

VB.NET

Interface utilisateur

Abdelali Saidi

saidi.a@ucd.ac.ma

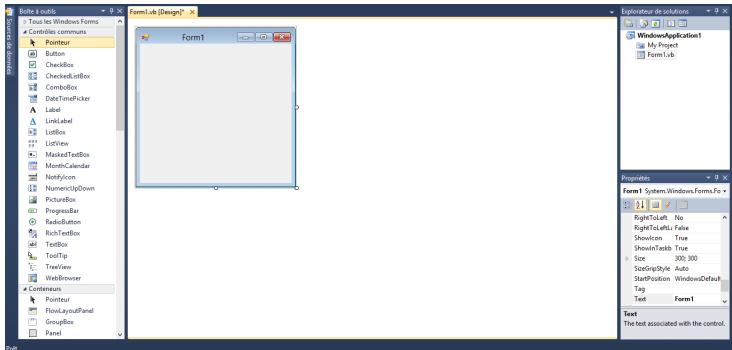
- 1 Les contrôles de base
- 2 Exemple d'une première application
- 3 Plus de précisions

Plan

- 1 Les contrôles de base
- 2 Exemple d'une première application
- 3 Plus de précisions

L'interface Utilisateur

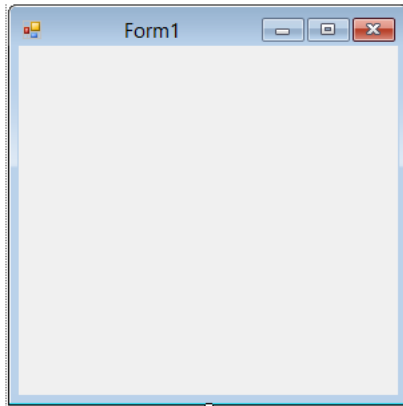
Un nouveau projet :



L'interface Utilisateur

La Form

Illustration



L'interface Utilisateur

La Form

Présentation

- Ce sont des formulaires Windows
- Une Form est générée automatiquement à la création de chaque nouveau projet
- Il est possible d'ajouter d'autres Forms dans un même projet
- Elle possède des propriétés (comme n'importe quel autre *contrôle*)

L'interface Utilisateur

La Form

Propriétés (1/2)

Propriété	Description
BackColor	Définis l'arrière plan du contrôle
BackgroundImage	L'image en arrière plan du contrôle
Cursor	Permet de changer le curseur lors du passage
Font	La font (police) utilisée dans le contrôle
ForeColor	La couleur du premier plan pour le texte
FormBorderStyle	Comportement de la form (bordure et barre de titre)
Text	Le titre de la form
AllowDrop	Indique si on autorise le glisser / coller
ContextMenuStrip	Associe un menu contextuel au contrôle
Enabled	Indique si le contrôle est activé ou pas
Name	Le nom du contrôle utilisé dans le codage
Size	La taille du contrôle en pixels

L'interface Utilisateur

La Form

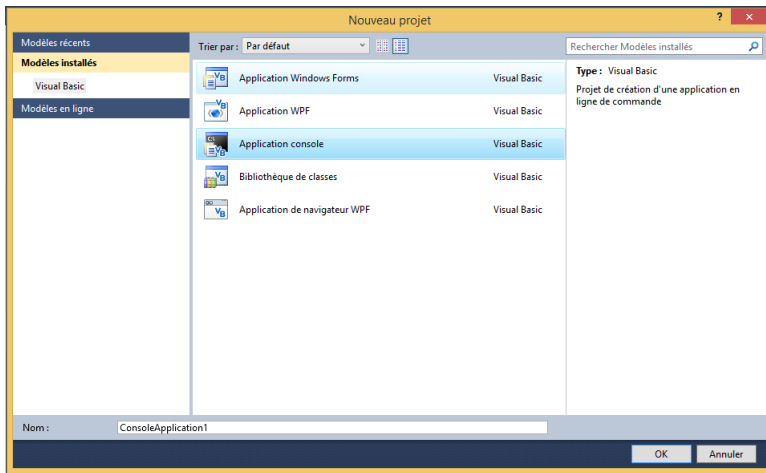
Propriétés (2/2)

