

Beste de savoir

Les énumérations en VBA

samedi 06 juillet 2024

Table des matières

	Introduction	1
1.	Définition	1
1.1.	Déclaration d'une énumération	2
1.2.	Affectation de valeurs aux variables d'énumération	4
1.3.	Cas des membres d'énumération masqués	5
2.	Utilisation	6
2.1.	Accès aux variables d'énumération	6
2.2.	Utilisation du type énuméré	7
2.3.	Itération	7
2.4.	Exemple final	8
	Conclusion	10

Introduction

En programmation, les [énumérations](#)  sont un moyen de regrouper des constantes ayant un sens commun.

Ces constantes, alors appelées variables d'énumération, doivent être entières en [VBA](#). Si vous avez déjà programmé via ce langage, vous en avez sans doute déjà utilisé. Par exemple, `xlCalculationAutomatic`, `xlCalculationManual` et `xlCalculationSemiautomatic`, correspondant chacune à une option de calcul, sont membres de l'énumération `xlCalculation` d'Excel.

Rassembler ainsi des valeurs simplifie aussi bien la conception que le développement.

Au cours de ce billet, nous allons voir comment fonctionnent les types énumérés en [VBA](#).

C'est parti !



Pour ce billet, nous nous placerons dans l'environnement Microsoft Office.

1. Définition

En introduction, nous avons dit que les énumérations permettent de regrouper des constantes entières en VBA.

Par exemple, nous pourrions vouloir gérer le statut de publication d'un contenu tel que sur Zeste de Savoir.

1. Définition

Sans énumération, cela donnerait :

```
1 Public Const STATUT_CONTENU_BROUILLON As Integer = 0
2 Public Const STATUT_CONTENU_BETA As Integer = 1
3 Public Const STATUT_CONTENU_VALIDATION As Integer = 2
4 Public Const STATUT_CONTENU_PUBLIE As Integer = 3
```

?

Et avec énumération ?

1.1. Déclaration d'une énumération

Voyons cela.

1.1.1. Syntaxe

Pour définir un type énuméré, il faut utiliser l'instruction `Enum` en indiquant un niveau de visibilité (`Public` par défaut), un nom et des constantes qui seront alors de type `Long`.

En reprenant le même exemple avec énumération, cela donnerait :

```
1 Public Enum EStatutContenu
2     eBrouillon
3     eBeta
4     eValidation
5     ePublie
6 End Enum
```

Cette déclaration crée à la fois un type énuméré `EStatutContenu` mais aussi des constantes `eBrouillon`, `eBeta`, `eValidation` et `ePublie`.

1.1.2. Emplacement

Il faut déclarer ses types énumérés en tête de fichier au risque de rencontrer des erreurs de compilation dans le cas contraire.

