



OFPPT

ROYAUME DU MAROC

مكتب التكوين المهني وإنعاش الشغل

Office de la Formation Professionnelle et de la Promotion du Travail

Technicien Spécialisé  
en Développement Informatique

Pratiquer



Réalisé par :

Aziz ELANZI

Encadré Par :

M. BENTALEB, Mohammed



# Plan Des Tp

---

<i>REMERCIEMENT :</i>	3
<i>INTRODUCTION :</i>	3
<i>INTERFACE VB.NET:</i>	3
<i>TP N° 1 CREATION D'UNE INTERFACE</i>	3
<i>TP N° 2 LES FONCTIONS</i>	3
<i>TP N° 3 LES FORMULAIRES</i>	3
<i>TP N° 4 STRUCTURES ET TABLEAUX</i>	3
<i>TP N° 5 ÉNUMERATION, STRUCTURE ET COLLECTION</i>	3
<i>TP N° 6 LES EVENEMENTS, LES CASES A COCHER ET LES CASES OPTIONS</i>	3
<i>TP N° 7 L'EVENEMENT CLOSING ET REGROUPEMENT DES CONTROLES</i>	3
<i>TP N° 8 LISTBOX ET L'EVENEMENT ENABLED</i>	3
<i>TP N° 9 LA BOUCLE FOR...NEXT</i>	3
<i>TP N° 10 L'ÉVÉNEMENT KEYPRESS</i>	3
<i>TP N° 11 LISTE SÉLECTION</i>	3
<i>TP N° 12 ARRAYLIST, CLASSE</i>	3
<i>TP N° 13 TABCONTROL</i>	3
<i>TP N° 14 RITCHTEXTBOX, CONTEXTMENU, COLORDIALOG, FONTDIALOG</i>	3
<i>TP N° 15 LES FICHIERS, LES MENUS, DATAGRID</i>	3
<i>NOTES :</i>	3
<i>CONCLUSION :</i>	3

## **Remerciement :**

Avant de commencer à présenter le contenu de ce modeste travail, je tiens à remercier tous mes amis de classe sauf les ennemis de la connaissance, ceux qui refusent de partager l'information pour la rendre disponible à tout le monde afin d'obtenir une génération dite « Génération d'informatique » capable de relever les défis de la modernisation.

Je remercie particulièrement notre cher formateur M. Mohammed BENTALEB pour sa créativité, sa compétence et sa rigueur, et pour tous ce qu'il a fait pour nous afin de nous mettre dans la bonne voie du développement.

Je remercie aussi tout le personnel de l'ISGI Beni Mellal et notamment Mme NAJAT.

## **Introduction :**

Puisque les langages de programmation ont devenu un outil nécessaire dans le développement informatique, j'ai opté la réalisation d'un petit travail qui met en œuvre l'un des langages les plus répondu au monde, c'est le VB.NET de Microsoft inclus dans la plateforme de Visual Studio.Net.

Le lecteur de ce petit rapport constatera que le point a été mis sur le côté pratique et non plus la théorie, la chose que le lecteur doit savoir avant de prendre mon travail en main.

Les fonctionnalités les plus utilisables de VB.NET sont presque toutes traitées avec soin afin de faciliter la compréhension et l'apprentissage.

**Bonne lecture ☺**

**Avec mes meilleurs voeux**

## Interface VB.NET:

Boîte à outils

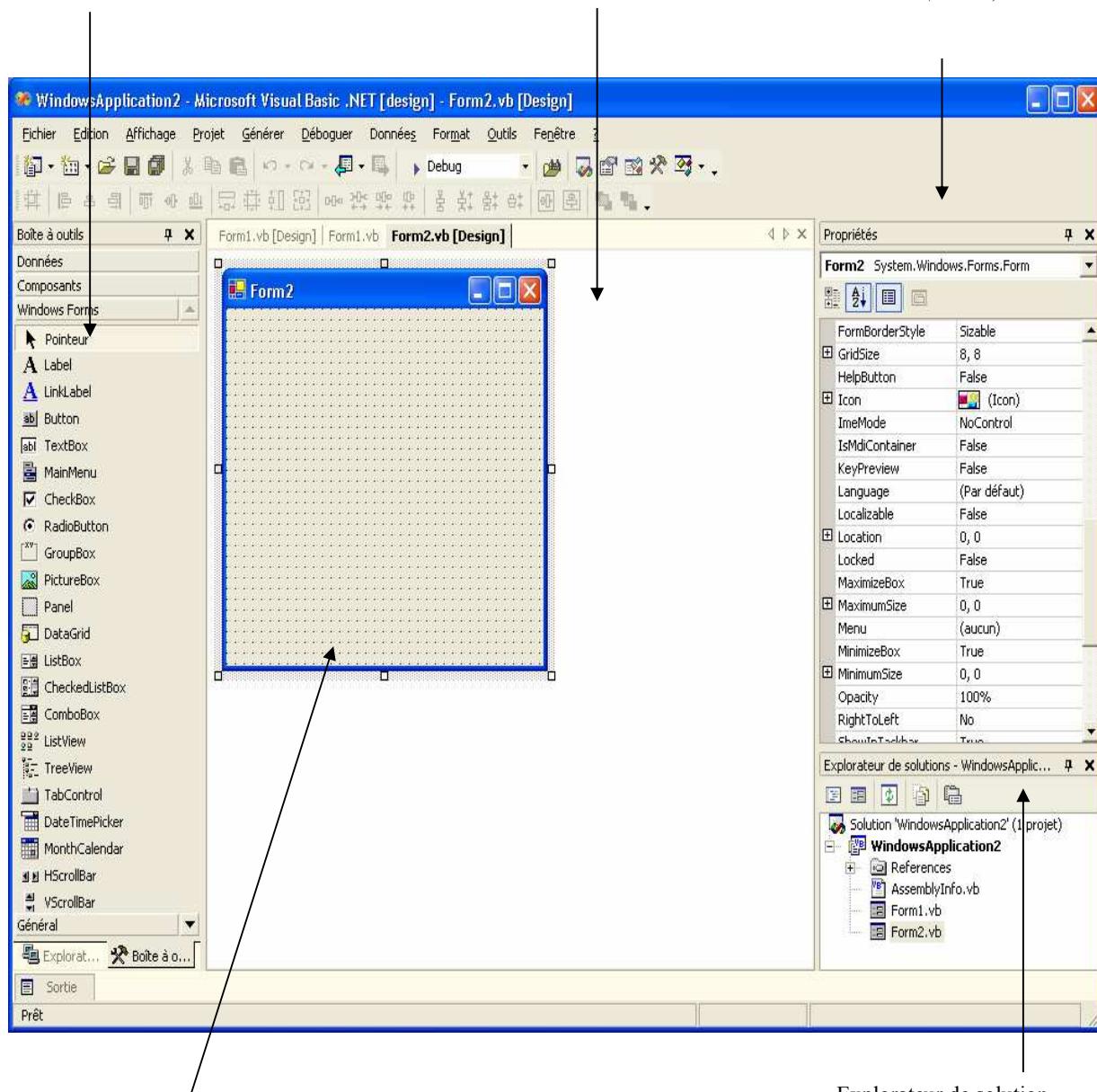
(Non active = barre grise)

Espace de travail VB

Fenêtre des propriétés

*Propriétés de l'objet*

*Sélectionné (Form2)*



Un formulaire (Form) sélectionné  
De nom 'Form1'

Explorateur de solution

Figure 1: Environnement de développement Visual Basic .NET



## TP N° 1 *Création d'une Interface*

### Objectifs

Dans cet exercice vous allez découvrir l'espace de travail VB STUDIO.NET, et mettre en œuvre la méthode « Cliquez et glissez ».

### Travail Demandé :

1. Créer l'interface ci-dessous

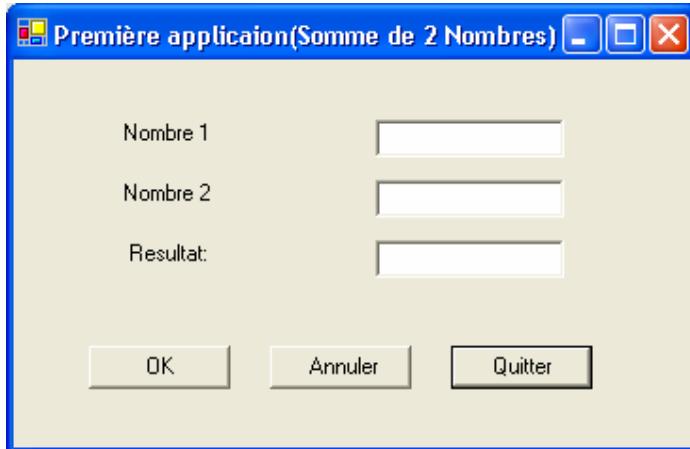


Figure 2.

2. modifier les propriétés de chaque contrôle.
3. le programme doit calculer la somme de deux nombres, écrire le code correspondant pour chaque bouton.
4. Tester votre application.

