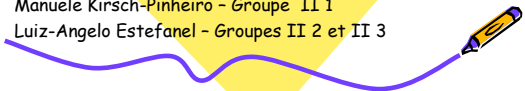


Introduction à L'Informatique


Professeur responsable : Hervé Martin

Intervenants :

- Manuele Kirsch-Pinheiro - Groupe II 1
- Luiz-Angelo Estefanel - Groupes II 2 et II 3



Plan du Cours






Environnement Visual Basic


- portée des variables
- fonctions auxiliaires
- interaction entre composants

Algorithmique




- actions itératives - WHILE
 - explication sémantique/syntaxe WHILE
- Actions itératives - FOR
 - explication sémantique/syntaxe WHILE

Portée des Variables



- En générale, une variable n'existe qu'à l'intérieur de la **procédure** qui la déclare
- Ce comportement est très gênant quand on doit garder des informations d'un appel de méthode à l'autre

Illustration

• Exercice:

- Faire un programme qui lit le contenu d'un objet TextBox et fait l'addition à une variable déclaré dans le méthode Bouton.click()

Solution



Portée des Variables

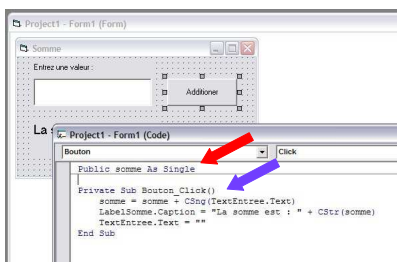
• Ces situations requièrent la déclaration des variables « globales »

- On les déclare à l'extérieur des méthodes
- On utilise le mot spécial **Public**

```
Public somme As Single
```

```
Private Sub Command1_Click()  
    somme = somme + CSng(Text1.Text)  
    Label2.Caption = "La somme est : " + CStr(somme)  
End Sub
```

Portée des Variables



Démonstration

Exercices

- Faire un programme qui lit des numéros entiers, un à la fois. Il garde le plus petit et le plus grand numéro donné par l'utilisateur, et les affiche sur la fenêtre
 - Utiliser des variables de portée globale
 - [Démonstration](#)
 - [Solution](#)



Exercices

- Vous devez réaliser un petit jeu de devinette.
 - Le but du jeu est de retrouver le nombre choisi par l'ordinateur (compris entre 1 et 50).
`Randomize`
`NbATrouver = Int((Rnd * 50) + 1)`
 - L'utilisateur saisie une valeur, et après l'examen de celle-ci, vous devez afficher si cette valeur est supérieure ou inférieure au nombre choisi.



Exercices

- Encore sur le jeu de devinette
 - Afficher le nombre de coups qu'il a fallu au joueur pour gagner
 - Ajouter un bouton permettant de recommencer le jeu
 - Défis : Comment empêcher le joueur de faire un autre coup quand il a deviné le numéro?
 - [Démonstration](#)
 - [Solution](#)



Fonctions auxiliaires

- Pour nos exercices, quelques fonctions seront très utiles
 - `InputBox("Titre", "Message")`
 - Ouvre une fenêtre pour entrer un texte
 - Retourne la string ou ""
 - `Len(String)`
 - Calcule la longueur de la chaîne en question
 - `Mid(String, position, quantité)`
 - Retourne une sub-string de caractères, avec une longueur et à partir de la position



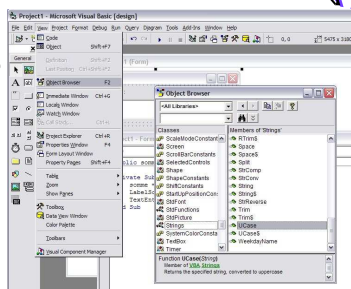
Exercices

- Faire un programme qui demande un « mot de passe » pour laisser l'utilisateur accéder au programme
 - Utiliser `InputBox`
 - Tester si le mot de passe est correct
 - Utiliser `form1.visible` pour cacher/montrer l'application
 - [Démonstration](#) / [Solution](#)



Comment trouver ?!

