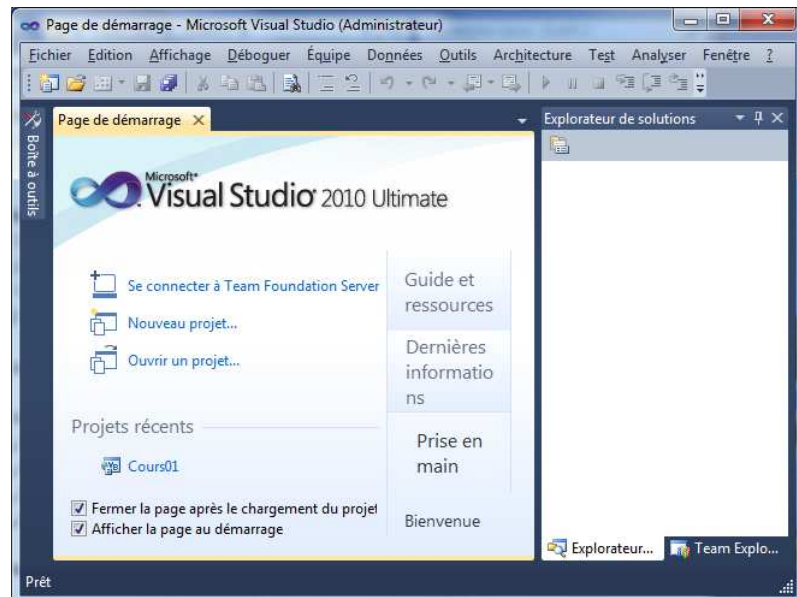


L'environnement de développement de VB.NET 2010

1 La page de démarrage



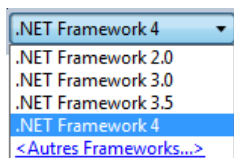
2 Les types d'application

On peut créer plusieurs types d'applications avec VB.NET : application Windows, bibliothèque de classes, bibliothèque de contrôles windows, application web, service web, bibliothèque de contrôles web, application console et service windows.

On sélectionne :

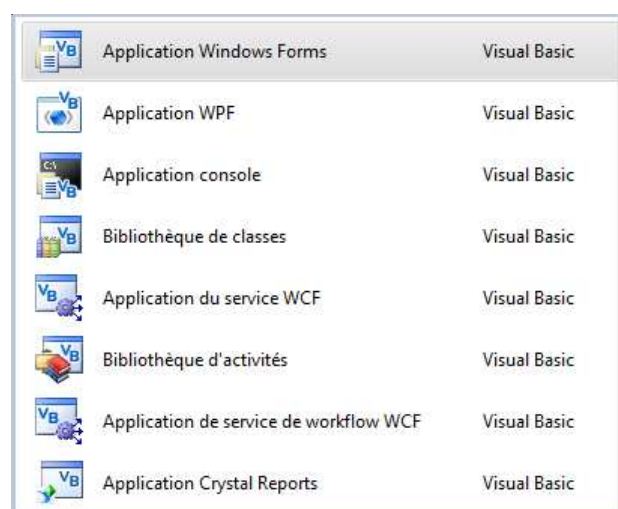
"Le type de l'application"

Et le framework :

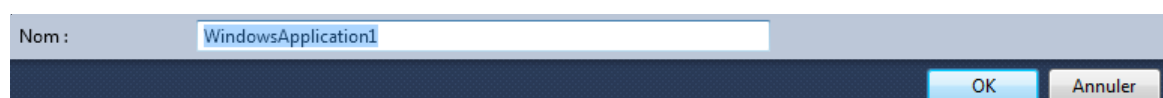


On donne un nom au projet :

"MonAppli" (par exemple)



Sauvegarder le projet en local sur le disque :



2.1 Application Windows



Une application Windows est la nouvelle manière de présenter une interface graphique utilisateur pour les applications Win32 et d'utiliser les formulaires Windows (Fenêtres).

2.2 Application Web



Les applications Web font partie d'ASP.NET ; au lieu de s'appuyer sur des formulaires Windows comme interface utilisateur, ces applications utilisent des formulaires Web, permettant ainsi de créer une interface web aussi facilement qu'une interface de bureau. Le principal concept des formulaires Web consiste à séparer le code d'affichage du code de logique.

Cette amélioration est obtenue en divisant un formulaire Web donné en deux parties : un modèle HTML contrôlant l'affichage de l'interface graphique et un composant gérant toute la logique de l'application. Les contrôles d'un formulaire Web s'exécutent sur le serveur, mais s'affichent sur le navigateur client.