### Solution :

```
Private Sub CmdAnnuler_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles CmdAnnuler.Click
    txt1.ResetText() ' vider la zone de texte
    txt2.ResetText()
    txt3.ResetText()
    txt1.Select() ' pour donner le focus a l'objet txt1
End Sub
```

```
Private Sub Cmdquit_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Cmdquit.Click
If MsgBox("Voulez vous vraiment quitter l'application
?", MsgBoxStyle.YesNo_ Or MsgBoxStyle.Exclamation,
"Sortir!!")=MsgBoxResult.Yes Then
    End ' fin du programme
End If
End Sub
```

```
Private Sub CmdOk_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles CmdOk.Click
' On va vérifier d'abord si les zones de textes ne sont pas vides.
If txt1.Text <> Space(0) And txt2.Text <> Space(0)
Then
    txt3.Text = CInt(txt1.Text) + CInt(txt2.Text)
End If
txt1.Focus()
End Sub
```



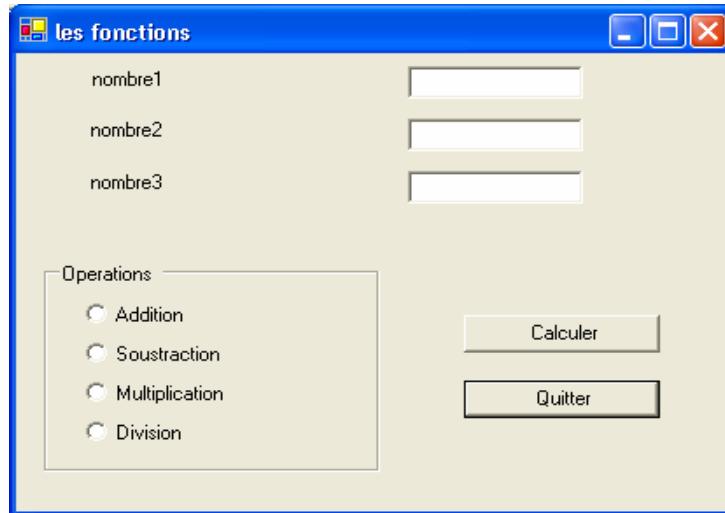
## ***TP N° 2 les fonctions***

### **Objectifs**

Dans ce TP vous allez être capable de manipuler les fonctions sous VB.NET.

### **Travail demandé :**

1. créer l'interface suivante.

**Figure 3.**

2. Modifier les propriétés de chaque contrôle.
3. créer les fonctions suivantes : addition, soustraction, multiplication, division.
4. en cliquant sur le bouton calculer, on doit avoir le résultat approprié selon la case option sélectionnée.
5. programmer le bouton quitter.
6. testez votre application.

**Solution :**

Voici une proposition pour la création des fonctions :

```
' On va passer deux arguments pour toutes les fonctions
' puisque nous avons deux nombres qui vont être manipulé.
Private Function addition(ByVal a As Integer, ByVal b As
Integer)
    Return a + b
' Dans une fonction le mot clé "RETURN" est obligatoire.
End Function

Private Function soustraction(ByVal a As Integer, ByVal b
As Integer)
    Return a - b
End Function

Private Function multiplication(ByVal a As Integer, ByVal b
As Integer)
    Return a * b
End Function

Private Function division(ByVal a As Integer, ByVal b As
Integer)
    If b = 0 Then
        MsgBox("division par ZERO impossible!!!", _
        MsgBoxStyle.Information, "ZERO!!")
    Else
```

```
        Return a / b
    End If
End Function
```

```
Private Sub cmdquitter_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles cmdquitter.Click
    Dim msg
    msg = MsgBox("voulez vous vraiment quitter
l'application ? ", MsgBoxStyle.YesNo Or MsgBoxStyle.Question,
"abdel application")
    If msg = vbYes Then
        End
    End If
End Sub
```

```
Private Sub cmdcalculer_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
cmdcalculer.Click
    If RAdd.Checked = True Then
        Call addition(CInt(txtnum1.Text),
CInt(txtnum2.Text))
    End If
    If Rsous.Checked = True Then
        Call soustraction(CInt(txtnum1.Text),
CInt(txtnum2.Text))
    End If
    If Rmulti.Checked = True Then
        Call multiplication(CInt(txtnum1.Text),
CInt(txtnum2.Text))
    End If
    If Rdiv.Checked = True Then
        Call division(CInt(txtnum1.Text),
CInt(txtnum2.Text))
    End If
End Sub
```

```
Private Sub txtnum1_TextChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
txtnum1.TextChanged
    If (txtnum1.TextLength > 0) Then
        If Not IsNumeric(txtnum1.Text) Then
            txtnum1.ResetText()
            MsgBox("veuillez entrez un nombre",
MsgBoxStyle.Information, "abdel & Aziz application")
        End If
    End If
End Sub
```

```

Private Sub txtnum2_TextChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
txtnum2.TextChanged
    If (txtnum2.TextLength > 0) Then
        If Not IsNumeric(txtnum2.Text) Then
            txtnum2.ResetText()
            MsgBox("veuillez entrez un nombre",
MsgBoxStyle.Information, "abdel & Aziz")
        End If
    End If
End Sub

```



## *TP N° 3 les Formulaires*

### **Objectifs**

Dans ce TP vous allez être capable d'instancier un formulaire.

### **Travail Demandé :**

1. créer les interfaces suivantes.

Figure 4

Figure 5

2. Modifier les propriétés de chaque contrôle.
3. Ajouter un module, dans lequel vous aller instancier form1 et form2 successivement avec les noms a et b.
4. à la clique sur le bouton « aller au form2 », du formulaire1 form2 sera afficher et form1 cacher.
5. à la clique sur le bouton « aller au form1 », du formulaire2 form1 sera afficher et form2 cacher.
6. programmer les boutons quitter des deux formes pour que l'application se ferme.
7. tester votre application.

### **Proposition de solution:**

```
Module Module1
    Public a As New Form1
    Public b As New Form2
    Public texte As String

    Sub main()
        a.Text = "Formulaire1"
        b.Text = "Formulaire2"
        Application.Run(a)
    End Sub
End Module
```

#### **Form1 :**

```
Private Sub allerauform2_Click(ByVal sender As System.Object,
    ByVal e As System.EventArgs) Handles allerauform2.Click
    texte = a.TextBox1.Text ' conserver le contenue du textbox1
                           dans la variable texte
    a.Hide() 'cacher form1
    b.Show() 'afficher form2
End Sub

Private Sub quitter_Click(ByVal sender As Object, ByVal e As
    System.EventArgs) Handles quitter.Click
    Application.Exit()
End Sub
```

#### **Form2 :**

```
Private Sub quitter_Click(ByVal sender As Object, ByVal e As
    System.EventArgs) Handles quitter.Click
    Application.Exit()
End Sub

Private Sub allerauform1_Click(ByVal sender As Object, ByVal e As
    System.EventArgs) Handles allerauform1.Click
    b.Hide()
    a.Show()
End Sub

Private Sub Form2_Activated(ByVal sender As Object, ByVal e As
    System.EventArgs) Handles MyBase.Activated
    b.TextBox2.Text = texte 'affecter la valeur de la variable
                           texte au contrôle textbox2
End Sub
```



## TP N° 4 Structures et Tableaux

### Objectifs

Dans ce TP en va voir ensemble l'utilisation des structures et des tableaux afin de remplir des listboxs.

### Travail Demandé :

1. créer les interfaces suivantes ?

**Figure 6**

**Figure 7**

2. ajouter un module dans lequel en va créer une structure client (nom, prénom, age) ?
3. déclarer un tableau du type client dans le même module ?
4. le click sur le bouton « ajouter » permettra de remplir le tableau
5. « Lister » affichera le deuxième formulaire.
6. « Afficher » du deuxième formulaire permettra de remplir la liste à partir du tableau.
7. les boutons « Quitter » et « Fermer » permettent de quitter l'application.
8. Testez votre application, trop marrant ☺.

### Proposition de solution

-Déclaration du module :

```
Module Module1

    'Création de la structure client:
    Structure client
        Dim nom As String
        Dim prenom As String
        Dim age As Integer
    End Structure

    'Déclaration d'une variable et un tableau du type client:
    Public enr As client
    Public tab() As client
    Public nbrclients As Integer = -1

    'Instanciation de form1 et form2:
    Public f1 As New Form1
    Public f2 As New Form2

    'Sub main:
    Public Sub main()
        Application.Run(f1)
    End Sub

End Module
```

```
Private Sub cmdajouter_Click(ByVal sender As System.Object,
    ByVal e As System.EventArgs) Handles cmdajouter.Click
    nbrclients += 1
    ReDim Preserve tab(nbrclients)
    With enr
        .nom = txtnom.Text
        .prenom = txtprenom.Text
        .age = CInt(txtage.Text)
    End With
    tab(nbrclients) = enr
    effacer()
End Sub
```

```
Private Sub cmdlister_Click(ByVal sender As System.Object,
    ByVal e As System.EventArgs) Handles cmdlister.Click
    f1.Hide()
    f2.Show()
End Sub
Sub effacer()
    txtnom.ResetText()
    txtprenom.ResetText()
    txtage.ResetText()
End Sub
```

```
        txtnom.Focus()
    End Sub

    Private Sub cmdeffacer_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
cmdeffacer.Click
    effacer()
End Sub

    Private Sub cmdquitter_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
cmdquitter.Click
    Dim a = MsgBox("voulez vous vraiment quitter ?",
MsgBoxStyle.YesNo Or MsgBoxStyle.Exclamation, "Sortir!!!")
    If a = vbYes Then
        End
    End If
End Sub

Public Sub Remplir(ByVal a As TextBox, ByVal b As TextBox,
ByVal c As TextBox)
    If a.Text <> Space(0) And b.Text <> Space(0) And
IsNumeric(c.Text) Then
        cmdajouter.Enabled = True
    Else
        cmdajouter.Enabled = False
    End If
End Sub

    Private Sub txtnom_TextChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
txtnom.TextChanged
    Remplir(txtnom, txtprenom, txtage)
End Sub

    Private Sub Form1_Load(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles MyBase.Load
    cmdajouter.Enabled = False
End Sub

    Private Sub txtprenom_TextChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
txtprenom.TextChanged
    Remplir(txtnom, txtprenom, txtage)
End Sub

    Private Sub txtage_TextChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
txtage.TextChanged
    Remplir(txtnom, txtprenom, txtage)
End Sub
```

```
End Sub
```