- Object Browser
 - Menu View -> Object Browser



Exercices

- Faire un programme qui dit s'un numéro binaire est paire ou impaire ([Démo](#) / [Solution](#))
- Faire un programme qui imprime les deux premiers et deux derniers caractères d'une phrase donnée ([Démo](#) / [Solution](#))
 - Utiliser `Len()` pour savoir la longueur des strings
 - Utiliser `Mid()` pour accéder à une position spécifique



Interaction entre Composants

- Nous avons déjà vu que on peut modifier les attributs des composants
Ex: `Text1.Text`
`Bouton1.Caption`
- D'autres interactions plus élaborées sont aussi possibles



Interaction entre Composants

- Propriétés Communes aux Composants
 - Habiliter ou désactiver un composant
 - `Bouton.Enabled`
 - Montrer ou cacher un composant
 - `Bouton.Visible`
 - Changer la Couleur
 - `Bouton.BackColor`
 - `Bouton.ForeColor`
 - `Bouton.Font`



Interaction entre Composants

- Autres Propriétés
 - Images sur les boutons
 - `Bouton.Picture`
 - `Bouton.DownPicture`
 - `Bouton.DisabledPicture`
 - Les images ne seront activées que si l'attribut `Bouton.Style=1.Graphical`



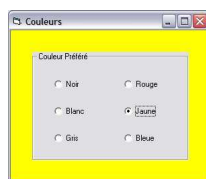
Exercice

- « Jean qui rit, Jean qui pleure »
 - Construire un formulaire avec une image que change à chaque fois qu'on click sur le bouton. ([Démo](#)/[Solution](#))



Interaction entre Composants

- Groupes de composants
 - Certains composants ont un comportement différent quand regroupés
 - Ex: Plusieurs `RadioButton` à l'intérieur d'un `Frame` (canevas)



Exercices

- Réaliser un « programme d'édition de texte », qui change le style d'une zone de texte
 - Alignement : Centré ou Gauche ou Droite
 - Style : Gras et/ou Italique et/ou Souligné
 - Couleur : Bleue ou Jaune ou Noir
 - [Démonstration](#) / [Solution](#)



Interaction entre Composants

- Interaction entre plusieurs **Forms**
 - Certaines applications nécessitent plusieurs **fenêtres**
 - Pour **ajouter** un form
 - menu Projet -> Ajouter une feuille -> Onglet Nouveau -> Form -> Ouvrir
 - Identifier l'objet de démarrage
 - Menu Projet -> Propriétés de Auto -> Objet de démarrage



Exercices

- Ajouter une feuille de Bienvenue à l'application Devinette
 - Elle présente une image de bienvenue
 - Quand on clique sur l'image, cette feuille est déchargé de la mémoire (**unload**) et la fenêtre de l'application apparaît (**show**)
 - [Démonstration](#) / [Solution](#)



Plan du Cours

Environnement Visual Basic

- portée des variables
- fonctions auxiliaires
- interaction entre composants

- Algorithmique

- actions itératives - WHILE
 - explication sémantique/syntaxe WHILE
- Actions itératives - FOR
 - explication sémantique/syntaxe WHILE



Actions itératives

- Parfois, on a besoin de répéter un ensemble d'opérations plusieurs fois (une **boucle**)
 - parcourir un vecteur de données
- Les langages de programmation en générale offrent plusieurs structures de contrôle pour les action itératives
 - FOR, WHILE, etc.



Structure WHILE

- WHILE est une structure qui permet l'exécution des instructions tant que la condition est vraie

While condition	→	While cont<10
...		cont=cont+1
Wend		Wend



Exemple de WHILE

- Reprendre l'exercice pour calculer la note moyenne d'un groupe de 16 étudiants
- Utiliser une boucle **while**

```
Dim i As Integer
i=1
While i < 16
  moyenne = (notes(i,1) + notes(i,2) +
            + notes(i,3)) / 3
  notes(i,4) = moyenne
  i = i + 1
Wend
```



