

Cours de Visual basic (vb.net)

DEFINITION VB : est un langage des programmations orienté objet

Une donnée : Une donnée est une information naturellement numérique ou alphanumérique représentée sous forme codée en vue d'y être enregistrée, traitée, conservée et communiquée qui est compréhensible par la seule machine.

Déclaration d'une variable : pour déclarer une variable on utilise le mot clé

Dim

Syntaxe : **Dim nom variable as type**

Exemple : **Dim age as Integer**

Nous avons les types numériques :

Type	
Byte	
Short	
Integer	
Long	
Single	
Double	
Decimal	

Chaîne des caractères :

Type	
Char	Stock un seul caractère
String	Chaîne caractères

Autres :

Type	
Boolean	True/false
Date	
Object	Universel n'importe lequel

Une constante : c'est une donnée identifiable par un nom qui est unique, dont la valeur est définie à la déclaration. Cette valeur lui reste affectée pendant toute la durée des vies du programme. Et n'est pas modifiable lors des exécutions

Déclaration : on utilise const

Syntaxe : **const nom_const as type = valeur**

Les opérateurs : Est un symbole permettant d'effectuer des opérations arithmétiques ou logiques sur les données.

Les types d'opérations (5) :

Affectation : un seul opérateur d'affectation existe, c'est ce que quel que soit le type concerné qui est : « = » il permet d'affecter une valeur à une variable.

Arithmétique : nous permet d'effectuer des calculs sur des variables et constante. (+, -, *, /, \ (division entier), %, ^ (puissance))

Comparaison : permet des comparer deux membres ou variables et en fonction du résultat retour True ou false (=, <, >, <>, =<,>=, (like seulement pour les chaine caractères)

CONCATENATION : (+ : il faut les deux chaines de la concaténation soit des types de chaine de caractères)

(& (appelé l'esperluette) dans ce cas l'opérateur effectuer une conversion implicite l'osque les deux chaines ne sont pas des chaines de caractère)

Logique : permet d'effectue des **combinaisons** entre les expression logique (à l'intérieur des condition)

Nous avons : (AND, OR, NOT, Xor ou exclusif, AndAlso et inclusif, OrElse ou inclusif)

Les exercices avec leurs corrigés :

Exo N°1: Ecrire un programme en VB qui demande le nom d'utilisateur et qui affiche un message pour lui souhaiter la bienvenue à l'IST de Mamou.

Exo N°2 : Ecrire un programme qui afficher le produit de deux nombre saisie au clavier.

Exo N°3 : On souhaite construire un programme qui permet de saisir le prix de vente, PAU, Qte et qui calcul le bénéfice la tva collectée et la tva déductible.

Exo N°4 : Ecrire un programme qui calcul la moyenne des trois nombres et le produit de deux premier nombre.

Exo N°5 : On souhaite réalise un programme qui calcul le poids d'un objet connaissant la masse et la pesanteur.

Exo N°6 : Soit r le rayon d'une sphère, écrire un programme qui permet d'affiché le volume soit $r=4\pi R^3/3$.

Exo N°7 : Réaliser un programme qui permet de demander le nom d'une matière et qui calcul la moyenne dans le système LMD.

Exo N°8 : Soit a, b et c des nombres réels écrire un programme la somme le produit et la moyenne des ces nombres.

Exo N°9 : Construire un programme qui calcul la distance entre deux points Sachant les coordonnées de ces points.

Exo N°10 : Ecrire un programme qui permet de saisir un nombre et qui affiche le produit, le double, et le tiers du même nombre.

Exo N°11 : Construire un programme qui permet de saisir deux nombres a et b, et qui affiche le max pour le plus grand et min pour le plus petit.

Exo N°12 : Réaliser un programme de saisir deux nombre a et b, qui affiche vrai si le deux sont égaux et faux dans le cas contraire.

Traité des exercices :

Module TP1

Sub Main ()

'déclaration de la variable

Dim nom As String

'Permet a user d'entrer le nom

Console.WriteLine ("Entrer votre nom")

Nom = Console.ReadLine ()

Console.WriteLine("Bienvenue " + nom + " à L'IST de Mamou")

Console.Read ()

End Sub

End Module

Module TP_2

Sub main ()

Dim a, b As Double

Console.WriteLine("Entrer nombre a")

a = Console.ReadLine()

Console.WriteLine("Entrer nombre b")

b = Console.ReadLine()

Console.WriteLine("le produit des deux nombre est:" & a * b)

Console.Read()

End Sub

End Module

Module TP_3

Sub main ()

Const TVA As Double = 0.18

Dim PA, PV, Qte As Double

Console.WriteLine("Entrer le PA")

PA = Console.ReadLine()

Console.WriteLine("Entrer le PV")

PV = Console.ReadLine()

Console.WriteLine("Entrer le Qte")

Qte = Console.ReadLine()

Console.WriteLine("Le bénéfice que vous pouvez obtenir est :" & (PV - PA) * Qte & " GNF")

```

    Console.WriteLine("Le TVAC que vous pouvez obtenir est :" & (PV * Qte * TVA & "
GNF"))
    Console.WriteLine("Le TVAD que vous pouvez obtenir est :" & PA * Qte * TVA & " GNF")
    Console.Read()
End Sub
End Module

