

MEMO DES BASES DU LANGAGE VISUAL BASIC

H. TSOUNGUI, INSA-UPHF, 2020

Types de données et déclarations

short, integer, long, single, double, decimal, boolean, string, char, date

Déclaration des variables et constantes :

Dim variable As **Type** : **Const PI As Decimal = 3,1415**

Public CLIENT(5,4) As Integer : tableau à 5 lignes et 4 colonnes d'entiers

Structures de contrôle

IF *expression_booléenne* **THEN**

instructions_si_vrai

[ELSE

instructions_si_faux

ENDIF

SELECT Case Valeur

Case condition-1 ou valeur1

Instructions (si le test est vrai)

Case condition-2

instructions

...

Case Else ' Autrement

instructions

END SELECT

WHILE *expression_booléenne*

instructions

WEND (ou END WHILE)

FOR *variable* = *valeur_initiale* **TO** *valeur_finale* [step *pas*]

instructions

NEXT *variable*

----- *Imbrication de boucles POUR* -----

FOR *variable-i* = *valeur_initiale* **to** *valeur_finale* [step *pas*]

instructions

FOR *variable-j* = *valeur_initiale* **to** *valeur_finale* [step *pas*]

instructions

NEXT *variable-j*

instructions

NEXT *variable-i*

DO [LOOP]

... ..

UNTIL *condition booléenne*

WITH *var*

.membre1.text = "Toto"

.membre2.text = "Lens"

END WITH

Déclaration et utilisation des structures (à déclarer dans un module de préférence)

Public Structure unclient

<VBFixedString(4)> **Public** cliNum **As** String

<VBFixedString(15)> **Public** cliNom **As** String

<VBFixedString(8)> **Public** clidatNaiss **As** Date

.....

<VBFixedString(5)> **Public** cliTaille **As** single

End Structure

Utilisation des variables structures:

Dim cli **As** **unClient**

With cli

.cliNum = "X63"

.cliNom = "Mamadou"

.clidatNaiss = "08/11/2007"

.....

.cliTaille = 1,87

End With

Déclaration de tableau dans un module

Module ModMatrices

Public MAT(m, n) **as** string

'Déclare un tableau de **m** lignes et **n** colonnes de chaînes

End Module

Initialisation du contenu d'une cellule du tableau **MAT(ligne, colonne)** :

MAT(2,3) = "pioche" 'chaîne en 3^{ème} ligne, 4^{ème} colonne

MAT(2,4) = "34" ' en ligne 3 et colonne 5

Utilisation de l'objet **ListBox**

Soit un objet liste appelé **lstClients**, pour y afficher des données (items), on utilise

lstClients.items.add(données)

ex : **lstClients.items.add**(txtNum.Text & "-" txtNom.Text)

Utilisation de l'objet **datagridView** (grille)

Soit un objet grille appelé **datagridClient**

-Déclaration du nombre de colonnes et nombre de lignes de la grille :

'Nbre de colonnes(5) et de lignes (10)

datagridClient.ColumnCount = 5

datagridClient.RowCount = 10

'Affichage des noms de colonnes

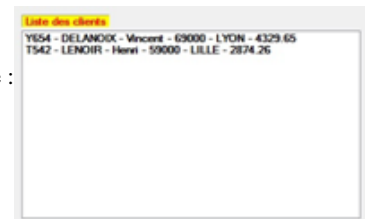
With **datagridClient**

.Columns(0).Name = "Numéro" ' Nom de la colonne 1

.Columns(1).Name = "Nom"

.Columns(2).Name = "Ville"

End With



-Initialisation du tableau et affichage de la grille (ci-dessous, à droite)
init_clients() 'Procédure de remplissage des valeurs du tableau MAT(10,5)

MAT(0,0) = "F43" : MAT(0,1) = "Leblanc":

'Affichage de 2 premiers clients dans la grille **datagridClient(colonne, ligne)**

datagridClient.Item(0, 0).Value = MAT(0, 0)

datagridClient.Item(1, 0).Value = MAT(0, 1)

L'objet **ComboBox** (capture ci-dessous)

Remplissage du comboBox

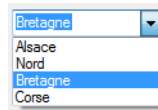
cboBoxCli.Items.Add("Alsace")

cboBoxCli.Items.Add("Nord")

cboBoxCli.Items.Add("Bretagne")

cboBoxCli.Items.Add("Corse")

cboBoxCli.SelectedItem.ToString contient le texte de l'item cliqué



| NUMERO | NOM | SECTEUR |
|--------|-----------|---------|
| F43 | Leblanc | s2 |
| E87 | Digard | s1 |
| P39 | Zouave | s2 |
| D53 | Amaurie | s1 |
| H64 | Line | s1 |
| Y65 | Kerviel | s2 |
| R28 | Tsou Ying | s2 |

