

S.G.B.D. 1

TD n° 9

Introduction à ACCESS – Modules (1)

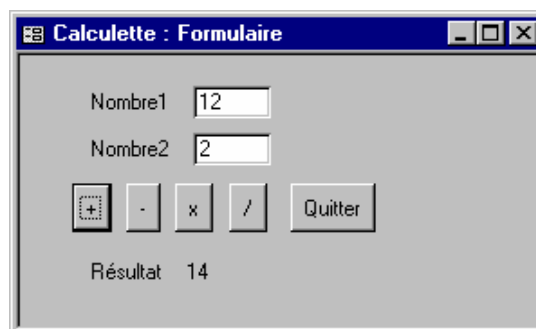
- CORRIGE -

Objectif : manipuler les contrôles de base, créer des procédures événementielles, utiliser SQL dans les procédures Visual Basic pour exécuter des requêtes Action.

1. Manipuler les contrôles de base

Note : vous pouvez afficher les propriétés d'un contrôle par un double clic sur ce contrôle lorsque vous êtes en Mode création ; vous pouvez aussi accéder à la référence Visual Basic d'une propriété : pour cela appuyer sur F1 lorsque le pointeur se trouve sur valeur de la propriété qui vous intéresse.

- 1.1. Créer un formulaire dans lequel l'utilisateur aura deux nombres à saisir, à la suite de quoi il devra cliquer sur l'un des quatre boutons représentant les opérations +, -, x, /. La procédure événementielle qui sera déclenchée devra effectuer l'opération choisie puis afficher le résultat. Celui ci sera inscrit en rouge s'il est négatif. Le bouton Quitter permettra de fermer le formulaire.



Ouvrir un nouveau formulaire en Mode Création.

Créer la zone de texte pour saisir le nombre 1, la nommer *nombre1* (propriété Name).

Créer la zone de texte pour saisir le nombre 2, la nommer *nombre2*.

Créer les boutons des 4 opérations et le bouton Quitter, les nommer *plus*, *moins*, *mult*, *div*, *quitter*.

Créer l'étiquette pour afficher le résultat d'une opération, la nommer *resultat*.

Ajouter à gauche de l'étiquette *resultat*, un autre champ étiquette pour inscrire le libellé « Résultat »

Créer le code événement Sur clic pour chacun des boutons :

```
Private Sub plus_Click()  
    resultat.Caption = Val(nombre1.Value) + Val(nombre2.Value)  
    If resultat.Caption < 0 Then  
        resultat.ForeColor = RGB(255, 0, 0)  
    End If  
End Sub  
  
Private Sub moins_Click()  
    resultat.Caption = Val(nombre1.Value) - Val(nombre2.Value)  
    If resultat.Caption < 0 Then  
        resultat.ForeColor = RGB(255, 0, 0)  
    End If  
End Sub  
  
Private Sub mult_Click()  
    resultat.Caption = Val(nombre1.Value) * Val(nombre2.Value)  
    If resultat.Caption < 0 Then  
        resultat.ForeColor = RGB(255, 0, 0)  
    End If  
End Sub
```

```

End If
End Sub

Private Sub div_Click()
    resultat.Caption = Val(nombre1.Value) / Val(nombre2.Value)
    If resultat.Caption < 0 Then
        resultat.ForeColor = RGB(255, 0, 0)
    End If
End Sub

Private Sub quitter_Click()
    DoCmd.Close
End Sub

```

- 1.2. Créer un module Général pour contenir des fonctions ou des procédures utilitaires qui pourront être exécutées en tout point de votre base de données.

Dans la fenêtre Base de Données sélectionnez l'onglet Modules, puis cliquez sur Nouveau.

- 1.3. Dans le module Général, rédiger la fonction dateDeSemaine (uneDate As Date) qui renvoie la date de début de la semaine (du lundi) d'une date spécifiée en paramètre (suggestion : utiliser la fonction DatePart de Visual Basic).

```

Function dateDeSemaine(uneDate As Date)
    dateDeSemaine = uneDate - DatePart("w", uneDate, 2) + 1
End Function

```

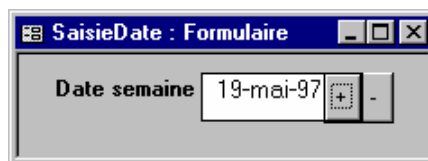
- 1.4. Tester la fonction dateDeSemaine dans la fenêtre de débogage Visual Basic.

Cliquer sur le bouton Fenêtre débogage dans la barre d'outil Visual Basic

Tester la fonction, par exemple :

```
? dateDeSemaine (#02/25/98#)
```

- 1.5. Créer un formulaire qui affiche à son ouverture la date de début de la semaine courante, et qui comporte deux boutons + et - pour faire avancer ou faire reculer la date d'une semaine.



Créer la zone de texte affichant la date, la nommer uneDateDeSemaine, définir sa propriété Verrouillé sur Oui.

Créer le Code événement qui sera exécuté à l'ouverture du formulaire.

```

Private Sub Form_Load()
    uneDateDeSemaine = dateDeSemaine(Date())
End Sub

```

Créer le Code événement Sur clic de chacun des boutons.

