A la découverte d ’access

Ecrire vos propres fonctions

MCours.com

G. Gesquière, Gilles.Gesquière@up.univ-mrs.fr

Objectifs

- Savoir créer un module standard
- Savoir écrire des procédures générales Sub et Function
- Savoir utiliser le passage de paramètres
- Savoir utiliser des procédures générales

⇒Vous utiliserez le fichier TP9.mdb

1- Comprendre les modules et les procédures

- Pour l'instant, nous avons écrit tout notre code dans les procédures événementielles associées à des objets.
- Les procédures générales (que nous allons écrire) ne s'exécutent pas automatiquement en réponse aux événements.
- Il y a deux types de procédures générales : **Sub** et **Function** alors que toutes les procédures événementielles sont de type Sub.

2- Pourquoi créer des procédures générales

- Utiliser des procédures générales pour :
  - Exécuter des opérations complexes qui ne rentrent pas dans une expression
  - réutiliser du code pour éviter de répéter une tâche
  - Diviser des tâches de programmation en unités plus facilement gérables

3- Modules standard et modules de formulaires

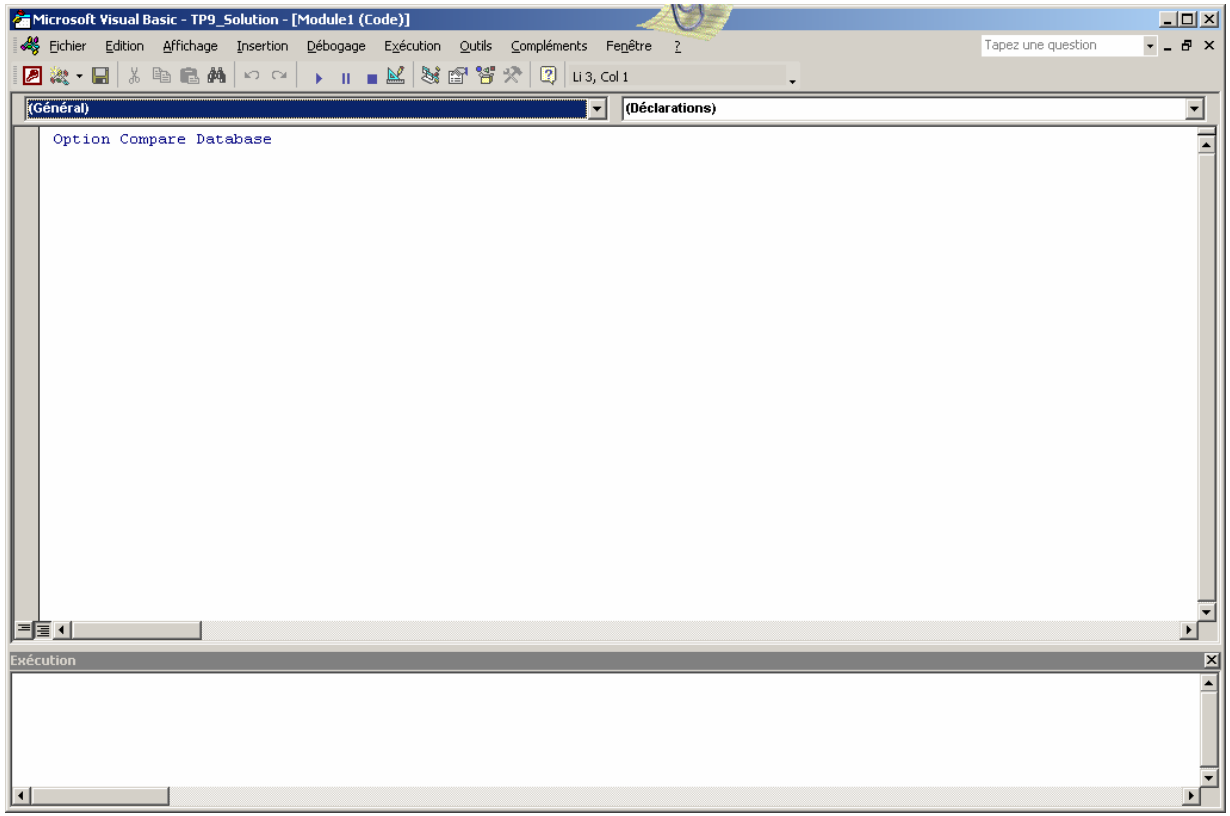
- Il est possible de mettre des procédures générales dans un module de formulaire (ou d'état)
  - Elles sont alors accessibles à toutes les procédures événementielles du formulaire (ou de l'état)
- Il est possible de mettre des procédures générales dans un module standard
  - Elles sont alors accessibles à toute l'application.

