

vBook

livre &
vidéo



VBA Excel 2013



Programmez sous Excel :
Macros et langage VBA



Développez un ruban
personnalisé
1 H 50 de vidéo

Michèle AMELOT
Claude DUIGOU

Téléchargement
www.editions-eni.fr



Les éléments à télécharger sont disponibles à l'adresse suivante :
<http://www.editions-eni.fr>
Saisissez la référence ENI de l'ouvrage **RI13EXCV** dans la zone de recherche et validez. Cliquez sur le titre du livre puis sur le bouton de téléchargement.

Avant-propos

Chapitre 1 Présentation

- 1. Présentation du langage VBA 17
 - 1.1 Objectifs du langage VBA 17
 - 1.2 Quelques définitions 18
 - 1.3 Écriture de code VBA 19
- 2. Les macros d'Excel 19
 - 2.1 Affichage de l'onglet DÉVELOPPEUR dans le ruban 19
 - 2.2 Description de l'onglet DÉVELOPPEUR 20
 - 2.2.1 Groupe Code 20
 - 2.2.2 Groupe Compléments 21
 - 2.2.3 Groupe Contrôles 21
 - 2.3 L'enregistrement de macros 22
 - 2.3.1 Enregistrement d'une première macro 22
 - 2.3.2 Exécuter une macro 23
 - 2.3.3 Enregistrer une macro avec des références de cellules relatives 23
 - 2.3.4 Définir le lieu de stockage d'une nouvelle macro 25
 - 2.3.5 Supprimer une macro 26
 - 2.3.6 Enregistrer un classeur avec des macros 26

2.4	Les macros et la sécurité.	27
2.4.1	Modification des paramètres de sécurité	27
2.4.2	Description des différentes options de sécurité	28
2.4.3	Activer les macros lorsque l'avertissement de sécurité est affiché	29
2.4.4	Activer les macros situées dans un emplacement donné.	30
2.4.5	Signatures électroniques de macros.	30
2.5	Modification du code d'une macro	31
3.	Les affectations de macros	33
3.1	Accéder à une macro à partir du ruban Office 2013.	33
3.2	Associer une macro à une icône de la barre d'outils Accès rapide	36
3.3	Associer une macro à un bouton de commande.	37
3.4	Associer une macro à une image	38
3.5	Associer une macro à une zone d'un objet graphique	38
3.6	Associer une macro à un contrôle ActiveX image	39
4.	L'environnement de développement VBE	40
4.1	Accès à l'environnement VBE	40
4.2	Fermeture de l'environnement VBE	40
4.3	Retour à l'environnement Excel.	41
4.4	Description de l'environnement VBE	41
4.5	Choix des fenêtres à afficher	44
5.	Configuration de l'éditeur VBA.	44
5.1	Paramétrage des polices	44
5.2	Paramétrage de la saisie du code	45
5.3	Gestion des erreurs	46
5.4	Ancrage d'une fenêtre.	47

Chapitre 2
Le langage VBA

- 1. Les modules 49
 - 1.1 Présentation 49
 - 1.2 Accès aux modules 51
 - 1.3 Import et export de code VBA 51
- 2. Les procédures 52
 - 2.1 Définitions 52
 - 2.2 Accès aux procédures 53
 - 2.3 Les procédures Sub 53
 - 2.4 Les procédures Function 54
 - 2.5 Déclaration des procédures 55
 - 2.6 Portée des procédures 56
 - 2.7 Arguments des procédures 56
 - 2.8 Les arguments nommés 57
 - 2.9 Appel d'une procédure 59
 - 2.10 Appel d'une fonction VBA dans une formule Excel 59
 - 2.11 Exemples de procédures et fonctions 62
- 3. Les variables 63
 - 3.1 Les types de variables 63
 - 3.2 Les déclarations de variables 68
 - 3.2.1 Déclarations implicites 68
 - 3.2.2 Déclarations explicites 69
 - 3.2.3 Syntaxe des instructions de déclaration 69
 - 3.3 Les déclarations des types de variables 70
 - 3.3.1 Déclarations explicites du type 70
 - 3.3.2 Déclarations implicites du type 71
 - 3.4 Les tableaux 73
 - 3.5 Les constantes 75
 - 3.5.1 Les constantes personnalisées 75
 - 3.5.2 Les constantes intégrées 76

