

Visual Basic pour Applications

Les concepts de base de la programmation, 2ème partie

Romain Tavenard
Kévin Huguenin
Christophe Avenel

Romain.Tavenard@irisa.fr

Kevin.Huguenin@gmail.com

Christophe.Avenel@irisa.fr

18 Septembre 2008

MCours.com

Plan

- 1 Quelques rappels
 - Que signifie "programmer" ?
 - Variables / constantes
 - Tableaux
 - Tests conditionnels
- 2 La programmation
 - Tests conditionnels
 - Boucles
 - Récursivité
 - Bonnes pratiques
- 3 Transcription en Visual Basic (VB)
 - Tests conditionnels
 - Boucles
 - Récursivité

Plan

- 1 Quelques rappels
 - Que signifie "programmer" ?
 - Variables / constantes
 - Tableaux
 - Tests conditionnels
- 2 La programmation
 - Tests conditionnels
 - Boucles
 - Récursivité
 - Bonnes pratiques
- 3 Transcription en Visual Basic (VB)
 - Tests conditionnels
 - Boucles
 - Récursivité

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.

Plan

- 1 Quelques rappels
 - Que signifie "programmer" ?
 - Variables / constantes
 - Tableaux
 - Tests conditionnels
- 2 La programmation
 - Tests conditionnels
 - Boucles
 - Récursivité
 - Bonnes pratiques
- 3 Transcription en Visual Basic (VB)
 - Tests conditionnels
 - Boucles
 - Récursivité

Variables / Constantes

- Déclaration
- Types de données
- Affectation
- Opérations

Plan

- 1 Quelques rappels
 - Que signifie " programmer " ?
 - Variables / constantes
 - **Tableaux**
 - Tests conditionnels
- 2 La programmation
 - Tests conditionnels
 - Boucles
 - Récursivité
 - Bonnes pratiques
- 3 Transcription en Visual Basic (VB)
 - Tests conditionnels
 - Boucles
 - Récursivité

Tableaux

- Tableaux de taille constante (tableaux statiques)

```
Dim intTableau (10) as Integer  
intTableau(10) = 5
```

- Tableaux de taille variable (tableaux dynamiques)

```
Dim intTableau () as Integer  
Redim intTableau (10)  
Redim Preserve intTableau (10)
```

- Attention aux indices en VBA :
 - à la déclaration ;
 - à l'utilisation.

Plan

- 1 Quelques rappels
 - Que signifie " programmer " ?
 - Variables / constantes
 - Tableaux
 - Tests conditionnels
- 2 La programmation
 - Tests conditionnels
 - Boucles
 - Récursivité
 - Bonnes pratiques
- 3 Transcription en Visual Basic (VB)
 - Tests conditionnels
 - Boucles
 - Récursivité

Test "Si...Alors"

- En pseudo-code :

```
si condition alors
  instruction1
  [...]
  instructionN
fin si
```

- En VB :

```
If condition Then
  instruction1
  [...]
  instructionN
End If
```

Test "Si...Alors...Sinon" # 1

- En pseudo-code :

```
si condition alors
  instruction1
  [...]
  instructionN
sinon
  instructionAlternative1
  [...]
  instructionAlternativeN
fin si
```

MCours.com

Test "Si...Alors...Sinon" # 2

- En VB :

```
If condition Then
    instruction1
    [...]
    instructionN
Else
    instructionAlternative1
    [...]
    instructionAlternativeN
End If
```

Plan

- 1 Quelques rappels
 - Que signifie "programmer" ?
 - Variables / constantes
 - Tableaux
 - Tests conditionnels
- 2 La programmation
 - Tests conditionnels
 - Boucles
 - Récursivité
 - Bonnes pratiques
- 3 Transcription en Visual Basic (VB)
 - Tests conditionnels
 - Boucles
 - Récursivité

Combinaison de conditions

Une condition, telle que `(intAge > 20)` est un booléen !

On peut alors combiner les conditions à l'aide des opérateurs `Ou`, `Et` ou `Non`.