```

Module tp4

```

Sub main ()
    Dim a, b, c, moy, prod As Double
    Console.WriteLine("Entrer le nombre a")
    a = Console.ReadLine
    Console.WriteLine("Entrer le nombre b ")
    b = Console.ReadLine
    Console.WriteLine("Entrer le nombre b ")
    c = Console.ReadLine
    moy = (a + b + c) / 3
    prod = a * b
    Console.WriteLine("La moyenne des trois nombres est " & moy)
    Console.WriteLine("Le produit des deux nombres est " & prod)
    Console.WriteLine("Appuyer sur Entrer pour quitter...")
    Console.Read()
    Console.WriteLine("appuyer sur entrer pour quitter merci...")
End Sub
End Module

```

Module TP_5

```

Sub main ()
    Const g As Double = 9.81
    Dim m, p As Double
    Console.WriteLine("Entrer la masse de l'objet")
    m = Console.ReadLine
    p = m * g
    Console.WriteLine("Le poids de l'objet est : " & p & " N")
    Console.WriteLine("Appuyer sur Entrer pour quitter...")
    Console.Read()
End Sub
End Module

```

Module TP_6

```

Sub main ()
    Const PI As Double = 3.14
    Const k As Double = 4 / 3
    Dim r, v As Double
    Console.WriteLine("le rayon :")
    r = Console.ReadLine
    v = k * PI * r ^ 3

```

```
Console.WriteLine("le volume de la sphère est: " & v)
Console.Read()
```

```
End Sub
End Module
```

```
Module TP_7
```

```
Sub main ()
```

```
Const K1 As Double = 0.25
```

```
Const K2 As Double = 0.35
```

```
Const K3 As Double = 0.4
```

```
Dim matiere As String
```

```
Dim n1, n2, n3, moy As Double
```

```
Console.WriteLine("Le nom de la matiere")
```

```
matiere = Console.ReadLine
```

```
Console.WriteLine("1er note")
```

```
n1 = Console.ReadLine
```

```
Console.WriteLine("2eme note")
```

```
n2 = Console.ReadLine
```

```
Console.WriteLine("3eme note")
```

```
n3 = Console.ReadLine
```

```
moy = n1 * K1 + n2 * K2 + n3 * K3
```

```
Console.WriteLine("la moyenne de l'étudiant avec LMD En : " + matiere + " est " & moy)
```

```
Console.Read()
```

```
End Sub
End Module
```

```
Module TP_9
```

```
Sub main ()
```

```
Dim AB, xA, xB, yA, yB As Double
```

```
Console.WriteLine("les coordonnées des A")
```

```
Console.WriteLine("les L'absice des A")
```

```
xA = Console.ReadLine
```

```
Console.WriteLine("Ordonnée de A")
```

```
yA = Console.ReadLine
```

```
Console.WriteLine("les coordonnées des B")
```

```
Console.WriteLine("les L'absice des B")
```

```
xB = Console.ReadLine
```

```
Console.WriteLine("Ordonnée de B")
```

```
yB = Console.ReadLine
```

```
AB = Math.Sqrt ((xB - xA) ^ 2 - (yB - yA) ^ 2)
```

```
Console.WriteLine("la distance AB est : " & AB)
```

```
Console.Read()
```

```
End Sub  
End Module
```

```
Module TP_10
```

```
Sub main ()  
    Const D As Integer = 2  
    Const T As Double = 1 / 3  
    Dim n As Double  
    Console.WriteLine("Entrer n")  
    n = Console.ReadLine  
    Console.WriteLine("le double de" & n & "est : " & n * D)  
    Console.WriteLine("le produit de" & n & "est : " & n * n)  
    Console.WriteLine("le tier de" & n & "est : " & n * T)  
    Console.Read()  
    Console.WriteLine("FIN DU PROGRAMME")
```

```
End Sub  
End Module
```

```
Module TP_11
```

```
Sub main ()  
    Dim a, b, min, max As Double  
    Console.WriteLine("saisissez le 1er nombre")  
    a = Console.ReadLine  
    Console.WriteLine("saisissez le 2em nombre")  
    b = Console.ReadLine  
    max = ((a + b) + Math.Abs (a - b)) / 2  
    min = ((a + b) - Math.Abs (a - b)) / 2  
    Console.WriteLine("le plus grand nombre est " & max)  
    Console.WriteLine("le plus petit nombre est " & min)  
    Console.Read()  
    Console.WriteLine("FIN DU PROGRAMME")
```

```
End Sub  
End Module
```

```
Module TP_12
```

```
Sub main ()  
    Dim a, b As Double  
    Dim resultat As Boolean  
    Console.WriteLine("Entrez le 1er nombre")  
    a = Console.ReadLine  
    Console.WriteLine("Entrez le 2em nombre")  
    b = Console.ReadLine  
    resultat = (a - b) = 0  
    Console.WriteLine("les deux nombres sont égaux " & resultat)
```


Console.Read()
End Sub End Module

A-LES STRUCTURE DE CONTROLES :

Permet de modifier les nombres d'exécution d'une instruction en fonction du résultat retourne True ou false. Elle exploite une conduction qui est une comparaison entre deux membres

Les structures conditionnelles : appelé structure des décisions dirige et oriente l'exécution de tel ou de tel bloc des codes en fonction d'une condition

Structure if : il n'y a plusieurs syntaxes

Forme simple : a utilise lorsqu'on n'est souhaité réaliser une seule instruction

Syntaxe : if condition then

.....

End if

Forme normale : elle permet d'exécuter plusieurs instructions lorsque la condition est vérifiée

Syntaxes : if condition then

Instruction

Instruction

End if

Forme évoluée : elle permet d'exécuter une ou plusieurs d'instruction lorsque la condition est vérifiée et exécuter d'autre dans le cas contraire

Syntaxe : if condition then

Instruction

Else

Instruction

End if

Exo N°13 : Ecrire un programme qui calcule la racine carre d'un nombre saisi au clavier.