## TP N° 5 Énumération, Structure et Collection

### Objectifs

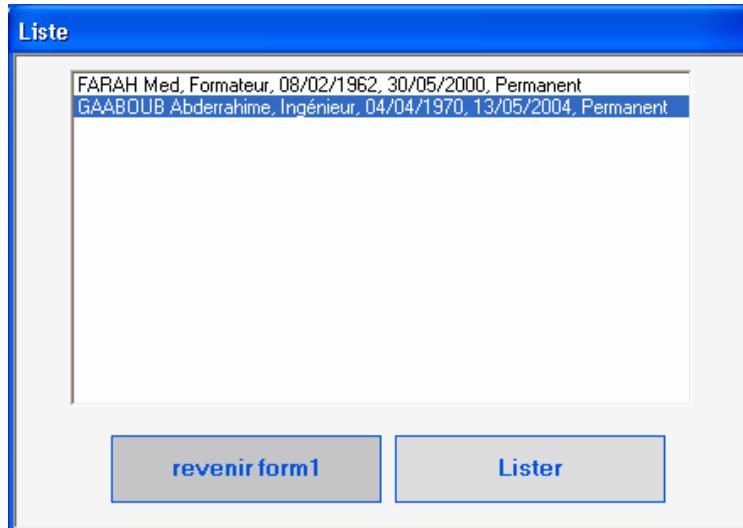
- ✓ Dans ce TP, Essayant de maîtriser l'utilisation de l'énumération, structure et collection.
- ✓ Et un nouveau contrôle, datetimepicker.

### Travail Demandé :

1. Créer les interfaces suivantes :

The screenshot shows a Windows application window titled "Saisie Employés". The window has a standard title bar with minimize, maximize, and close buttons. Inside, there are five text input fields arranged vertically. The first field is labeled "Nom Employé" and contains "GAABOUB Abderrahime". The second field is "DateNaissance" with the value "samedi 4 avril 1970". The third field is "DateEmbauche" with the value "jeudi 13 mai 2004". The fourth field is "Fonction" with the value "Ingénieur". The fifth field is "type Employé" with the value "Permanent". At the bottom of the window are four buttons: "Ajouter" (highlighted in blue), "Annuler", "Afficher", and "Quitter".

Figure 8

**Figure 9**

2. Dans un module créer une énumération TE (Permanent, Temporaire, Intermittant), et une structure Employé (Nom, DateEmbauche, DateNaissance, Fonction, TypeEmployé de type TE).
3. le combobox fonction doit être rempli à partir de l'énumération TE.
4. en cliquant sur le bouton « ajouter », les données saisies seront ajoutées dans une collection.
5. le bouton « annuler » vide les champs.
6. le bouton « Afficher » Affiche le deuxième formulaire.
7. le bouton « Lister» Affichera les données -contenues dans la collection-dans la liste.
8. « revenir form1 » permet le retour au form1.
9. programmer le bouton quitter à ta guise😊.
10. Tester votre application.

### **Proposition de solution:**

```
Module Module1
```

```
Public Structure Employé
    Dim NomEmployé As String
    Dim DateEmbauche As Date
    Dim DateNaissance As Date
    Dim Fonction As String
    Dim type As TE
End Structure
```

```
Public Enum TE
    Permanent = 0
    Temporaire = 1
    Intermittant = 2
End Enum
```

```

Function message() As MsgBoxResult
    Return MessageBox.Show("Voulez vous vraiment
sortir???", _
    "Sortir!!!!", MessageBoxButtons.YesNo, _
    MessageBoxIcon.Exclamation)
End Function

Public f2 As New Form2
Public f1 As New Form1
Public col As New Collection
Public emp As Employé

End Module

```

**Form1:**

```

Private Sub cmdquitter_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles cmdquitter.Click
    If message() = MsgBoxResult.Yes Then
        End
    End If
End Sub

Private Sub CmdAjouter_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles CmdAjouter.Click

    With emp
        .DateEmbauche = DateTimePicker2.Value.Date
        .DateNaissance = DateTimePicker1.Value.Date
        .NomEmployé = txt1.Text
        .Fonction = txt2.Text
        .type = ComboBox1.SelectedItem
    End With
    Col.add(emp)
    vider()
End Sub

public Sub Vider()
    Me.txt1.ResetText()
    Me.txt2.ResetText()
    Me.DateTimePicker1.ResetText()
    Me.DateTimePicker2.ResetText()
    txt1.Focus()
End Sub

Private Sub CmdAfficher_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
CmdAfficher.Click

```

```
f2.Show()
Me.Hide()
End Sub

Private Sub Form1_Load(ByVal sender As Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    ComboBox1.Items.Add(TE.Permanent)
    ComboBox1.Items.Add(TE.Temporaire)
    ComboBox1.Items.Add(TE.Intermittant)
    ComboBox1.SelectedIndex = 1
End Sub

Private Sub CmdAnnuler_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles CmdAnnuler.Click
    Vider()
End Sub

Private Sub Form1_Closing(ByVal sender As Object, ByVal e As
System.ComponentModel.CancelEventArgs) Handles
MyBase.Closing
    If message() = MsgBoxResult.No Then
        e.Cancel = True
    End If
End Sub
```

## **Form2:**

```
Private Sub Button1_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button1.Click
    f1.Show()
    f2.Hide()
End Sub

Private Sub Button2_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button2.Click
    ListBox1.Items.Clear()
    For Each emp In col
        Me.ListBox1.Items.Add(emp.NomEmployé & ", " &
emp.Fonction & ", " & emp.DateNaissance
        & ", " & emp.DateEmbauche.Date & ", " &
emp.type.ToString())
    Next
End Sub
```



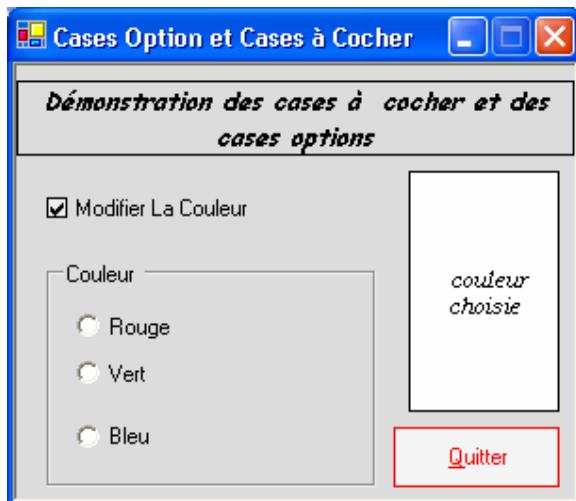
## TP N° 6 Les événements, les Cases à cocher et les cases options

### Objectifs

Dans ce TP on va voir ensemble l'utilisation des cases options et des cases à cocher afin de préciser quelques événements.

### Travail Demandé :

1. créer l'interface suivant.



**Figure 10**

2. si la case à cocher « Modifier Le Couleur » est inactive, le groupbox « couleur » doit être invisible.
3. « couleur choisie » doit prendre la couleur sélectionnée dans le groupbox.
4. sur l'événement MouseMove, « couleur choisie » deviendra blanche. Et sur l'événement MouseLeave, prendra la couleur sélectionnée.
5. avant de quitter l'application demandera une confirmation (utiliser l'événement closing du formulaire).
6. tester votre application.

### Proposition de solution :

```
Dim a As System.Drawing.Color = Color.White
-----
Function sortir() As MsgBoxResult
    Return MessageBox.Show("voulez vous vraiment quitter
    ???", "Quitter!!!!",
```

```
    MessageBoxButtons.YesNo, MessageBoxIcon.Question)
End Function

Private Sub cmdquitter_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
cmdquitter.Click
    If sortir() = MsgBoxResult.Yes Then
        End
    End If
End Sub

Private Sub Form1_Load(ByVal sender As System.Object,
 ByVal e As System.EventArgs) Handles MyBase.Load
    fracouleur.Visible = False
End Sub

Private Sub chkmodifier_CheckedChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
chkmodifier.CheckedChanged
    fracouleur.Visible = chkmodifier.Checked
    If Not (chkmodifier.Checked) Then
        optbleu.Checked = chkmodifier.Checked
        optrouge.Checked = chkmodifier.Checked
        optvert.Checked = chkmodifier.Checked
        lblcouleur.BackColor = Color.White
    End If
    a = Color.White
End Sub

Private Sub cole(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles optrouge.CheckedChanged,
optbleu.CheckedChanged, optvert.CheckedChanged
    If sender Is optbleu Then
        a = Color.Blue
    ElseIf sender Is optrouge Then
        a = Color.Red
    ElseIf sender Is optvert Then
        a = Color.Green
    End If
    lblcouleur.BackColor = a
End Sub

Private Sub lblcouleur_MouseMove(ByVal sender As Object,
 ByVal e As System.Windows.Forms.MouseEventArgs) Handles
lblcouleur.MouseMove
    lblcouleur.BackColor = Color.White
End Sub
```

```
Private Sub lblcouleur_MouseLeave(ByVal sender As Object,  
ByVal e As System.EventArgs) Handles lblcouleur.MouseLeave  
    lblcouleur.BackColor = a  
End Sub
```

```
Private Sub Form1_Closing(ByVal sender As Object, ByVal e  
As System.ComponentModel.CancelEventArgs) Handles  
 MyBase.Closing  
    If sortir() = MsgBoxResult.No Then  
        e.Cancel = True  
    End If  
End Sub
```



## ***TP N° 7 L'événement Closing et regroupement des contrôles***

### **Objectifs**

Dans ce TP on va voir ensemble comment une procédure répond à plusieurs contrôles.

