

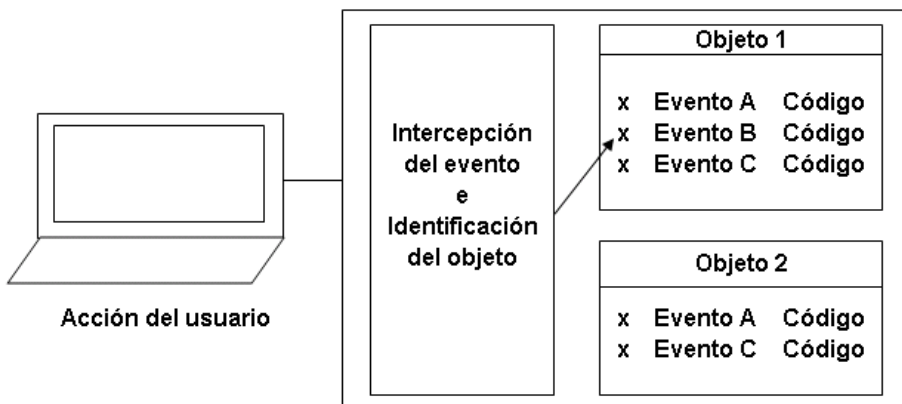
Capítulo 9

Administración de eventos

1. Presentación

Un evento es una **acción** del usuario o del sistema reconocido por un objeto de Microsoft Excel. El evento desencadena un procedimiento, asociado al evento del objeto activo.

Estos procedimientos le permiten asociar un código personalizado en respuesta a un evento que se produce en un objeto de Excel (libro, hoja, formulario, gráfico, etc.).



2. Escritura de eventos

2.1 Eventos de libro, de hoja o de formulario

Usted puede acceder a los procedimientos de eventos asociados a un objeto de la siguiente manera:

- ▶ En la ventana **Explorador de proyectos**, haga doble clic en el objeto deseado (libro, hoja o formulario) para hacer que aparezca la ventana de código correspondiente.
- ▶ Abra la lista desplegable a la izquierda de la ventana de código y seleccione **Workbook**, **Worksheet** o **UserForm**, según el objeto seleccionado.
- ▶ También puede seleccionar un evento vinculado al objeto seleccionado en la lista desplegable de la derecha, para asociarle un código personalizado.

■ Observación

*La ejecución de los procedimientos de eventos se puede desactivar en cualquier momento asignando el valor **False** a la propiedad **EnableEvents** del objeto **Application**.*


2.2 Eventos del objeto Application

Se necesitan tres etapas para la escritura y ejecución de los eventos del objeto **Application**.

Etapa 1

▣ Inserte un módulo de clase:

Insertar - Módulo de clase

o abra la lista  y haga clic en **Módulo de clase**.

▣ Una vez insertado el módulo, asígnele un nombre.

Ejemplo

Dele el nombre **ObjApplication** al módulo de clase.

Etapa 2

▣ En el módulo de clase, cree un objeto **Application** con el siguiente código:

```
Public WithEvents NomObjeto As Application
```

Ejemplo

Creación del objeto **oMiAplicación** como aplicación.

```
Public WithEvents oMiAplicacion As Application
```

El objeto así creado queda disponible en la lista de la izquierda del módulo.

▣ Seleccione el objeto creado en la lista de la izquierda del módulo y luego seleccione el evento esperado en la lista de la derecha. Escriba el código de los procedimientos que desea generar.