WindowState	L'état de la form au démarrage (minimize...)
AcceptButton	Le bouton d'acceptation du formulaire
CancelButton	Le bouton d'annulation du formulaire
ControlBox	Indique si le menu système est activée ou pas
HelpButton	Indique si le bouton d'aide est activé ou pas
Icon	Définit l'icône du contrôle
MainMenuStrip	Définit le menu principal du formulaire
MaximizeBox	Indique si le bouton d'agrandissement est activé
ShowIcon	Indique si l'icône est visible
ShowInTaskBar	Indique si le formulaire est visible dans la TaskBar
TopMost	Indique si le formulaire est au premier plan

L'interface Utilisateur

La Console

Illustration



L'interface Utilisateur

La Console

Présentation

- C'est une fenêtre de type DOS (invite de commande)
- On ne peut y afficher que du texte

L'interface Utilisateur

La Console

Exemple

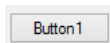
```
'Exemple de lignes de code utilisant la console
Dim LigneTapee As String = Console.In.ReadLine() 'permet de lire du texte
Console.ReadLine() 'marche aussi
Console.Out.WriteLine("Ligne à afficher") 'pour afficher du texte
Console.WriteLine 'est aussi accepté

'Petit programme en application console
Console.Out.Write("Tapez une ligne : ")
Dim LigneTapee As String = Console.In.ReadLine()
Console.Out.WriteLine("Je viens de taper : " & LigneTapee)
```

L'interface Utilisateur

Le Bouton

Illustration



Présentation

- C'est le composant le plus important
- Il permet une interaction entre l'application et son utilisateur
- En général, un bouton permet de valider ou d'annuler une opération

L'interface Utilisateur

Le Bouton

Propriétés

Propriété	Description
BackColor	Définit l'arrière plan du contrôle
BackgroundImage	L'image en arrière plan du contrôle
Cursor	Permet de changer le curseur lors du passage de la souris
FlatStyle	Le style du bouton
Font	Le font (police) utilisé dans le contrôle
ForeColor	La couleur du premier plan pour le texte
Text	Le texte du bouton
TextAlign	L'alignement du texte du bouton
AllowDrop	Indique si on autorise le glisser / coller
ContextMenuStrip	Associe un menu contextuel au contrôle
Enabled	Indique si le contrôle est activé ou pas
Visible	Indique si le contrôle est visible ou pas
Name	Le nom du contrôle utilisé dans le codage
Size	La taille du contrôle en pixels
AutoSize	Définit si le contrôle est dimensionné automatiquement

L'interface Utilisateur

Le Label

Illustration



Label1

Présentation

- C'est un élément informateur
- Il apporte une indication pour l'utilisateur de l'application
- Il est non modifiable

L'interface Utilisateur

Le Label

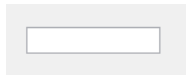
Propriétés

Propriété	Description
BackColor	Définit l'arrière plan du contrôle
Image	L'image affichée sur le label
Cursor	Permet de changer le curseur lors du passage de la souris
FlatStyle	Le style du bouton
Font	Le font (police) utilisé dans le contrôle
ForeColor	La couleur du premier plan pour le texte
Text	Le texte du bouton
TextAlign	L'alignement du texte du bouton
AllowDrop	Indique si on autorise le glisser / coller
ContextMenuStrip	Associe un menu contextuel au contrôle
Enabled	Indique si le contrôle est activé ou pas
Visible	Indique si le contrôle est visible ou pas
Name	Le nom du contrôle utilisé dans le codage
Size	La taille du contrôle en pixels
AutoSize	Définit si le contrôle est dimensionné automatiquement

L'interface Utilisateur

La Textbox

Illustration



Présentation

- C'est un emplacement pour la saisie du texte
- Elle permet à l'application de récupérer des données
- Elle est donc modifiable par l'utilisateur

L'interface Utilisateur

La Textbox

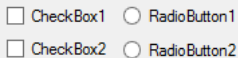
Propriétés

Propriété	Description
BackColor	Définit l'arrière plan du contrôle
Cursor	Permet de changer le curseur lors du passage de la souris
Font	Le font (police) utilisé dans le contrôle
ForeColor	La couleur du premier plan pour le texte
Text	Le texte initial visible dans la TextBox
TextAlign	L'alignement du texte
ContextMenuStrip	Associe un menu contextuel au contrôle
Enabled	Indique si le contrôle est activé ou pas
MaxLength	Indique le nombre maximal de caractères dans la TextBox
Multiline	Indique si la TextBox est multi ligne ou pas
PasswordChar	Caractère à afficher pour la saisie du mot de passe
ReadOnly	Indique si la TextBox est en lecture seule
Visible	Indique si le contrôle est visible ou pas
Name	Le nom du contrôle utilisé dans le codage
Size	La taille du contrôle en pixels
AllowDrop	Indique si on autorise le glisser / coller