### **Travail Demandé :**

1. créer l'interface ci-dessous.

**Figure 11**

2. créer une fonction qui calcule la remise tel que la remise est de 0% pour une quantité de livres inférieure à 5, et 5% pour une quantité entre 5 et 10, et 8% pour une quantité entre 10 et 20, et une remise de 10% pour une quantité supérieure à 20.
3. programmer le bouton « RAZ » pour que les zones de texte reçoivent la valeur 0.
4. programmer le bouton quitter à ta manière amigos☺.
5. tester votre application.

### Proposition de solution :

```
Private Sub typecollection_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles h.CheckedChanged, s.CheckedChanged
    If sender Is h Then
        sel(Qteh, h)
    ElseIf sender Is s Then
        sel(Qtes, s)
    End If
End Sub
```

```
Private Sub sel(ByVal a As TextBox, ByVal b As CheckBox)
    a.Visible = b.Checked
    a.Text = 0
    a.Focus()
    a.SelectionStart = 0
```

```
End Sub
```

```
Public Function calcrem(ByVal qte As Integer) As Single
Dim r As Single 'une variable dans laquelle on va calculer la
                 remise
    If qte >= 20 Then
        r = 10 / 100
    ElseIf qte >= 10 Then
        r = 8 / 100
    ElseIf qte >= 5 Then
        r = 5 / 100
    Else
        r = 0
    End If
    Return r
End Function
```

```
Private Sub Cmd_Raz_Click(ByVal sender As System.Object,
                           ByVal e As System.EventArgs) Handles Cmd_Raz.Click
    remise.ResetText()
    Qteh.ResetText()
    Qtes.ResetText()
    map.ResetText()
    remise.Text = 0
    Qteh.Text = 0
    Qtes.Text = 0
    map.Text = 0
    Qtes.Visible = False
    Qteh.Visible = False
    h.Checked = False
    s.Checked = False
End Sub
```

```
Private Sub Cmd_Calculer_Click(ByVal sender As
                               System.Object, ByVal e As System.EventArgs) Handles
                               Cmd_Calculer.Click
    Dim remis As Single
    Dim mnt As Single
    If h.Checked = True Then
        remise.Text = calcrem(CInt(Qteh.Text))
        mnt = 100 * CInt(Qteh.Text)
    End If
    If s.Checked = True Then
        remise.Text = calcrem(CInt(Qtes.Text))
        mnt = 120 * CInt(Qtes.Text)
    End If
    If h.Checked And s.Checked Then
        remise.Text = calcrem(CInt(Qteh.Text)) +
CInt(Qtes.Text))
    End If
End Sub
```

```
mnt = 120 * CInt(Qtes.Text) + 100 *  
CInt(Qteh.Text)  
End If  
map.Text = mnt * (1 - remise.Text)  
End Sub  
  
Private Sub Form1_Closing(ByVal sender As Object, ByVal e  
As System.ComponentModel.CancelEventArgs) Handles  
 MyBase.Closing  
    Dim a = MsgBox("voulez vous vraiment nous quitter ?",  
    MsgBoxStyle.YesNo, or MsgBoxStyle.Information, "Quitter..!!")  
    If a = vbYes Then  
        e.Cancel = False  
    Else : e.Cancel = True  
    End If  
End Sub  
  
Private Overloads Sub Cmd_quitter_Click(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
Cmd_quitter.Click  
    Dim a = MsgBox("voulez vous vraiment nous quitter ?",  
    MsgBoxStyle.YesNo or MsgBoxStyle.Information, "Quitter..!!!!")  
    If a = vbYes Then  
        End  
    End If  
End Sub
```



## ***TP N° 8 Listbox et l'événement enabled***

### **Objectifs**

Dans ce TP on va utiliser pas mal de propriétés du contrôle listbox, et la manipulation de la propriété enabled des boutons.

### **Travail Demandé :**

1. créer l'interface ci-dessous.

**Figure 12**

2. au démarrage du formulaire tous les bouton seront désactivés.
3. au clique sur le bouton « → » les éléments sélectionnés dans la liste se trouve à gauche seront déplacer dans la liste à droite et l'inverse pour le bouton « ← ».
4. les deux boutons « effacer » permet de supprimer les éléments sélectionnés dans la liste située au dessus du bouton.
5. tester votre application.

### **Proposition de solution :**

```

Private Sub cmdajouter_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles cmdajouter.Click
    List1.Items.Add(txt.Text)
    txt.Clear()
    txt.Select()
End Sub

Private Sub Form1_Load(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles MyBase.Load
    txt.Select()
End Sub

Private Sub r_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles r.Click
    For i As Integer = 0 To List1.SelectedItems.Count - 1
        List2.Items.Add((List1.SelectedItem))
        List1.Items.Remove(List1.SelectedItem)
    Next i
    r.Enabled = List1.Items.Count
    f1.Enabled = List1.Items.Count
End Sub

```

```
-----  
Private Sub txt_TextChanged(ByVal sender As System.Object,  
    ByVal e As System.EventArgs) Handles txt.TextChanged  
    cmdajouter.Enabled = txt.TextLength  
End Sub
```

```
-----  
Private Sub List1_SelectedIndexChanged(ByVal sender As  
    System.Object, ByVal e As System.EventArgs) Handles  
    List1.SelectedIndexChanged  
    r.Enabled = List1.Items.Count  
    f1.Enabled = List1.Items.Count  
    List1.SelectionMode = SelectionMode.MultiExtended  
End Sub
```

```
-----  
Private Sub l_Click(ByVal sender As System.Object, ByVal e  
    As System.EventArgs) Handles l.Click  
    For i As Integer = 0 To List2.SelectedItems.Count - 1  
        List1.Items.Add((List2.SelectedItem))  
        List2.Items.Remove(List2.SelectedItem)  
    Next i  
    l.Enabled = List2.Items.Count  
    f2.Enabled = List2.Items.Count  
End Sub
```

```
-----  
Private Sub List2_SelectedIndexChanged(ByVal sender As  
    System.Object, ByVal e As System.EventArgs) Handles  
    List2.SelectedIndexChanged  
    List2.SelectionMode = SelectionMode.MultiExtended  
    l.Enabled = List2.Items.Count  
    f2.Enabled = List2.Items.Count  
End Sub
```

```
-----  
Private Sub f1_Click(ByVal sender As System.Object, ByVal  
    e As System.EventArgs) Handles f1.Click  
    For i As Integer = 0 To List1.SelectedItems.Count - 1  
        List1.Items.RemoveAt(List1.SelectedIndex)  
    Next i  
    f1.Enabled = List1.Items.Count  
    r.Enabled = List1.Items.Count  
End Sub
```

```
-----  
Private Sub f2_Click(ByVal sender As System.Object, ByVal  
    e As System.EventArgs) Handles f2.Click  
    For i As Integer = 0 To List2.SelectedItems.Count - 1  
        List2.Items.RemoveAt(List2.SelectedIndex)  
    Next i  
    f2.Enabled = List2.Items.Count  
    l.Enabled = List2.Items.Count  
End Sub
```

```
Private Sub txt_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles txt.KeyPress
    If AscW(e.KeyChar) = 13 Then
        List1.Items.Add(txt.Text)
        txt.Clear()
        txt.Select()
    End If
End Sub
```



## TP N° 9 La boucle For...Next

### Objectifs

Dans ce TP vous allez utiliser la boucle for afin de rechercher un caractère dans une chaîne de texte.

### Travail Demandé :

1. créer l'interface ci-dessous.



Figure 13

2. programmer le bouton quitter premièrement ☺.
3. le bouton « Lancer la recherche » permet de compter le nombre de fois se répète un caractère dans un texte.
4. tester votre application.

### Une Solution

```
Private Sub Rechercher_Click(ByVal sender As System.Object,
    ByVal e As System.EventArgs) Handles Rechercher.Click
    Dim a As Integer = 0
    Dim longTexte As Integer = TextBox1.TextLength
    Try
        For i As Integer = 0 To longTexte - 1
            If UCase(TextBox1.Text.Chars(i)) =
    UCase(TextBox2.Text.Chars(0)) Then
                a = a + 1
            End If
    Next
    TextBox3.Text = a.ToString
    Catch ex As Exception
        MsgBox("Introduire du texte SVP!!!!",
    MsgBoxStyle.Information, "Erreur!")
    End Try
End Sub
```

```
Private Sub cmd_Quitter_Click(ByVal sender As Object, ByVal
    e As System.EventArgs) Handles cmd_Quitter.Click
    End
End Sub
```



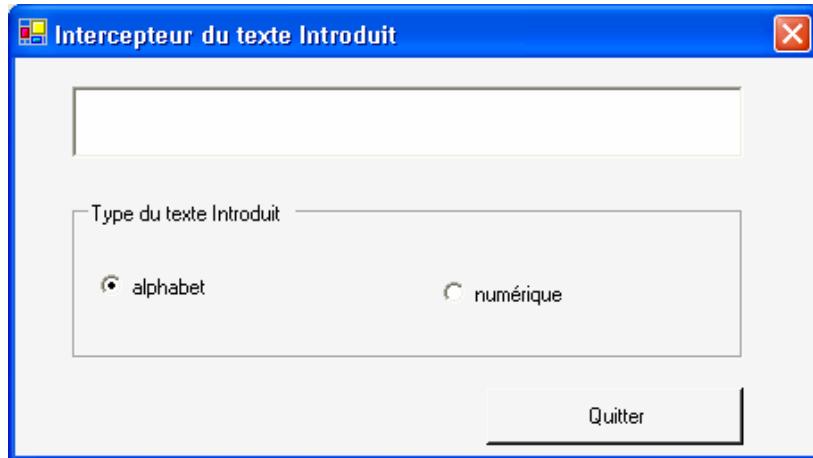
## ***TP N° 10 L'événement keypress***

### **Objectifs**

Dans ce TP vous allez utiliser l'événement keypress d'un contrôle textbox.