Exercices

- Réaliser un programme « Distributeur Automatique de Billets » ([Démo](#) / [Solution](#))
 - L'automate contient des billets de 10€, 20€, 50€ et 100€
 - Étant donné une demande de retrait, le programme calcule et affiche le nombre de billets de chaque valeur
 - Le programme minimise le nombre de billets distribués



Exercices

- Sur le programme « Distributeur Automatique de Billets » ([Démo](#) / [Solution](#))
 - L'automate contient un **nombre finit** de billets de 10€, 20€, 50€ et 100€
 - Lors du retrait, l'automate bancaire mettra à jour les piles de billets
 - Le programme minimise le nombre de billets distribués, dans la mesure où il y a encore de billets d'une valeur



Structure FOR

- FOR est une structure qui permet l'exécution des instructions un nombre déterminé de fois

For compteur=valeur To valeurfin [Step incrément]

[Exit For]

Next compteur



Exemple de FOR

Dim notes (1 to 4, 1 to 16) As Integer

Dim moyenne As Single

notes(1,1) = 10

notes(1,2) = 14

...

notes(16,3) = 12

For i = 1 To 16 Step 1

moyenne = (notes(i,1) + notes(i,2) + notes(i,3)) / 3

notes(i,4) = moyenne

Next i

• Calculer la moyenne entre
3 notes pour chaque
étudiant d'un groupe



Exercices

- Réaliser un programme qui compte le nombre de fois que les lettres apparaissent dans une phrase. Imprimer la fréquence d'occurrence de ces lettres

- Utiliser un vecteur
- Il n'y a pas différence entre majuscules et minuscules
- Solution



Exercices

- Réaliser un programme qui déplace une image dans la fenêtre
 - de gauche à droite
 - de droite à gauche
 - de gauche à droite et de haut en bas
 - [Démonstration](#) / [Solution](#)



Organisation du Cours

✓ Solutions TP



Exercice

- Faire un programme qui lit le contenu d'un objet TextBox et fait l'addition à une variable déclarée dans le méthode Bouton.click()

```
Private Sub Bouton_Click()  
    Dim somme As Single  
    somme = somme + CSng(TextEntree.Text)  
    LabelSomme.Caption = "La somme est : " + CStr(somme)  
    TextEntree.Text = ""  
End Sub
```

[Retour](#)



Exercice - Variables de Portée Globale

- Faire un programme qui lit des numéros entiers, un à la fois. Il garde le plus petit et le plus grand numéro donné par l'utilisateur, et les affiche sur la fenêtre



Exercice - Variables de Portée Globale

```
Public Min As Integer 'plus petit numero
Public Max As Integer 'plus grand numero
Public deja As Boolean 'on a deja utilisee le form?
```

```
Private Sub BoutonMinMax_Click()
    Dim valeur As Integer
    valeur = CInt(Text1.Valeur.Text) 'valeur que l'utilisateur vient de rentrer
    If (deja = False) Then 'premiere fois que l'utilisateur propose un numero
        Min = valeur
        Max = valeur
        deja = True
    Else
        If (valeur > Max) Then
            Max = valeur
        End If
        If (valeur < Min) Then
            Min = valeur
        End If
    End If
    LabelMin.Caption = "Minimum = " & Min
    LabelMax.Caption = "Maximum = " & Max
End Sub
Private Sub Form_Load()
    deja = False
End Sub
```

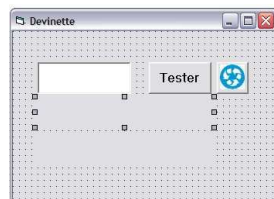
[Retour](#)

Exercices - Devinette

```
'Variables globales
Dim NbATrouver As Integer 'numero a trouver
Dim NbCoups As Integer 'coups
```

```
Private Sub Form_Load()
    'Initialisation
    Randomize
    NbATrouver = Int((Rnd * 50) + 1)
    NbCoups = 0
End Sub
```

