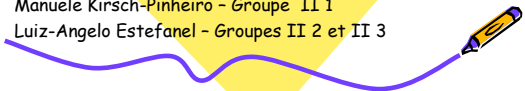



Introduction à L'Informatique

Professeur responsable : Hervé Martin



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




Plan du Cours





- Introduction à Visual Basic
 - présentation à l'environnement
 - éléments du langage
- Algorithmique
 - actions conditionnelles - IF
 - explication sémantique/syntaxe IF



Visual Basic



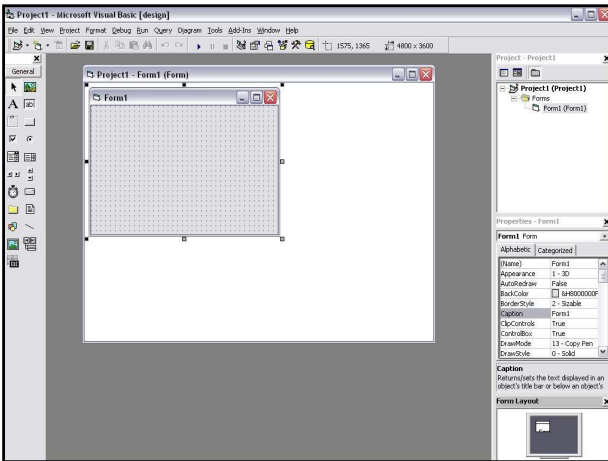
- VB comme environnement de programmation
 - Environnement graphique
 - Composants graphiques déjà prêts
 - Boutons, zones de texte, fenêtres, ...
- VB comme langage de programmation
 - Manipulation des données
 - Description des actions



Environnement Graphique

- VB est *a priori* utilisé pour le développement des applications pour Windows
- L'environnement VB permet la construction rapide des fenêtres, avec boutons, menus, et tout ce qu'une application Windows besoin





Le Contrôle *form*

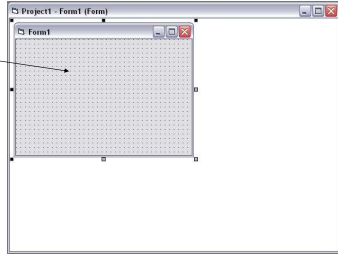
- C'est la feuille ou formulaire (*form*, en anglais) sur laquelle les composants graphiques sont placés
- Un *form* normalement représente une fenêtre sous Windows
- Une application VB contient au moins un *form*



Le contrôle *form*

Fenêtre *form*

Correspond exactement à la fenêtre de l'application qu'on construit



La boîte à outils

- Contient les **composants** traditionnels d'une application Windows
- En choisissant ces composants, on peut dessiner rapidement une fenêtre d'application

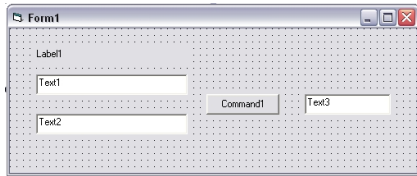


Composants?

- Composant est comme un **type de données** « complexe » qui représente, en générale, une entité réelle
 - Composant Bouton
 - Attributs
 - Nom
 - Couleur
 - Actions
 - Clique simple ()
 - Clique double ()

Exercices

- Faire connaissance avec les composants de la boîte à outils
- Construire une fenêtre comme celle ci



Attributs des Composants

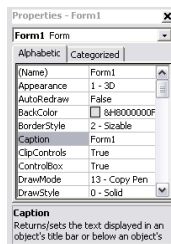
- Un composant a des attributs et des actions
- Les attributs sont des données propres à un objet spécifique

Composant Bouton1
Name = Bouton1
Caption = Texte sur bouton1

Attributs des Composants VB

- La fenêtre *Propriétés* permet l'accès aux attributs des composants

Exemples :
name : nom du objet
caption : texte sur l'objet



Exercices

- Reprendre l'exercice précédent
- Changer le nom des objets
 - On y change pour des noms plus représentatifs
 - Command1 -> BtnCalculer
 - Label1 -> LabelTitre
 - TextField1 -> TxtValeur1
 - TextField2 -> TxtValeur2
 - TextField2 -> TxtResultat