Exemples de combinaison de conditions

- Tester si la variable `intAge` est comprise entre les valeurs 12 et 25 incluses :

```
si ( ( intAge >= 12 ) et ( intAge <= 25 ) ) alors  
    [...]  
fin si
```

- Tester si la variable `intAge` est comprise en dehors de l'intervalle [12,25] :

```
si ( ( intAge < 12 ) ou ( intAge > 25 ) ) alors  
    [...]  
fin si
```

Utilisation de "cas"

Utile si l'on veut tester toutes les valeurs possibles d'une variable.

```
selection nomDeLaVariable
  cas valeur1:
    listeInstructions1
  fin cas
  cas valeur2:
    listeInstructions2
  fin cas
  [...]
  default:
    listeInstructionsParDefaut
fin selection
```


Exemple de cas

Tester le mois stocké dans la variable `intMois` :

```
selection intMois
  cas 1:
    strSaison = "Hiver"
  fin cas
[...]
```

```
cas 12:
  strSaison = "Automne-Hiver"
fin cas
```

```
default:
  strSaison = ""
```

```
fin selection
```

Plan

- 1 Quelques rappels
 - Que signifie "programmer" ?
 - Variables / constantes
 - Tableaux
 - Tests conditionnels
- 2 La programmation
 - Tests conditionnels
 - **Boucles**
 - Récursivité
 - Bonnes pratiques
- 3 Transcription en Visual Basic (VB)
 - Tests conditionnels
 - Boucles
 - Récursivité

Principe des boucles

But de la programmation : automatisation de tâches.

Exemple : remplir les cases vides d'un tableau Excel.

⇒ On souhaite appliquer une même suite d'instructions à chaque case du tableau.

Boucle "tant que"

Une boucle "tant que" (boucle `while`) est définie par :

- une condition d'arrêt qui sera testée avant chaque itération.

Exemple de boucle "tant que"

```
capital = 1000
tant que capital < 2000
  capital = capital * 1.25
fin tant que
```

Boucle "pour"

Une boucle "pour" (boucle `for`) est définie par :

- un indice de boucle ;
- une valeur initiale pour cet indice ;
- une valeur finale pour cet indice ;
- un pas.

Exemple de boucle "pour"

```
pour i parcourant les indices du tableau t
  si t(i) = "" alors
    t(i) = 0
  fin si
fin pour
```

Plan

- 1 Quelques rappels
 - Que signifie "programmer" ?
 - Variables / constantes
 - Tableaux
 - Tests conditionnels
- 2 La programmation
 - Tests conditionnels
 - Boucles
 - **Récursivité**
 - Bonnes pratiques
- 3 Transcription en Visual Basic (VB)
 - Tests conditionnels
 - Boucles
 - Récursivité

Notion de procédure

Une procédure c'est :

- un ensemble d'instructions ;
- qui peut prendre une ou plusieurs valeur(s) en paramètre.

Exemple d'appel de procédure

```
procedure afficheSomme (Entier a, Entier b)
  afficher (a+b)
fin procedure
```

```
procedure Main ()
  afficheSomme(12, 14)
fin procedure
```

Notion de fonction

Une fonction c'est :

- un ensemble d'instructions ;
- qui peut prendre une ou plusieurs valeur(s) en paramètre ;
- et qui peut retourner une valeur.

Exemple d'appel de fonction

```
fonction calculeSomme (Entier a, Entier b)  
    retourner (a+b)  
fin fonction
```

```
procedure Main ()  
    somme = calculeSomme(12, 14)  
fin procedure
```

Principe de récursivité

Un appel de fonction est dit récursif si la fonction qu'on appelle est celle depuis laquelle l'appel est effectué.

⇒ intérêt : calcul de suites de la forme $u_{n+1} = f(u_n)$.

Exemple du principe de récursivité

Calcul de factorielle :

```
fonction factorielle (Entier a)
  si a = 1 alors
    retourner 1
  sinon
    retourner a * factorielle ( a - 1 )
  fin si
fin fonction
```

```
procedure Main ()
  afficher factorielle (5)
fin procedure
```

Plan

- 1 Quelques rappels
 - Que signifie "programmer" ?
 - Variables / constantes
 - Tableaux
 - Tests conditionnels
- 2 La programmation
 - Tests conditionnels
 - Boucles
 - Récursivité
 - **Bonnes pratiques**
- 3 Transcription en Visual Basic (VB)
 - Tests conditionnels
 - Boucles
 - Récursivité

Rappel des bonnes pratiques déjà énoncées

Voici la première liste de bonnes pratiques énoncée la semaine dernière :

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

Un peu plus de bonnes pratiques

Nous pouvons y ajouter deux nouvelles bonnes pratiques liées à ce que nous avons appris :

- indenter son code ;
- organiser ses fonctions/procédures.