4- Créer des procédures générales dans un module standard

- Une application peut contenir plusieurs modules et chacun d'eux peut contenir plusieurs procédures
- Cela permet de créer des modules spécialisés réutilisables dans de multiples applications (il suffit d'importer le module)

5- Créer un module standard

- Fenêtre base de données / Onglet Modules / Bouton Nouveau
- La fenêtre Module1 : Module s'affiche

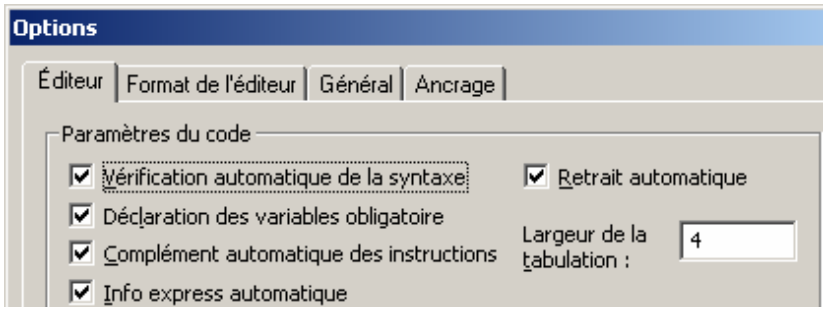


5- Créer un module standard

- « Option Compare Database » ne peut être utilisée qu'au sein de Microsoft Access. Elle génère des comparaisons de chaînes fondées sur l'ordre de tri déterminé par l'ID des paramètres nationaux de la base de données dans laquelle a lieu la comparaison
- Avec l'instruction Option Explicit, toutes les variables doivent être déclarées explicitement à l'aide d'une instruction Dim, Private, Public, ReDim, Static
- Si vous tentez d'utiliser une variable non déclarée, une erreur se produit lors de la compilation

5- Créer un module standard

- Remarque :
  - Indiquer toujours l'option Explicit
  - Menu Outil / Option … / Dans l'onglet Editeur / activer Déclaration des variables obligatoire



- A l'ouverture, un module est vide ou presque.

6- Déclarer des valeurs constantes

- Déclaration :

  Const conNomApp = « Gestion des clients »

- Cette instruction rend accessible la constante conNomApp à tout votre module.
- Si vous souhaitez qu'elle soit connue de votre application entière, il faut ajouter le mot Public devant le mot Const

7- Créer une procédure Sub

- Si pour les procédures événementielles liées aux objets de votre applications les noms et les paramètres vous sont imposés, là dans votre module, c'est vous qui devez décider du nom et des paramètres éventuellement nécessaires
- Créer la procédure AfficheMessage qui prendra en charge l'affichage des messages que nous adresserons aux utilisateurs
  - Cliquez sur le bouton « Insérer une procédure » dans le menu Insérer de la barre d'outils de la fenêtre VB
  - Nom « AfficheMessage » / type procédure (Sub)
  - Laisser étendue sur Publique
  - Appuyer sur Ok