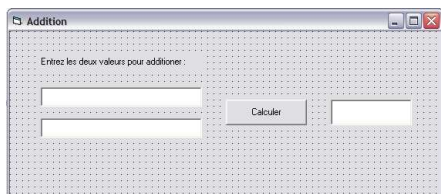


Exercices


- Changer les attributs des objets
 - Mettre un titre
 - LabelTitre.caption = «Entrez les deux valeurs pour additionner : »
 - Donner un « nom » au bouton
 - BtnCalculer.caption = « Calculer »
 - Laisser vides les zones de texte
 - TxtValeur1.text
 - ...



Exercices

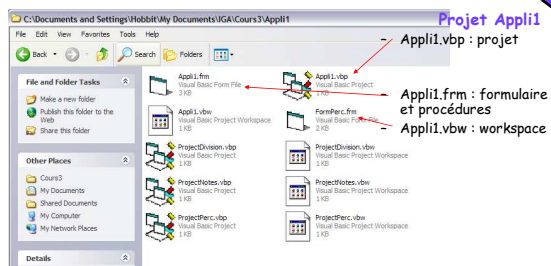


Sauvegarder le Projet


- Très important!!!
 - Il faut toujours **sauvegarder** avant de tester l'application
- Cliquez sur Fichier -> Sauvegarder Projet
 - ou sur le bouton 
- Les fichiers créés sont des fichiers texte qui décrivent les objets et leurs attributs



Fichiers d'un Projet



Exécution de l'Application

- Deux façons de le faire
 - À l'intérieur de Visual Basic
 - Cliquez sur « play » 
 - Ou sur le menu Run -> Start
 - À travers un fichier exécutable
 - Une « vraie » application
 - Fichier -> Make ____ .exe



Exercices

1. Tester l'application à l'intérieur de l'environnement Visual Basic
2. Créer l'exécutable et l'exécuter

Oh-oh...
Cliquer sur le bouton ne fait rien!!!



Actions

- Pour être fonctionnel, un composant nécessite la définition des actions qui seront effectuées
 - Pour un composant graphique, on appelle ces actions les « événements »
 - Ex. : un clique sur un bouton



Écrire des Actions

- Le clique du bouton « Calculer »
 - Cliquez deux fois sur le bouton ou sur le menu Affichage->Code Source
 - Sélectionner, dans la fenêtre de code, l'objet « BtnCalculer » et l'action « Click »
 - L'algorithme pour l'action click est écrit à l'intérieur de la procédure

```
Private Sub BtnCalculer_Click()  
End Sub
```



Exercices

- Écrire l'algorithme pour l'action « click »
 - `TxtResultat.Text = TxtValeur1.Text + TxtValeur2.Text`
- Tester votre application
- Pourquoi il ne fait pas la somme des valeurs?
 - Types de données



Conversion de Données

- On a vu précédemment que les opérations dépendent du **type** des données
 - Le signe « + »
 - Addition d'entiers ou réels
 - Concaténation de caractères
- Parfois on est obligé de faire la **conversion** des données



Conversion de Données

- Conversion vers **Entiers**
 - On utilise l'appel `CInt()`
 - Ex: `CInt(TxtValeur1.Text)`
- Conversion vers **Caractères**
 - Appel `CStr()`
- Conversion vers **Réels**
 - Appel `CSng()`
- Conversion vers **Booléen**
 - Appel `CBool()`



Exercices

- Réécrire l'algorithme pour « click »
 - Utiliser l'appel **CInt()**
 - `TxtResultat.Text = CInt(TxtValeur1.Text) + CInt(TxtValeur2.Text)`
 - Utiliser l'appel **CSng()**
 - `TxtResultat.Text = CSng(TxtValeur1.Text) + CSng(TxtValeur2.Text)`
- Quel est la différence entre **CInt()** et **CSng()**?