Ejemplo

Creación de dos procedimientos de eventos: el primero realiza la inserción de una nueva hoja; el segundo, la creación de un nuevo libro.

```
Public WithEvents oMiAplicacion As Excel.Application
```

```
Private Sub oMiAplicacion_WorkbookNewSheet _  
    (ByVal Wb As Workbook, ByVal Sh As Object)  
    Dim oNomHoja As String  
    ' Cada vez que se agrega una hoja, se pide al usuario  
    ' que introduzca un nombre que a continuación se asignará a la hoja  
    ' insertada tras las hojas existentes  
    oNomHoja = InputBox("Introduzca el nombre de la hoja")  
    ActiveSheet.Name = oNomHoja  
    ActiveSheet.Move After:=Sheets(Sheets.Count)  
End Sub
```

```
Private Sub oMiAplicacion_NewWorkbook(ByVal Wb As Workbook)  
    Dim iNbHojas As Integer  
    Dim iNbActual As Integer  
    Dim iDiferencia As Integer  
    ' Por cada nuevo libro,  
    ' solicitamos al usuario la cantidad de hojas  
    ' Según el caso, se agregan o eliminan las hojas necesarias  
    Do  
        iNbHojas = Application.InputBox _  
            ("¿Cantidad de hojas?", Type:=1)  
    Loop While iNbHojas = False  
    iNbActual = Sheets.Count  
    iDiferencia = iNbActual - iNbHojas  
    ' Eliminar las hojas de más  
    ' Eliminar los mensajes de alerta con el fin  
    ' de no obtener mensajes en la eliminación de hojas  
    Do While iDiferencia > 0  
        Application.DisplayAlerts = False  
        Sheets.Item(iDiferencia).Select  
        ActiveWindow.SelectedSheets.Delete  
        iDiferencia = iDiferencia - 1  
    Loop  
  
    ' Agregar hojas si es necesario  
    ' Se desactivan los eventos para  
    ' no indicar los nombres de las nuevas hojas  
    Do While iDiferencia < 0
```

```
Application.EnableEvents = False
Sheets.Add After:=Sheets(Sheets.Count)
    iDiferencia = iDiferencia + 1
Loop
'   Reactivar eventos y alertas
Application.EnableEvents = True
Application.DisplayAlerts = True
End Sub
```

Etapa 3

- Active un módulo cualquiera y conecte el objeto declarado en el módulo de clase con el objeto **Application** para las siguientes instrucciones:

```
Dim oNomVariable As New NomModuloDeClase

Sub NomProced ()
Set oNomVariable.NomObjeto = Application
End Sub
```

Ejemplo

Agregue el siguiente código en el módulo Declaraciones.

```
Option Explicit
Dim oApp As New ObjApplication

Sub InicializaMiAplicacion()
    Set oApp.oMiAplicacion = Application
End Sub
```

Finalmente, llame al procedimiento InicializaMiAplicacion al abrir el libro (módulo de clase ThisWorkbook).

```
Private Sub Workbook_Open()
    InicializaMiAplicacion
End Sub
```

Cuando se abra el libro, se ejecutarán automáticamente los procedimientos de eventos creados durante la etapa 2 y se agregarán los libros o las hojas. Estos procedimientos se desactivarán al cerrar el libro.

Chapitre 4

Structures de contrôle

Durée : 1 heure 55

Mots-clés

condition, choix, test, alternative, branchement conditionnel, compteur, itération, incrémenter, décrémenter, sortie

Objectifs

Maîtriser les structures de décision afin de tester des conditions puis effectuer des actions différentes selon le résultat obtenu. Maîtriser les instructions d'itération qui, associées aux instructions conditionnelles, permettent d'écrire du code Visual Basic pour la prise de décision et la répétition des actions. Vous retrouverez ces structures dans la suite du livre. Dans ce chapitre, nous nous limitons à l'emploi des boîtes de dialogue déjà rencontrées.

