

VB.NET

Introduction

Abdelali Saidi

saidi.a@ucd.ac.ma

Plan

- 1 Introduction
- 2 Structure d'un programme
- 3 Le langage VISUAL BASIC
- 4 Exercice

Plan

- 1 Introduction
- 2 Structure d'un programme
- 3 Le langage VISUAL BASIC
- 4 Exercice

Introduction

Présentation

Visual Basic .NET est un langage de programmation à la syntaxe similaire à celle de Visual Basic 6. Il permet de développer en .NET via Visual Studio, sur les systèmes d'exploitation Windows.

Caractéristiques

- Destiné à tout le monde
- Il fonctionne sous le framework .NET
 - Ce framework est un intermédiaire entre les programmes et le système d'exploitation
 - Il permet à plusieurs langages de programmation de collaborer entre eux

Plan

- 1 Introduction
- 2 Structure d'un programme**
- 3 Le langage VISUAL BASIC
- 4 Exercice

Structure d'un programme

Les objets

Un objet est une entité de programmation qui peut posséder plusieurs propriétés. Ces propriétés sont stockés dans des variables qui sont reliées à l'objet. On appelle ces variables des champs. Pour créer un objet, on fait appel à un ensemble qui décrit de façon abstraite ses objets appelé une classe.

Exemple

- Création (déclaration)
 - Dim mon_objet As New ma_classe
- Accès aux propriétés
 - mon_objet.propriété_1 = valeur_1
 - mon_objet.propriété_2 = valeur_2

Structure d'un programme

Les méthodes

Une méthode est une fonction liée à une classe donnée qui permet un traitement spécifique aux objets de cette classe. Il est possible qu'une méthode puisse être appelée avec des paramètres ou sans (on appelle ce concept la surcharge).

Exemple d'appel d'une méthode

- `mon_objet.maMéthode()`
- `mon_objet.maMéthode(para1)`
- `mon_objet.maMéthode(para1,para2)`

Structure d'un programme

Instructions

Une instruction est une ligne de code qui s'exécute lors du fonctionnement du programme. Elle permet d'effectuer une opération, une déclaration ou bien une définition. Le retour à la ligne détermine la fin d'une instruction.

Exemple

- Dim A As Integer 'une déclaration
- A=1 'une affectation
- Dim B As String : B="Bonjour" 'deux instructions sur une même ligne

NB : Tous ce qui se trouve derrière un apostrophe est un commentaire (Ce sont des lignes qui sont ignorées lors de la compilation)

Structure d'un programme

Procédures

Une procédure est un ensemble d'instructions regroupées pour un même objectif. En VB.NET, il en existe deux :

- 1 Les sub : ceux là ne retournent aucune valeur
- 2 Les fonction : ceux là si

Pour appeler un sub ou une fonction il suffit d'utiliser son nom

Structure d'un programme

Modules

Un module est un ensemble de sub et de fonctions qui commence par le mot clé *Module* et se termine par le mot clé *End Module*

NB : Dans un même module, les sub et les fonctions ne peuvent avoir le même nom que si et seulement s'ils ont des nombres de paramètres différents

Plan

- 1 Introduction
- 2 Structure d'un programme
- 3 Le langage VISUAL BASIC**
- 4 Exercice

Le langage VISUAL BASIC

Les variables

Définition

Une variable est une partie de la mémoire destinée à stocker des données. Elle s'appelle variable vu qu'elle est susceptible d'être modifiée au cours de l'exécution du programme.

Le langage VISUAL BASIC

Les variables

Le nom

Pour le choix du nom d'une variable, il est possible d'utiliser des lettres (minuscule et majuscule) et des chiffres. Les espaces et autres caractères spéciaux ne sont pas acceptés. Le nom d'une variable doit toujours commencer par une lettre.

Exemple

- Noms acceptés
 - MaVariable
 - COOL
 - Je_Suis_Une_Variable
- Noms refusés
 - 2_MaVariable
 - Ma Variable
 - différent

Le langage VISUAL BASIC

Les variables

Déclaration

- `Dim MaVariable1 As String`
- `Dim MaVariable2 As String = "texte"`
- `Dim nombre1 As Integer`
- `Dim nombre2 As Integer = 11`
- `Dim resultat As Integer = fonctionPourCalculerLaSomme()`
- `Dim Test As New Class`

Pour la dernière déclaration, il s'agit d'une instantiation d'un objet.

Le langage VISUAL BASIC

Les variables

Types de variables

Le type d'une variable permet de spécifier au système d'exploitation la taille de la partie qu'on veut réserver sur la mémoire pour tel ou tel programme.

Type de variable	Mémoire occupée	Plage de nombres acceptés	Utilisation principale
Boolean	2 octets	True (0) ou False (1)	Pour les conditions
Byte	1 octet	De 0 à 255	Pour les fichiers
Integer	4 octets	De - 2 147 483 648 à 2 147 483 647	Pour les nombres entiers
Long	8 octets	De - 9 223 372 036 854 775 808 à 9 223 372 036 854 775 807	Pour les entiers longs
String	Variable	0 à 2 milliards de caractères Unicode	Pour les chaîne de caractères

Le langage VISUAL BASIC

Les variables

La portée d'une variable

La portée d'une variable (les sub et les fonctions également) permet de spécifier sa visibilité dans tout un programme.