Exercices

- Faire en VB les exercices du cours précédant :
 - calculer la valeur moyenne de deux numéros
 - calculer la valeur de la division entière de deux nombres entiers
 - calculer une réduction de 10% sur un prix donnée

Solutions

Plan du Cours

- Introduction à Visual Basic
 - présentation à l'environnement
 - éléments du langage
- Algorithmique
 - actions conditionnelles - IF
 - explication sémantique/syntaxe IF

L'opération conditionnelle IF (si)

- Parfois, le déroulement d'une action dépend des conditions spécifiques

```
Si employe.salaire > 5000€ alors
  Impot = 10%
Sinon
  Impot = 5%
Fin Si
```



L'opération conditionnelle IF

- En Visual Basic, l'opération **IF** a la syntaxe suivante :

```
If condition Then
  ...actions...
Else
  ...actions...
End If
```



Condition?

- L'opération **IF** doit évaluer une **condition** pour déterminer la suite de l'opération
- Une condition est une opération logique (**vrai** ou **faux**?)

```
Salaire > 5000
Bouton.name = Calculer
Salaire > 5000 AND age < 25
```



Exercices

- Faire en VB un programme pour calculer le montant d'impôt à payer, selon le revenu mensuel :
 - 0-900€ -> 0% d'impôt
 - 901-5000€ -> 10% d'impôt
 - Plus de 5000€ -> 25% d'impôt

Solution



Exercices

- Ajouter au programme précédent une réduction des impôts dus selon l'âge du déclarant
 - Moins de 18 ans -> imprimer un message d'erreur (« Non imposable !!! »)
 - 18 - 25 ans -> subvention de 50€ (mais pas de remboursement si la subvention est plus grande que les impôts)
 - Plus de 26 ans -> aucune subvention

Solution



Exercices

- Faire en VB un programme pour calculer la note moyenne d'un étudiant dont :
 - On fournit les 5 notes
 - La note la plus basse et la plus élevée ne sont pas considérées pour faire la moyenne

Solution



Exercices

- Faire en VB un programme que compare deux numéros donnés. Le programme indique si :
 - Numéro1 < Numéro2
 - Numéro1 = Numéro2
 - Numéro1 > Numéro2

Solution



Exercices

- Faire en VB un programme pour trier trois numéros
 - En ordre croissante
 - En ordre décroissante
- **Extra** : Réfléchissez un algorithme (sans l'implémenter en VB) pour un nombre arbitraire d'éléments

Solution



Organisation du Cours

✓ Solutions TPs



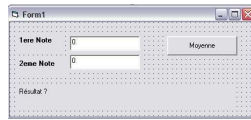
Solutions Exercices VB

- calculer la valeur moyenne de deux numéros

```
Private Sub BtnMoyenne_Click()  
Dim a As Integer  
Dim b As Integer  
Dim moyenne As Single
```

```
a = CInt(TxtNote1.Text)  
b = CInt(TxtNote2.Text)  
moyenne = (a + b) / 2  
LabelResult.Caption = "La moyenne est de : " + CStr(moyenne)
```

```
End Sub
```



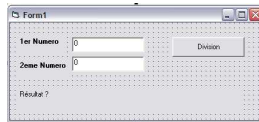
Solutions Exercices VB

- calculer la valeur de la division entière de deux nombres entiers

```
Private Sub BtnDiv_Click()  
Dim a As Integer  
Dim b As Integer  
Dim c As Integer
```

```
a = CInt(TxtNote1.Text)  
b = CInt(TxtNote2.Text)  
c = a \ b 'division entière  
LabelResult.Caption = "Le résultat de division est : " +  
CStr(c)
```

```
End Sub
```