1. Définition

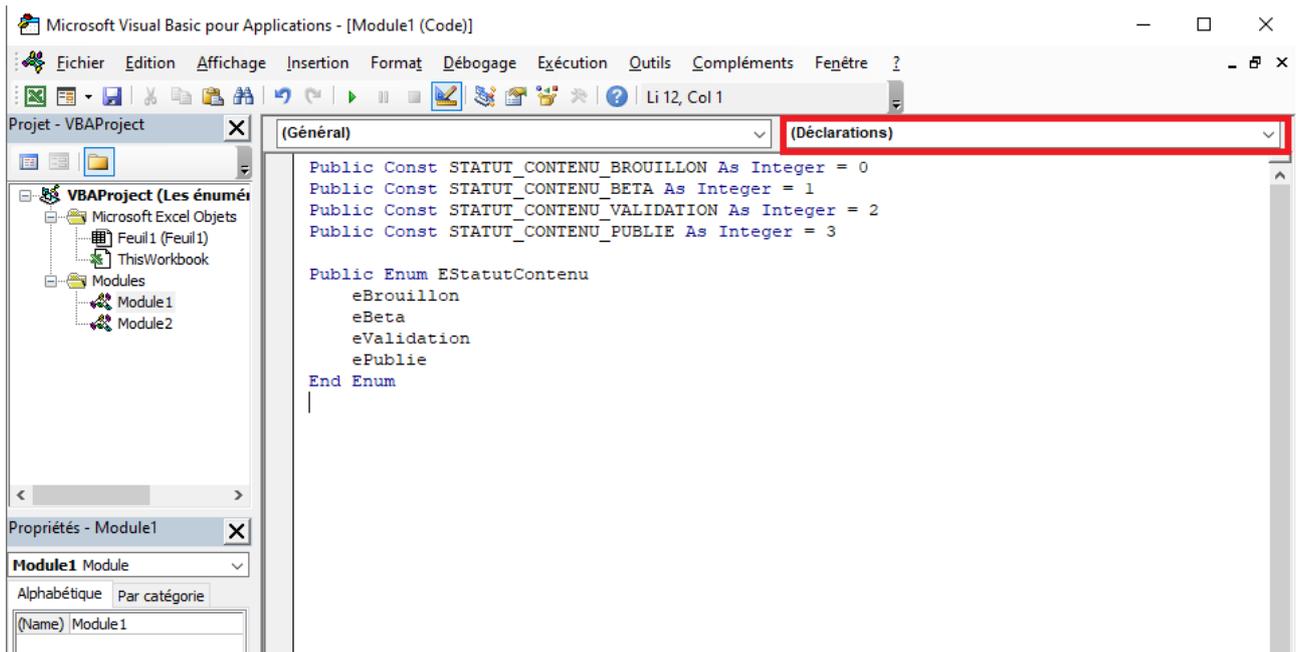


FIGURE 1.1. – Zone "Déclarations"

Il est possible de déclarer des énumérations dans n'importe quel type de fichier, y compris les classes, mais les énumérations déclarées dans ces dernières ne sont pas considérées comme membres de la classe.

Nous pouvons constater que les énumérations sont listées à part à travers l'explorateur d'objets (raccourci **F2**) pour le projet en cours :