### **Travail Demandé :**

1. créer l'interface ci-dessous.

**Figure 14**

2. l'utilisateur ne doit introduire que le type du texte sélectionné parmi les deux boutons radio « numérique » et « alphabet ».
3. programmer le bouton « quitter » pour sortir de votre vaste application ☺.
4. tester votre application.

### **Proposition de solution :**

```

Private Sub alphabet_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles alphabet.CheckedChanged
    TextBox1.Focus()
End Sub

Private Sub TextBox1_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles TextBox1.KeyPress
    If alphabet.Checked = True Then
        If Char.IsDigit(e.KeyChar) Then
            e.Handled = True
        End If
    End If
    If numérique.Checked = True Then
        If Char.IsLetter(e.KeyChar) Then
            e.Handled = True
        End If
    End If
End Sub

Private Sub numérique_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles numérique.CheckedChanged
    TextBox1.Focus()
End Sub

```

```
Private Sub quitter_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles quitter.Click
    Me.Close()
End Sub
```

```
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
    Me.MaximizeBox = False
    Me.MinimizeBox = False
    Me.ShowInTaskbar = False
End Sub
```



## *TP N° 11 Liste sélection*

### Objectifs

Dans ce TP vous allez connaître les différents types de sélection dans une listbox.

### Travail Demandé :

1. créer l'interface ci-dessous.

The screenshot shows a Windows application window titled "Listes sélection". It features two list boxes: "Liste de gauche" on the left containing items like Beni mellal, Tadla, Fes, Tanger, Azilal, Fkikh Ben Salah, Rabat, Casa, and Agadir; and "Liste de droite" on the right containing Beni mellal, Agadir, and Tanger. Below the list boxes are two buttons: "remplir liste" and "élément Selectionnés". Underneath these buttons is a "selection mode" section with four radio buttons: "MultiExtended", "None", "MultiSimple" (selected), and "One". At the bottom left is a checkbox labeled "Multicolonne" and at the bottom right is a signature "By Abdel / Aziz".

Figure 15

2. en cliquant sur le bouton « remplir liste », la liste à gauche sera remplie par des éléments de votre choix, pour nous c'est une liste des villes marocaines.
3. le bouton élément sélectionné permet d'ajouter le ou les éléments sélectionnés ; dans la liste ; au textbox (RTF).
4. le type de sélection dans la liste et celui de l'option sélectionnée dans « sélection mode » :
  - None** : pas de sélection permise.
  - One** : sélection d'un élément à la fois.
  - MultiSimple** : Plusieurs éléments peuvent être sélectionnés.
  - MultiExtended** : Plusieurs éléments peuvent être sélectionnés et tu peux utiliser les touches MAJ et CTRL, ainsi que les touches de direction, pour effectuer les sélections.
5. la case à cocher « Multicolonne » permet à la liste de prendre en charge les colonnes multiples.
6. tester votre application, à ta santé.

### **Proposition de solution :**

```

Private Sub remplir_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles remplir.Click
    Dim i As Integer
    List.Items.Clear()
    Dim tableau() As String = {"Beni mellal", "Tadla",
    "Fes", "Tanger", "Azilal", "Fkih Ben Salah", "Rabat", "Casa",
    "Agadir"}
    List.Items.AddRange(tableau)
End Sub

Private Sub multicolonne_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles multicolonne.CheckedChanged
    If CheckBox1.Checked = True Then
        List.MultiColumn = True
    Else
        List.MultiColumn = False
    End If
End Sub

Private Sub MultiExtended_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MultiExtended.CheckedChanged
    If RadioButton1.Checked = True Then
        List.SelectionMode = SelectionMode.MultiExtended
    End If
End Sub

```

```

Private Sub elementsélectionné_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
elementsélectionné.Click

    For i As Integer = 0 To List.SelectedItems.Count - 1
        Text1.Text += List.SelectedItems(i) & vbCrLf
    Next

End Sub

Private Sub MultiSimple _CheckedChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
MultiSimple.CheckedChanged
    If RadioButton2.Checked = True Then
        List.SelectionMode = SelectionMode.MultiSimple
    End If
End Sub

Private Sub One _CheckedChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
One.CheckedChanged
    If RadioButton4.Checked = True Then
        List.SelectionMode = SelectionMode.One
    End If
End Sub

Private Sub None _CheckedChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
None.CheckedChanged
    If RadioButton3.Checked = True Then
        List.SelectionMode = SelectionMode.None
    End If
End Sub

```



## *TP N° 12 ArrayList, Classe*

### Objectifs

Dans ce TP vous allez découvrir les fonctionnalités de ArrayList et aussi l'utilisation d'une classe.

### Travail Demandé :

1. créer l'interface ci-dessous :

The screenshot shows a Windows application window titled "Gestion de cyber". The interface is in French. It has several input fields and dropdown menus:

- N°règlement:** 1
- date:** 10/05/2006
- client:** Leuf Aziz Nour
- type client:** abonné
- N°poste:** 1
- montant:** 48
- service utilisé:** visioconférence
- nombre d'heure:** 3

Below the input fields are several buttons:

- Quitter** (Quit)
- enregistrer**
- nouveau**
- supprimer**
- précédent**
- premier**
- modifier**
- suivant**
- dernier**
- chercher**

**Figure 16**

Cette application a pour but la gestion des règlements. Elle permet entre autres le calcul du montant que doit payer le client après une séance de navigation.

Il existe trois types de clients :

- ➡ Etudiant : il paye 15 DH /heure
- ➡ Fonctionnaire : il paye 20 DH /heure
- ➡ Abonné : il paye 6 DH /heure

Le Cyber café offre trois services :

- ➡ Navigation : 0DH de plus pour chaque client /heure
- ➡ Visioconférence : 20DH de plus pour chaque client /heure
- ➡ Navigation Assistance : 10DH de plus pour chaque client /heure

2. créer une classe comprenant les éléments de l'interface comme attributs et créer les méthodes et les propriétés.
3. créer une fonction qui calcule le montant.
4. le montant sera calculer automatiquement et afficher dans la zone de montant après avoir remplir la zone nombre d'heures et sélectionner le type client et le service utilisé.
5. Programmer les boutons «nouveau», «précédent», «suivant», «premier», «dernier».
6. écrire le code des boutons modifier, supprimer et chercher par le N° du règlement.
7. programmer le bouton quitter pour mettre fin l'application.

```
Public Class Gestion
    'Déclaration des attributs de type privé.
    Private m_nreg As Integer
    Private m_client As String
    Private m_typeclient As String
    Private m_nump As Integer
    Private m_serut As String
    Private m_nbrh As Integer
    Private m_date As Date

    'Création du constructeur
    Sub New(ByVal a As Integer, ByVal b As String, ByVal c As
String, ByVal d As Integer, ByVal e As String, ByVal f As
Integer, ByVal g As Date)
        m_nreg = a : m_client = b : m_typeclient = c : m_nump = d
        m_serut = e : m_nbrh = f : m_date = g
    End Sub

    'création des propriétés pour l'accès aux attributs.
    Public Property nreg() As Integer
        Get
            Return m_nreg
        End Get
        Set(ByVal Value As Integer)
            m_nreg = Value
        End Set
    End Property

    Public Property client() As String
        Get
            Return m_client
        End Get
        Set(ByVal Value As String)
            m_client = Value
        End Set
    End Property

    Public Property typeclient() As String
        Get
            Return m_typeclient
        End Get
        Set(ByVal Value As String)
            m_typeclient = Value
        End Set
    End Property

    Public Property nump() As Integer
        Get
            Return m_nump
        End Get
    End Property
```

```

        Set(ByVal Value As Integer)
            m_nump = Value
        End Set
    End Property

    Public Property serut() As String
        Get
            Return m_serut
        End Get
        Set(ByVal Value As String)
            m_serut = Value
        End Set
    End Property

    Public Property nbrh() As Integer
        Get
            Return m_nbrh
        End Get
        Set(ByVal Value As Integer)
            m_nbrh = Value
        End Set
    End Property

    Public Property datee() As Date
        Get
            Return m_date
        End Get
        Set(ByVal Value As Date)
            m_date = Value
        End Set
    End Property

End Class

```

Form1 :

```

Public gest As Gestion
Public pos As Integer
Public list As New ArrayList

Private Sub Form1_Load(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles MyBase.Load
    Dim tab() as string={"etudiant","fonctionnaire",
    "abonné"}
    tt.Items.AddRange(tab)

    tab={"navigation","visioconférence","navigation assistance"}
    tsu.Items.AddRange(tab)
    tt.SelectedIndex = 0
    tsu.SelectedIndex = 0