4 **VBA Excel 2013**

Programmer sous Excel : macros et langage VBA

4.	Les structures de décisions	77
4.1	L'instruction If	77
4.2	L'instruction Select Case	80
5.	Les structures en boucles	82
5.1	L'instruction Do...Loop	82
5.2	L'instruction While...Wend	84
5.3	L'instruction For...Next	84
5.4	L'instruction For Each...Next	86
5.5	Quitter les structures de contrôle	87
6.	Les opérateurs	88
6.1	Les opérateurs arithmétiques	88
6.2	Les opérateurs de comparaison	89
6.3	Les opérateurs logiques	90
6.4	L'opérateur de concaténation	90
6.5	Priorité des opérateurs	91
7.	Les règles d'écriture du code	92
7.1	Les commentaires	92
7.2	Le caractère de continuation	92
7.3	Les retraits	93
7.4	Les noms des procédures, variables et constantes	93

Chapitre 3

La programmation objet sous Excel

1.	Présentation	95
2.	Le modèle objet d'Excel	96
2.1	Présentation	96
2.2	Objets et collections	97
3.	Principes d'utilisation des objets et collections	100
3.1	Les propriétés	100
3.2	Propriétés représentant des objets	100
3.3	Les méthodes	102

- 3.4 Les événements 103
- 3.5 Les collections 104
- 3.6 Complément automatique des instructions 106
- 4. Instructions utilisées avec les objets 108
 - 4.1 L'instruction With 108
 - 4.2 L'instruction For Each...Next 109
 - 4.3 L'instruction If TypeOf 109
 - 4.4 L'instruction Set 110
- 5. L'explorateur d'objets 111
 - 5.1 Présentation 111
 - 5.2 Recherche dans l'Explorateur d'objets 113

Chapitre 4
Les objets d'Excel

- 1. L'objet Application 115
 - 1.1 Propriétés représentant les options d'Excel 116
 - 1.1.1 Options de la catégorie Standard 116
 - 1.1.2 Options de la catégorie Formules 117
 - 1.1.3 Options de la catégorie Vérification 119
 - 1.1.4 Options de la catégorie Enregistrement 121
 - 1.1.5 Options de la catégorie Options avancées 122
 - 1.2 Propriétés relatives à la présentation de l'application 129
 - 1.3 Propriétés diverses 131
 - 1.4 Méthodes de l'objet Application 135
 - 1.4.1 Méthodes agissant sur les formules et calculs 135
 - 1.4.2 Méthodes agissant sur les cellules 136
 - 1.4.3 Méthodes agissant sur les listes personnalisées 136
 - 1.4.4 Méthodes affichant des boîtes de dialogue 136
 - 1.4.5 Méthodes se rapportant aux actions dans Excel 137
 - 1.4.6 Méthodes relatives à la messagerie 138
 - 1.4.7 Autres méthodes 138

1.5	Exemples de codes utilisant l'objet Application	140
1.5.1	Modification de l'interface d'Excel.	140
1.5.2	Création d'une liste personnalisée.	140
1.5.3	Sélection de colonnes disjointes	141
1.5.4	Évaluation du résultat d'une formule	141
2.	L'objet Workbook	142
2.1	Objets et collections.	143
2.2	Propriétés.	145
2.2.1	Propriétés relatives à la mise à jour et l'enregistrement de classeurs	145
2.2.2	Propriétés relatives aux classeurs partagés	146
2.2.3	Autres propriétés	149
2.3	Liste des méthodes	151
2.3.1	Méthodes agissant directement sur les classeurs	151
2.3.2	Méthodes relatives à la sécurité.	153
2.3.3	Méthodes relatives aux classeurs partagés	153
2.3.4	Méthodes se rapportant aux données liées.	154
2.3.5	Méthodes relatives à l'envoi de classeur	155
2.3.6	Autres méthodes.	156
2.4	Exemples de codes utilisant l'objet Workbook	156
2.4.1	Création d'un classeur Excel	156
2.4.2	Import d'une base de données et export au format HTML	157
2.4.3	Affichage des propriétés d'un classeur.	158
2.4.4	Export du classeur au format PDF.	159
3.	L'objet Worksheet.	159
3.1	Liste des objets et collections.	160
3.2	Objets et collections.	160
3.3	Propriétés.	163
3.4	Méthodes	165

