

Visual Basic pour Applications

Les concepts de base de la programmation, 1ère partie

Romain Tavenard
Kévin Huguenin
Christophe Avenel

Romain.Tavenard@irisa.fr

Kevin.Huguenin@gmail.com

Christophe.Avenel@irisa.fr

11 Septembre 2008

- 1 Plan général du cours
- 2 La programmation
 - Généralités
 - Concepts de base
 - Bonnes pratiques
- 3 Le cas particulier de Visual Basic (VB)
 - Intérêt de VB
 - Application des concepts de base en VB
 - Les boîtes de dialogue et l'interface utilisateur

Plan général du cours (6 × 2h de cours magistral)

- 1 Les concepts de base de la programmation (2 séances)
- 2 La notion d'objet
- 3 Les objets spécifiques de la suite Microsoft Office (2 séances)
- 4 Les interfaces graphiques avec VB

Plan

- 1 Plan général du cours
- 2 La programmation
 - Généralités
 - Concepts de base
 - Bonnes pratiques
- 3 Le cas particulier de Visual Basic (VB)
 - Intérêt de VB
 - Application des concepts de base en VB
 - Les boîtes de dialogue et l'interface utilisateur

Pourquoi programmer ?

Pour **automatiser** une tâche.

Exemples (issus de Visual Basic pour Applications) :

- Microsoft Word : mettre en italique toutes les locutions latines d'un texte ;
- Microsoft Excel : remplir toutes les cases vides d'un tableau ;
- Microsoft Access : insérer des champs à partir de valeurs d'un tableau Excel.

Que signifie "programmer" ?

Cela consiste à :

- 1 **comprendre** les actions à exécuter et les décisions à prendre ;
- 2 **décomposer** une action en instructions élémentaires ;
- 3 **traduire** ces instructions élémentaires en langage compréhensible par une machine.

Que signifie " programmer " ? (*Exemple*)

- Exemple de programme : tri d'un tableau de nombres :
`monTableau = [31 6 4 2 78 3 5 0] ;`
- Comprendre les actions à exécuter
 - parmi ceux qu'on n'a pas encore traités :
 - chercher le plus petit
 - le mettre à la fin des éléments traités
 - recommencer jusqu'à ce qu'on ait tout traité

Que signifie "programmer" ? (*Exemple*)

- Exemple de programme : tri d'un tableau de nombres ;
- Décomposer ces actions en instructions élémentaires

```
pour i allant de 1 au nombre d'éléments du tableau
    valeurMin = tableau[i]
    indexMin = i
    pour j allant de i+1 au nombre d'éléments de la liste
        si tableau[j] < valeurMin
            valeurMin = tableau[j]
            indexMin = j
        fin si
    fin pour
    tableau[indexMin] = tableau[i]
    tableau[i] = valeurMin
fin pour
```


Que signifie "programmer" ? (*Exemple*)

- Exemple de programme : tri d'un tableau de nombres ;
- Traduction en langage compréhensible par une machine *via* VB

```
For i = 1 to N-1
    valeurMin = tableau(i)
    indexMin = i
    For j = i+1 to N
        If tableau(j) < valeurMin Then
            valeurMin = tableau(j)
            indexMin = j
        Endif
    Next j
    tableau(indexMin) = tableau(i)
    tableau(i) = valeurMin
Next i
```

Plan

1 Plan général du cours

2 La programmation

- Généralités
- Concepts de base
- Bonnes pratiques

3 Le cas particulier de Visual Basic (VB)

- Intérêt de VB
- Application des concepts de base en VB
- Les boîtes de dialogue et l'interface utilisateur

Variables

Une variable sert à stocker et manipuler des données dans un programme. Elle est définie par :

- un nom ;
- un type de donnée qu'elle contiendra.

L'instruction de définition d'une variable est appelée **déclaration**. Par exemple, en VB, si on veut déclarer que `maVariable` contiendra un nombre entier, ça donne :

```
Dim maVariable as Integer
```

Il est possible de modifier la valeur d'une variable au cours de l'exécution du programme.

Constantes

Une constante possède les mêmes caractéristiques qu'une variable sauf que, par définition, **on ne peut pas la modifier** au cours de l'exécution du programme.

Types primitifs

Différents types primitifs :

- type **booléen** : VRAI/FAUX ;
- type **entier** : nombre entier ;
- type **flottant** : nombre à virgule ;
- type **chaîne de caractères** : morceau de texte.

Affectations

Une affectation est l'instruction qui permet de donner une valeur à une variable. On utilise, en VB, le signe = pour cela :

```
maVariable = 12
```

La valeur d'une constante est affectée lors de sa déclaration ! Le code suivant produirait donc une erreur :

```
maConstante = 12
```

Opérations

Pour programmer, il faut pouvoir manipuler les variables dont on dispose. Cela se fait à partir d'opérations élémentaires qui sont :

- pour les variables numériques :
 - +, -, ×, /;
- pour les variables de type chaîne de caractères :
 - la concaténation : "mettre bout-à-bout des morceaux de texte"
⇒ *par exemple, la concaténation de "bon" et "jour" donne "bonjour"* ;
- pour les variables booléennes :
 - ET, OU, NON ;

Plan

1 Plan général du cours

2 La programmation

- Généralités
- Concepts de base
- Bonnes pratiques

3 Le cas particulier de Visual Basic (VB)

- Intérêt de VB
- Application des concepts de base en VB
- Les boîtes de dialogue et l'interface utilisateur

Pourquoi des bonnes pratiques ?

Pour avoir un code facile à relire et à modifier...