```

```
End Sub

Sub calcul()
    Dim t As Single
    Select Case tt.Text
        Case "etudiant"
            t = 15
        Case "fonctionnaire"
            t = 20
        Case "abonné"
            t = 6
    End Select
    Select Case tsu.Text
        Case "navigation"
            tm.Text = (t * CDbl(tnh.Text)).ToString
        Case "visioconférence"
            tm.Text = CDbl(t * tnh.Text + (tnh.Text * 10))
        Case "navigation assistance"
            tm.Text = CDbl(t * tnh.Text + tnh.Text * 20)
    End Select
End Sub

Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button3.Click
    raz()
End Sub

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    Dim i As Integer = 0
    If MsgBox("Voulez vous vraiment Supprimer l'enregistrement courant??!!", MsgBoxStyle.YesNo Or MsgBoxStyle.Question, "Confirmation de Suppression...?") = MsgBoxResult.Yes Then
        For Each gest In list 'col
            If gest.nreg = tn.Text Then
                list.RemoveAt(i)
                Try
                    lire(i)
                    pos = i
                Catch ex As Exception
                    If list.Count < 1 Then
                        raz()
                    Else
                        lire(i - 1)
                        pos = i - 1
                    End If
                End Try
                Exit Sub
            End If
        End For
    End If
End Sub
```

```
i += 1
    Next
End If
End Sub

Sub raz()
    tn.Text = ""
    tc.Text = ""
    tt.SelectedIndex = 0
    tn.Text = ""
    tsu.SelectedIndex = 0
    tnh.Text = ""
    td.Text = ""
    tnp.Text = ""
    tn.Select()
End Sub

Private Sub tt_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles tt.SelectedIndexChanged
    If tt.Text <> "" And tsu.Text <> "" And IsNumeric(tnh.Text) Then
        calcul()
    Else
        tm.Text = ""
    End If
End Sub

Private Sub tsu_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles tsu.SelectedIndexChanged
    If tt.Text <> "" And tsu.Text <> "" And IsNumeric(tnh.Text) Then
        calcul()
    Else
        tm.Text = ""
    End If
End Sub

Public Sub remplirzones()
    tn.Text = gest.nreg
    tc.Text = gest.client
    tt.Text = gest.typeclient
    tn.Text = gest.nump
    tsu.Text = gest.serut
    tnh.Text = gest.nbrh
    td.Text = gest.datee
End Sub
```

```

Private Sub tnh_TextChanged(ByVal sender As System.Object,
    ByVal e As System.EventArgs) Handles tnh.TextChanged
    If tt.Text <> "" And tsu.Text <> "" And
    IsNumeric(tnh.Text) Then
        calcul()
    Else
        tm.Text = ""
    End If
End Sub

Private Sub enregistrer_Click(ByVal sender As
    System.Object, ByVal e As System.EventArgs) Handles
    enregistrer.Click
    For Each gest In list
        If gest.nreg = CInt(tn.Text) Then
            MsgBox("cet enregistrement " & tn.Text & " existe déjà",
                MsgBoxStyle.Information, "Enregistrer...")
            Exit Sub
        End If
    Next
    gest = New Gestion(CType(tn.Text, Integer), tc.Text,
        tt.Text, CType(tnp.Text, Integer), tsu.Text,
        CType(tnh.Text, Integer), CType(td.Text, Date))
    list.Add(gest)
    pos = list.Count - 1
End Sub

Private Sub Modifier_Click(ByVal sender As System.Object,
    ByVal e As System.EventArgs) Handles Modifier.Click
    Dim i As Integer = 0
    If MsgBox("Voulez vous vraiment modifier l'enregistrement courant??!!",
        MsgBoxStyle.YesNo Or MsgBoxStyle.Question, "Confirmation de Modification...?") =
    MsgBoxResult.Yes Then
        For Each gest In list
            If gest.nreg = tn.Text Then
                gest = New Gestion(CType(tn.Text, Integer),
                    tc.Text, tt.Text, CType(tnp.Text, Integer),
                    tsu.Text, CType(tnh.Text, Integer), CType(td.Text, Date))
                Try
                    list.RemoveAt(i)
                    If list.Count < 1 Then
                        list.Add(gest)
                    Else
                        list.Insert(i, gest)
                    End If
                Catch ex As Exception
                    MsgBox(ex.Message, MsgBoxStyle.Information)
                End Try
                Exit For
            End If
        Next
    End If
End Sub

```

```
        End If
        i += 1
    Next
End If
End Sub

Sub lire(ByVal p As Integer)
    gest = CType(list.Item(p), Gestion)
    remplirzones()
End Sub

Private Sub Premier_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Premier.Click
    pos = 0
    lire(pos)
End Sub

Private Sub Dernier_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Dernier.Click
    pos = list.Count - 1
    lire(pos)
End Sub

Private Sub Suivant_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Suivant.Click
    Try
        pos += 1
        lire(pos)
    Catch ex As Exception
        pos -= 1
    End Try
End Sub

Private Sub Précédent_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Précédent.Click
    Try
        pos -= 1
        lire(pos)
    Catch ex As Exception
        pos += 1
    End Try
End Sub

Private Sub Chercher_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Chercher.Click
    Dim i As Integer
    Dim a As Integer = InputBox("entrer le code à
rechercher:", "Recherecher...")
    For Each gest In list
        If gest.nreg = a Then
```

```

        Remplirzones()
        pos = i
        Exit Sub
    End If
    i += 1
Next
MsgBox("l'élément recherché n'existe pas ????!!!!",
       MsgBoxStyle.Critical, "Recherecher....")
End Sub

```

```

Private Sub Quitter_Click(ByVal sender As Object, ByVal e As
System.Windows.Forms.MouseEventArgs) Handles Quitter.Click
    MsgBoxStyle.Question Or MsgBoxStyle.YesNo, "Sortir de
l'application") = MsgBoxResult.Yes Then
        Application.exit()
    End If
End Sub

```



## *TP N° 13 tabcontrol*

### Objectifs

Dans ce TP on va mettre en œuvre « tabcontrol » et ses astuces.

1. Créer l'interface ci-dessous :

Figure 17

**Figure 18**

2. écrire une procédure qui permet d'intercepter l'écriture des lettres dans les zones de texte.
3. « ok » permet de calculer la somme des deux nombres et le produit des deux nombre dans la feuille « produit ».
4. « Annuler » permet de vider les zones de texte.
5. « Quitter » pour sortir de l'application.

**Solution :**

```

Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button3.Click
    Dim reponse As Integer
    reponse = MessageBox.Show("Voulez vous vraiment quitter l'application", "TSDI1", MessageBoxButtons.YesNo, MessageBoxIcon.Question)
    If reponse = vbYes Then
        End
    End If
End Sub

Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button4.Click
    Dim reponse As Integer
    reponse = MessageBox.Show("Voulez vous vraiment quitter l'application", "TSDI1", MessageBoxButtons.YesNo, MessageBoxIcon.Question)
    If reponse = vbYes Then
        End
    End If
End Sub

```

```
Sub intercepter(ByVal e As System.Windows.Forms.KeyPressEventArgs)
    If Char.IsDigit(e.KeyChar) Or AscW(e.KeyChar) = 8 Or
e.KeyChar = "," Then
        e.Handled = False
    Else
        e.Handled = True
    End If
End Sub
```

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    Me.TextBox3.Text = CInt(Me.TextBox1.Text) +
CInt(Me.TextBox2.Text)
End Sub
```

```
Private Sub TabPage2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TabPage1.Click
    Me.TextBox1.Text = Me.TextBox4.Text
    Me.TextBox2.Text = Me.TextBox5.Text
End Sub
```

```
Private Sub TextBox1_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles TextBox1.KeyPress
    Call intercepter(e)
End Sub
```

```
Private Sub TextBox2_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles TextBox2.KeyPress
    Call intercepter(e)
End Sub
```

```
Private Sub TextBox6_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles TextBox6.KeyPress
    intercepter(e)
End Sub
```

```
Private Sub TextBox5_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles TextBox5.KeyPress
    Call intercepter(e)
End Sub
```

```
Private Sub TabControl1_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TabControl1.SelectedIndexChanged
```

```
        TextBox6.Text = TextBox1.Text
        TextBox5.Text = TextBox2.Text
        TextBox4.Text = ""
    End Sub

    Private Sub Button6_Click(ByVal sender As System.Object,
    ByVal e As System.EventArgs) Handles Button6.Click
        Me.TextBox4.Text = CInt(Me.TextBox5.Text) *
        CInt(Me.TextBox6.Text)
    End Sub

    Private Sub Button5_Click(ByVal sender As System.Object,
    ByVal e As System.EventArgs) Handles Button5.Click
        Call anul(TextBox4, TextBox5, TextBox6)
    End Sub

    Public Sub anul(ByVal a As TextBox, ByVal b As TextBox, ByVal
    c As TextBox)
        a.ResetText()
        b.ResetText()
        c.ResetText()
    End Sub