- 3.5 Exemples de codes utilisant l'objet Worksheet 168
 - 3.5.1 Tri des feuilles de calcul d'un classeur 168
 - 3.5.2 Protection des feuilles de calcul d'un classeur 169
 - 3.5.3 Tri d'un tableau 169
- 4. L'objet Range 170
 - 4.1 Propriétés et méthodes renvoyant un objet Range 170
 - 4.2 Syntaxes des propriétés renvoyant un objet Range 173
 - 4.3 Liste des objets et collections 177
 - 4.4 Propriétés 179
 - 4.4.1 Propriétés se rapportant à la position et au format des cellules 179
 - 4.4.2 Propriétés se rapportant au contenu des cellules et aux formules 180
 - 4.4.3 Autres propriétés 181
 - 4.5 Méthodes 182
 - 4.5.1 Méthodes renvoyant un objet 182
 - 4.5.2 Méthodes se rapportant à la présentation des cellules . . 182
 - 4.5.3 Méthodes se rapportant au contenu des cellules 184
 - 4.5.4 Méthodes se rapportant aux noms des cellules 186
 - 4.5.5 Méthodes se rapportant aux filtres 186
 - 4.5.6 Méthodes se rapportant au mode plan 187
 - 4.5.7 Méthodes se rapportant à l'outil d'Audit 187
 - 4.5.8 Autres méthodes 188
- 5. Exemples d'utilisation des objets 189
 - 5.1 Calcul du montant d'une prime 189
 - 5.2 Affectation de commentaires à des cellules 191
 - 5.3 Création d'un tableau croisé dynamique avec des sparklines . 193

Chapitre 5 **Les boîtes de dialogue**

1. Présentation	197
2. Les boîtes de dialogue intégrées	198
2.1 L'objet Dialog	198
2.2 Les méthodes GetOpenFileName et GetSaveAsFileName. . . .	199
3. Les boîtes de dialogue prédéfinies	201
3.1 La fonction InputBox	201
3.2 La méthode InputBox	202
3.3 La fonction MsgBox	203
3.4 Constantes utilisées dans les boîtes de dialogue	207

Chapitre 6 **Les formulaires**

1. Présentation	209
2. Créer un formulaire	210
3. Personnaliser un formulaire	219
3.1 Écrire des procédures	219
3.2 Liste des événements associés aux principaux contrôles	220
3.3 Exécution et fermeture d'un formulaire	226
4. Exemple de formulaire personnalisé	227
4.1 Présentation	227
4.2 Code associé au bouton macro de la fiche Employés	229
4.3 Code VBA associé au formulaire	229

Chapitre 7
Amélioration de l'interface utilisateur

- 1. Présentation 235
- 2. Personnalisation du ruban au moyen
de l'utilitaire Custom UI Editor 236
 - 2.1 Présentation de l'utilitaire Custom UI Editor 236
 - 2.2 Exemple de code XML de personnalisation 238
 - 2.3 Balises XML correspondant aux différents éléments
du ruban 241
 - 2.3.1 Onglets et groupes 241
 - 2.3.2 Principaux contrôles du ruban 242
 - 2.3.3 Les attributs des contrôles du ruban 245
 - 2.3.4 Récapitulatif des attributs par type de contrôle 247
 - 2.3.5 Images de la galerie des icônes Microsoft Office 249
 - 2.3.6 Les fonctions d'appels Callbacks 249
 - 2.3.7 Utilisation des fonctions d'appels Callbacks 252
- 3. Exemple de ruban personnalisé au moyen
de l'utilitaire Custom UI Editor 253
 - 3.1 Présentation 253
 - 3.2 Code XML du ruban 255
 - 3.3 Code VBA de personnalisation du ruban (module "Ruban") . . 257
- 4. Personnalisation du ruban au moyen
de la collection CommandBars 261
- 5. Exemples de barres de commandes 262
 - 5.1 Barres d'outils personnalisées 262
 - 5.2 Commandes de menu au format Office 2013 262
 - 5.3 Ajouter le groupe à la barre d'outils Accès rapide 263
- 6. Les barres de commandes 264
 - 6.1 Terminologie 264
 - 6.1.1 Barre de commandes 264
 - 6.1.2 Contrôle 264