Mot clé	Portée de la variable
Public	Partout dans le programme
Private	Partout dans la classe ou le module où elle a été créée
Dim	Seulement dans la procédure ou la fonction créée
Protected	Les membres Protected sont similaires aux Private, mais ils ont une particularité en cas d'héritage. Ce point ne sera pas abordé
Friend	Les membres Friend ne sont accessibles qu'à l'intérieur du projet, et pas par des éléments extérieurs au projet en cours
Shared	Permet de partager une variable. N'est pas indispensable

Les constantes

Définition

Une constante est une variable qui a une valeur fixe tout au long de l'exécution du programme. L'avantage majeur des constantes est le fait qu'elles améliorent la lisibilité du code.

Exemple

```
'Exemples de constantes
Private Const Pi As Decimal = 3.14
Public Const Plat_Prefere As String = "Pates"

'Ensuite on peut les utiliser
Dim Rayon As Decimal = 5 'Soit un cercle de rayon 5
Dim AireCercle, PerimetreCercle As Decimal

AireCercle = Pi * Rayon ^ 2
PerimetreCercle = 2 * Pi * Rayon

MsgBox(Plat_Prefere) 'Affiche mon plat préféré !!
```

Les opérateurs

Les opérateurs d'arithmétiques et de concaténation

La liste

Opérateur	Opération réalisée par l'opérateur
+	Additionne deux nombres
-	Soustrait deux nombres
*	Multiplie deux nombres
/	Divise deux nombres et retourne un nombre à virgule flottante (décimal)
\	Divise deux nombres et retourne un nombre entier
Mod (Modulo)	Divise deux nombres et retourne seulement le reste
^	Elève à la puissance un nombre
&	Additionne (concatène) deux chaînes

Les opérateurs

Les opérateurs d'arithmétiques et de concaténation

Exemple

```
'Exemples
Dim x, y, Somme As Single
X = 10
Y = 15.4
Somme = X + Y 'Somme retourne 25,4

Dim Revenu, Taxes, Revenu_Reel As Integer
Revenu = 2000
Taxes = 2500
Revenu_Reel = Revenu - Taxes 'retourne -500

Dim a, b, c As Integer
a = 12
b = 3
c = a Mod b 'c = 0 car la division tombe juste donc le reste = 0

Dim Prenom, Nom, NomCompleet As String
Prenom = "Florent"
Nom = "Aspic"
NomCompleet = Prenom & Nom 'retourne "FlorentAspic"
```

Les opérateurs

Les opérateurs de logique

La liste

Opérateurs	Syntaxe	Emploi
And (et)	Var_1 And Var_2	True si Var_1 et Var_2 sont vraies
Or (ou)	Var_1 Or Var_2	True si une des deux est vraie
Xor (ou exclusif)	Var_1 Xor Var_2	True si une et une seule est vraie
Not (non)	Not Var_1	True si Var est faux et vice versa

Les opérateurs

Les opérateurs de logique

Exemple

```
'Exemples
Dim a, b As Integer
Dim Var, Var_1 As String
a = 13
b = - 15
Var = "Cool"
Var_1 = "Snif"

If (a Or b) < 0 Then
    'Evaluate la condition si a ou b est négative : ici c'est Vrai
End If

If (a And b) > 0 Then
    'Evaluate la condition si a et b sont positifs : ici c'est Faux
End If

If Not Var = Var_1 Then
    'Evaluate la condition si Var est différent de Var_1 : Ici c'est Vrai
End If
```

Les opérateurs

Les opérateurs de comparaison

Exemple

Opérateurs	Signification
>	Strictement supérieur à
<	Strictement inférieur à
>=	Supérieur ou égal à
<=	Inférieur ou égal à
<>	Différent de
=	Egal à

Les opérateurs

Les opérateurs de comparaison

Exemple

```
'Exemples
Dim Age, AgeMin, AgeMax As Integer
Dim EntreeBoiteDeNuit As Boolean

Age = 12
AgeMin = 16
AgeMax = 45

If AgeMin < Age AndAlso Age < AgeMax Then
    EntreeBoiteDeNuit = True 'La condition est Vraie
End If

If Age < AgeMin Then
    EntreeBoiteDeNuit = False 'La condition est Fausse
End If

If Age = AgeMin then
    EntreeBoiteDeNuit = True 'La condition est Vraie
End If

If Age >= 46 Then
```

Les structures conditionnelles

If - Then

Structure 1

```
If condition 1 vraie Then
    'exécuter instructions
Else
    'exécuter autre intructions
End If
```

Structure 2

```
If Condition1 vraie Then
    ..
ElseIf condition2 Then 'sinon on teste la condition 2
    ..
ElseIf condition3 Then 'la condition 3
    ..
End If
```