7- Créer une procédure Sub

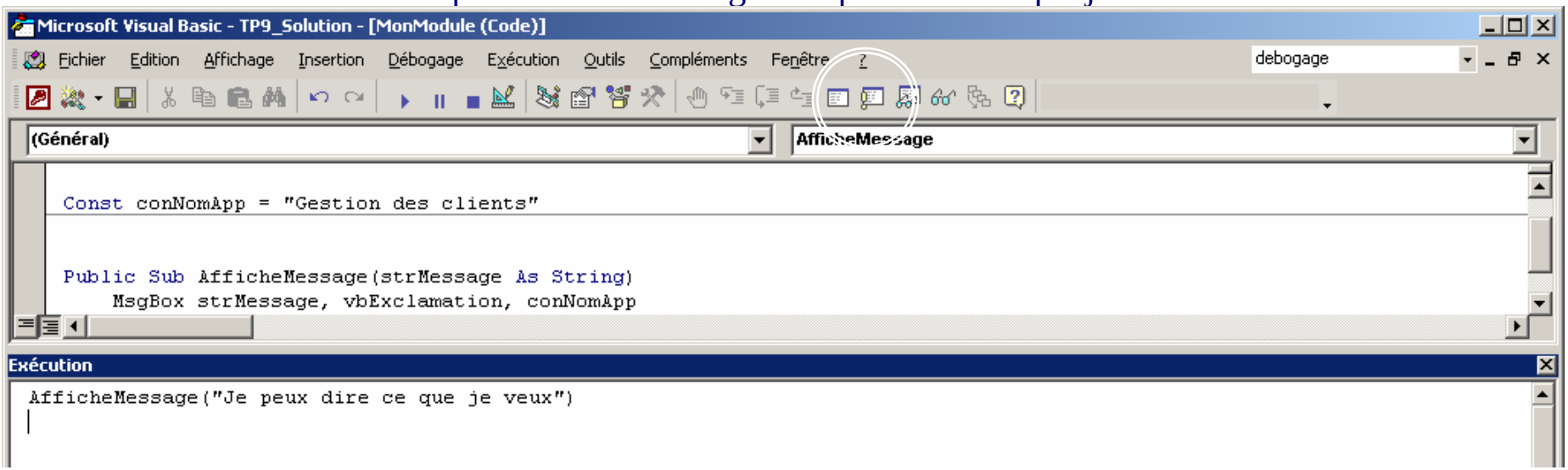
- Taper l'instruction suivante :

  MsgBox « Voici mon message », vbExclamation, conNomApp

- Tester la procédure en utilisant la fenêtre de débogage
  - Cliquer sur le bouton de lancement du code (ou sélectionner l'option Exécuter du menu Exécution
  - Choisir la procédure AfficheMessage
  - Votre procédure s'exécute

8- Ajouter des arguments à une procédure

- Pour passer le message en paramètres
  - Entrez strMessage As String dans les parenthèses de l'instruction Sub
  - Modifier la ligne MsgBox en remplaçant la chaîne « voici mon message » par le nom de la variable qui contiendra le texte à afficher. Cette variable est strMessage
  - Tester avec la fenêtre débogage / taper le nom de la procédure suivi du message que vous désirez afficher
    - Exemple : AfficheMessage « Je peux dire ce que je veux »