6.2	Créer une barre de commandes	264
6.3	Supprimer une barre de commandes	265
6.4	Afficher une barre de commandes	266
7.	Contrôles (options ou boutons de commande) des barres de commandes	267
7.1	Ajouter un contrôle	267
7.2	Préciser l'intitulé d'un contrôle	268
7.3	Supprimer un contrôle	268
7.4	Associer une procédure à un contrôle	268
7.5	Autres propriétés	269
7.6	Listes des images associées aux boutons de commande	270
8.	Exemples de menus personnalisés	272
8.1	Présentation	272
8.2	Code des exemples	273
8.3	Code du module de classe ThisWorkbook	274
8.4	Code de la feuille "Note de Frais"	274
8.5	Code du module ProcMenus	275
8.6	Code du module ProcActions	280

Chapitre 8

Gestion des événements

1.	Présentation	283
2.	Écriture des événements	284
2.1	Événements de classeur, de feuille ou de formulaire	284
2.2	Événements de l'objet Application	286
2.3	Événement d'un graphique incorporé	289
3.	Les événements de l'objet Application	291
4.	Les événements de l'objet Workbook	296
5.	Les événements de l'objet Worksheet	299
6.	Les événements de l'objet Chart	301

Chapitre 9
Débogage et gestion des erreurs

- 1. Les différents types d'erreur 303
 - 1.1 Les erreurs de syntaxe 303
 - 1.2 Les erreurs de compilation 304
 - 1.3 Les erreurs d'exécution 305
 - 1.4 Les erreurs de logique 306
- 2. Débogage 307
 - 2.1 Présentation 307
 - 2.2 La barre d'outils Débogage 307
 - 2.3 L'objet Debug 309
- 3. Gestion des erreurs en VBA 310
 - 3.1 L'objet Err 312

Chapitre 10
Communication avec les applications Office 2013

- 1. La technologie Automation 315
 - 1.1 Présentation 315
 - 1.2 Utilisation de la technologie Automation 317
- 2. Communiquer avec Word depuis Excel 318
 - 2.1 Le modèle objet Word 318
 - 2.2 Principales collections du modèle objet Word 319
 - 2.3 Principaux objets du modèle objets de Word 320
 - 2.4 La collection Documents 322
 - 2.5 Les objets Document 322
 - 2.6 Exemple 325
- 3. Communiquer avec Access depuis Excel 327
 - 3.1 Le modèle objet Access 327
 - 3.2 Principales collections du modèle objet Access 328
 - 3.3 Principaux objets du modèle objet Access 328

3.4	Exemples	330
3.4.1	Liste des tables d'une base Access	330
3.4.2	Affichage d'une table Access dans Excel	331
3.4.3	Ouverture d'une table ou requête Access dans un nouveau classeur	332
4.	Communiquer avec Outlook depuis Excel	333
4.1	Les objets d'Outlook	333
4.2	Accès aux objets Outlook	334
4.2.1	Création d'un objet (e-mail, contact...) dans Outlook .	334
4.2.2	Accès aux objets (contacts, rendez-vous...) d'Outlook .	335
4.3	Exemple d'utilisation de l'objet MailItem	336
5.	Les objets liés ou incorporés	337
5.1	Les méthodes de l'objet OLEObject	338
5.2	Les propriétés de l'objet OLEObject	338
6.	Méthodes et propriétés relatives aux liaisons Excel	339
6.1	Méthodes et propriétés de l'objet Workbook	339
6.2	Méthodes et propriétés des autres objets	341

Chapitre 11 **Internet**

1.	Requêtes sur Internet	343
2.	L'objet QueryTable	345
2.1	Propriétés de l'objet QueryTable	346
2.2	Exemples	350
3.	Publication de pages Web	352
3.1	Association d'un élément de classeur à une page Web	353
3.2	Publication de la page Web	354
3.3	Exemple	354

4. Les objets WebOptions et DefaultWebOptions	355
4.1 Propriétés.	356
4.1.1 Options de l'onglet Général	356
4.1.2 Options de l'onglet Navigateurs	357
4.1.3 Options de l'onglet Fichiers	358
4.1.4 Autres propriétés	358
4.2 Méthode de l'objet WebOptions	359
5. Import, export et mappage de fichiers XML.	359
5.1 Collections	360
5.2 Méthodes de l'objet Workbook	361
5.3 Événements de l'objet Workbook	362
5.4 Méthodes de l'objet XmlMap	363
6. L'objet HyperLink.	363
6.1 Propriétés.	363
6.2 Méthodes.	364