    Private Sub Button2_Click(ByVal sender As System.Object,
    ByVal e As System.EventArgs) Handles Button2.Click
        Call anul(TextBox1, TextBox2, TextBox3)
    End Sub
```



## TP N° 14

### Richtextbox, ContextMenu, ColorDialog, FontDialog

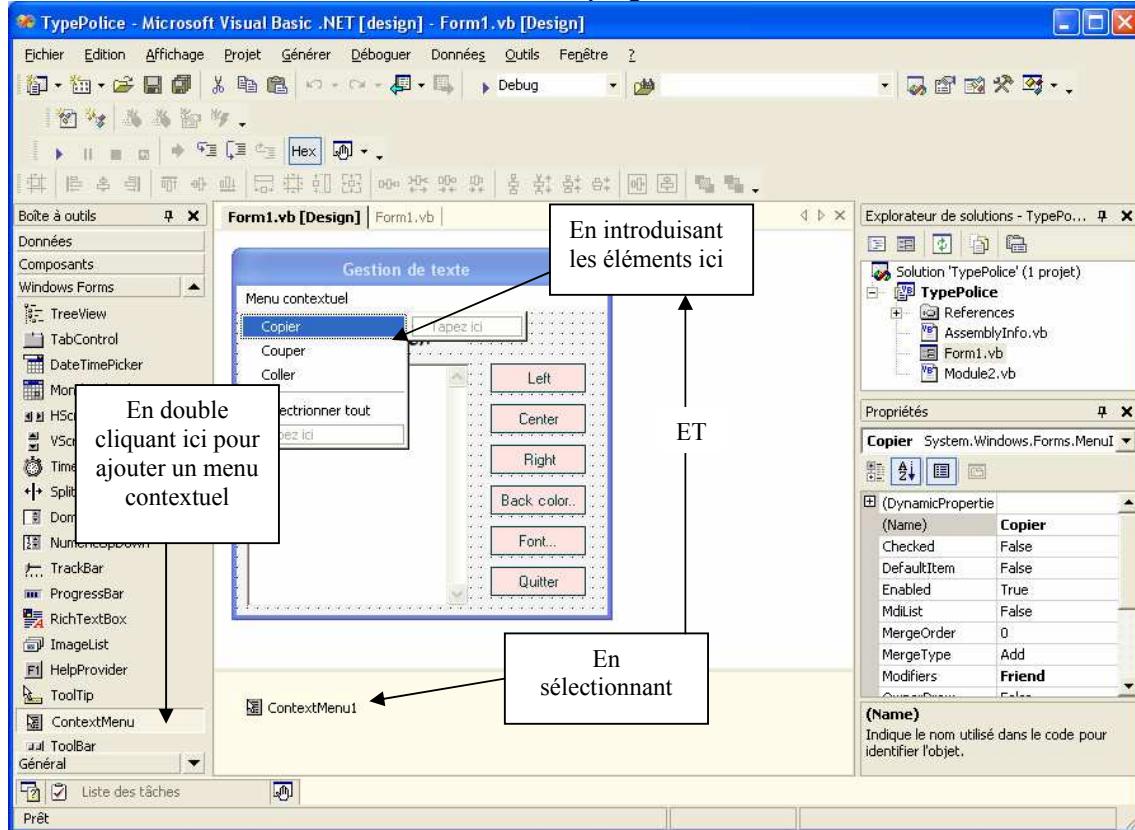
#### Objectifs

Dans ce TP on va voir ensemble le fonctionnement de la classe FontDialog, ColorDialog et CotextMenu. Et on va mettre les points sur quelques propriétés du contrôle RichTextBox.

1. Créer l'interface ci-dessous :

Figure 19

2. Créer un menu contextuel contenant les éléments : « Copier », « Couper », « Coller » et « Sélectionner tout » et programmer Chacun d'eux.

Figure 20

3. programmer chaque bouton à ta guise☺.
4. tester votre application.

**Solution :**

```
Private Sub BackColor_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles BackColor.Click
    Dim a As New ColorDialog
    a.AnyColor = True
    a.FullOpen = True
    a.ShowHelp = True
    a.Color = Texte.BackColor
    If a.ShowDialog = DialogResult.OK Then
        Texte.BackColor = a.Color
    End If
End Sub
```

```
Private Sub Font_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Fontt.Click
    Dim a As New FontDialog
    a.ShowApply = True
    a.ShowColor = True
    a.ShowHelp = True
    a.ShowEffects = True
    a.AllowVerticalFonts = True
    a.Font = Texte.SelectionFont
    a.Color = Texte.SelectionColor
    If (a.ShowDialog() = DialogResult.OK) Then
        Texte.SelectionFont = a.Font
        Texte.SelectionColor = a.Color
    End If
End Sub
```

```
Private Sub Centre_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Centre.Click
    Texte.SelectionAlignment = HorizontalAlignment.Center
End Sub
```

```
Private Sub Left_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Leftt.Click
    Texte.SelectionAlignment = HorizontalAlignment.Left
End Sub
```

```
Private Sub Right_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Rightt.Click
    Texte.SelectionAlignment = HorizontalAlignment.Right
End Sub
```

```
Private Sub Copier_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Copier.Click
    Clipboard.SetDataObject(Texte.SelectedText)
End Sub
```

```
Private Sub couper_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles couper.Click
```

```
Texte.Cut()
End Sub

Private Sub Coller_Click(ByVal sender As System.Object,
 ByVal e As System.EventArgs) Handles Coller.Click
 Texte.Paste()
End Sub

Private Sub Selectrionnertout_Click(ByVal sender As
 System.Object, ByVal e As System.EventArgs) Handles
 Selectrionnertout.Click
 Texte.SelectAll()
End Sub

Private Sub Button1_Click(ByVal sender As System.Object,
 ByVal e As System.EventArgs) Handles Button1.Click
 Dim a As MessageBox
 Dim b As MsgBoxResult
 b = a.Show("Voulez vous vraiment Sortir ????",
 "Confirmation!!", MessageBoxButtons.YesNo,
 MessageBoxIcon.Question, MessageBoxDefaultButton.Button1)
 If b = MsgBoxResult.Yes Then
 End
 End If
End Sub
```

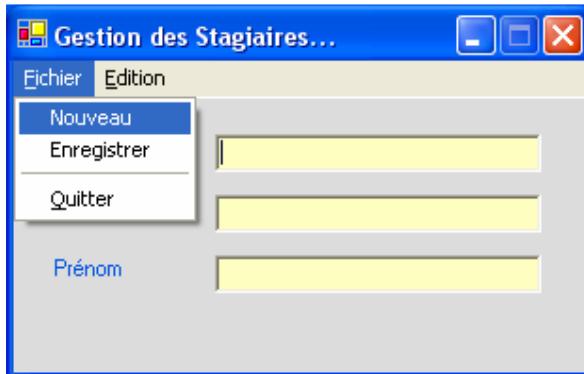


## ***TP N° 15 Les Fichiers, Les Menus, Datagrid***

### **Objectifs**

Dans ce TP nous vous proposons une interface contenant des menus dans le but de gérer des stagiaires à l'aide des fichiers textes, et en listant les stagiaires dans une datagrid.

1. Créer l'interface ci-dessous :

Figure 21Figure 22Figure 23Figure 24

Liste des Stagiaires		
Liste des Stagiaires		
	Code	Nom
▶	25	ELANZI
	20	ACHAHBAR
*	45	OUKHITA
		NourEddine

Figure 25

2. programmer les menus du premier formulaire « gestion des stagiaire » :
  - « Nouveau » pour vider les zones de texte.
  - « Enregistrer » pour ajouter les stagiaires dans un fichier texte.
  - « parcourir »pour afficher le formulaire « Consultation ....».
3. la consultation des stagiaires ce fait dans une collection remplie d'après le fichier texte.
4. la mise à jour ce fait sur le fichier : « Supprimer », « modifier ».

5. la rechercher ce fait par le code ou le nom du stagiaire.
6. le menu lister permet d'afficher les stagiaires contenus dans le fichier dans une datagrid.
7. tester votre application.

**NB :** Le remplissage du contrôle Datagrid ce fait par le biais d'une datatable.

### **Proposition de solution :**

#### **Module :**

```
Module Module1
    Public Structure stagiaire
        Private m_code As Long
        Private m_nom As String
        Private m_prenom As String
        Public Property code() As Long
            Get
                Return m_code
            End Get
            Set(ByVal Value As Long)
                m_code = Value
            End Set
        End Property
        Public Property Nom() As String
            Get
                Return m_nom
            End Get
            Set(ByVal Value As String)
                m_nom = Value
            End Set
        End Property
        Public Property Prenom() As String
            Get
                Return m_prenom
            End Get
            Set(ByVal Value As String)
                m_prenom = Value
            End Set
        End Property
    End Structure
    Public enr As stagiaire
    Public fs As FileStream
    Public rd As StreamReader
    Public wr As StreamWriter
    Public f1 As New Form1
    Public f2 As New Form2
    Sub main()
        Application.Run(f1)
    End Sub
End Module
```

**Form1 :**

```
Public tab() As String
    Private Sub Form1_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles MyBase.Load
        If Not File.Exists("c:\aziz.dat") Then
            fs = New FileStream("c:\aziz.dat", FileMode.CreateNew)
            wr = New StreamWriter(fs)
            MenuItem7.Enabled = False
        Else
            fs = New FileStream("c:\aziz.dat", FileMode.Append)
            wr = New StreamWriter(fs)
        End If
    End Sub

    Sub vider()
        Txtcode.ResetText()
        Ttxnom.ResetText()
        Txtprenom.ResetText()
        Txtcode.select()
    End Sub

    Private Sub Parcourir_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Parcourir.Click
        wr.Close()
        f2.Show()
        f1.Hide()
    End Sub

    Private Sub Nouveau_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Nouveau.Click
        Vider()
    End Sub

    Private Sub Quitter_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Quitter.Click
        End
    End Sub