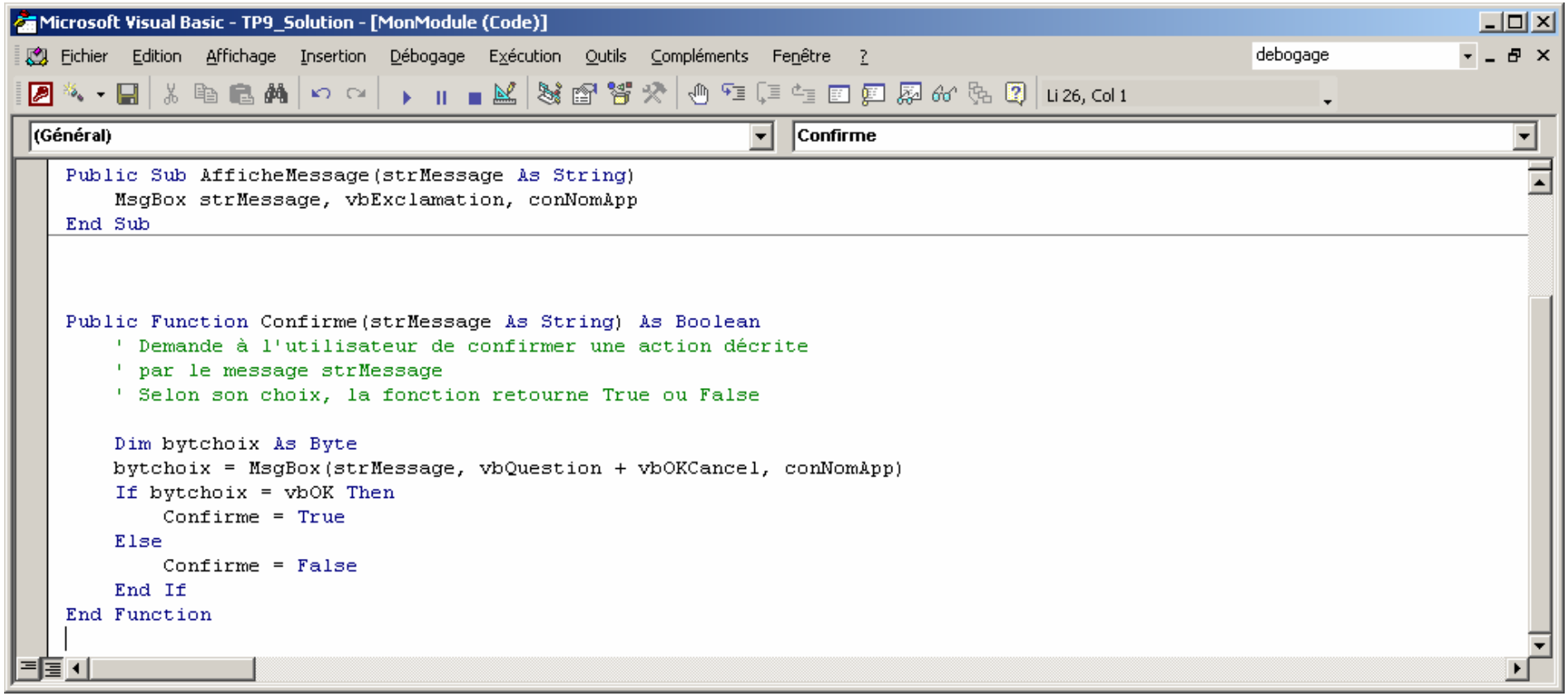
9- Créer une fonction

- Une fonction permet de retourner une valeur
- Dans le code, il doit toujours exister une ligne ayant la syntaxe :

  NomDeLaFonction = ValeurARetourner

- Si nous voulons créer une fonction qui pose des questions :
  - Fermer la fenêtre de débogage
  - Insérer une procédure
  - Nom : Confirme / Type function / Bouton Ok
  - Taper le code du transparent suivant

9- Créer une fonction



G. Gesquière

9- Créer une fonction

- Tester dans la fenêtre de débogage
  - Taper l'instruction suivante :

