine.List = Sheets("Réserve").Range("A2:A6").Value
            .Txt_date.Value = " "
            .Txt_début.Value = " "
            .Txt_fin.Value = " "
            .Cbx_inter.List = Sheets("Réserve").Range("B2:B5").Value
            .Opb_prev = xlOn
            .Opb_meca = xlOn
        End With
        USF_Inter.Show
    End Sub
```

On peut constater que le remplissage des zones combinées déroulantes s'opère lors de l'initialisation, par une programmation.

- Sortie de la boîte de saisie sans action (Annuler).

  La macro qui permet de sortir de l'UserForm est une macro privée de l'objet du genre :

```
    Private Sub Cmb_annuler_Click()
        Unload USF_Inter
         ... /...
    End Sub
```

- Sortie de la boîte de saisie avec action (OK).

  De même, lorsque l'on agit sur le bouton "OK", il faut prévoir par programmation :
  - la récupération (transfert dans des variables) des données de la boite de saisie.
  - le traitement (calcul, si nécessaire).
  - le remplissage de la base de données.

```
    Private Sub Cmb_ok_Click()
        With USF_Inter
            Machine = .Cbx_machine.Value
            H_Date = .Txt_date.Value
            ... /...
        End With
        Unload USF_Inter
        Mise_A_Jour
    End Sub
```

- Traitement et remplissage.

  idem méthode simplifiée

- Terminer le programme.

  idem méthode simplifiée

Il ne vous reste plus qu'à associer vos macros aux boutons de la base et de tester votre application.
Si tout fonctionne bien, ... Bravo!!!
Sinon faites les modifications nécessaires....

Fichier Excel 2000 correspondant à cet exercice.
(Planning_Intervention2.xls)

Maj. page : 17/03/04