MCours.com

Indentation du code

Indenter son code c'est le décaler d'un "cran" vers la droite à chaque :

- entrée dans une boucle ;
- entrée dans une fonction/procédure ;
- entrée dans un test conditionnel.

Exemple d'indentation du code

```
fonction nettoyerTableau (tableau t)
  pour i parcourant les indices du tableau t
    si t(i) == "" alors
      t(i) = 0
    fin si
  fin pour
  retourner t
fin fonction
```

Organisation au sein des fonctions et procédures

Pour mieux se repérer au sein des fonctions (respectivement procédures), il est préférable de définir les variables et constantes utilisées en début de fonction (respectivement procédure).

Plan

- 1 Quelques rappels
 - Que signifie "programmer" ?
 - Variables / constantes
 - Tableaux
 - Tests conditionnels
- 2 La programmation
 - Tests conditionnels
 - Boucles
 - Récursivité
 - Bonnes pratiques
- 3 Transcription en Visual Basic (VB)
 - Tests conditionnels
 - Boucles
 - Récursivité

If...Then...End If

```
If ( ( intAge >= 12 ) And ( intAge <= 25 ) ) Then  
    doubleReducSNCF = 0.5  
End If
```

If...Then...Else...End If

```
If ( ( intAge >= 12 ) And ( intAge <= 25 ) ) Then
    doubleReducSNCF = 0.5
Else
    doubleReducSNCF = 0
End If
```

If...Then...Else If...Else...End If

```
If ( ( intAge >= 12 ) And ( intAge <= 25 ) ) Then
    doubleReducSNCF = 0.5
Else If ( intAge >= 60 ) Then
    doubleReducSNCF = 0.25
Else
    doubleReducSNCF = 0
End If
```


Select Case

```
Select Case intMois
  Case 1:
    strSaison = "Hiver"
  [...]
  Case 12:
    strSaison = "Automne-Hiver"
  Case Else:
    strSaison = ""
End Select
```

Plan

- 1 Quelques rappels
 - Que signifie " programmer " ?
 - Variables / constantes
 - Tableaux
 - Tests conditionnels
- 2 La programmation
 - Tests conditionnels
 - Boucles
 - Récursivité
 - Bonnes pratiques
- 3 Transcription en Visual Basic (VB)
 - Tests conditionnels
 - **Boucles**
 - Récursivité

Boucle for

```
Dim t (10) As Integer
For i=LBound(t) to UBound(t)
    t(i) = t(i) * 2
Next
```

Boucle while

```
capital = 1000  
Do While capital < 2000  
    capital = capital * 1.25  
Loop
```

Plan

- 1 Quelques rappels
 - Que signifie "programmer" ?
 - Variables / constantes
 - Tableaux
 - Tests conditionnels
- 2 La programmation
 - Tests conditionnels
 - Boucles
 - Récursivité
 - Bonnes pratiques
- 3 Transcription en Visual Basic (VB)
 - Tests conditionnels
 - Boucles
 - Récursivité

Déclaration et appel de procédure

```
Sub afficheSomme (a As Integer, b As Integer)
    Call MsgBox ( Prompt:=a+b, Buttons:=vbOKOnly, _
        Title:="afficheSomme")
End Sub
```

```
Sub Main ()
    Call afficheSomme(a:=12, b:=14)
End Sub
```

```
Sub Main ()
    afficheSomme 12, 14
End Sub
```

Déclaration et appel de fonction

```
Function calculeSomme(a As Integer, b As Integer) _  
    As Integer  
    calculeSomme = a + b  
End Function  
  
Sub Main ()  
    x = calculeSomme(a:=12,b:=14)  
End Sub
```

Procédures / Fonctions – Précision

Deux façons possibles de passer un argument à une fonction / procédure :

- par valeur : ByVal ;
- par référence : ByRef (par défaut).

Appels récursifs

```
Function factorielle(a As Integer) As Integer
    If (a = 1) Then
        factorielle = 1
    Else
        factorielle = a * factorielle(a - 1)
    End If
End Function

Sub Main()
    x = factorielle(a:=3)
End Sub
```

Où télécharger ce cours ?

À l'adresse suivante :

<http://www.irisa.fr/texmex/people/tavenard/>
Rubrique "Enseignement"

MCours.com