→



Exercices - Devinette

```
Private Sub BoutonParier_Click()
    Dim NbTape As Integer

    'On recupere la valeur tape par le joueur
    NbTape = Val(TextPari.Text)

    'On compare les 2 valeurs
    If (NbTape > NbATrouver) Then
        LabelMessage.Caption = "Le nombre à trouver est plus petit"
    ElseIf (NbTape < NbATrouver) Then
        LabelMessage.Caption = "Le nombre à trouver est plus grand"
    Else
        LabelMessage.Caption = "Bravo, vous avez gagné !!!"
        'On desactive le bouton Command1
        BoutonParier.Enabled = False
    End If

    'On ajoute 1 coup
    NbCoups = NbCoups + 1
    LabelCoups.Caption = "Vous avez joué " & Str(NbCoups) & " coups"
End Sub

Private Sub BoutonRecommencer_Click()
    'On initialise les variables en appelant
    'la fonction Form_Load
    Call Form_Load

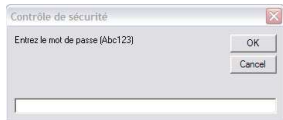
    'On décore l'interface
    'On active le bouton Command1
    BoutonParier.Enabled = True
    TextPari.Text = ""
    LabelMessage.Caption = ""
    LabelCoups.Caption = ""
End Sub
```

[Retour](#)

Exercices - Mot de Passe

```
Private Sub Form_Load()
    Dim MotdePasse As String
    Form1.Visible = False
    MotdePasse = InputBox("Entrez le mot de passe (Abc123)", _
        "Contrôle de sécurité")

    If MotdePasse = "Abc123" Then
        Form1.Visible = True
    Else
        MsgBox "Désolé !!!", vbOKOnly, "Alerte de Sécurité"
    End If
End Sub
```



[Retour](#)

Exercices - Binaire Paire/Impaire

- Faire un programme qui dit s'un numéro binaire est paire ou impaire



Exercices - Binaire Paire/Impaire

```

Private Sub BoutonPaire_Click()
    Dim binaire As String
    Dim dernier As String
    Dim taille As Integer

    binaire = TextBinaire.Text
    taille = Len(binaire) 'taille de la string
    If taille > 0 Then 'est-elle superieur a zero ?
        dernier = Mid(binaire, taille, 1) 'on copie sur binaire le dernier caractere
        If dernier = "0" Then 's'il est 0, c'est un binaire paire
            LabelMessage.Caption = "Le numéro est paire"
        Else
            If dernier = "1" Then 's'il est 1, c'est un binaire impaire
                LabelMessage.Caption = "Le numéro est impaire"
            Else
                'dans un binaire, il n'y a que 0 et 1
                LabelMessage.FontBold = True
                LabelMessage.Caption = "Erreur : le numéro n'est pas binaire"
            End If
        End If
    End If
End Sub

```

[Retour](#)

Exercices - premiers/derniers

- Faire un programme qui imprime les deux premiers et deux derniers caractères d'une phrase donnée

```

Private Sub BoutonCouper_Click()
    Dim phrase As String
    Dim taille As Integer

    phrase = TextPhrase.Text
    taille = Len(phrase)

    LabelPremiers.Caption = Mid(phrase, 1, 2)
    LabelDerniers.Caption = Mid(phrase, taille - 1, 2)
End Sub

```

[Retour](#)

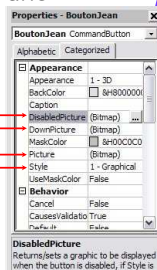


Exercices - Jean

- Construire un formulaire avec une image que change à chaque fois qu'on click sur le bouton

- Propriétés du bouton

- Style
- Picture
- DownPicture → bouton appuyé
- DisabledPicture → bouton désactivé



Exercices - Jean

- Construire un formulaire avec une image que change à chaque fois qu'on click sur le bouton
- Pour activer / deactiver → **CheckBox**

```
Private Sub CheckActive_Click()  
If CheckActive.Value = 0 Then  
    BoutonJean.Enabled = True  
Else  
    BoutonJean.Enabled = False  
End If
```

End Sub

[Retour](#)



Exercices - éditeur

- Réaliser un « programme d'édition de texte », qui change le style d'une zone de texte