2.3 Services Web



Les services Web permettent d'exposer nos composants à tout internaute grâce au protocole HTTP et ce indépendamment de la plate forme utilisée.

2.4 Application Console



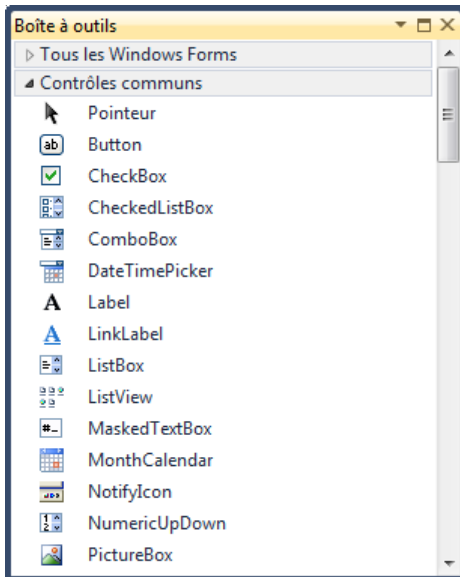
La plupart des applications d'administration de Windows fonctionne en mode console (l'ancien mode MS-DOS). Maintenant en VB.NET on peut créer facilement des applications fonctionnant avec une interface de type caractère.

3 Fonctionnalités de l'IDE

Affichage	Projet	Générer	Déboguer	Équipe	Données
Code					F7
Concepteur					Maj+F7
Explorateur de solutions					Ctrl+Alt+L
Team Explorer					Ctrl+*, Ctrl+M
Explorateur de serveurs					Ctrl+Alt+S
Navigateur de l'architecture					Ctrl+*, Ctrl+R
Explorateur d'objets					F2
Liste d'erreurs					Ctrl+W, Ctrl+E
Page de démarrage					
Boîte à outils					Ctrl+Alt+X
Autres fenêtres ▶					
Barres d'outils ▶					
Plein écran					Maj+Alt+Entrée
Ordre de tabulation					
Fenêtre Propriétés					F4
Pages de propriétés					Maj+F4

3.1) La boîte à outils

Ctrl+Alt+X



Elle est semblable à celle de VB6. Les contrôles sont regroupés par type dans des onglets à ajouter ou à supprimer selon les besoins. Il s'utilisent de la même façon : sélectionnez un contrôle en cliquant dessus, puis faites-le glisser sur le formulaire.

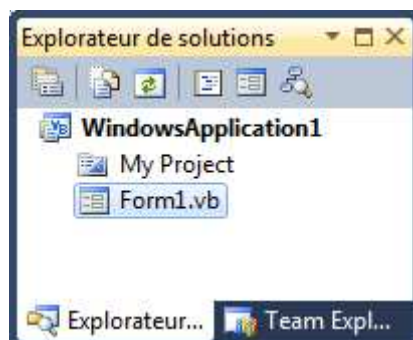
Elle renferme tous les contrôles standard et, éventuellement, des contrôles personnalisés que vous avez créés ou téléchargés.

- Ils sont classés aussi par catégories «**refermables**» et «**étendables**» grâce au principe du signe «**+**» et «**-**» afin de faciliter votre recherche d'un contrôle précis.

- La boîte à outils peut être affichée en permanence ou masquée lorsque le curseur se trouve dans l'environnement de développement. Pour ce faire, utilisez la petite « punaise » près de la croix de fermeture :

- si elle est pointée vers le bas, la boîte à outils est figée
- si elle est horizontale, la boîte à outils se masquera automatiquement quand le focus la quittera

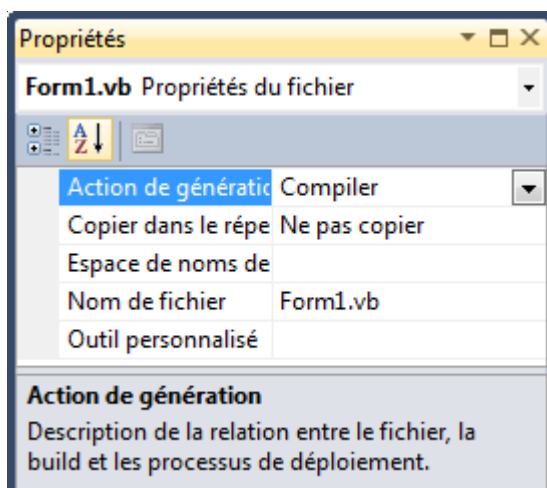
3.2) L'explorateur de solutions **Ctrl+Alt+L**



L'explorateur de solutions permet de visualiser exactement le contenu de la solution en cours. On peut regrouper plusieurs projets dans une solution, un peu comme fonctionnaient les groupes de projets dans VB6. La fenêtre de l'explorateur de solutions propose un affichage au niveau des fichiers et de ce qui se passe dans la solution.