Exo N°14 : Ecrire un programme qui traite la racine carre de la différence de deux nombre.

Exo N°15 : on souhaite réaliser un programme qui affiche la mention d'un étudiant en fonction de la moyenne, l'étudiant redouble lorsque sa moyenne est inferieur a 5 il affiche passable lorsque sa moyenne est comprise entre 5 et

6, BIEN si la moyenne est comprise entre 6 et 7, très bien entre 7 et 8, excellente supérieur ou égal à 8.

Exo N°16 : écrire un programme qui calcul la racine distincte

D'un polynôme du second degré : $ax^2+bx+c=0$

Exo N°17 :(énoncé exo 11)

Module TP_13

Sub main ()

Dim n, res As Double

Console.WriteLine("Entrer le nombre")

n = Console.ReadLine

If (n > 0) Then

res = Math.Sqrt(n)

Console.WriteLine("la racine care de " & n & " est : " & res)

Else

Console.WriteLine("la racine carre d'un nombre négatif négatif n'existe pas")

End If

Console.WriteLine("Fin du programme")

Console.Read()

End Sub

End Module

Module TP_14

Sub main ()

Dim a, b, rac As Double

Console.WriteLine("donnez le 1er nombre")

a = Console.ReadLine

Console.WriteLine("donnez le 2eme nombre")

b = Console.ReadLine

If (a - b) >= 0 Then

rac = Math.Sqrt(a - b)

Console.WriteLine("la racine de " & a & "-" & b & "=" & rac)

Else

Console.WriteLine("Impossible")

End If

Console.Read()

End Sub

End Module

Module TP_15

Sub main ()

Dim moy As Double

Console.WriteLine("la moyenne :")

moy = Console.ReadLine

If (moy >= 0 And moy < 5) Then

Console.WriteLine("redouble")


```

Elseif (moy >= 5 And moy < 6) Then
    Console.WriteLine("passable")
Elseif (moy >= 6 And moy < 7) Then
    Console.WriteLine("bien")
Elseif (moy >= 7 And moy < 8) Then
    Console.WriteLine("très bien")
Elseif (moy >= 8 And moy < 10) Then
    Console.WriteLine("Excellente")
Else
    Console.WriteLine("erreur")

End If
Console.Read()
End Sub
End Module

Module TP_16
Sub main ()
    Dim a, b, c, delta, x1, x2 As Double
    Console.WriteLine("Donnez la valeur de a")
    a = Console.ReadLine
    Console.WriteLine("Donnez la valeur de b")
    b = Console.ReadLine
    Console.WriteLine("Donnez la valeur de c")
    c = Console.ReadLine
    If (a <> 0) Then
        delta = b * b - 4 * a * c
        If delta > 0 Then
            x1 = (-b - Math.Sqrt(delta)) / 2 * a
            x2 = (-b + Math.Sqrt(delta)) / 2 * a
            Console.WriteLine("X1= " & x1)
            Console.WriteLine("X2= " & x2)
        ElseIf delta = 0 Then
            x1 = x2 = -b / 2 * a
            Console.WriteLine("X= " & x1)
        Else
            Console.WriteLine("Pas de solution dans R")
        End If
    Else
        x1 = -c / a
        Console.WriteLine("x= " & x1)
    End If
    Console.Read()
End Sub
End Module

```

Module TP_17

```

Sub main ()
    Dim a, b As Double
    Console.WriteLine("Donnez à :")
    a = Console.ReadLine
    Console.WriteLine("Donnez b :")
    b = Console.ReadLine
    If (a > b) Then
        Console.WriteLine("le plus grand est :" & a)
        Console.WriteLine("le plus petit est :" & b)
    ElseIf (a < b) Then
        Console.WriteLine("le plus grand est :" & b)
        Console.WriteLine("le plus petit est :" & a)
    Else
        Console.WriteLine("les deux nombres sont égaux")
    End If
    Console.Read()
End Sub
End Module

```

B-Les structures select case :

Aussi appelé choix multiples permet d'effectuer un ensemble des tests sur une seule valeur. Cette valeur peut être le contenu d'une variable, le résultat d'un calcul ou d'une fonction. Le principal intérêt de cette structure est clarifier le code : En effet l'utilisation de cette structure peut être remplacé par ensemble de structure if

Les différentes possibilités sont :

Syntaxe :

Case constante : test valide, si la valeur est une constante

Exemple :

```
Dim choix As Integer
```

```
Select case choix
```

```
Case 6 :
```

```
End select
```

Case valeur minimale to valeur maximale

Exemple : case 3 to 10

Pour les valeurs numériques, permet de définir d'un intervalle

Case is> ou bien <valeur. Pour les valeurs numériques définition un intervalle non fermé

EXEMPLE :

Case is > 6

Case else : cette condition sera validée si tous les tests de la structure (select case) sont faux.

Exo N°18 :

Ecrire un programme qui permet d'afficher le nom du jour de la semaine correspondant au numéro du jour de la semaine.