L'interface Utilisateur

Les CheckBox et les RadioButton

Illustration



Présentation

- Ce sont des cases à cocher
- Permettent de savoir si l'utilisateur veut ou pas activer une fonction quelconque
- Avec les CheckBox, l'utilisateur peut cocher plusieurs fonctions
- Avec les RadioButton, l'utilisateur ne peut choisir qu'une seule fonction

L'interface Utilisateur

Les CheckBox et les RadioButton

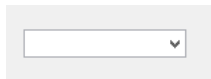
Propriétés

Propriété	Description
Appearance	Indique l'apparence du contrôle
BackColor	Définit l'arrière plan du contrôle
BackgroundImage	L'image en arrière plan du contrôle
CheckAlign	Emplacement de la case à cocher
Checked	Indique si la case est cochée
CheckState	Indique l'état de la case à cocher
Cursor	Permet de changer le curseur lors du passage de la souris
FlatStyle	Le style du bouton
Font	Le font (police) utilisé dans le contrôle
ForeColor	La couleur du premier plan pour le texte
Text	Le texte de la CheckBox
TextAlign	L'alignement du texte
AllowDrop	Indique si on autorise le glisser / coller
AutoCheck	Indique si la case se coche automatiquement
ContextMenuStrip	Associe un menu contextuel au contrôle
Enabled	Indique si le contrôle est activé ou pas
ThreeState	Indique si la case à 3 états d'activation
Visible	Indique si le contrôle est visible ou pas
Name	Le nom du contrôle utilisé dans le codage
Size	La taille du contrôle en pixels

L'interface Utilisateur

La Combobox

Illustration



Présentation

- C'est un menu déroulant
- Elle permet de choisir un élément entre plusieurs

L'interface Utilisateur

La Combobox

Propriétés

Propriété	Description
BackColor	Définit l'arrière plan du contrôle
Cursor	Permet de changer le curseur lors du passage de la souris
DropDownStyle	Apparence de la zone du menu déroulant
FlatStyle	Le style de la ComboBox
Font	Le font (police) utilisé dans le contrôle
ForeColor	La couleur du premier plan pour le texte
Text	Le texte de la ComboBox
AllowDrop	Indique si on autorise le glisser / coller
ContextMenuStrip	Associe un menu contextuel au contrôle
DrawMode	Indique si le code gère le dessin des éléments de la liste
Enabled	Indique si le contrôle est activé ou pas
Sorted	Indique comment sont triés les éléments de la liste
Visible	Indique si le contrôle est visible ou pas
Name	Le nom du contrôle utilisé dans le codage
Size	La taille du contrôle en pixels
Items	Permet d'ajouter les éléments dans la liste déroulante
Tag	Données définies par l'utilisateur

L'interface Utilisateur

Les listes

Présentation

Il existe trois types de listes :

① Les ListBox

- C'est le contrôle le plus facile à maîtriser
- Il permet d'ajouter en ligne les éléments

② Les ListView

- C'est le contrôle le plus complexe à maîtriser
- Il permet d'ajouter plusieurs colonnes

③ Les TreeView

- C'est une arborescence
- Il y a toujours une racine, puis des noeuds, puis des sous-noeuds ...

L'interface Utilisateur

Les listes

Propriétés d'une ListBox

Propriété	Description
BackColor	Définit l'arrière plan du contrôle
Cursor	Permet de changer le curseur lors du passage de la souris
Font	Le font (police) utilisé dans le contrôle
ForeColor	La couleur du premier plan pour le texte
AllowDrop	Indique si on autorise le glisser / coller
DrawMode	Indique si le code gère le dessin des éléments de la liste
ContextMenuStrip	Associe un menu contextuel au contrôle
MultiColumn	Indique comment doivent être affichées les données
Sorted	Indique comment sont triés les éléments de la liste
Visible	Indique si le contrôle est visible ou pas
Name	Le nom du contrôle utilisé dans le codage
Size	La taille du contrôle en pixels
Items	Permet d'ajouter les éléments dans la ListBox

