

PROGRAMMER AVEC VISUAL BASIC APPLICATIONS.

→ L'écriture du code se fait à l'intérieur du module.

→ Les instructions

DIM : Déclare les variables

Syntaxe : DIM « nom de variable » As type

Ex : DIM Nombre AS Integer.

↓
déclare le type de données d'une variable.

les opérateurs :

permettent de relier des variables entre elles afin d'exécuter l'opération souhaitée.

REM : inclut des commentaires explicatifs ds un programme.

Ex : REM Programme permettant la soustraction de 2 entiers.

INPUTBOX : saisi une donnée ds une variable avec un msg d'invite.

Syntaxe : Variable = INPUTBOX (« msg d'invite »)

Ex : Nombre = INPUTBOX ("Donner la valeur de ...")

Debug.Print : affiche un msg et le contenu d'une variable

Syntaxe : Debug.Print (« msg » & variable)

Ex : Debug.Print "Le total est : " & Total.

MsgBox : affiche le résultat ds une boîte de msg. C'est mieux que l'affichage par la fenêtre d'exécution.

Syntaxe : MsgBox (« msg » & variable)

→ les boucles (ou itérations)

A = FAIRE TANT QUE

Répète un bloc d'instruction aussi longtemps qu'une condition est vraie.

DO WHILE CONDITION
Bloc d'instruction

LOOP

DO

Bloc d'instruction

B - REPETER JUSQU'À

DO UNTIL CONDITION
Bloc d'instruction
LOOP

DO
Bloc d'instruction
LOOP WHILE CONDITION

} Répète un bloc d'instruction jusqu'à ce qu'une condition devienne vraie.

C POUR FIN DU POUR

FOR compteur = début To fin
Bloc d'instructions
NEXT compteur

} Répète un certain nb de fois un groupe d'instructions.

Le nb de x est connu à l'avance.

Compteur : variable numérique utilisée à compter de boucles
Début : valeur initiale du compteur
Fin : " fin de "

Lecture séquentielle

Il s'agit de lire une table et cela à partir du 1^{er} enregistrement. La lecture se déroulera jusqu'au pr d'arrêt par défaut, la fin de fichier.

Boucle

Il s'agit de répéter un certain nb d'opérations, d'instructions, un nb fini de fois.

Cumul

Il s'agit de totaliser le contenu d'une variable à la somme des résultats précédents.

→ Les instructions (2)

Ex: Dim SKILL As Database

LD Declare "SKILL" en tant que base de données

SET SKILL = CurrentDB

LD Declare "SKILL" comme base de donnée courante

Dim Salaire As Recordset

LD Declare la table "Salaire" en tant que table de W

Salary: NewFirst

↳ Demande au pointeur d'enregistrement de se rendre sur le 1^{er} enregistrement.

Salary: EOF

↳ Demande au pointeur d'enregistrement de se rendre à la fin de fichier, (dernier enregistrement)

EOF = End of file

Salary: NewNext

↳ Demande ... sur l'enregistrement suivant

Salary: Close

↳ ordonne la fermeture de la table "Salary"

SQL close

↳ ordonne la fermeture de la base de données "SQL".