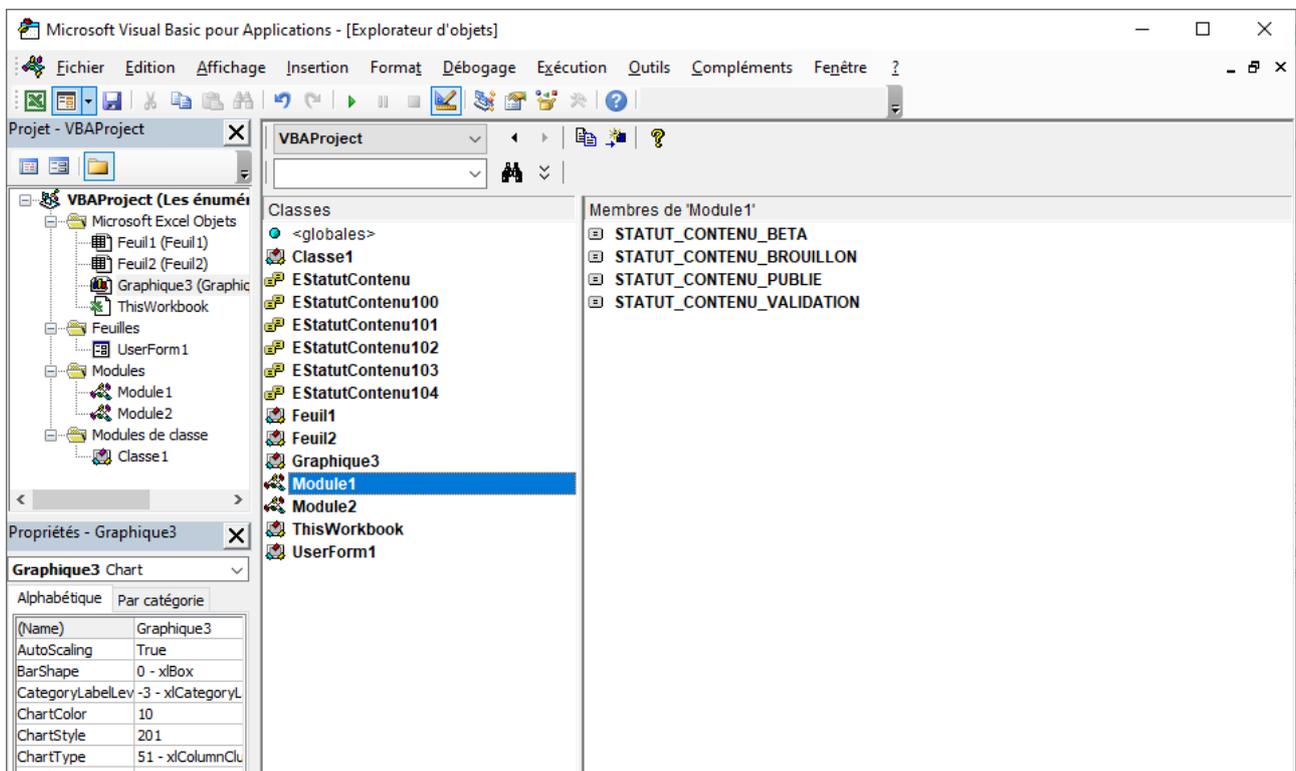


FIGURE 1.2. – Explorateur d'objets et types énumérés

1. Définition

1.1.3. Nommage

Concernant le nommage, une solution répandue est de préfixer le nom de l'énumération par `E` et les membres de celle-ci par `e`. Cela peut être un identifiant de bibliothèque aussi.

Quant aux variables d'énumération, certaines reprennent une partie du nom de l'énumération dans le leur (`XlCalcul` vu en introduction par exemple), d'autres non (`XlDisplayUnit` par exemple).

Vous pouvez regarder des exemples à travers l'explorateur d'objets.



L'important c'est d'avoir des noms parlants et d'éviter les conflits de nommage.

1.2. Affectation de valeurs aux variables d'énumération

Voyons maintenant comment gérer les valeurs des membres.

1.2.1. Valeurs par défaut

Par défaut, des valeurs incrémentées à partir de zéro sont affectées aux variables d'énumération :

```
1 Public Enum EStatutContenu
2     eBrouillon ' 0
3     eBeta ' 1
4     eValidation ' 2
5     ePublie ' 3
6 End Enum
```

1.2.2. Affectation partielle ou intégrale

Bien sûr, nous pouvons choisir partiellement ou intégralement les valeurs. Les valeurs non décidées sont déduites des précédentes via l'incrémentation encore une fois.

```
1 Public Enum EStatutContenuAffectationPartielle
2     eStatutContenuAffectationPartielleBrouillon = -2 ' -2
3     eStatutContenuAffectationPartielleBeta ' -1
4     eStatutContenuAffectationPartielleValidation = 5 ' 5
5     eStatutContenuAffectationPartiellePublie ' 6
6 End Enum
7
8 Public Enum EStatutContenuAffectationComplete
```

1. Définition

```
9     eStatutContenuAffectationCompleteBrouillon = -20 ' -20
10    eStatutContenuAffectationCompleteBeta = -10 ' -10
11    eStatutContenuAffectationCompleteValidation = 0 ' 0
12    eStatutContenuAffectationCompletePublie = 10.7 ' 11
13 End Enum
```



Remarquez que nous pouvons utiliser des valeurs négatives et que les nombres flottants sont automatiquement arrondis.