3.3) Fenêtre Propriétés

F4



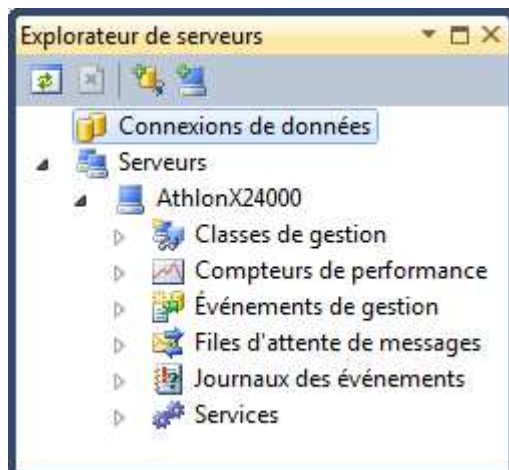
La fenêtre Propriétés permet de regrouper les propriétés par catégorie. On trouve notamment Apparence, Comportement, Design, et Disposition. Toutes ces catégories peuvent être réduites.

- L'icône A-Z classe toutes les propriétés par ordre alphabétique sans distinction d'appartenance à une catégorie.

- Par contre, en cliquant sur l'icône de gauche, les propriétés seront classées par ordre alphabétique de catégories et conserveront l'ordre A-Z à l'intérieur de leur catégorie.

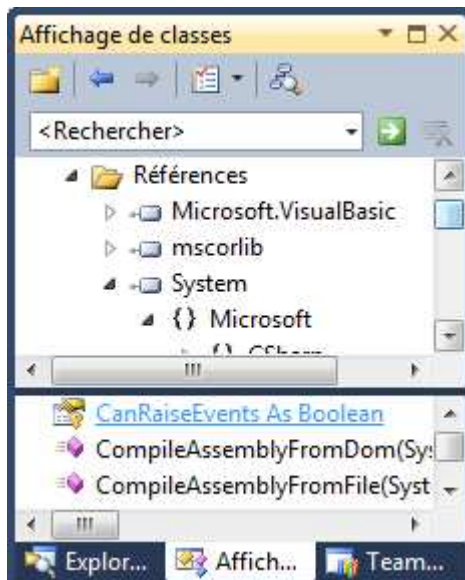
- L'icône suivante affiche le choix des événements du contrôle concerné et la dernière revient à l'affichage des propriétés.

3.4) L'explorateur de serveur

Ctrl+Alt+S

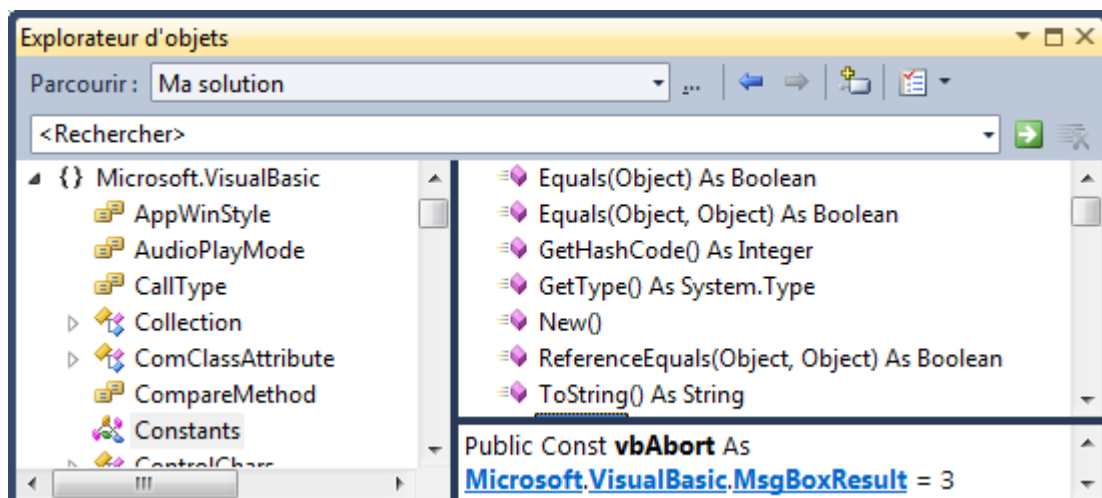
L'explorateur de serveur permet d'afficher et de modifier les informations d'authentification utilisées pour accéder à un serveur et à ses bases de données déterminent les types de sources de données. Plus le niveau d'accès d'un utilisateur est important, plus la quantité de données susceptibles d'être extraites et utilisées est élevée.