Chapitre 12

Programmation Windows

1. Présentation des API.	365
2. Appel d'une fonction de l'API Windows	367
2.1 Syntaxe de l'instruction Declare	367
2.2 Passage des arguments	368
3. Liste de fonctions API Windows	368
4. Exemples d'utilisation de fonctions API Windows.	369
4.1 Récupération du répertoire Windows	369
4.2 Lancement de la calculatrice Windows.	370
5. L'objet FileSystemObject	372
5.1 Méthodes.	372
5.2 Propriété	373
5.3 Exemple : copie de fichiers Excel	374

Chapitre 13**Code d'une mini-application**

1. Présentation générale	375
2. Description de l'application GestDevis	376
3. Ruban Office 2013 personnalisé	378
3.1 Présentation	378
3.2 Code XML du ruban personnalisé	379
4. Module ThisWorkBook	380
4.1 Présentation	380
4.2 Code VBA du module ThisWorkbook	380
5. Formulaire NouveauDevis	381
5.1 Présentation	381
5.2 Liste des contrôles	381
5.3 Liste de cellules nommées du modèle Devis.xltx	382
5.4 Code VBA du formulaire NouveauDevis	382
6. Formulaire RechDevis	385
6.1 Présentation	385
6.2 Liste des contrôles	385
6.3 Code VBA du formulaire RechDevis	386
7. Modules ProcRuban et ProcGene	389
7.1 Code VBA du module ProcRuban	389
7.2 Code VBA du module ProcGene	391

Annexes

1. Liste des instructions	395
1.1 Chaînes de caractères	395
1.2 Date Heure/Mathématique	396
1.3 Déclaration	396
1.4 Erreur	398

1.5	Fichier	399
1.6	Structuration	403
1.7	Système.....	404
1.8	Divers	405
2.	Liste des fonctions	406
2.1	Conversions	406
2.2	Chaînes de caractères	408
2.3	Mathématiques.....	411
2.4	Financières.....	413
2.5	Dates et heures	415
2.6	Fichiers, Système	418
2.7	Vérifications de variables	420
2.8	Interaction.....	421
2.9	Tableau	421
2.10	SQL	422
2.11	Divers	423
2.12	Solveur.....	425
3.	Constantes VBA	426
3.1	Constantes de couleur	426
3.2	Constantes de date	427
3.3	Constantes des touches clavier lettres et chiffres.....	427
3.4	Constantes de touches de fonction	428
3.5	Constantes de touches diverses.....	428
	Index	431

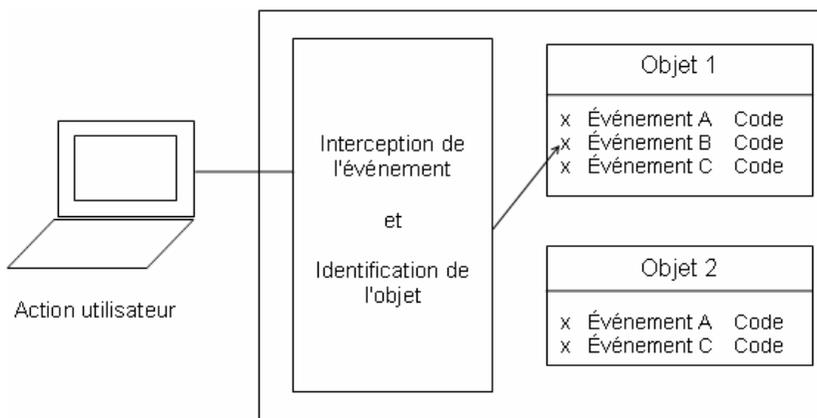
Chapitre 8

Gestion des événements

1. Présentation

Un événement est une **action** utilisateur ou système reconnue par un objet Microsoft Excel. Il déclenche la procédure événementielle associée à l'événement de l'objet activé.

Les procédures événementielles vous permettent d'associer un code personnalisé en réponse à un événement qui se produit sur un objet Excel (classeur, feuille, formulaire, graphique...).



2. Écriture des événements

2.1 Événements de classeur, de feuille ou de formulaire

Vous pouvez accéder aux procédures événementielles associées à un objet de la façon suivante :

- ▣ Dans la fenêtre **Explorateur de projet**, double cliquez sur l'objet souhaité (classeur, feuille, ou formulaire) afin de faire apparaître la fenêtre de code correspondante.
- ▣ Ouvrez la liste déroulante de gauche de la fenêtre de code et sélectionnez **Workbook**, **Worksheet** ou **UserForm** en fonction de l'objet sélectionné.
- ▣ Vous pouvez alors sélectionner l'un des événements liés à l'objet sélectionné dans la liste déroulante de droite afin de lui associer un code personnalisé.