L'interface Utilisateur

Les listes

Propriétés d'une ListView

Propriété	Description
BackColor	Définit l'arrière plan du contrôle
BackgroundImage	L'image en arrière plan du contrôle
BorderStyle	Indique le style de la bordure de la ListView
Cursor	Permet de changer le curseur lors du passage de la souris
CheckBoxes	Indique si des CheckBox sont visibles à coté des éléments
Font	Le font (police) utilisé dans le contrôle
ForeColor	La couleur du premier plan pour le texte
FullRowSelect	Indique si toute la ligne est sélectionnée
GridLines	Affiche un quadrillage autour des éléments en mode détails
View	Définit la vue dans laquelle les éléments seront affichés
AllowDrop	Indique si on autorise le glisser / coller
Columns	Gère les colonnes de la ListView
ContextMenuStrip	Associe un menu contextuel au contrôle
Items	Gère les éléments de la ListView

L'interface Utilisateur

Les listes

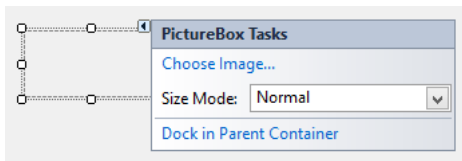
Propriétés d'une TreeView

Propriété	Description
BackColor	Définit l'arrière plan du contrôle
BorderStyle	Indique le style de la bordure de la TreeView
Cursor	Permet de changer le curseur lors du passage de la souris
CheckBoxes	Indique si des CheckBox sont visibles à coté des éléments
Font	Le font (police) utilisé dans le contrôle
ForeColor	La couleur du premier plan pour le texte
AllowDrop	Indique si on autorise le glisser / coller
ContextMenuStrip	Associe un menu contextuel au contrôle
DrawMode	Indique si le code gère le dessin des éléments de la liste
FullRowSelect	Indique si la surbrillance s'étend sur la largeur du TreeView
Nodes	Gère l'ensemble des noeuds du contrôle TreeView
Visible	Indique si le contrôle est visible ou pas
Name	Le nom du contrôle utilisé dans le

L'interface Utilisateur

La PictureBox

Illustration



Présentation

- Permet d'afficher et de redimensionner des images

L'interface Utilisateur

La PictureBox

Propriétés

Propriété	Description
BackColor	Définit l'arrière plan du contrôle
BackgroundImage	L'image en arrière plan du contrôle
Cursor	Permet de changer le curseur lors du passage de la souris
Image	L'image a affichée dans la PictureBox
ErrorImage	Icone à afficher lorsque le chargement d'une image échoue
InitialImage	Image à afficher pendant le chargement d'une autre image
ContextMenuStrip	Associe un menu contextuel au contrôle
Enabled	Indique si le contrôle est activé ou pas
SizeMode	Manière dont l'image va être redimensionnée
Visible	Indique si le contrôle est visible ou pas
Name	Le nom du contrôle utilisé dans le codage
Size	La taille du contrôle en pixels

L'interface Utilisateur

La ProgressBar

Présentation

- Permet de suivre la progression d'une application
- Indique à l'utilisateur l'avancement d'une opération (une recherche de fichiers par exemple)
- Il faut indiquer une valeur minimale et une autre maximale pour faire fonctionner ce contrôle (une représentation en pourcentage par exemple)

L'interface Utilisateur

La ProgressBar

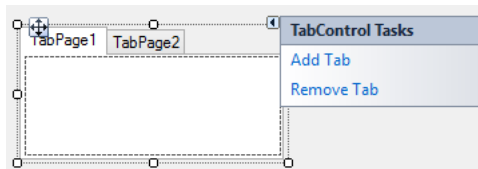
Propriétés

Propriété	Description
BackColor	Définit l'arrière plan du contrôle
Cursor	Permet de changer le curseur lors du passage de la souris
ForeColor	La couleur du premier plan pour le texte
Maximum	Limite supérieure de la plage que la ProgressBar utilise
Minimum	Limite inférieure de la plage que la ProgressBar utilise
Style	Définit le style de la ProgressBar
Value	Valeur actuelle pour la ProgressBar
ContextMenuStrip	Associe un menu contextuel au contrôle
Visible	Indique si le contrôle est visible ou pas
Name	Le nom du contrôle utilisé dans le codage
Size	La taille du contrôle en pixels
Tag	Données définies par l'utilisateur associées à l'objet
Enabled	Indique si le contrôle est activé ou pas

L'interface Utilisateur

Le TabControl

Illustration



Présentation

- C'est un outil très important et bien pratique
- Il permet de gérer beaucoup de choses en onglets
- Il fait parti des conteneurs