DataGridView

L'objet ListView

Dim Ligne As ListViewItem = New ListViewItem(New String() {agNum, agNom, agSecteur})

'Affichage des NOMS des colonnes (entêtes: columnHeads)

LstVAgent.View = **View.Details**

LstVAgent.GridLines = **True** 'On affiche la grille du listView

LstVAgent.Columns.Add("Numéro", 60, **HorizontalAlignment.Left**)

LstVAgent.Columns.Add("Nom", 90, **HorizontalAlignment.Left**)

LstVAgent.Columns.Add("Secteur", 60, **HorizontalAlignment.Left**)

LstVAgent.Items.Add(Ligne)

'On affiche des « Lignes » dans la listView appelée **lstVAgent**

ListView

| Numéro | Nom | Secteur |
|--------|-----------|---------|
| F43 | Leblanc | s2 |
| E87 | Digard | s1 |
| P39 | Zouave | s2 |
| D53 | Amaurie | s1 |
| H64 | Line | s1 |
| Y65 | Kerviel | s2 |
| R28 | Tsou Ying | s2 |

Accès aux BDD par VBnet (cas de MySQL)

-Attention : ne pas oublier les références aux dll (Menu Projet - Ajouter une Référence...)

Imports **mysql.data**

Imports **MySQL.Data.MySqlClient**

Ces imports sont à ajouter **avant** la classe principale ou les modules

'Définition de la chaîne de Connexion « **ChaineConx** »

Dim ChaineConx **As String** = "Data Source=127.0.0.1;Database=CLIENTS;" & "User
 Id=xxx; Password=xxx;" & "Connection Timeout=20" 'cas de la connexion locale (IP 127.0.0.1)

'Dans les procédures utilisant la connexion

Dim requete **As String** = "SELECT * FROM table" 'Requête SQL nécessaire

Dim conx **As MySqlConnection**

Dim cmde **As MySqlCommand**

Dim reader **As MySqlDataReader**

conx = **New MySqlConnection**(ChaineConx)

conx.Open() 'Ouverture de la connexion

cmde = **New MySqlCommand**(requete, conx)

reader = cmde.ExecuteReader() 'Execution du Reader

While (reader.Read())

'Affichage champs cliNum, cliNom et date dans les textboxes ou le listBox

'Les indices commencent à 0.

txtNum.Text = ((reader.GetString(0)))

txtNom.Text = ((reader.GetString(1)))

lstClients.Items.Add(reader.GetString(0) & " - " & reader.GetString(1) & " - "...

Me.Refresh()

Me.Show()

End While

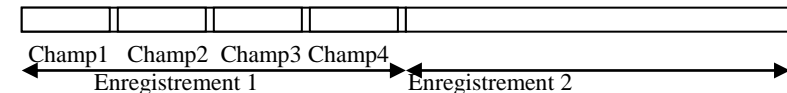
reader.Close() 'Fermeture du Reader et de la connexion à la base

conx.Close() 'Fermeture de la connexion à la base

Gestion des fichiers à accès aléatoire (random files)

Un fichier à accès aléatoire est constitué d'enregistrements homogènes et identiques. Pour lire un fichier ou y écrire, il faut d'abord l'**ouvrir, en lecture, en écriture** ou en **lecture/écriture**.

Quand on a fini de lire ou écrire, il faut **fermer** le fichier.



Les procédures pour ouvrir et fermer le fichier

Public Sub ouvrir_clients2() 'On ouvre en lecture/écriture

Dim vccli **As New client**

'Ouvrir le fichier en lecture/écriture.

FileOpen(1, "C:\CLIENTS2.DAT", **OpenMode.Random**, **OpenAccess.ReadWrite**,

OpenShare.LockWrite, Len(vccli))

End Sub

Public Sub fermer_clients2()

FileClose(1)

End Sub

Ecriture d'enregistrement (création/écriture dans le fichier)

ouvrir_clients2()

Dim vccli **As New client** : **Dim** nenreg **As Integer**

nenreg = **FileLen**("C:\clients2.dat") \ **Len**(vccli) 'Nombre d'enregistrements déjà présents

dans le fichier

nenreg = nenreg + 1 'Position d'écriture du record on écrit toujours à une position donnée

FilePut(1, vccli, nenreg) 'Ecriture de l'enregistrement (record)

fermer_clients2()

Lecture d'enregistrement (lecture du fichier)

'Ouvrir le fichier en lecture/écriture (ReadWrite)

FileOpen(1, "C:\CLIENTS2.DAT", **OpenMode.Random**, **OpenAccess.ReadWrite**,

OpenShare.LockWrite, Len(vccli))

FileGet(1, vccli, i) 'Lecture enregistrement (record)

FileClose(1) 'Fermeture du fichier