    Private Sub Enregistrer_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Enregistrer.Click
        wr.Close()
        fs = New FileStream("c:\aziz.dat", FileMode.Open)
        rd = New StreamReader(fs)
        Dim ligne As String
        Do While rd.Peek <> -1
            ligne = rd.ReadLine
            tab = ligne.Split(";".Chars(0))
```

```

        If CInt(Txtcode.Text) = CInt(tab(0)) Then
            MsgBox("l'enregistrement n° " & Txtcode.Text & " Existe
                    déjà!!")
            rd.Close()
            Txtcode.Focus() : Txtcode.SelectAll()
            Exit Sub
        End If
    Loop
    rd.Close()
    fs = New FileStream("c:\aziz.dat", FileMode.Append)
    wr = New StreamWriter(fs)
    enr.code = CInt(Txtcode.Text)
    enr.Nom = Txtnom.Text
    enr.Prenom = Txprenom.Text
    wr.WriteLine(enr.code & ";" & enr.Nom & ";" &
enr.Prenom)
    Parcourir.Enabled = True
End Sub

```

**Form2 :**

```

Public tab() As String
Public list As New Collection
Public Shared pos As Integer
Private Sub Form2_Load(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles MyBase.Load
    If Not File.Exists("c:\aziz.dat") Then
        MsgBox("File Not Found", MsgBoxStyle.Information)
    Else
        fs = New FileStream("c:\aziz.dat", FileMode.Open)
        rd = New StreamReader(fs)
        Dim ligne As String
        Do While rd.Peek <> -1
            ligne = rd.ReadLine
            tab = ligne.Split(";")
            enr.code = CInt(tab(0))
            enr.Nom = tab(1)
            enr.Prenom = tab(2)
            list.Add(enr, enr.code.ToString)
        Loop
    End If
    pos = 1
    Try
        lire(pos)
    Catch ex As Exception

    End Try
    rd.Close()
End Sub

```

```
Private Sub Dernier_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles Dernier.Click
    pos = list.Count
    lire(pos)
End Sub

Private Sub Precedent_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles Precedent.Click
    pos -= 1
    Try
        lire(pos)
    Catch ex As Exception
        pos += 1
    End Try
End Sub

Sub lire(ByVal p As Integer)
    enr = list.Item(pos)
    Txtcode.Text = enr.code
    Txtnom.Text = enr.Nom
    Txtprenom.Text = enr.Prenom
End Sub

Private Sub Suivant_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles Suivant.Click
    pos += 1
    Try
        lire(pos)
    Catch ex As Exception
        pos -= 1
    End Try
End Sub

Private Sub Premier_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles Premier.Click
    pos = 1
    lire(pos)
End Sub

Private Sub cmd_Supprimer_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cmd_Supprimer.Click
    If MsgBox("Voulez vous vraiment supprimer l'enregistrement courant??", MsgBoxStyle.YesNo Or MsgBoxStyle.Critical Or MsgBoxStyle.DefaultButton2, "Supprimer!!!") = MsgBoxResult.Yes Then
        If File.Exists("c:\tmp.dat") Then
            File.Delete("c:\tmp.dat")
        End If
    End If

```

```

        fs = New FileStream("c:\tmp.dat",
 FileMode.CreateNew)
        wr = New StreamWriter(fs)
        fs = New FileStream("c:\aziz.dat", FileMode.Open)
        rd = New StreamReader(fs)
        Dim ligne As String
        Do While rd.Peek <> -1
            ligne = rd.ReadLine
            tab = ligne.Split(";")
            If Not CInt(Txtcode.Text) = CInt(tab(0)) Then
                wr.WriteLine(ligne)
            End If
        Loop
        Try
            list.Remove(Txtcode.Text.ToString)
        Catch ex As Exception
            MsgBox("Pas d'élément à supprimer!!!!!!",
        MsgBoxStyle.Critical, "Erreur!!!!")
        End Try
        rd.Close() : wr.Close()
        File.Delete("c:\aziz.dat")
        Rename("c:\tmp.dat", "c:\aziz.dat")
    '
    Try
        lire(pos)
    Catch ex As Exception
        If list.Count < 1 Then
            vider()
        Else
            pos -= 1
            lire(pos)
        End If
    End Try
    End If
End Sub

Sub vider()
    Txtcode.ResetText()
    Txtnom.ResetText()
    Txtprenom.ResetText()
End Sub

```

```

Private Sub cmd_Modifier_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles cmd_Modifier.Click
    Dim i As Integer
    If MsgBox("Voulez vous vraiment modifier
l'enregistrement courant???", MsgBoxStyle.YesNo Or
MsgBoxStyle.Information Or MsgBoxStyle.DefaultButton2, "Modifier!!!") =
MsgBoxResult.Yes Then

```

```

        If list.Count > 0 Then
            fs = New FileStream("c:\prov.dat", FileMode.CreateNew)
            wr = New StreamWriter(fs)
            fs = New FileStream("c:\aziz.dat", FileMode.Open)
            rd = New StreamReader(fs)
            Dim ligne As String
            Do While rd.Peek <> -1
                i += 1
                ligne = rd.ReadLine
                tab = ligne.Split(";")
                If Not CInt(Txtcode.Text) = CInt(tab(0)) Then
                    wr.WriteLine(ligne)
                Else
                    wr.WriteLine(Txtcode.Text & " ;" &
Txtnom.Text & ";" & Txtprenom.Text)
                    enr.code = CInt(Txtcode.Text) :
enr.Nom = Txtnom.Text : enr.Prenom = Txtprenom.Text
                    Try
                        list.Add(enr, enr.code.ToString, i)
                    Catch ex As Exception
                        If list.Count < 1 Then
                            list.Add(enr, enr.code)
                        Else
                            list.Add(enr, enr.code, , i - 1)
                        End If
                    End Try
                End If
            Loop
            rd.Close() : wr.Close()
            File.Delete("c:\aziz.dat")
            Rename("c:\prov.dat", "c:\aziz.dat")
        Else
            MsgBox("Pas d'élément à modifier!!!!!!",
MsgBoxStyle.Critical, "Erreur!!!!")
        End If
    End If
End Sub

```

```

Private Sub RechercherCode_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
RechercherCode.Click
    Try
        Dim rech As Integer = InputBox("Entrer le code à
rechercher!! :", "Rechercher....!!")
        For Each enr In list
            If enr.code = rech Then
                Txtcode.Text = enr.code : Txtnom.Text =
enr.Nom : Txtprenom.Text = enr.Prenom
                Exit Sub
            End If
        Next
    
```

```

        MsgBox("L'enregistrement " & rech & " est
introuvable!!!", MsgBoxStyle.Information, "Rechercher...!")
    Catch ex As Exception
        Txtcode.Focus()
    End Try
End Sub

Private Sub RechercherNom_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
RechercherNom.Click
    Try
        Dim rech As String = InputBox("Entrer le NOM à
rechercher!! :", "Rechercher....!!")
        For Each enr In list
            If enr.Nom = rech Then
                Txtcode.Text = enr.code : Txtnom.Text =
enr.Nom : Txtprenom.Text = enr.Prenom
                Exit Sub
            End If
        Next
        MsgBox("L'enregistrement dont le nom :" & rech & "
est introuvable!!!", MsgBoxStyle.Information, "Rechercher...!")
    Catch ex As Exception
        Txtcode.Focus()
    End Try
End Sub

Private Sub Quitter_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Quitter.Click
    f1.Show() : f2.Hide()
End Sub

Private Sub Lister_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Lister.Click
    Dim frm As New Form3(list)
    frm.Show()
End Sub

```

**Form3 :****Surcharger le constructeur :**

```

Public Sub New(ByVal CollStag As Collection)
    MyBase.New()
    'Cet appel est requis par le Concepteur Windows Form.
    InitializeComponent()
    MaColl = CollStag
    'Ajoutez une initialisation quelconque après l'appel
    InitializeComponent()
End Sub

```

```
Dim MaColl As Collection
    Dim dt As New DataTable
' Procedure pour créer une datatable
Sub CreaterTable()
    Dim Col As DataColumn
    Col = New DataColumn
    Col.ColumnName = "Code"
    Col.DataType = Type.GetType("System.Int64")
    dt.Columns.Add(Col)
    ' ---
    Col = New DataColumn
    Col.ColumnName = "Nom"
    Col.DataType = Type.GetType("System.String")
    Col.MaxLength = 60
    dt.Columns.Add(Col)
    ' ---
    Col = New DataColumn
    Col.ColumnName = "Prénom"
    Col.DataType = Type.GetType("System.String")
    Col.MaxLength = 60
    dt.Columns.Add(Col)
End Sub

' Procedure pour remplir Datable et après datagrid
Sub RemplirGrid()
    Dim Stag As stagiaire, dr As DataRow
    For Each Stag In MaColl
        dr = dt.NewRow()
        dr("Code") = Stag.code
        dr("Nom") = Stag.Nom
        dr(2) = Stag.Prenom: dt.Rows.Add(dr)
    Next
    'Remplir datagrid
    DataGridView1.DataSource = dt
End Sub

Private Sub Form3_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
    CreaterTable():RemplirGrid()
End Sub

Private Sub ButQuitter_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ButQuitter.Click
    Close()
End Sub
```



## **Conclusion :**

Je crois que c'est la fin de ce document...!!!! Je rigole ☺:

Et oui c'est la vérité, c'est vraiment la fin.

J'espère que vous avez trouvé au moins ce que vous cherchiez, et que ce document était vraiment utile pour les newbies du développement.