Exercices - éditeur

```
Private Sub CheckGras_Click()  
If TextInput.FontBold = True Then  
    TextInput.FontBold = False  
Else  
    TextInput.FontBold = True  
End If  
End Sub
```

```
Private Sub CheckItalique_Click()  
If TextInput.FontItalic = True Then  
    TextInput.FontItalic = False  
Else  
    TextInput.FontItalic = True  
End If  
End Sub
```

```
Private Sub CheckSouligne_Click()  
If TextInput.FontUnderline = True Then  
    TextInput.FontUnderline = False  
Else  
    TextInput.FontUnderline = True  
End If  
End Sub
```

```
Private Sub OptionCentre_Click()  
    TextInput.Alignment = vbCenter  
End Sub
```

```
Private Sub OptionGauche_Click()  
    TextInput.Alignment = vbLeftJustify  
End Sub
```

→

→



Exercices - éditeur

```
Private Sub OptionDroite_Click()  
    TextInput.Alignment = vbRightJustify  
End Sub
```

```
Private Sub OptionBleue_Click()  
    TextInput.ForeColor = vbBlue  
End Sub
```

```
Private Sub OptionJaune_Click()  
    TextInput.ForeColor = vbYellow  
End Sub
```

```
Private Sub OptionNoir_Click()  
    TextInput.ForeColor = vbBlack  
End Sub
```

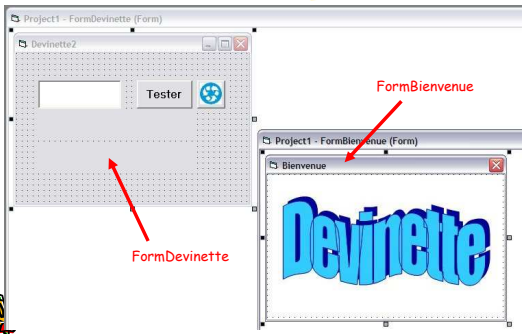


[Retour](#)

Exercices - forms

- Ajouter une feuille de Bienvenue à l'application Devinette

Exercices - forms



Exercices - forms

```
Private Sub ImageBienvenue_Click()  
    FormDevinette.Show  
    Unload FormBienvenue  
End Sub  
  
Private Sub FormBienvenue_Click()  
    FormDevinette.Show  
    Unload FormBienvenue  
End Sub
```

[Retour](#)



Exercices - DAB

- Distributeur Automatique de Billets



Exercices - DAB

```
Private Sub BoutonRetirer_Click()  
    Dim valeur As Integer  
    Dim num100 As Integer  
    Dim num50 As Integer  
    Dim num20 As Integer  
    Dim num10 As Integer  
  
    valeur = CInt(TextValeur.Text)  
    If (valeur Mod 10) <> 0 Then  
        valeur = (valeur \ 10) * 10  
        TextValeur.Text = CStr(valeur)  
        LabelMessage.Caption = "Valeur arrondie à " & CStr(valeur)  
    End If  
  
    While valeur >= 100  
        num100 = num100 + 1  
        valeur = valeur - 100  
    Wend  
  
    While valeur >= 50  
        num50 = num50 + 1  
        valeur = valeur - 50  
    Wend  
  
    While valeur >= 20  
        num20 = num20 + 1  
        valeur = valeur - 20  
    Wend  
  
    While valeur >= 10  
        num10 = num10 + 1  
        valeur = valeur - 10  
    Wend  
  
    Label100.Caption = num100  
    Label50.Caption = num50  
    Label20.Caption = num20  
    Label10.Caption = num10  
  
End Sub
```



[Retour](#)