Module TP_18

Sub main ()

Dim choix As Integer

Console.WriteLine("Entrez svp le numéro du jour")

Choix = Console.ReadLine

Select Case choix

Case 1

Console.WriteLine("Lundi")

Case 2

Console.WriteLine("Mardi")

Case 3

Console.WriteLine("Mercredi")

Case 4

Console.WriteLine("Jeudi")

Case 5

Console.WriteLine("Vendredi")

Case 6

Console.WriteLine("Samedi")

Case 7

Console.WriteLine("Dimanche")

Case Else

Console.WriteLine("le numéro ne correspond à aucun jour de la semaine")

End Select

Console.Read()

End Sub

End Module

Exo N°19 :

Ecrire un programme en utilisant la structure case(exo15)

Module Tp_19

Sub main ()

Dim moy As Double

Console.WriteLine("Entrez la moyenne de l'étudiants")

moy = Console.ReadLine

Select Case moy

Case 0 To 4.99

```

    Console.WriteLine("l'étudiant a redoublé")
Case 5.0 To 5.99
    Console.WriteLine("passable")
Case 6 To 6.99
    Console.WriteLine("Bien")
Case 7 To 7.99
    Console.WriteLine("très bien")
Case 8 To 10
    Console.WriteLine("Excellent")
Case Else
    Console.WriteLine("erreur")
End Select
Console.Read()
End Sub
End Module

```

C-Les boucles :(Structures répétitives)

Permet d'exécuter un bloc d'instruction un certain nombre des
Fois

Structure While : permet d'exécuter les instructions contenues entre While et
End While tant que la condition est vérifiée.

Syntaxe :

While condition

End While

Exo N°20 : écrire un programme qui demande l'étudiant de répondre par oui
ou non s'il comprend son cours de vb.net.

```

Module TP_20
Sub main ()
    Dim reponse As String
    Console.WriteLine("svp comprenez-vous le cours de vb.net ?")
    reponse = Console.ReadLine
    While (reponse <> "oui" And reponse <> "Non")
        Console.WriteLine("svp répondez oui ou non")
        Reponse = Console.ReadLine
    End While
    Console.WriteLine("reponse accepté")
    Console.WriteLine(reponse)
    Console.Read()
End Sub
End Module

```

Exo N°21 : écrire un programme qui demande un nombre entre 13 et 19 jusqu'à ce que la réponse convienne. Il affiche : donnez une valeur plus grand si le nombre saisi est inférieur 13 et inversement si le nombre supérieur à 19.

Module TP_21

```
Sub main ()
    Dim nb As Double
    Console.WriteLine("Entrez le nombre")
    nb = Console.ReadLine
    While nb < 13 Or nb > 19
        If (nb < 13) Then
            Console.WriteLine("Donnez une valeur plus grand")
            nb = Console.ReadLine
        ElseIf (nb > 19) Then
            Console.WriteLine("Entrez une valeur plus petit")
            nb = Console.ReadLine
        End If
    End While
    Console.WriteLine("Bravo !!!!")
    Console.Read()
End Sub
End Module
```

C-Les structures Do Loop :

Elle peut être paramétrée par la structure while (tant que la condition est vérifiée) et until (jusqu'à ce que la condition soit vérifiée).

Syntaxes :

Do while condition

Loop

Do until condition

Loop

Dans ces deux cas la condition est évaluée avant l'entrée dans la boucle c'est-à-dire il est possible que les instructions ne s'exécutent aucune fois

Do

Loop while condition

Do

Loop until condition

Dans ces deux cas les instructions sont exécuter ont moins une fois avant l'évaluation de la condition.

Exo N°22 : soit a et b deux entier naturels écrit un programme qui calcul la somme de ces deux nombres saisi au clavier veuille à ce que l'utilisateur ne saisis des erreurs.

Module TP_22

```
Sub Main ()  
    Dim a As Integer = 0  
    Dim b As Integer = 0  
    Dim c As String = ""  
    Console.WriteLine("Entrez le 1er nombre")  
    c = Console.ReadLine()  
    Do While Not IsNumeric(c)  
        Console.WriteLine("Erreur, svp Donnez un nombre")  
        c = Console.ReadLine()  
    Loop  
    a = c  
    Console.WriteLine("Entrez le second nombre")  
    c = Console.ReadLine()  
    Do While Not IsNumeric(c)  
        Console.WriteLine("Erreur ! Entrez le 2eme nombre svp")  
        c = Console.ReadLine()  
    Loop  
    b = c  
    Console.WriteLine("La somme est : " & a & "+" & b & "=" & a  
+ b)  
    Console.Read()  
End Sub  
End Module
```

D- Les structures for :

La structure for est utilisé pour exécuter un bloc instruction un nombre de fois détermine toute en gérant un compteur qui sera automatiquement incrémenter. Le nombre d'exécution est fonction de l'intervalle définie par le compteur

Syntaxes :

For compteur valeur Minimale to valeur Maximale step valeur
d'incréméntation

Next

Exemple :

For i=0 to 5

Msgbox ("très bien")

Next

Exo N°23 : écrire un programme qui affiche les entiers naturels impaire inférieur à 20.

```
Module TP_23
    Sub Main ()
        Dim i As Integer
        Console.WriteLine("Les entiers naturels impaire inférieur à 20 sont :")
        For i = 1 To 20 Step 2
            Console.WriteLine(i)
        Next
        Console.Read()
    End Sub
End Module
```

Exo N°24 : écrire un programme qui affiche la table de multiplication d'un nombre saisi au clavier.

```
Module TP_24
    Sub Main ()
        Dim n, i As Integer
        Console.WriteLine("Donnez un nombre")
        n = Console.ReadLine()
        Console.WriteLine("La table de multiplication par " & n)
        For i = 1 To 10
            Console.WriteLine(n & "*" & i & "=" & n * i)
        Next
        Console.Read()
    End Sub
End Module
```

E-Les procédures et les fonctions :

Dans une application de Visual basic de vb.net les instructions doivent obligatoirement figure dans des procédures et des fonctions. C'est sont des unités logiques de code qui seront ensuite appelé. Utilisation des fonctions et des procédures présente plusieurs intérêts.