1.2.3. Utilisation de constantes

Il est aussi possible d'utiliser des constantes :

```
1 Public Enum EStatutContenuAvecConstantes
2     eStatutContenuAvecConstantesBrouillon =
3         STATUT_CONTENU_BROUILLON ' 0
4     eStatutContenuAvecConstantesBeta = STATUT_CONTENU_BETA ' 1
5     eStatutContenuAvecConstantesValidation =
6         STATUT_CONTENU_VALIDATION ' 2
7     eStatutContenuAvecConstantesPublie = STATUT_CONTENU_PUBLIE ' 3
8 End Enum
```

1.3. Cas des membres d'énumération masqués

Comme nous le verrons par la suite, il est faisable d'itérer sur des valeurs d'une énumération. Mais les noms peuvent changer. Et puis, des constantes peuvent être ajoutées ou supprimées en cours de route, impactant notre programme. Ce n'est pas pratique.

C'est un intérêt des membres masqués.

Ils s'écrivent entre crochets et sont préfixés d'un `[_]`, comme ceci :

```
1 Public Enum EIndexColonneContenu
2     [_Debut] ' 0
3     eIndexColonneTitre ' 1
4     eIndexColonneDescription ' 2
5     eIndexColonneStatut ' 3
6     [_Fin] ' 4
7 End Enum
```

Ils sont alors masqués dans l'explorateur d'objets et la complétion de code :