Exercices - DAB

- Distributeur Automatique de Billets - nombre finit de billets

Exercices - DAB

```

Public Pile100 As Integer
Public Pile50 As Integer
Public Pile20 As Integer
Public Pile10 As Integer

Private Sub Form_Load()
    Pile100 = 20
    Pile50 = 20
    Pile20 = 20
    Pile10 = 20
    LabelPile100.Caption = Pile100
    LabelPile50.Caption = Pile50
    LabelPile20.Caption = Pile20
    LabelPile10.Caption = Pile10
End Sub

Private Sub BoutonRetrait_Click()
    Dim valeur As Integer
    Dim num100 As Integer
    Dim num50 As Integer
    Dim num20 As Integer
    Dim num10 As Integer
    Dim EnCaisse As Integer
    Label100.Caption = ""
    Label50.Caption = ""
    Label20.Caption = ""
    Label10.Caption = ""
    LabelMessage.Caption = ""

    valeur = CInt(TextRetrait.Text)
    If (valeur Mod 10) <> 0 Then
        valeur = (valeur \ 10) * 10
        LabelMessage.Caption = "Valeur arrondi à " & _
            & CStr(valeur)
    End If

    EnCaisse = Pile100 * 100 + Pile50 * 50 + Pile20 * 20 + Pile10 * 10
    
```

Exercices - DAB

```

If valeur < EnCaisse Then
    While valeur >= 100 And Pile100 > 0
        num100 = num100 + 1
        valeur = valeur - 100
        Pile100 = Pile100 - 1
    Wend

    While valeur >= 50 And Pile50 > 0
        num50 = num50 + 1
        valeur = valeur - 50
        Pile50 = Pile50 - 1
    Wend

    While valeur >= 20 And Pile20 > 0
        num20 = num20 + 1
        valeur = valeur - 20
        Pile20 = Pile20 - 1
    Wend

    While valeur >= 10 And Pile10 > 0
        num10 = num10 + 1
        valeur = valeur - 10
        Pile10 = Pile10 - 1
    Wend

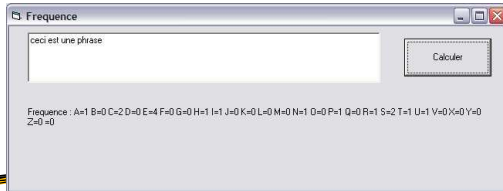
    Label100.Caption = num100
    Label50.Caption = num50
    Label20.Caption = num20
    Label10.Caption = num10
    LabelPile100.Caption = Pile100
    LabelPile50.Caption = Pile50
    LabelPile20.Caption = Pile20
    LabelPile10.Caption = Pile10
Else
    LabelMessage.Caption = "Pas assez d'argent en caisse:" & _
        & "maximum " & CStr(EnCaisse)
End If
End Sub

```

Retour

Exercices - Fréquence

- Réaliser un programme qui compte le nombre de fois que les lettres apparaissent dans une phrase.



Exercices - Fréquence

```
Private Sub BoutonCalculer_Click()
    Dim lettres As String
    Dim frequence(1 To 26) As Integer
    Dim phrase As String
    Dim char1 As String
    Dim char2 As String
    Dim resultat As String
    Dim i As Integer
    Dim j As Integer

    phrase = TextPhrase.Text

    lettres = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
    For j = 1 To 26 Step 1
        frequence(j) = 0
    Next

    'on prend une lettre de la phrase
    For i = 1 To Len(phrase) Step 1
        char1 = UCase(Mid(phrase, i, 1))

        'et on voit a quelle lettre elle correspond
        For j = 1 To 26 Step 1
            char2 = UCase(Mid(lettres, j, 1))
            If char1 = char2 Then
                frequence(j) = frequence(j) + 1
            End If
        Next
    Next

    resultat = "Frequence : "
    For j = 1 To 26 Step 1
        resultat = resultat + Mid(lettres, j, 1) + "="
        resultat = resultat + CStr(frequence(j)) + " "
    Next
    LabelResultat.Caption = resultat
End Sub
```

[Retour](#)

Exercices - image

- Réaliser un programme qui déplace une image dans la fenêtre

```
Private Sub BoutonDroite_Click()
    Dim i As Integer
    For i = 1 To 4000 Step 2
        Image1.Move i + 240, 360
    Next i
End Sub

Private Sub BoutonGauche_Click()
    Dim i As Integer
    For i = 4000 To 1 Step -2
        Image1.Move i + 240, 360
    Next i
End Sub

Private Sub BoutonBas_Click()
    Dim i As Integer
    For i = 1 To 4000 Step 2
        Image1.Move 240 + i, 360 + i / 2
    Next i
End Sub
```



[Retour](#)