- Lisibilité du code
- Modularité
- Evolutivité

Création d'une fonction ou d'une procédure :

La création d'une fonction ou d'une procédure comporte deux étapes :

- **Déclaration** : consiste à définir le comportement de la fonction ou la procédure
- **Appelé** : consiste à demande l'exécution du comportement de la fonction ou de la procédure

Syntaxe de la procédure :

Syntaxe de la Déclaration :

Sub Nom Procédure ()

End sub

- **Appelé** : consiste à écrire le nom de la procédure

Syntaxe de la Déclaration d'une fonction :

Fonction nom fonction (argument) as type retour

Retourn

End fonction

Exo N°25 : écrire un programme qui fait la somme de deux nombre saisi au clavier

```
Module Exo_25
    Sub Main ()
        sommeProcure ()
    End Sub
    Public Sub sommeProcure ()
        Dim a, b As Integer
        Console.WriteLine("Donnez le 1er nombre")
        a = Console.ReadLine()
        Console.WriteLine("Saisissez le 2eme nombre")
        b = Console.ReadLine()
        Console.WriteLine("La somme de " & a & "+" & b & "=" & a +b)
        Console.Read()
    End Sub
End Module
```

Exo N°26 : sur les fonctions

```
Module Exo_26
    Sub Main ()
        Dim a, b As Integer
        Console.WriteLine(somme fonction (a, b))
    End Sub
```

```

Function sommefonction (ByVal a As Integer, ByVal b As Integer)
As Integer
    Dim somme As Integer
    Console.WriteLine("Donnez le 1er nombre")
    a = Console.ReadLine()
    Console.WriteLine("Saisissez le 2eme nombre")
    b = Console.ReadLine()
    Somme = a + b
    Console.WriteLine("La somme est :" & (somme))
    Console.Read()
    Return somme
End Function
End Module

```

F-Tableau et Structure :

Est un regroupement des variables accessible par le même nom et différenciable par leur indice.

Un tableau peut avoir jusqu'à 32 dimension l'indice d'un tableau commence toujours par zéro lors de la déclaration d'un tableau, l'indice maximum est précise il comporte l'indice maximum plus une valeur

Pour la déclaration d'un tableau

Syntaxe : Dim NomTableau (taille) as type

Exo N°27 : écrire un programme un programme qui demande à l'utilisateur de saisir les 12 mois de l'année est qui encore les affiche

```

Module Exo_27
    Sub Main ()
        Dim MoisAnnee (11) As String
        Dim i As Integer
        For i = 0 To MoisAnnee.Length - 1
            Console.WriteLine("Donnez le nom du " & (i + 1) & "e" &
"mois de l'année")
            MoisAnnee(i) = Console.ReadLine()
        Next
        Console.WriteLine()
        Console.WriteLine("Les 12 mois de l'année sont : ")
        For i = 0 To MoisAnnee.Length - 1
            Console.WriteLine(MoisAnnee(i))
        Next
        Console.Read()
    End Sub
End Module

```

Une structure est un regroupement des données de différent type

Syntaxe : Structure NomStructure

Champ1 as type

Champ2 as type

Champn as type

Exo N°28 : écrire un programme qui permet à l'utilisateur de saisir les informations d'un étudiant puis les affiche

Module Exo_28

```
Sub Main ()
    Dim O_B As Etudiant
    Console.WriteLine("Donnez le matricule de l'étudiant")
    O_B.matricule = Console.ReadLine()
    Console.WriteLine("Saisissez le nom de l'étudiant")
    O_B.nom = Console.ReadLine()
    Console.WriteLine("Entrez le prénom de l'étudiant")
    O_B.prenom = Console.ReadLine()
    Console.WriteLine("Veuillez saisir la data de naissance de l'étudiant")
    O_B.date_naiss = Console.ReadLine()
    Console.WriteLine()
    Console.WriteLine("Matricule :" & O_B.matricule)
    Console.WriteLine("Nom :" & O_B.nom)
    Console.WriteLine("Prénom :" & O_B.prenom)
    Console.WriteLine("Date_naissance : " & O_B.date_naiss)
    Console.Read()
End Sub
Public Structure Etudiant
    Dim matricule As String
    Dim nom As String
    Dim prenom As String
    Dim date_naiss As Date
End Structure
End Module
```

G- Les Interfaces :