Solutions Exercices VB

- calculer une réduction de 10% sur un prix donnée

```
Private Sub BtnPerc_Click()  
Dim prix As Single  
prix = CSng(TxtNote1.Text)  
prix = prix - (prix * 0.1)  
LabelResult.Caption = "Réduction de 10% sur Prix : " + CStr(prix)  
End Sub
```



[Retour](#)



Solutions Exercices VB

```
Private Sub BtnCalculer_Click()
    Dim revenu As Single
    Dim impot As Single

    revenu = CSng(TxtRevenu.Text)

    If revenu <= 900 Then
        impot = 0
    Else
        If revenu <= 5000 Then
            impot = revenu * 0.1
        Else
            impot = revenu * 0.25
        End If
    End If

    TxtImpot.Text = CStr(impot)
End Sub
```

- Calculer le montant d'impôt à payer, selon le revenu mensuel



[Retour](#)

Solutions Exercices VB

- Calculer le montant d'impôt à payer, selon le revenu mensuel et l'âge

```
Private Sub BtnCalculer_Click()
    Dim revenu As Single
    Dim impot As Single
    Dim age As Integer
    Dim subvention As Integer

    revenu = CSng(TxtRevenu.Text)
    age = CInt(TxtAge.Text)
    subvention = 0
    impot = 0

    If age < 18 Then
        TxtImpot.Text = "Non imposable!"
    Else
        If age >= 18 And age <= 25 Then
            subvention = 50
        End If
    End If
```



→

Solutions Exercices VB

- Calculer le montant d'impôt à payer, selon le revenu mensuel et l'âge

```

If revenu <= 900 Then
    impot = 0
Else
    If revenu <= 5000 Then
        impot = revenu * 0.1
    Else
        impot = revenu * 0.25
    End If
End If

If impot > subvention Then
    impot = impot - subvention
End If

TxtImpot.Text = CStr(impot)
End If

End Sub
```



[Retour](#)

Solutions Exercices VB

- Faire en VB un programme pour calculer la note moyenne d'un étudiant dont :
 - On fournit les 5 notes
 - La note la plus basse et la plus élevée ne sont pas considérées pour faire la moyenne

Solutions Exercices VB

```
Private Sub BtnCalculer_Click()
    Dim notes(1 To 5) As Single
    Dim moyenne As Single
    Dim min As Single
    Dim max As Single

    min = 20
    max = 0

    notes(1) = CSng(TxtNote1.Text)
    notes(2) = CSng(TxtNote2.Text)
    notes(3) = CSng(TxtNote3.Text)
    notes(4) = CSng(TxtNote4.Text)
    notes(5) = CSng(TxtNote5.Text)

    ' d'abord on decouvre le min
    If notes(1) <= min Then
        min = notes(1)
    End If
    If notes(2) <= min Then
        min = notes(2)
    End If
    If notes(3) <= min Then
        min = notes(3)
    End If
    If notes(4) <= min Then
        min = notes(4)
    End If
    If notes(5) <= min Then
        min = notes(5)
    End If
    LabelMin.Caption = "Minimum écarté : " + CStr(min)
End Sub
```

Solutions Exercices VB

```
moyenne = (notes(1) + notes(2) + _
notes(3) + notes(4) + _
notes(5))
moyenne = moyenne - min
moyenne = moyenne - max
moyenne = moyenne / 3

TxtMoyenne.Text = CStr(moyenne)
End Sub
```

[Retour](#)

Solutions Exercices VB

- Faire en VB un programme que compare deux numéros donnés. Le programme indique si :
 - Numéro1 < Numéro2
 - Numéro1 = Numéro2
 - Numéro1 > Numéro2



Solutions Exercices VB

```
Private Sub BtnComparer_Click()
    Dim num1 As Single
    Dim num2 As Single
    Dim result As String
```

```
num1 = TxtNumero1.Text
num2 = TxtNumero2.Text
result = "Je n'en sais rien"
```

```
If (num1 > num2) Then
    result = CStr(num1) + " > " + CStr(num2)
ElseIf (num1 = num2) Then
    result = CStr(num1) + " = " + CStr(num2)
Else
    result = CStr(num1) + " < " + CStr(num2)
End If
```

```
LabelResultat.Caption = "Comparaison: " + result
```

```
End Sub
```