L'interface Utilisateur

Le TabControl

Propriétés

Propriété	Description
Cursor	Permet de changer le curseur lors du passage de la souris
Font	La font (police) utilisée dans le contrôle
ImageList	Objet qui permet de contenir des images pour les onglets
AllowDrop	Indique si on autorise le glisser / coller
DrawMode	Indique si l'utilisateur ou le système peint les légendes
HotTrack	Indique si l'aspect des onglets change lors du survol de la souris
ContextMenuStrip	Associe un menu contextuel au contrôle
MultiLine	Indique si plusieurs lignes d'onglets sont autorisées
SizeMode	Indique la façon dont les onglets sont redimensionnés
TabPage	La liste des TabPages contenus dans le TabControl
Visible	Indique si le contrôle est visible ou pas
Name	Le nom du contrôle utilisé dans le

L'interface Utilisateur

Les boîtes de dialogue

Présentation

Les boîtes de dialogue sont des outils très utilisés chez les applications Windows. En général, il en existe 5, chacune avec un rôle spécifique :

- **OpenFileDialog** : demande à l'utilisateur d'ouvrir un fichier
- **FolderBrowserDialog** : permet de chercher un dossier spécifique
- **ColorDialog** : permet d'affecter une couleur parmi d'autres à un élément donné
- **SaveFileDialog** : permet l'utilisateur d'indiquer l'emplacement de ce qu'il veut sauvegarder
- **FontDialog** : affecte une certaine police à un texte

L'interface Utilisateur

Les boîtes de dialogue

Exemple

```
'Exemples avec les boîtes de dialogues
Dim Selection As String
Dim Couleur As Color
Dim MonFont As Font

FolderBrowserDialog.ShowDialog()
Selection = FolderBrowserDialog.SelectedPath 'récupère le chemin du dossier

OpenFileDialog.ShowDialog()
Selection = OpenFileDialog.FileName 'récupère le nom et le chemin du fichier

ColorDialog.ShowDialog()
Couleur = ColorDialog.Color 'récupère la couleur sélectionnée

FontDialog.ShowDialog()
MonFont = FontDialog.Font 'récupère le font (police, taille et style)
```

Plan

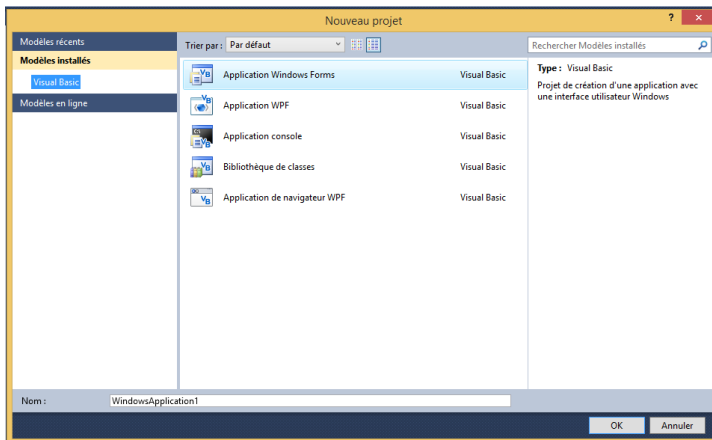
- 1 Les contrôles de base
- 2 Exemple d'une première application
- 3 Plus de précisions

Exemple d'une première application

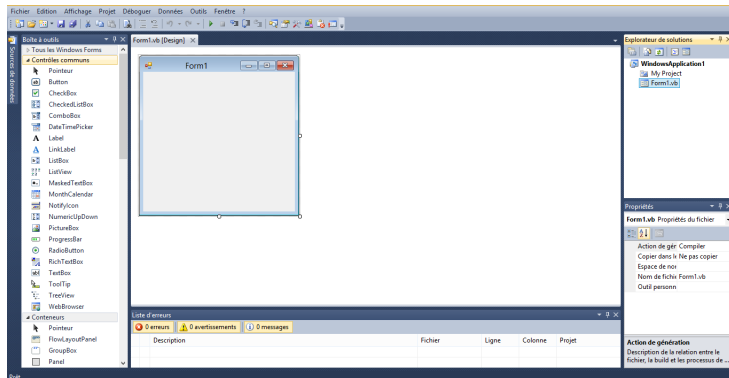
On désire un formulaire qui récupère deux nombre et qui affiche leur somme

The image shows a screenshot of a Windows application window titled "Form1". The window has a standard Windows XP-style title bar with minimize, maximize, and close buttons. Inside the form, there are three labels: "Label1", "Label2", and "Label3", arranged vertically. To the right of each label is a text box. At the bottom of the form, there are three buttons: "Button1", "Button3", and "Button2", arranged horizontally. The form has a light gray background and a blue border.