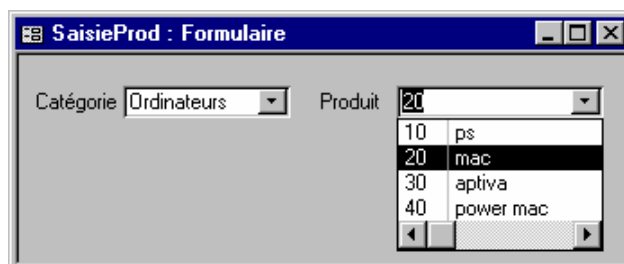
```

Private Sub plus_Click()
    uneDateDeSemaine = uneDateDeSemaine + 7
End Sub
Private Sub moins_Click()
    uneDateDeSemaine = uneDateDeSemaine - 7
End Sub

```

Pour la suite, vous utiliserez le jeu d'essai <\\chopin\prof\dn\pub\MICRO.MDB> que vous copierez dans votre répertoire de travail.

- 1.6. Créer un formulaire comportant deux zones de liste modifiable : l'une pour sélectionner une catégorie de produits, l'autre pour sélectionner un produit de la catégorie qui a été sélectionnée. La première liste sera alimentée avec les champs *IdCat* et *Désignation* de la table Catégorie. La deuxième liste sera alimentée sur l'événement Après MAJ de la liste catégorie. Elle affichera les champs *IdPro* et *Désignation* des produits de la catégorie sélectionnée dans la première liste.



Créer la zone de liste modifiable pour sélectionner une catégorie :

```

Nom..... w_IdCat
Contenu ..... SELECT C.IdCat, C.Désignation FROM Categorie C ;
Nbre colonnes ..... 2
Largeurs colonnes ..... 0cm ; 2,5cm
Colonne liée ..... 1
Limiter à liste ..... Oui
Après MAJ ..... [Procédure événementielle]
Private Sub w_IdCat_AfterUpdate()
    Dim chSQL As String
    chSQL = "SELECT P.IdPro, P.Désignation "
    chSQL = chSQL + "FROM Produit P "
    chSQL = chSQL + "WHERE P.IdCat = w_IdCat;"
    w_IdPro.Value = ""
    w_IdPro.RowSource = chSQL
End Sub

```

Créer la zone de liste modifiable pour sélectionner un produit :

```

Nom..... w_IdPro
Nbre colonnes ..... 2
Largeurs colonnes ..... 0,75cm ; 2,5cm
Colonne liée ..... 1
Limiter à liste ..... Oui

```

2. Requêtes Action à l'intérieur de procédures Visual Basic

Note : pour exécuter une requête Action à l'intérieur d'une procédure Visual Basic, vous utiliserez la méthode *Execute* de l'objet base de donnée courante. La phrase de requête SQL est passée en argument de cette méthode sous la forme d'une chaîne de caractères.

- 2.1. Créer un formulaire pour saisir la quantité livrée d'un produit et mettre à jour la quantité en stock dans la table Produit.



Créer les deux zones de liste modifiable comme dans l'exercice 1.3

Créer la zone de texte pour saisir la quantité livrée, la nommer *uneQte*.

Créer les deux boutons de commande, les nommer Appliquer et Annuler

Créer le Code événement Sur clic de chacun des deux boutons :

```

Private Sub Appliquer_Click()
    Dim oBD As Database
    Dim chSQL As String
    ' Retourne une référence à la base de données en cours.
    Set oBD = CurrentDb
    ' Construire la phrase de requête SQL
    chSQL = "UPDATE produit P " _
        & "SET Qstock = Qstock + " & CStr(uneQte) & " " _
        & "WHERE P.IdPro = '" & w_IdPro & "';"
    oBD.Execute (chSQL)
    MsgBox ("MAJ EFFECTUEE"+Chr(10)+Str(oBD.RecordsAffected)+"lignes affectées")
    Set oBD = Nothing
End Sub

Private Sub Annuler_Click()
    DoCmd.Close
End Sub

```

- 2.2. Créer un formulaire permettant de saisir un nouveau produit. La quantité en stock sera initialisée à 0.

Créez les zones de texte pour saisir les différentes informations, les nommer w_IdPro, w_IdCat, w_Désignation, w_Marque, w_PrixUht .

Créer les deux boutons et le Code événement Sur clic qui leur sont associées :

```

Private Sub Appliquer_Click()
Dim bds As Database
Dim chSQL As String
' Retourne une référence à la base de données en cours.
Set bds = CurrentDb
' Construire la phrase de requête SQL
chSQL = "INSERT INTO produit VALUES ( " _
& "'" & w_IdPro & "', " _
& "'" & w_IdCat & "', " _
& "'" & w_Désignation & "', " _
& "'" & w_Marque & "', " _
& w_PrixUht _
& ", 0 );"
bds.Execute (chSQL)
MsgBox ("AJOUT EFFECTUE")
Set bds = Nothing
End Sub

Private Sub Annuler_Click()
DoCmd.Close
End Sub

```