Calculer

```
Public Class Form1
    Private Sub BtCalculer_Click (sender As Object, e As EventArgs)
        Handles BtCalculer.Click
            If Tbnb1.Text = "" Then
                MessageBox.Show ("Veuillez saisir le premier nombre",
                "Premier Nombre", MessageBoxButtons.OK, MessageBoxIcon.
                Information)
                Exit Sub
            End If
            If Tbnb2.Text = "" Then
                MessageBox.Show ("Veuillez saisir le second nombre",
                "Deuxième Nombre", MessageBoxButtons.OK, MessageBoxIcon.
                Information)
                Exit Sub
            End If
            If Rbaddition.Checked = False And Rbmultiplication.Checked
            = False And Rbsoustraction.Checked = False And Rbdivision.Checked
            = False Then
                MessageBox.Show ("Vous devez choisir un opérateur svp",
                "Choix opérateur", MessageBoxButtons.OK, MessageBoxIcon.Error)
                Exit Sub
            End If

            Timer1.Start()

            Try
                If Tbnb1.Text = IsNumeric (Tbnb1.Text) And Tbnb2.Text =
                IsNumeric (Tbnb2.Text) Then
                    End If

                    If Rbaddition.Checked = True Then
                        TbResultat.Text = Val (Tbnb1.Text) + Val
                        (Tbnb2.Text)
                        Lbresultat_somme.Text = TbResultat.Text

                        Cacherresultat ()
                        Lbsomme.Visible = True
                        Lbresultat_somme.Visible = True

                        Timer1.Start()
                    ElseIf Rbsoustraction.Checked = True Then
                        TbResultat.Text = Val (Tbnb1.Text) - Val
                        (Tbnb2.Text)
                        Lbresultat_difference.Text = TbResultat.Text

                        Cacherresultat ()
                        Lbdifference.Visible = True
```

```

        Lbresultat_difference.Visible = True

        Timer1.Start()
    ElseIf Rbmultiplication.Checked = True Then
        TbResultat.Text = Val (Tbnb1.Text) * Val
(Tbnb2.Text)
        Lbresultat_produit.Text = TbResultat.Text

        Cacherresultat ()
        Lbproduit.Visible = True
        Lbresultat_produit.Visible = True

        Timer1.Start()
    ElseIf Rbdivision.Checked = True Then
        TbResultat.Text = Val (Tbnb1.Text) / Val
(Tbnb2.Text)
        Lbresultat_quotient.Text = TbResultat.Text

        Cacherresultat ()
        Lbquotient.Visible = True
        Lbresultat_quotient.Visible = True
    End If

    Catch ex As Exception
        MessageBox.Show ("Attention, vous devez saisir des
valeurs numérique", "Erreur de saisie", MessageBoxButtons.OK,
MessageBoxIcon.Error)
    End Try
End Sub
➤ Création d'une procédure :
Sub cacherresultat ()
    Lbsomme.Visible = False
    Lbresultat_somme.Visible = False

    Lbdifference.Visible = False
    Lbresultat_difference.Visible = False

    Lbproduit.Visible = False
    Lbresultat_produit.Visible = False

    Lbquotient.Visible = False
    Lbresultat_quotient.Visible = False
End Sub

```

Nouveau

```

Private Sub Btnnouveau_Click (sender As Object, e As EventArgs)
Handles Btnnouveau.Click
    Cacherresultat ()
    Tbnb1.Clear()

```



```

Tbnb2.Clear()
TbResultat. Clear ()

Rbaddition. Checked = False
Rbsoustraction. Checked = False
Rbmultiplication. Checked = False
Rbdivision. Checked = False

```

```

End Sub
End Class

```

Quitter

```

Private Sub Btquitter_Click (sender As Object, e As EventArgs)
Handles Btquitter. Click
    Dim confirmation As DialogResult
    Confirmation = MessageBox. Show ("Voulez-vous quitter
l'application", "Quitter", MessageBoxButtons. YesNo, MessageBoxIcon.
Question)
    If confirmation = DialogResult. Yes Then
        Me. Close ()
    End If
End Sub

```

```

Private Sub Timer1_Tick (sender As Object, e As EventArgs) Handles
Timer1.Tick
    If Lbresultat_somme. Text = "" Then
        Lbresultat_somme. Text = TbResultat. Text
    Else
        Lbresultat_somme. Text = ""
    End If

    If Lbresultat_difference. Text = "" Then
        Lbresultat_difference. Text = TbResultat. Text
    Else
        Lbresultat_difference. Text = ""
    End If

    If Lbresultat_produit. Text = "" Then
        Lbresultat_produit. Text = TbResultat. Text
    Else
        Lbresultat_produit. Text = ""
    End If

    If Lbresultat_quotient. Text = "" Then
        Lbresultat_quotient. Text = TbResultat. Text
    Else
        Lbresultat_quotient. Text = ""
    End If