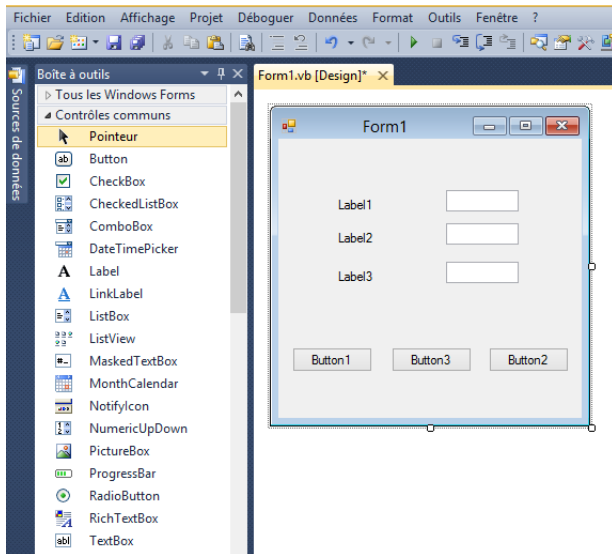
Exemple d'une première application



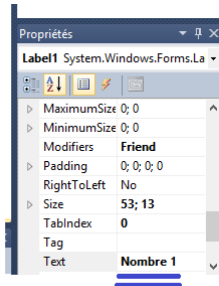
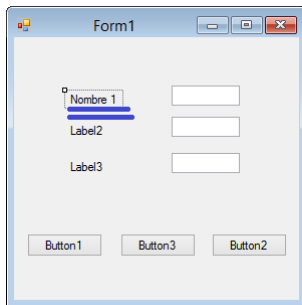
Exemple d'une première application



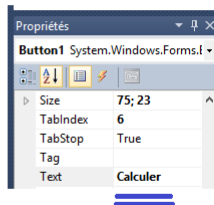
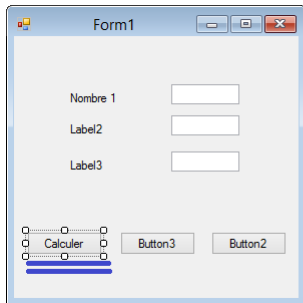
Exemple d'une première application



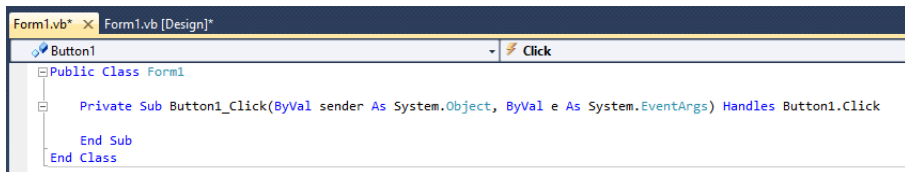
Exemple d'une première application



Exemple d'une première application



Exemple d'une première application



Exemple d'une première application

```
Public Class Form1
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
        If TextBox1.Text <> Space(0) And TextBox2.Text <> Space(0) Then
            TextBox3.Text = CInt(TextBox1.Text) + CInt(TextBox2.Text)
        End If
        TextBox1.Focus()
    End Sub
End Class
```

Exemple d'une première application

```
Public Class Form1

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
        If TextBox1.Text <> Space(0) And TextBox2.Text <> Space(0) Then
            TextBox3.Text = CInt(TextBox1.Text) + CInt(TextBox2.Text)
        End If
        TextBox1.Focus()
    End Sub

    Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button3.Click
        TextBox1.ResetText()
        TextBox2.ResetText()
        TextBox3.ResetText()
        TextBox1.Select()
    End Sub

    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles quitter.Click
        Dim quitter As Boolean
        quitter = MsgBox("Voulez vous vraiment quitter l'application ?", MsgBoxStyle.YesNo, "!!!")
        If quitter = True Then
            End
        End If
    End Sub
End Class
```