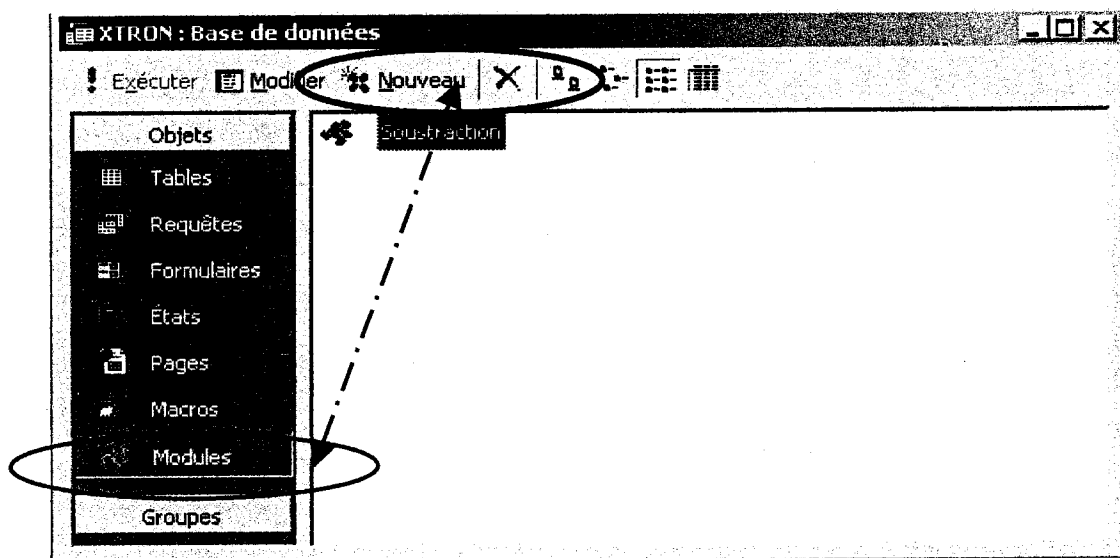
Access : Programmer avec Visual Basic Application (1)

Les fonctionnalités de Microsoft Access sont déjà relativement évoluées en raison des formulaires, tables, requêtes, macros ou états, mais elles ne suffisent pas toujours pour correspondre parfaitement aux exigences plus pointues et spécifiques des clients souhaitant une application taillée sur mesure. A cet effet, vous disposez d'Access Basic grâce auquel les capacités de Microsoft Access peuvent être élargies quasiment sans limite.