3.5) L'affichage de classes

Ctrl+Maj+C

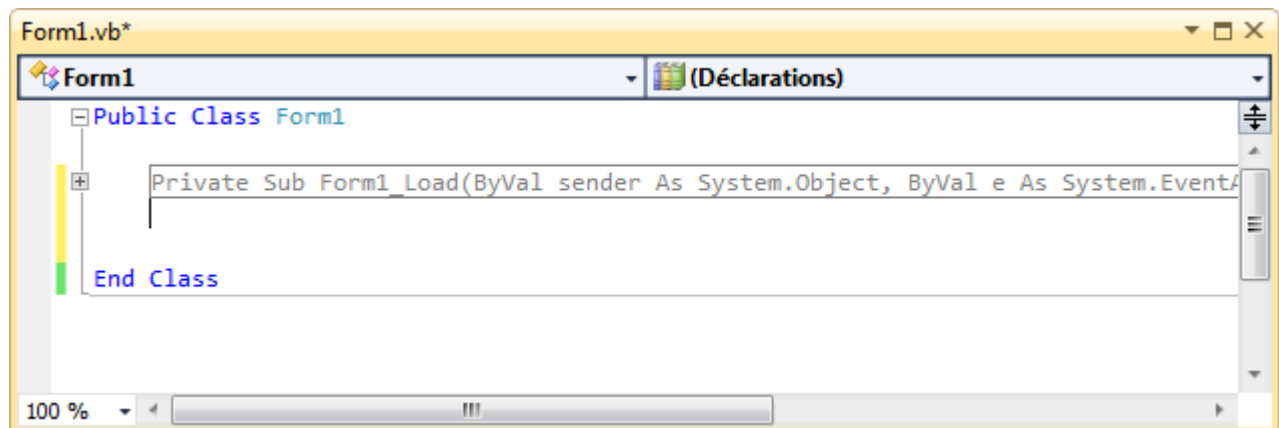
L'affichage de classes permet de voir toutes les classes de la solution en cours, ainsi que leurs propriétés et leurs méthodes.

3.6) L'explorateur d'objet

Ctrl+Alt+J

L'explorateur d'objets vous permet de rechercher et d'examiner des objets (espaces de noms, classes, structures, interfaces, types, énumérateurs...) et leurs membres (propriétés, méthodes, événements, variables, constantes, éléments d'énumérateurs...) à partir de divers composants.

3.7) Fenêtre de code



Il est désormais possible de réduire toutes les fonctions et sous-programmes à une ligne de déclaration uniquement. Pour les fichiers sources plus importants, le fait de se concentrer sur une seule fonction, un seul sous-programme ou une seule classe peut constituer une aide au développement.

3.8) La liste des erreurs

Ctrl+W puis Ctrl+E

Situés en bas d'écran, on trouve un outil qui vous permettra de suivre la « bonne tenue » du code. Voyons un exemple de la gestion des "erreurs" et des "avertissements" :

- Lors de la compilation (Building) ou de l'écriture de l'application, les erreurs importantes sont listées dans cet onglet. Ce type d'erreur ne permet pas l'exécution du programme. Par défaut, VB.NET exécute alors la dernière compilation sans erreurs.

Liste d'erreurs					
<div> 1 erreur 0 avertissements 0 messages </div>					
	Description	Fichier	Ligne	Colonne	Projet
1	Fin d'instruction attendue.	Form1.vb	6	12	WindowsApplication1

- Lorsqu'une erreur de syntaxe est détectée, un « warning » sera levé mais n'empêche pas le bon déroulement de la compilation et le code s'exécutera. Il faut éviter de laisser s'accumuler ce type « d'attention » car, comme dans l'exemple donné, cela permet aussi d'alléger le code en supprimant des variables déclarées mais non utilisées.