Plan

- 1 Les contrôles de base
- 2 Exemple d'une première application
- 3 Plus de précisions

Plus de précisions

Manipulation de plusieurs formulaires

Ouverture

- L'ouverture d'un formulaire par le biais d'un autre se fait en :
 - Créant un objet de type formulaire : `Dim f As New FormN()`
 - Appelant la méthode : `f.ShowDialog()`

Fermeture

- `f.Close()` permet de fermer le formulaire
- `f.Hide()` permet de le cacher

Remarques :

- La fenêtre `f` sera modale (on ne peut aller à une autre fenêtre que si on ferme celle là)
- La méthode `f.Show()` permet d'ouvrir une fenêtre non modale

Plus de précisions

Manipulation de plusieurs formulaires

Initialiser le formulaire

Au cours du lancement du projet, le formulaire chargé exécute la procédure `FormN_Load()`. Par exemple :

```
Private Sub Form1_Load()  
  
    TextBox1.Text= "bonjour"  
  
End Sub
```

Plus de précisions

Les boutons

Créer un évènement

Pour indiquer au programme ce qu'il doit faire suite à l'évènement "Click sur le bouton", on doit renseigner la procédure :

```
Private Sub Button_Click(..)  
  
End Sub
```

Remarque : On peut aller directement à ce bout de code en double cliquant le bouton en mode conception

Plus de précisions

Les boutons

Formulaire avec le comportement d'une boîte de dialogue

Il se peut qu'on voudrait que notre formulaire aie un comportement d'une boîte de dialogue. Pour cela, il faut modifier les propriétés suivantes :

- **AcceptButton** : Pour que la touche ENTRÉE agisse directement sur un bouton donné
- **CancelButton** : Pour que la touche ESC agisse directement sur un bouton donné

Plus de précisions

Les boutons

Création d'un bouton par code

```
Private Sub InitializeMonButton()  
    Dim button1 As New Button1()  
    button1.Text="Ok"  
    ' Ajouter le bouton à la Form  
    Controls.Add(button1)  
End Sub
```

Plus de précisions

Les textBox

Quelques propriétés

- `TextBox.Text` : permet de récupérer le texte saisi par l'utilisateur ou d'afficher du texte
- `ReadOnly` : rend le text de la textBox non modifiable
- `MaxLenght` : limite le nombre de caractères qu'on peut saisir sur la textBox
- `PasswordChar` : permet d'afficher des étoiles au lieu du texte saisi

Plus de précisions

Les textBox

Validation de saisie

Il est possible de contrôler ce que saisisse l'utilisateur lettre par lettre. Quand on tape une nouvelle touche, les évènements suivants se déclenchent :

- ❶ `textBox_KeyDown()` : Au moment de l'appuie
- ❷ `textBox_KeyPress()` : Au moment de l'enfoncement
- ❸ `textBox_KeyUp()` : Au moment du relâchement

1 et 3 retournent un paramètre `e` dont l'attribut `e.KeyCode` permet de récupérer le caractère saisi. 2 retourne un paramètre `e` dont l'attribut `e.KeyChar` permet de récupérer le caractère saisi.

Plus de précisions

Les textBox

Comment interdire la saisie d'un caractère ?

```
Private Sub TextBox1_KeyPress(ByVal sender As Object, ByVal e As  
    System.Windows.Forms.KeyPressEventArgs) Handles TextBox1.KeyPress  
    If e.KeyChar = "A" Then e.Handled = True  
End Sub
```

Comment interdire la saisie de plusieurs caractères ?

```
Private Sub TextBox1_KeyPress  
    If Not "<>=/*-+".Contains(e.KeyChar) Then e.KeyChar = Nothing
```

Plus de précisions

Les textBox

Ne permettre que la saisie de chiffres

```
Private Sub TextBox1_KeyPress(ByVal sender As Object, ByVal e As  
    System.Windows.Forms.KeyPressEventArgs)  
-Handles TextBox1.KeyPress  
  
If IsNumeric(e.KeyChar) Then  
    e.Handled = False  
  
Else  
    e.Handled = True  
  
End If  
  
End Sub
```