§ 1 - Présentation générale

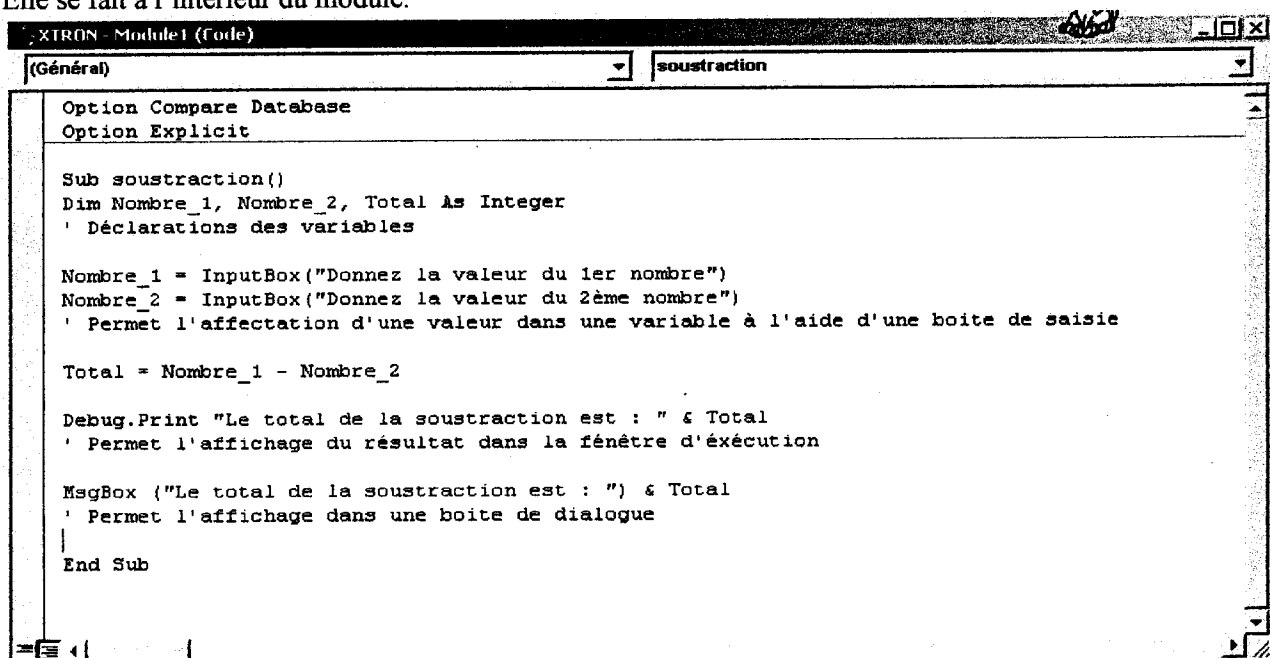
A - Déclaration du programme

Il faut utiliser un **module**. Dans la fenêtre Base de données, choisissez l'icône du module. Si la base en comporte déjà, leurs noms seront affichés dans une boîte de dialogue, mais si vous souhaitez en créer un nouveau, vous validez le bouton de commande *Nouveau* pour accéder à la fenêtre du module.



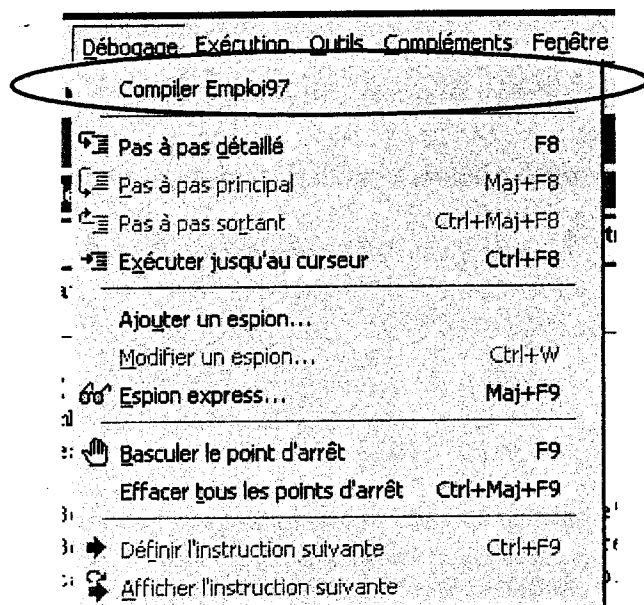
B - Ecriture du code

Elle se fait à l'intérieur du module.



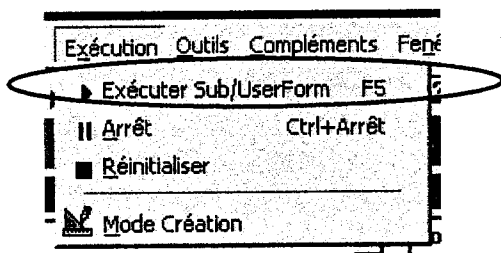
C - Mise au point du programme

Il s'agit de l'étape de compilation qui va permettre de vérifier la syntaxe du module. Cette étape n'est pas primordiale. En effet, Access 2000 montre les erreurs de syntaxe dès la frappe de l'instruction concernée. Un message d'erreur apparaît, la ligne entière est marquée en rouge et l'instruction est sélectionnée pour modification.



D - Exécution d'un programme

Pour s'exécuter, l'opération est la suivante :



Remarque : la touche « F5 » donne le même résultat que l'instruction « Exécuter Sub/UserForm ».

Cela doit donner ceci :

- Saisie des 2 valeurs :

Microsoft Access

Donnez la valeur du 1er nombre

OK

Annuler

123

Microsoft Access

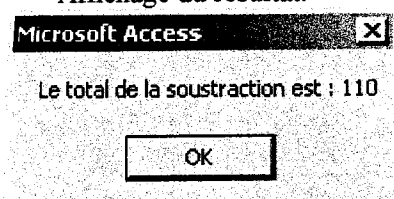
Donnez la valeur du 2ème nombre

OK

Annuler

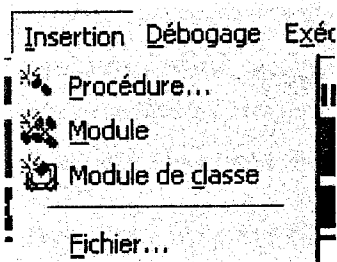
13

- Affichage du résultat.