```

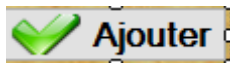
End Sub

Pour établir une connexion entre le SGBD et Visual Basic :

```
Imports System
Imports System.Data
Imports System.Data. Odbc
Imports System.IO

Module Module1
    Public conx As OleDb.OleDbConnection
    Dim ad As Stream

    Public Sub connection ()
        Conx = New
OleDb.OleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=C:\Users\Ingénieur Diallo\Desktop\menuserie.mdb")
        Try
            conx.Open ()
        Catch ex As Exception
            MsgBox (ex. Message)
        End Try
    End Sub
    Public Function TabImg (ByVal Tab As Byte ()) As Image
        Try
            Dim ad As MemoryStream = New MemoryStream (Tab)
            Dim Img As Image
            Img = Image.FromStream (ad)
            Return Img
        Catch ex As Exception
            Return Nothing
        End Try
    End Function
    Public Function Imgtab (ByVal img As Image) As Byte ()
        Try
            Dim ab As New MemoryStream
            img. Save (ab, Drawing.Imaging. ImageFormat.Jpeg)
            Dim tab As Byte () = ab. GetBuffer
            Return tab
        Catch ex As Exception
            Return Nothing
        End Try
    End Function
End Module
```



```

Private Sub BtAjouer_Click (sender As Object, e As EventArgs)
Handles BtAjouer_Click

    'Rendre les champs obligatoires'

    If TbCodeClt.Text = "" Then
        MessageBox.Show ("Veuillez entrer le code du client",
"CODE CLIENT", MessageBoxButtons.OK, MessageBoxIcon.Information)
        TbCodeClt.Focus ()
        Exit Sub
    End If

    If TbNomClt.Text = "" Then
        MessageBox.Show ("Veuillez entrer le nom du client",
"NOM CLIENT", MessageBoxButtons.OK, MessageBoxIcon.Information)
        TbNomClt.Focus ()
        Exit Sub
    End If

    If TbPrenomClt.Text = "" Then
        MessageBox.Show ("Veuillez entrer le prénom du client",
"PRENOM CLIENT", MessageBoxButtons.OK, MessageBoxIcon.Information)
        TbPrenomClt.Focus ()
        Exit Sub
    End If

    If TbAdresseClt.Text = "" Then
        MessageBox.Show ("Veuillez entrer l'adresse du client",
"ADRESSE CLIENT", MessageBoxButtons.OK, MessageBoxIcon.Information)
        TbAdresseClt.Focus ()
        Exit Sub
    End If

    If TtelephoneClt.Text = "" Then
        MessageBox.Show ("Veuillez entrer le numéro de téléphone
du client", "TELEPHONE CLIENT", MessageBoxButtons.OK,
MessageBoxIcon.Information)
        TtelephoneClt.Focus ()
        Exit Sub
    End If

    Connection ()

    'Gérer les doublons'

    Dim verifierDoublon As OleDb.OleDbCommand
    VerifierDoublon = New OleDb.OleDbCommand
    verifierDoublon.Connection = conx

```

```

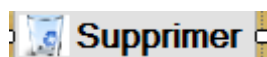
verifierDoublon.CommandText = "SELECT * FROM t_client WHERE
id_client = '" & TbCodeClt.Text & "'"
Dim readerDoublon As OleDb.OleDbDataReader
ReaderDoublon = verifierDoublon.ExecuteReader
    If readerDoublon.Read Then
        MessageBox.Show("Ce code du client existe déjà, veuillez
saisir un autre", "DOUBLON", MessageBoxButtons.OK, MessageBoxIcon.
Warning)
        TbCodeClt.Focus()
    Else

        'Enregistrement du client'

Dim commande As OleDb.OleDbCommand = New OleDb.OleDbCommand
Commande.Connection = conx
Commande.CommandText = "INSERT INTO t_client VALUES ('" &
TbCodeClt.Text & "', '" & TbNomClt.Text & "', '" & TbPrenomClt.Text
& "', '" & TbAdresseClt.Text & "', '" & TtelephoneClt.Text & "')"
Commande.ExecuteNonQuery ()
MessageBox.Show("Enregistrement effectué avec succès", "Nouveau
client", MessageBoxButtons.OK, MessageBoxIcon.Information)
Actualiser () 'Actualiser les enregistrements'

        'Vider les champs
        TbCodeClt.Text = ""
        TbNomClt.Text = ""
        TbPrenomClt.Text = ""
        TbAdresseClt.Text = ""
        TtelephoneClt.Text = ""
    End If
End Sub

```



```

Private Sub BtSupprimer_Click (sender As Object, e As EventArgs)
Handles BtSupprimer_Click

    Connection ()
    Dim confirmation As DialogResult
    Confirmation = MessageBox.Show("Voulez-vous supprimer ce client ?",
"Suppression", MessageBoxButtons.YesNo, MessageBoxIcon.Question)
    If confirmation = DialogResult.Yes Then
        Connection ()
        Dim cmd As New OleDb.OleDbCommand
        Cmd.Connection = conx
        Cmd.CommandText = "Delete *from t_client where id_client = '" &
TbCodeClt.Text & "'"
        Cmd.ExecuteNonQuery ()
    End If
End Sub

```

```

    MessageBox.Show("Suppression effectuée avec succès", "Suppression",
    MessageBoxButtons.OK, MessageBoxIcon.Information)
        Actualiser ()
    End If
End Sub

```



```

Private Sub BtModifier_Click (sender As Object, e As EventArgs)
Handles BtModifier_Click

Dim confirmation As DialogResult
Confirmation = MessageBox.Show("Voulez-vous modifier les données de
ce client ?", "Modification", MessageBoxButtons.YesNo,
MessageBoxIcon.Question)
If confirmation = Windows.Forms.DialogResult.Yes Then
    Connection ()

```

'Modifier l'enregistrement'

```

Dim cmd As New OleDb.OleDbCommand
Cmd.Connection = conx
Cmd.CommandText = "Update t_client set id_client=@code,
nom_client=@nom, penom_client=@prenom, adresse_client=@adresse,
tel_client=@tel where id_client='" & TbCodeClt.Text & "'"
Cmd.Parameters.Add ("@code", OleDb.OleDbType. Varchar). Value =
TbCodeClt.Text
Cmd.Parameters.Add ("@nom", OleDb.OleDbType. VarChar). Value =
TbNomClt.Text
Cmd.Parameters.Add ("@prenom", OleDb.OleDbType. VarChar). Value =
TbPrenomClt.Text
Cmd.Parameters.Add ("@adresse", OleDb.OleDbType. VarChar). Value =
TbAdresseClt.Text
Cmd.Parameters.Add ("@tel", OleDb.OleDbType. VarChar). Value =
TbtelephoneClt.Text
Cmd.ExecuteNonQuery ()
MessageBox.Show("Modification effectuée avec succès", "Modification
", MessageBoxButtons.OK, MessageBoxIcon.Information)
DGV_ListClt. Rows. Clear ()
Actualiser ()