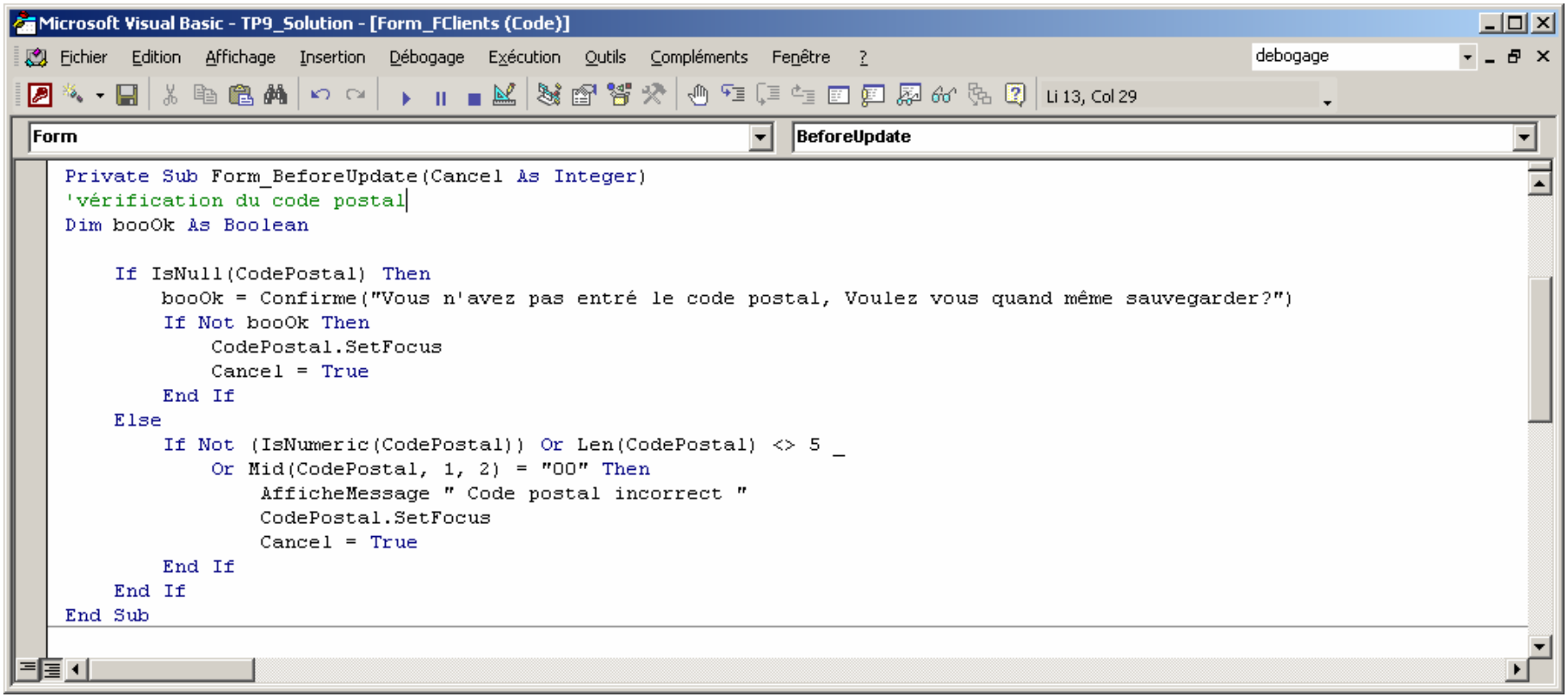
    ?Confirme(« Est-ce Ok ?»)

    - La boîte de dialogue s'affiche / Cliquer sur Ok / la valeur vrai s'affiche dans la fenêtre débogage
    - Si vous recommencez en choisissant le bouton Annuler, la fonction retourne Faux et affiche cette valeur dans la fenêtre de débogage

10- Utiliser des procédures générales dans un formulaire

- Les procédures étant déclarées en public, elles sont consultables par toute l'application
- Utilisation de la fonction Confirme
  - Onglet Formulaire / Sélectionner FClients / Code
  - liste de gauche du module : sélectionner l'objet Form / Proc : BeforeUpdate
  - Modifier le code comme sur le transparent suivant

10- Utiliser des procédures générales dans un formulaire



MCours.com

G. Gesquière