2. Utilisation

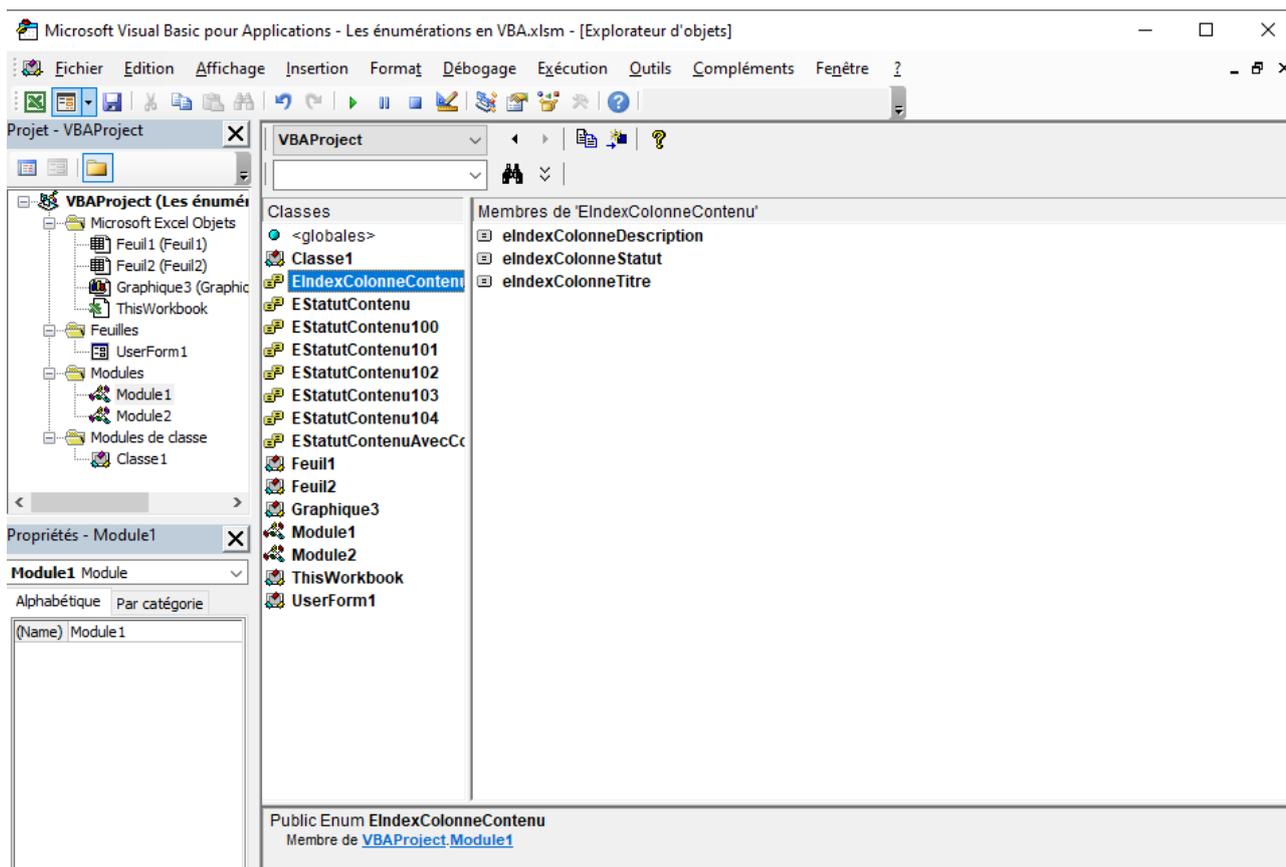


FIGURE 1.3. – Membres non affichés dans l'explorateur d'objets

Pendant cette première section, nous avons vu comment définir des énumérations en VBA.

2. Utilisation

Maintenant que nous avons défini une énumération, nous allons pouvoir voir les façons de l'utiliser.

D'une part, le type énuméré regroupe les différentes constantes. D'autre part, comme son nom l'indique, c'est un type dont nous pouvons nous servir pour des variables, des paramètres ou encore des valeurs de retour.

2.1. Accès aux variables d'énumération

Nous accédons aux constantes via la notation `NomEnumeration.NomVariableEnumeration` ou `NomVariableEnumeration` tout simplement.

```
1 Debug.Print (EStatutContenu.ePublie) ' 3
2 Debug.Print (ePublie) ' 3
```

2. Utilisation

Un avantage de la première approche est l'aide au développement (IntelliSense) quand on ne connaît pas forcément les noms :

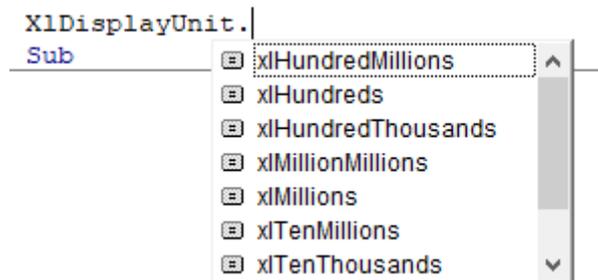


FIGURE 2.4. – Aide au développement (IntelliSense)

2.2. Utilisation du type énuméré

Nous pouvons utiliser notre nouveau type pour déclarer des variables ainsi que définir le type de paramètres ou retours :

```
1 Dim eStatut As EStatutContenu
2 eStatut = ePublie
3
4 Private Function StatutToString(ByVal eStatut As EStatutContenu)
5     As String
6     ' ...
7 End Function
8 Private Function ChoixStatut() As EStatutContenu
9     ' ...
10 End Function
```

2.3. Itération

L'itération sur les valeurs est rendue possible par la boucle `For` avec les bornes voulues :

```
1 Private Sub Exemple_Iteration()
2     Dim eIndexColonne As EIndexColonneContenu
3     For eIndexColonne = EIndexColonneContenu.[_Debut] + 1 To
4         EIndexColonneContenu.[_Fin] - 1
5         Debug.Print (eIndexColonne)
6         ' 1
7         ' 2
8         ' 3
9     Next eIndexColonne
```

2. Utilisation

```
9 End Sub
```



Cela ne fonctionne pas s'il y a discontinuité entre les valeurs comme dans notre exemple EStatutContenuAffectationPartielle !

2.4. Exemple final

Voici un exemple dans lequel l'utilisateur choisit un code de statut de contenu, ce code de statut est ensuite transposé en texte et les informations sont insérées dans une [liste de données](#) :

```
1 Public Enum EStatutContenu
2     eBrouillon ' 0
3     eBeta ' 1
4     eValidation ' 2
5     ePublie ' 3
6 End Enum
7
8 Private Function StatutsPossibles() As String
9     Dim sResult As String
10
11     Dim eStatut As EStatutContenu
12     For eStatut = eBrouillon To ePublie
13         sResult = sResult & CStr(eStatut) & " - " &
14             StatutToString(eStatut) & vbNewLine
15     Next eStatut
16
17     StatutsPossibles = sResult
18 End Function
19
20 Private Function StatutToString(ByVal eStatut As EStatutContenu)
21     As String
22     Select Case eStatut
23     Case 0
24         StatutToString = "Brouillon"
25     Case 1
26         StatutToString = "Bêta"
27     Case 2
28         StatutToString = "Validation"
29     Case 3
30         StatutToString = "Publié"
31     Case Else
32         StatutToString = ""
33     End Select
34 End Function
```