- ...par soi-même ;
- ...et par les autres.

Quelles bonnes pratiques ?

Voici une première liste (non-exhaustive) de bonnes pratiques :

- utiliser des noms de variables explicites ;
- utiliser, lorsque c'est adapté, des constantes ;
- commenter son code.

Les noms de variables

Les noms de variables utilisés doivent être explicites. Le nom d'une variable doit représenter :

- son type ;
- son utilisation.

Par exemple, on doit à tout prix éviter des noms du type : `variable1` et leur préférer des noms tels que `intAge`.

L'utilisation de constantes

L'intérêt de l'utilisation de constantes réside en deux points :

- modification du code simplifiée, ce qui permet d'éviter :

```
Dim intRayon as Integer
Dim doubleAire as Double
Dim doublePerimetre as Double
intRayon = 5
doubleAire = 3.14 * intRayon * intRayon
doublePerimetre = 3.14 * 2 * intRayon
```

- utilisation des erreurs générées, ce qui permet d'éviter :

```
Dim doublePi as Double
doublePi = 3.1
[...]
doublePi = 3.14
```

Les commentaires

Les commentaires sont des lignes de code qui ne seront pas interprétées par la machine. Ils servent donc de "pense-bête" aux programmeurs. En VB, les commentaires sont toujours précédés du symbole '. Par exemple :

```
Dim intRayon as Integer
Dim doubleAire as Double
Dim doublePerimetre as Double
Const constDoublePi as Double
intRayon = 5 'le rayon du cercle vaut 5
constDoublePi = 3.14 'on conserve 2 chiffres après la virgule
'calcul de l'aire du disque
doubleAire = constDoublePi * intRayon * intRayon
doublePerimetre = constDoublePi * 2 * intRayon
```

Plan

- 1 Plan général du cours
- 2 La programmation
 - Généralités
 - Concepts de base
 - Bonnes pratiques
- 3 Le cas particulier de Visual Basic (VB)
 - Intérêt de VB
 - Application des concepts de base en VB
 - Les boîtes de dialogue et l'interface utilisateur

Intérêt de VB

L'intérêt d'utiliser VB pour programmer réside dans les deux points suivants :

- facilité de la programmation de fenêtres et boîtes de dialogue ;
- Visual Basic pour Applications (VBA) : extension de VB spécifique à la suite Microsoft Office.

Plan

- 1 Plan général du cours
- 2 La programmation
 - Généralités
 - Concepts de base
 - Bonnes pratiques
- 3 Le cas particulier de Visual Basic (VB)
 - Intérêt de VB
 - **Application des concepts de base en VB**
 - Les boîtes de dialogue et l'interface utilisateur

Types de données

Implémentation des types primitifs :

- type **booléen** (VRAI/FAUX) : Boolean;
- type **entier** (nombre entier) : Byte, **Integer**, Long;
- type **flottant** (nombre à virgule) : Single, **Double**;
- type **chaîne de caractères** (morceau de texte) : String;
- autres types : Date, Currency.

Déclaration de variables et de constantes

La déclaration d'une variable en VB se fait de la manière suivante :

```
Dim nomDeLaVariable as TypeDeLaVariable
```

Par exemple, pour déclarer une variable `intRayon` de type entier :

```
Dim intRayon as Integer
```

La déclaration d'une constante en VB se fait de la manière suivante :

```
Const nomDeLaConstante as TypeDeLaConstante = valeurDeLaConstante
```

Remarquez que l'affectation se fait à la déclaration ! Par exemple, pour déclarer une constante `constDoublePi` de type nombre à virgule qui vaille 3.14 :

```
Const constDoublePi as Double = 3.14
```

Affectation de variables et de constantes

L'affectation d'une valeur à une variable en VB se fait de la manière suivante :

```
nomDeLaVariable = valeurAffectee
```

Par exemple, pour déclarer une variable `strNomDeFamille` de type chaîne de caractères et lui attribuer la valeur "Tavenard", en VB, on a :

```
Dim strNomDeFamille as String  
strNomDeFamille = "Tavenard"
```

Opérations sur les variables

- Variables numériques (nombres entiers et nombres à virgule) :

⇒ +, -, *, /;

```
doubleAireDisque = constDoublePi * intRayon * intRayon
```

- Variables de type chaînes de caractères :

⇒ &;

```
strNomComplet = strPrenom & " " & strNomDeFamille
```

- Variables booléennes :

⇒ And, Or et Not.

```
boolIlFaitBeau = boolLeSoleilBrille And Not(boolIlPleut)
```

Plan

- 1 Plan général du cours
- 2 La programmation
 - Généralités
 - Concepts de base
 - Bonnes pratiques
- 3 Le cas particulier de Visual Basic (VB)
 - Intérêt de VB
 - Application des concepts de base en VB
 - Les boîtes de dialogue et l'interface utilisateur

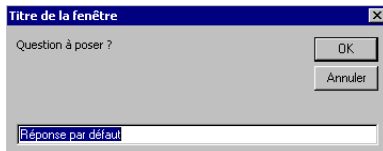
Affichage d'une boîte de dialogue simple

```
intRetour = MsgBox("Texte à afficher", _  
    vbOKOnly,"Titre de la fenêtre")
```



Affichage d'une boîte de dialogue avec zone de saisie

```
strRetour = InputBox("Question à poser ? ", _  
    "Titre de la fenêtre", "Réponse par défaut")
```



Où télécharger ce cours ?

À l'adresse suivante :

`http://www.irisa.fr/texmex/people/tavenard/htbin/teaching.php`