Prérequis

Pour valider les prérequis nécessaires, avant d'aborder le TP, répondez aux questions ci-après (certaines questions peuvent nécessiter plusieurs réponses) :

1. Les structures suivantes sont des structures de décision :
 - a. If ... Then ... Else ... End If
 - b. Do ... Loop
 - c. Select Case ... Case ... End Select
2. Résultat = IIf(7 / 2 > 3, IIf(2.8 * 3.3 < 11, "X", "Y"), "Z")
La variable Résultat contient la valeur :
 - a. X
 - b. Y
 - c. Z

3. Les instructions suivantes appartenant chacune à une structure de contrôle différente sont correctes :
 - a. Case If N1 > N2
 - b. Case A, B, C
 - c. Case 1 to 10
 - d. Case Nombre, Is > 50
4. Le mot-clé `ElseIf` :
 - a. peut apparaître de suite après une clause `Else`.
 - b. est facultatif.
 - c. peut être utilisé plusieurs fois dans un bloc `If`.
5. Répétition d'instructions tant qu'une condition a la valeur `True` :
 - a. For Each ... Next
 - b. For ... Next
 - c. Do ... Loop
 - d. While ... Wend
 - e. With ... End With
6. Utilisation d'un compteur pour exécuter des instructions un certain nombre de fois :
 - a. For Each ... Next
 - b. For ... Next
 - c. Do ... Loop
 - d. While ... Wend
 - e. With ... End With
7. Répéter un groupe d'instructions pour chaque élément d'un tableau ou d'une collection :
 - a. For Each ... Next
 - b. For ... Next
 - c. Do ... Loop
 - d. While ... Wend
 - e. With ... End With

8. Répéter un groupe d'instructions le nombre de fois indiqué :
 - a. For Each ... Next
 - b. For ... Next
 - c. Do ... Loop
 - d. While ... Wend
 - e. With ... End With
9. Exécuter une série d'instructions appliquées à un seul objet ou à un type défini par l'utilisateur :
 - a. For Each ... Next
 - b. For ... Next
 - c. Do ... Loop
 - d. While ... Wend
 - e. With ... End With

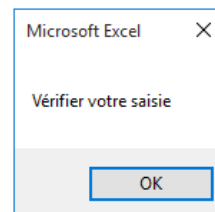
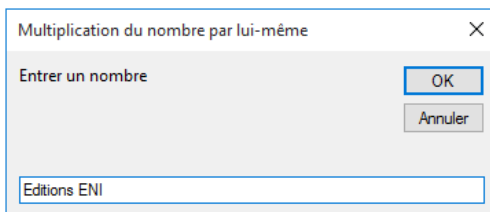
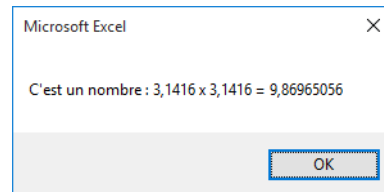
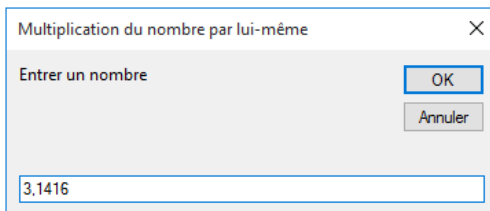
Corrigé p. 259

Énoncé 4.1 Vérifier que la saisie est numérique

Exercice 1

Durée estimative : 10 minutes

Créez la procédure **ContrôleSaisie** qui propose une boîte de saisie et vérifiez qu'il s'agit d'un nombre. Si c'est le cas, effectuez l'opération qui consiste à multiplier le nombre par lui-même, sinon affichez un message. Exemple :



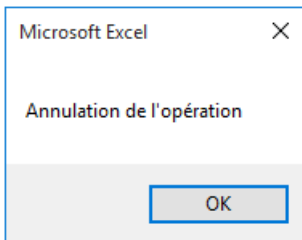
Indice

Utilisez la fonction `IsNumeric` pour tester la saisie.