Les structures conditionnelles

If - Then

Imbrication

```
If condition_1 vraie Then
    If condition_2 vraie Then
        MsgBox("Condition_1 Vraie et Condition_2 Vraie")
    Else
        If condition_3 Vraie Then
            MsgBox("Condition_1 Vraie et Condition_2 Fausse et Condition_3 Vraie")
        End If
    End If
End If
```

Les structures conditionnelles

Select - End Select

Structure

Quand on évalue plusieurs fois la même condition, cette structure sera d'une bonne utilité

```
Select Case X
    Case 1 : 'faire Y
    Case 2 : 'faire Z
    Case Else : 'faire A
End Select
```

Les boucles

For - Next

Structure

```
For Variable = Debut To Fin  
    'Exécute la boucle  
Next Variable
```

Exemple 1

```
Dim i As Integer  
For i = 3 To 10 Step 2  
    MsgBox(i.ToString) 'affiche 3 puis 5 puis 7 et enfin 9  
Next i
```

Exemple 2 : Quitter la boucle prématurément

```
'Exemple de sortie de boucle  
For Test As Integer = 0 To 43  
    If Test = 3 then Exit For 'Quand Test sera égale à 3 alors on sort  
Next
```

Les structures conditionnelles

Do - Loop

Structure

```
Do  
    'Instructions  
Loop Until Condition  
  
Do  
    'Instructions  
Loop While Condition
```

Les structures conditionnelles

Do - Loop

Structure

```
'Exemple avec Until
Dim i As Integer = 0
Do
    i = i + 1 'incrmente i de 1
    MsgBox(i.ToString) 'affiche la valeur de i
Loop Until i = 10 'Quand i = 10 on sort de la boucle.
```

```
'Exemple avec While
Dim Fichier As String
Do
    'Liste les fichiers d'un repertoire
Loop While
    Fichier <> "NomDeMonFichier " 'Tant que Fichier n'est pas égal à NomDeMonFichier
```

Les structures conditionnelles

While - End While

Structure

```
While Condition
    'Exécute la boucle
End While
```

Exemple

```
'Exemple :
Dim Compteur As Integer = 0
While Compteur < 20 'Dès que Compteur = 20, on sort de la boucle.
    Compteur = Compteur + 1 'incrémente le compteur de 1
    MsgBox(Compteur.ToString) 'affiche le compteur
End While
```

Les structures conditionnelles

For Each - Next

Structure

```
For Each Item In Collection  
    'boucle  
Next Item
```

Exemple

```
'Exemple  
Dim Var_Item As String  
For Each Item As Object In MonTableau  
    Var_Item = Var_Item +  
    Item 'Var_Item va contenir tous les éléments du tableau en admettant que le tab  
Next Item  
  
'Autre exemple  
Dim Chaine As String = "Toto va bien"  
Dim Caractere As String  
For Each Caractere In Chaine  
    MsgBox(Caractere.ToString) 'affiche chaque caractère de Chaine  
Next
```

Les tableaux

Présentation

Un tableau est une variable qui peut regrouper plusieurs valeurs de même type.

Création d'un tableau

- Dim Tableau(6) As Integer

Les tableaux

Affectation

```
'Exemple d'assignations  
Tableau(2) = 3 'assigne la valeur 3 à l'élément d'indice 2  
Tableau(0) = 98 'assigne la valeur 98 à l'élément d'indice 0  
Tableau(7) = 3 'provoquera une erreur car l'indice est supérieur à 6
```

Effacement

```
'Efface le tableau  
Erase Tableau  
Tableau = Nothing  
'Ces deux commandes sont équivalentes.
```

Les tableaux

Réinitialisation

```
'Réinitialisation du tableau de l'index 0 au dernier index du tableau  
Array.Clear(Tableau, 0, Tableau.Length)
```

Parcours

```
'Lister les éléments d'un tableau  
Dim Tableau(5) As Integer  
For Each Item As Object In Tableau  
    MsgBox(Item.ToString) 'affiche chaque valeur du tableau  
Next Item  
  
'une autre méthode  
For i As Integer = 0 To Tableau.Length - 1  
    MsgBox(tableau(i).ToString)  
Next
```

Plan

- 1 Introduction
- 2 Structure d'un programme
- 3 Le langage VISUAL BASIC
- 4 Exercice

Que fait le programme suivant ?

```
Sub Main()  
    Dim val As String = ""  
    Dim x As Double = 0  
    Dim y As Double = 0  
  
    Do  
        Console.WriteLine("Entrez la première valeur")  
        val = Console.ReadLine()  
  
        Loop Until IsNumeric(val)  
  
        x = val  
  
        Do  
            Console.WriteLine("Entrez la seconde valeur")  
            val = Console.ReadLine()  
  
            Loop Until IsNumeric(val)  
  
            y = val  
  
            Console.WriteLine(x & " + " & y & " = " & x + y)  
  
            Console.Read()  
        End Sub
```