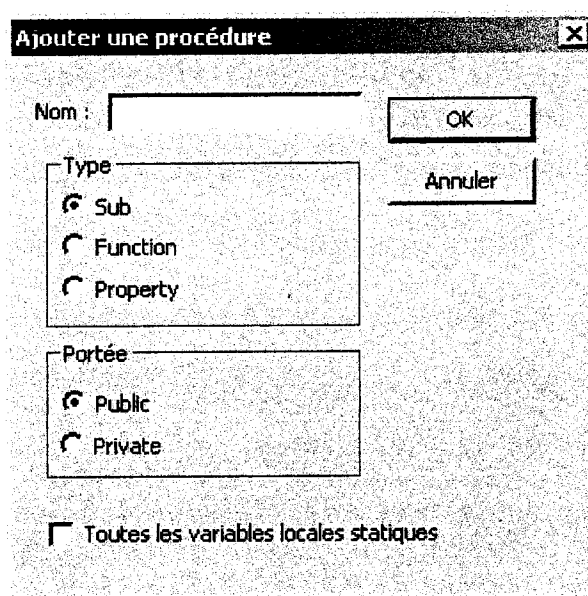
E - Reprendre le programme ultérieurement

Cliquez sur le module désiré. ACCESS ouvre la fenêtre de création de module. Il est possible de le modifier de créer une fonction ou un nouveau module.



Lors de l'ajout d'une procédure, vous obtenez la fenêtre avec les choix suivants :

- Sub : il s'agit d'une procédure, un programme indépendant.
- Fonction : c'est un programme qui renvoie à une procédure le résultat de ses opérations (voir séquence sur les fonctions).
- Property : ce sont des propriétés, hors programme Terminale STT.



§ 2 - Les instructions

A - Instruction Dim

Elle est utilisée pour déclarer des variables et allouer de l'espace mémoire.

Syntaxe : Dim « nom de variable » As type

As : mot réservé utilisé pour déclarer le type de données d'une variable.

Exemple : DIM NUMCLI,CATEG AS INTEGER

Remarque : il est possible de déclarer sur la même ligne plusieurs variables *de même type*, chaque variable étant séparée par une virgule.

Type de données	Taille d'enregistrement (en octets)	Plage
Integer	2	-32 768 à 32 767.
Long (entier long)	4	-2 147 483 648 à 2 147 483 647
Single (valeur à virgule flottante en simple précision)	4	-3,402823E38 à -1,401298E-45 pour les valeurs négatives ; 1,401298E-45 à 3,402823E38 pour les valeurs positives et 0.
Currency (entier à l'échelle)	8	-922 337 203 685 477,5808 à 922 337 203 685 477,5807
String (chaîne de caractères)	1 par caractère	0 à environ 65535 octets

B - LES OPERATEURS :

Ils permettent de relier des variables entre elles afin d'exécuter l'opération souhaité.

1 - OPERATEURS ARITHMETIQUES

Opérateur	Exemple
Addition	A+B
Soustraction	A-7
Multiplication	C*D
Division	A/F
Puissance	X^2

2 - OPERATEURS DE COMPARAISON

Les opérateurs de comparaison permettent de comparer les variables en fonction de leur importance.

Opérateur	Signification
<	Plus petit que
<=	Plus petit ou égal à
=	Egal
>	Plus grand que
>=	Plus grand ou égal à
<>	Différent de

3 - OPERATEURS LOGIQUES

Les opérateurs logiques se rencontrent souvent dans le contexte des opérations de comparaison.

Opérateur	Signification
AND	Et
OR	Ou

C - Structure alternative

Instruction : If...

Then...

Else...

End If

Permet d'exécuter conditionnellement des instructions en fonction du résultat d'une expression.

Syntaxe :

```
If condition
    Then
    [bloc d'instructions A]
    Else
    [bloc d'instructions B]
End If
```

Exemple :

```
If CA > 25000 then
    COM = CA * 0.15
Else
    COM = CA * 0.05
End If
```

Remarque : il est possible d'imbriquer plusieurs conditions les unes dans les autres.

D - Instruction Rem

Utilisée pour inclure des commentaires explicatifs dans un programme.