Exercice 2

Durée estimative : 5 minutes

Complétez la procédure précédente qui vérifie que l'utilisateur a cliqué sur le bouton **OK**. Dans le cas contraire, affichez un message. Exemple :

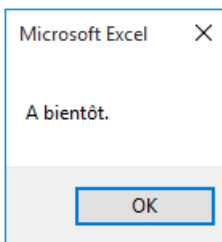
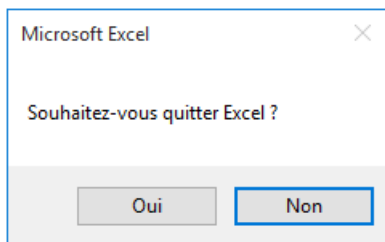


Corrigé p. 259

Énoncé 4.2 Dire au revoir

Durée estimative : 5 minutes

Complétez la procédure **AuRevoir**. Celle-ci propose une boîte de dialogue demandant si l'on souhaite quitter Excel. Elle affiche le message "A bientôt" si l'utilisateur clique sur le bouton **Oui**, et le message "Poursuivons" sinon. Exemple :



```
Sub AuRevoir()  
    Dim strMessage As String  
    Dim intStyle As Integer  
    Dim intChoix As Integer  
    strMessage = "Souhaitez-vous quitter Excel ?"  
    intStyle = vbYesNo + vbDefaultButton2  
    ... = MsgBox(strMessage, intStyle)  
    If ... = vbYes Then  
        MsgBox "A bientôt."  
    ...  
        MsgBox "Poursuivons."  
    ...  
End Sub
```

Indice

Voici un extrait de l'aide VBA en ligne :

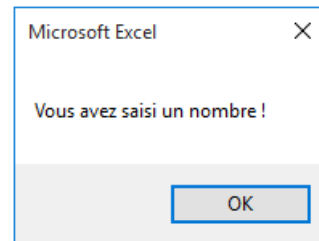
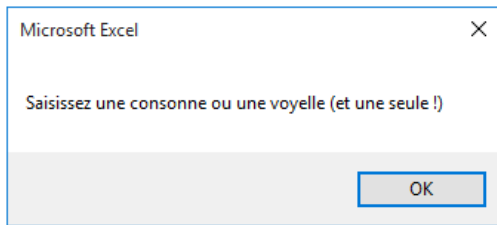
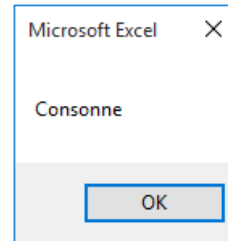
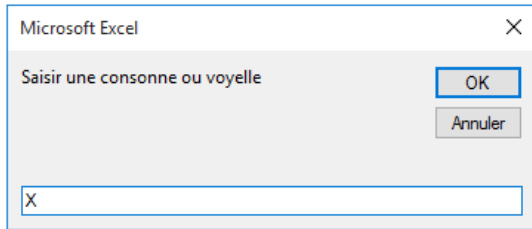
Constante	Valeur	Description
vbOKOnly	0	Bouton OK uniquement (par défaut)
vbOKCancel	1	Boutons OK et Annuler
vbAbortRetryIgnore	2	Bouton Abandonner, Réessayer et Ignorer
vbYesNoCancel	3	Boutons Oui, Non et Annuler .
vbYesNo	4	Boutons Oui et Non

Corrigé p. 260

Énoncé 4.3 Contrôler la saisie d'une consonne ou voyelle

Durée estimative : 10 minutes

Créez la procédure **ConsonneVoyelle** qui demande de saisir une voyelle ou une consonne. Affichez un message selon la saisie : "Voyelle", "Consonne" ou un message approprié en cas d'erreur. Exemple :



Indice

Pour tester la lettre en majuscule, on effectuera la transformation suivante :

```
varChoix = VBA.UCase(varChoix)
```

Corrigé p. 260

Énoncé 4.4 Afficher un message selon l'âge et le genre

Durée estimative : 15 minutes

Créez la procédure **VotreAge** qui demande le sexe et l'âge de l'utilisateur. Les messages seront différents selon qu'il s'agisse d'un homme ou d'une femme. Prenez également en considération l'abandon de la saisie et la saisie de lettres pour l'âge.

0 - 17 ans : Gamin

18 - 30 ans : Vous êtes jeune

31 - 50 ans : Vous êtes encore jeune

> 50 ans : Vous commencez à vieillir