2. Utilisation

```
34 Private Function ChoixStatut() As EStatutContenu
35     On Error GoTo GestionErreur
36     Dim sMsg As String
37     sMsg = _
        "Merci de choisir une valeur de statut parmi la liste suivante :"_
        & vbNewLine &
        StatutsPossibles
38
39 SaisieStatut:
40     ChoixStatut = CLng(InputBox(sMsg))
41     If (ChoixStatut < eBrouillon Or ChoixStatut > ePublie) Then
42         Err.Raise vbObjectError + 1000
43     End If
44     Exit Function
45
46 GestionErreur:
47     sMsg = "/!\ " & sMsg
48     Resume SaisieStatut
49 End Function
50
51 Private Sub AjouteLigneContenu(ByVal vContenuLigne As Variant)
52     Dim loContenus As ListObject
53     Set loContenus = Sheets(1).ListObjects("TableauContenus")
54
55     Dim lrLigne As ListRow
56     Set lrLigne = loContenus.ListRows.Add
57     lrLigne.Range = vContenuLigne
58 End Sub
59
60 Private Sub Main()
61     Dim eStatut As EStatutContenu
62     eStatut = ChoixStatut()
63
64     Dim vContenuLigne As Variant
65     vContenuLigne = Array("Un super tuto",
        "pour apprendre des supers trucs", StatutToString(eStatut))
66
67     AjouteLigneContenu (vContenuLigne)
68 End Sub
```

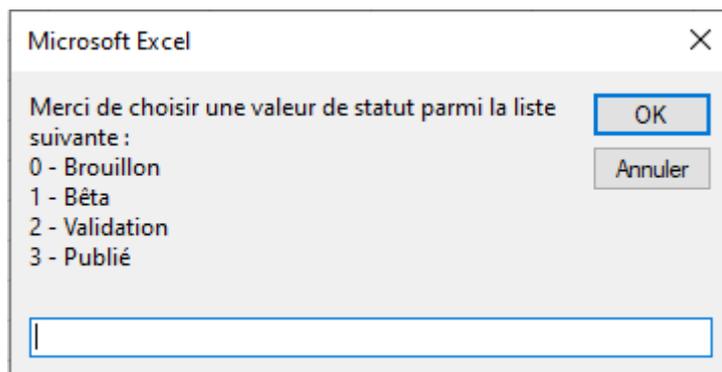


FIGURE 2.5. – Exécution du programme avec fenêtre saisie code statut

	A	B	C	D	E	F
1	Titre	Description	Statut			
2	Un super tuto	pour apprendre des supers trucs	Bêta			
3	Un super tuto	pour apprendre des supers trucs	Brouillon			
4	Un super tuto	pour apprendre des supers trucs	Validation			
5	Un super tuto	pour apprendre des supers trucs	Publié			
6						
7						
8						
9						
10						

FIGURE 2.6. – Résultat après plusieurs exécution de Main pour les différents statuts

Au fil de cette seconde section, nous avons vu comment nous servir des énumérations en VBA.

Conclusion

C'est déjà la fin de ce billet.

Au cours de celui-ci, nous avons vu comment définir et utiliser des énumérations en VBA.

D'autres langages tels que Python offrent davantage de possibilités, comme pouvoir créer des énumérations de chaîne de caractères ou encore accéder aux noms des constantes.

À bientôt !

Quelques ressources :

Conclusion

- [La documentation](#) ↗
- [Cet article](#) ↗

Liste des abréviations

VBA Visual Basic for Applications. 1