[Retour](#)



Solutions Exercices VB

- Faire en VB un programme pour trier trois numéros
 - En ordre croissante
 - En ordre décroissante



Solutions Exercices VB



```
Private Sub BtnCroissante_Click()
    Dim ordre(1 To 3) As Single
    Dim nums(1 To 3) As Single

    nums(1) = CSng(TextNumero1.Text)
    nums(2) = CSng(TextNumero2.Text)
    nums(3) = CSng(TextNumero3.Text)

    If (nums(1) > nums(2)) Then
        If (nums(1) > nums(3)) Then
            If (nums(2) > nums(3)) Then
                ordre(1) = nums(3)
                ordre(2) = nums(2)
                ordre(3) = nums(1)
            Else
                ordre(1) = nums(2)
                ordre(2) = nums(3)
                ordre(3) = nums(1)
            End If
        Else
            ordre(1) = nums(2)
            ordre(2) = nums(3)
            ordre(3) = nums(1)
        End If
    Else
        If (nums(2) > nums(3)) Then
            If (nums(1) > nums(3)) Then
                ordre(1) = nums(3)
                ordre(2) = nums(1)
                ordre(3) = nums(2)
            Else
                ordre(1) = nums(1)
                ordre(2) = nums(3)
                ordre(3) = nums(2)
            End If
        Else
            ordre(1) = nums(1)
            ordre(2) = nums(2)
            ordre(3) = nums(3)
        End If
    End If

    LabelResultat.Caption = "Ordre Croissante : " & _
        + CStr(ordre(1)) & _
        + " " & CStr(ordre(2)) & _
        + " " & CStr(ordre(3))
End Sub
```

Solutions Exercices VB



```
Private Sub BtnDecroissante_Click()
    Dim ordre(1 To 3) As Single
    Dim nums(1 To 3) As Single

    nums(1) = CSng(TextNumero1.Text)
    nums(2) = CSng(TextNumero2.Text)
    nums(3) = CSng(TextNumero3.Text)

    If (nums(1) > nums(2)) Then
        If (nums(1) > nums(3)) Then
            If (nums(2) > nums(3)) Then
                ordre(3) = nums(3)
                ordre(2) = nums(2)
                ordre(1) = nums(1)
            Else
                ordre(3) = nums(2)
                ordre(2) = nums(3)
                ordre(1) = nums(1)
            End If
        Else
            ordre(3) = nums(2)
            ordre(2) = nums(3)
            ordre(1) = nums(1)
        End If
    Else
        If (nums(2) > nums(3)) Then
            If (nums(1) > nums(3)) Then
                ordre(3) = nums(3)
                ordre(2) = nums(1)
                ordre(1) = nums(2)
            Else
                ordre(3) = nums(1)
                ordre(2) = nums(2)
                ordre(1) = nums(3)
            End If
        Else
            ordre(3) = nums(1)
            ordre(2) = nums(2)
            ordre(1) = nums(3)
        End If
    End If

    LabelResultat.Caption = "Ordre Décroissante : " & _
        + CStr(ordre(1)) & _
        + " " & CStr(ordre(2)) & _
        + " " & CStr(ordre(3))
End Sub
```

[Retour](#)

Solutions Exercices VB



- **Extra** : Réfléchissez un algorithme (sans l'implémenter en VB) pour un nombre arbitraire d'éléments
 - Solution choisi : *Bubble Sort*

```

k = 4
for i=1 to k-1
    for j=(i+1) to k
        if top(j)>top(i)
            tmpvalue=top(i)
            top(i)=top(j)
            top(j)=tmpvalue
        end if
    next j
next i
                
```

[Retour](#)