Syntaxe : REM suivi de la remarque

Exemple :

```
REM Programme permettant la soustraction de deux entiers
' Déclarations des variables
' Permet l'affectation d'une valeur dans une variable à l'aide d'une boîte de saisie
```

Remarque : on peut la remplacer par l'apostrophe « ' ».

E - Instruction InputBox

Utilisée pour saisir une donnée dans une variable avec un message d'invite.

Syntaxe : Variable = InputBox («message d'invite »)

Exemple :

```
Nombre_1 = InputBox("Donnez la valeur du 1er nombre")
Nombre_2 = InputBox("Donnez la valeur du 2ème nombre")
```

F - Instruction Debug.Print

Utilisée pour afficher un message ainsi que le contenu d'une variable.

Syntaxe : Debug.print (« message » & variable)

Exemple :

```
Debug.Print "Le total de la soustraction est : " & Total
' Permet l'affichage du résultat dans la fenêtre d'exécution
```

Remarque : l'opérateur & permet d'afficher le message et le contenu de la variable sur la même ligne.
Nota bene: le résultat sera visible dans une fenêtre d'exécution. Celle-ci s'appelle dans le menu « Affichage », « Fenêtre d'exécution » ou par le raccourci clavier « Ctrl+G ». Cela donne le résultat suivant.

Exécution

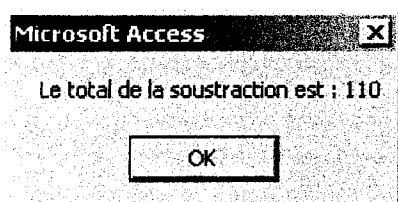
```
Le total de la soustraction est : 110
```

G - L'instruction MsgBox

Celle-ci permet l'affichage du résultat dans une boîte de message (Message Box). C'est plus élégant que l'affichage par la fenêtre d'exécution.

Exemple :

MsgBox ("Le total de la soustraction est : ") & Total
' Permet l'affichage dans une boîte de dialogue



§ 3 - *Les boucles (ou itérations)*

A - BOUCLE : FAIRE TANT QUE

DO WHILE CONDITION
 Bloc d'instructions
LOOP

Répète un bloc d'instructions aussi longtemps qu'une condition est vraie.
Il existe une variante possible :

DO
 Bloc d'instructions
LOOP WHILE CONDITION

B - BOUCLE : RÉPÉTER JUSQU'À

DO UNTIL CONDITION
 Bloc d'instructions
LOOP

Répète un bloc d'instructions jusqu'à ce qu'une condition devienne vraie.
Il existe une variante possible :

DO
 Bloc d'instructions
LOOP WHILE CONDITION

C - BOUCLE : POUR FIN DU POUR

Répète un certain nombre de fois un groupe d'instructions.
Le nombre de fois est connu à l'avance.

For compteur = début To fin
 Bloc d'instructions
NEXT compteur

Compteur : variable numérique utilisée comme compteur de boucles.
Début : valeur initiale du compteur.
Fin : valeur finale du compteur.

A chaque passage de la boucle, sans autre précision, le compteur augmente de 1.

Access : Programmer avec Visual Basic Application (1)1

§ 1 - Présentation générale 1

A - Déclaration du programme 1

B - Ecriture du code..... 1

C - Mise au point du programme 2

D - Exécution d'un programme 2

E - Reprendre le programme ultérieurement 3

§ 2 - Les instructions 3

A - Instruction Dim..... 3

B - LES OPERATEURS : 4

1 - *Opérateurs arithmétiques*..... 4

2 - *Opérateurs de comparaison*..... 4

3 - *Opérateurs logiques*..... 4

C - Structure alternative..... 4

D - Instruction Rem 5

E - Instruction InputBox..... 5

F - Instruction Debug.Print 5

G - L'instruction MsgBox..... 6

§ 3 - Les boucles (ou itérations) 6

A - BOUCLE : FAIRE TANT QUE 6

B - BOUCLE : RÉPÉTER JUSQU'À 6

C - BOUCLE : POUR FIN DU POUR..... 6