```

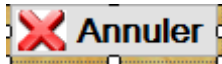
'Vider les champs
TbCodeClt.Text = ""
TbNomClt.Text = ""
TbPrenomClt.Text = ""
TbAdresseClt.Text = ""
TbtelephoneClt.Text = ""

```

Else
    IsNothing (0)

```

```
End If
End Sub
```



```
Private Sub BtAnnuler_Click (sender As Object, e As EventArgs)
Handles BtAnnuler_Click
```

```
'Vider les champs
TbCodeClt.Text = ""
TbNomClt.Text = ""
TbPrenomClt.Text = ""
TbAdresseClt.Text = ""
TbtelephoneClt.Text = ""
```

```
End Sub
```

'Cette procédure va actualiser la liste des clients'

```
Public Sub Actualiser ()
    Connection ()
    Dim cmd As OleDb.OleDbCommand = New OleDb.OleDbCommand
    Cmd. Connection = conx
    Cmd. CommandText = "SELECT * FROM t_client"

    Dim lire As OleDb.OleDbDataReader
    Lire = cmd. ExecuteReader

    'Nettoyer la datagreed view avant d'actualiser
    DGV_ListClt. Rows. Clear ()

    While lire. Read
        DGV_ListClt. Rows. Add (lire (0), lire (1), lire (2),
lire (3), lire (4))
    End While
End Sub
```



```
Private Sub BtRecherche_Click (sender As Object, e As EventArgs)
Handles BtRecherche_Click
```

```
If CbRecherche.Text = "" Then
    MessageBox.Show("veuillez saisir un parametre de recherche",
"Rechercher Client", MessageBoxButtons.OK,
MessageBoxIcon.Information)
End If
If CbRecherche.SelectedItem = "Code Client" Then
    Connection ()
    Dim cmd As OleDb.OleDbCommand
```



```

Cmd = New OleDb.OleDbCommand
Cmd. Connection = conx
Cmd.CommandText = "Select *from t_client where id_client='" &
TbRecherche.Text & "'"

Dim lire As OleDb.OleDbDataReader
Lire = Cmd. ExecuteReader
DGV_ListClt. Rows. Clear ()
If lire. Read Then
DGV_ListClt. Rows. Add (lire (0), lire (1), lire (2), lire (3),
lire (4))
Else
MessageBox.Show("Ce code ne correspond à aucun client",
"RECHERCHER", MessageBoxButtons.OK, MessageBoxIcon.Information)

End If
End If
If CbRecherche.SelectedItem = "Téléphone" Then
Connection ()
Dim cmd As OleDb.OleDbCommand
Cmd = New OleDb.OleDbCommand
Cmd. Connection = conx
Cmd. CommandText = "Select *from t_client where
tel_client='" & TbRecherche.Text & "'"

Dim lire As OleDb.OleDbDataReader
Lire = Cmd. ExecuteReader
DGV_ListClt. Rows. Clear ()
If lire. Read Then
DGV_ListClt. Rows. Add (lire (0), lire (1), lire (2), lire (3), lire
(4))
Else
MessageBox.Show("Ce code ne correspond à aucun
client", "RECHERCHER", MessageBoxButtons.OK,
MessageBoxIcon.Information)

End If
End If
End Sub

Private Sub F_Client_Load (sender As Object, e As EventArgs) Handles
MyBase. Load
Actualiser () 'Actualiser les enregistrements à l'ouverture
du formulaire
End Sub

Private Sub DGV_ListClt_CellMouseClick (sender As Object, e As
DataGridViewCellEventArgs) Handles DGV_ListClt_CellMouseClick

```

```

'chargement des informations sur le DataGridView'
If DGV_ListClt. CurrentRow. Cells (0). ToString <> "" Then
TbCodeClt.Text = DGV_ListClt. CurrentRow. Cells (0). Value
TbNomClt.Text = DGV_ListClt. CurrentRow. Cells (1). Value
TbPrenomClt.Text = DGV_ListClt. CurrentRow. Cells (2). Value
TbAdresseClt.Text = DGV_ListClt. CurrentRow. Cells (3). Value
TbtelephoneClt.Text = DGV_ListClt. CurrentRow. Cells (4). Value
TbCodeClt. Enabled = False
End If
End Sub

```

```

Private Sub BtnImprClt_Click (sender As Object, e As EventArgs)
Handles BtnImprClt_Click

    Connection ()
    Dim cmd As New OleDb.OleDbCommand
    Cmd. Connection = conx
    Cmd. CommandText = "Select *from t_client where id_client=@code"

    Dim code As New OleDb. OleDbParameter
    Code. ParameterName = "@code"
    Code. OleDbType = OleDb. OleDbType. VarChar
    Code. Value = TbCodeClt.Text
    Cmd. Parameters. Add (code)
    Try
        Dim lire As OleDb.OleDbDataReader
        Lire = Cmd. ExecuteReader
        If lire. Read Then
            Lire. Close ()

            Dim ad As New OleDb. OleDbDataAdapter (cmd)
            Ad. Fill (F_ImprUnClt.menuserieDataSet.t_client1)
        Else
            MessageBox.Show("Aucune n'existe")
        End If
    Catch ex As Exception
        MessageBox.Show(ex. Message)
    End Try

    F_ImprUnClt. ReportViewImprUnClt. RefreshReport ()
    F_ImprUnClt. ShowDialog ()
    F_ImprUnClt. Dispose ()
End Sub
End Class

```

Génie Info P13 à l'IST/Mamou