■ Remarque

*Vous pouvez à tout moment désactiver l'exécution des procédures événementielles en affectant **False** à la propriété **EnableEvents** de l'objet **Application**.*

2.2 Événements de l'objet Application

Trois étapes sont nécessaires à l'écriture et à l'exécution des événements de l'objet **Application**.

Étape 1

▣ Insérez un module de classe :

Insertion - Module de classe

ou ouvrez la liste  et cliquez sur **Module de classe**.

▣ Une fois le module inséré, nommez-le.

Exemple

Nommez le module de classe ObjApplication.

Étape 2

▣ Dans le module de classe, créez un objet **Application** par le code suivant :

```
Public WithEvents NomObjet As Application
```

Exemple

Création de l'objet appelé **MonAppli** en tant qu'application.

```
Public WithEvents MonAppli As Application
```

L'objet ainsi créé devient disponible dans la liste de gauche du module.

▣ Sélectionnez l'objet ainsi créé dans la liste de gauche du module puis sélectionnez l'événement attendu dans la liste de droite. Écrivez le code des procédures à générer.

Exemple

Création de deux procédures événementielles : la première concerne l'insertion d'une nouvelle feuille, la seconde la création d'un nouveau classeur.

```
Public WithEvents MonAppli As Excel.Application
```

```
Private Sub MonAppli_WorkbookNewSheet _  
    (ByVal Wb As Workbook, ByVal Sh As Object)  
    Dim NomFeuille As String  
    ' A chaque ajout de feuille, on demande à l'utilisateur  
    ' de saisir un nom qui sera ensuite affecté à la feuille insérée  
    ' après les feuilles existantes  
    NomFeuille = InputBox("Entrez le nom de la feuille")  
    ActiveSheet.Name = NomFeuille  
    ActiveSheet.Move After:=Sheets(Sheets.Count)  
End Sub
```

```
Private Sub MonAppli_NewWorkbook(ByVal Wb As Workbook)  
    Dim NbFeuilles As Integer  
    Dim NbActuel As Integer  
    Dim Différence As Integer  
    ' Pour chaque nouveau classeur,  
    ' on demande à l'utilisateur le nombre de feuilles  
    ' Suivant les cas, on ajoute ou on supprime des feuilles  
    Do  
        NbFeuilles = Application.InputBox _  
            ("Nombre de feuilles ?", Type:=1)  
        Loop While NbFeuilles = False  
        NbActuel = Sheets.Count  
        Différence = NbActuel - NbFeuilles  
        ' Suppression des feuilles en trop  
        ' Suppression des messages d'alerte afin de ne pas  
        ' avoir de message lors de la suppression de feuilles  
        Do While Différence > 0  
            Application.DisplayAlerts = False  
            Sheets.Item(Différence).Select  
            ActiveWindow.SelectedSheets.Delete  
            Différence = Différence - 1  
        Loop  
  
        ' Ajout de feuilles si nécessaire  
        ' Les événements sont désactivés afin de ne pas avoir  
        ' à saisir le nom des nouvelles feuilles  
        Do While Différence < 0  
            Application.EnableEvents = False
```

```
        Sheets.Add
        Différence = Différence + 1
    Loop
    ' Réactivation des événements et alertes
    Application.EnableEvents = True
    Application.DisplayAlerts = True
End Sub
```

Étape 3

- Activez un module quelconque et connectez l'objet déclaré dans le module de classe avec l'objet **Application** par les instructions suivantes :

```
Dim NomVariable As New NomModuleDeClasse
```

```
Sub NomProcédure ()
Set NomVariable.NomObjet = Application
End Sub
```

Exemple

Ajoutez le code suivant dans le module Déclarations.

```
Option Explicit
Dim app As New ObjApplication

Sub InitializeMonAppli()
    Set app.MonAppli = Application
End Sub
```

Enfin appelez la procédure InitializeMonAppli lors de l'ouverture du classeur (module de classe ThisWorkbook).

```
Private Sub Workbook_Open()
    InitializeMonAppli
End Sub
```

Lorsque ce classeur sera ouvert, les procédures événementielles créées au cours de l'étape 2 s'exécuteront automatiquement lors de l'ajout de classeurs ou de feuilles. Ces procédures seront désactivées à la fermeture du classeur.