Plus de précisions

Les textBox

Modifier le caractère saisi

```
Private Sub TextBox1_KeyPress(ByVal sender As Object, ByVal e As  
    System.Windows.Forms.KeyPressEventArgs) _  
Handles TextBox1.KeyPress  
  
Dim pos As Integer  
  
pos = TextBox1.SelectionStart 'on mémorise la position du curseur  
  
If e.KeyChar = "," Then  
  
e.Handled = True 'on ne valide pas le caractère ',' qui n'apparaît  
  
TextBox1.Text = TextBox1.Text.Insert(pos, ".") 'on insère un '.'  
  
TextBox1.SelectionStart = pos + 1 'on avance le curseur d'un caract  
  
End If
```

Plus de précisions

Les textBox

Valider le texte en quittant le contrôle

```
Private Sub TextBox1_Validated(ByVal sender As Object, ByVal e As System.EventArgs)
    TextBox1.Validated
    Dim Contenu As String = TextBox1.Text
    '*** Enlever les blancs
    Contenu = System.Text.RegularExpressions.Regex.Replace(Contenu, "[\s]*", " ")

    '*** N'accepter que les entiers ou les décimaux avec 3 chiffres derrière la virgule
    Dim pattern As String
    pattern = "^[([0-9]+)|([([0-9]+)(,)([0-9]{0,3}))?)$)"
    If Contenu = System.Text.RegularExpressions.Regex.Match(Contenu,
pattern).ToString AndAlso Contenu <> "" Then
        TextBox1.Text = Contenu
    Else
        Contenu = ""
        TextBox1.Text = ""
        TextBox1.Focus()
    End If
End Sub
```

Plus de précisions

Les cases à cocher

RadioButton et CheckBox

- Il faut regrouper les radios boutons (pareillement les checkbox) dans des "GroupBox" par exemple pour rendre les groupes indépendants. Sinon, cliquer sur un bouton décochera tous les autres boutons

The image shows two separate Windows Forms controls, each a GroupBox. The first GroupBox, titled "Format", contains two radio buttons: "RTF" and "TEXTE BRUT". The second GroupBox, titled "Sauvegarde auto", contains two radio buttons: "OUI" and "NON". This visualizes how to group related radio buttons into independent containers to prevent them from being mutually exclusive across different parts of the application.

Plus de précisions

Les cases à cocher

RadioButton et CheckBox

- `RadioButton.text` : permet d'afficher du texte à côté du bouton
- `RadioButton.CheckAlign` : permet de mettre la case à cocher à droite ou à gauche du texte
- `RadioButton.TextAlign` : permet d'aligner le texte
- `RadioButton.Checked` : permet de savoir si le bouton est coché ou pas

Plus de précisions

Les cases à cocher

Les listBox

L'utilisation d'une listBox entraîne l'utilisation d'une collection d'Items.

- `ListBox.Items` contient tous les éléments chargés dans la liste.
- `ListBox.Items.Count` indique le nombre d'éléments contenus dans la liste
- `ListBox.Items.Add()`, `Items.Insert()` : permettent d'ajouter des éléments à la liste
- `ListBox.Items.Clear()` : permet de la vider
- `ListBox.Items.Remove()` : permet de supprimer un élément quelconque

Plus de précisions

Les cases à cocher

Charger des nombres

```
Dim i As Integer
For i = 1 To 100

    ListBox1.Items.Add(i.ToString)

Next i
```

Lire un élément

```
Dim t As String

t=ListBox1.Items(3).ToString
```

Plus de précisions

Les cases à cocher

Charger une listBox à partir d'un fichier texte

```
ListBox1.Items.AddRange(System.IO.File.ReadAllLines("c:\\list.txt"))
```

Enlever des éléments

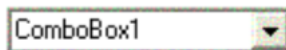
```
' Enlever l'élément d'index 5:  
ListBox1.Items.RemoveAt(5)  
  
' Enlever l'élément sélectionné:  
ListBox1.Items.Remove(ListBox1.SelectedItem)  
  
' Enlever l'élément "Tokyo":  
ListBox1.Items.Remove("Tokyo")
```

Plus de précisions

Les comboBox

Les comboBox possèdent deux caractéristiques essentielles par rapport aux listBox

- Ils sont modifiables
- Déroulantes ou déjà déroulées



Plus de précisions

Les comboBox

- `Items.Add()` : ajoute un élément à une liste.
- `Items.Clear()` : efface tous les éléments d'une liste
- `Items.Count` : renvoie le nombre d'éléments d'une liste
- `Multiselect` : permet la sélection multiple
- `Item.Remove()` : supprime un élément de la liste
- `Sorted` : trie les éléments d'une liste
- `ComboBox1.Text` : récupère le texte de la zone