Liste d'erreurs					
<div> 0 erreurs 1 avertissement 0 messages </div>					
	Description	Fichier	Ligne	Colonne	Projet
1	Variable locale inutilisée : 'i'.	Form1.vb	4	13	WindowsApplication1

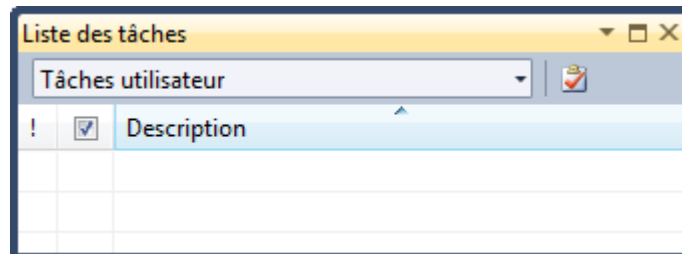
3.9) La liste des tâches

Ctrl+Alt+K

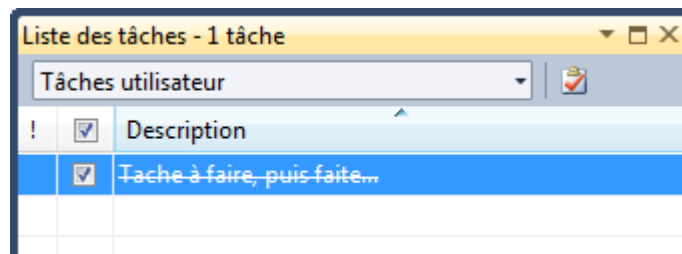
Deux possibilités sont offertes pour commenter le travail :

Dans la partie commentaires, on place un «TODO» (choses à faire) comme dans l'exemple :
'TODO test de la liste des taches

Dans les colonnes File et Line s'affichent automatiquement l'endroit auquel se réfère le commentaire.



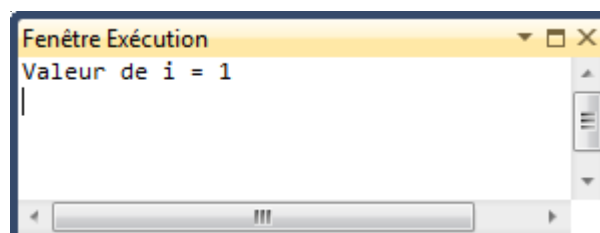
Dans la partie User Tasks (Tâches utilisateurs), on place les descriptions. Il suffit de cliquer sur l'icône et de taper une phrase.



3.10) La fenêtre d'exécution

Ctrl+G

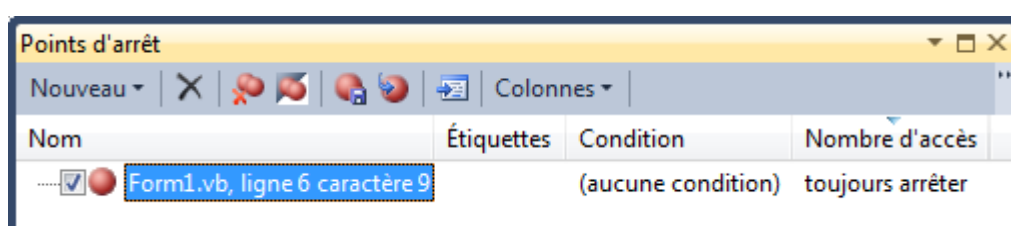
En utilisant la l'instruction Debug . Print, on peut afficher un message dans la " Fenêtre Exécution" :



3.11) Les points d'arrêts

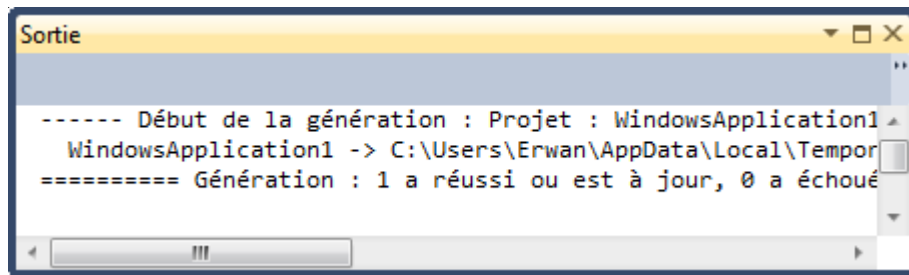
Ctrl+Alt+B

Permet de visualiser les points d'arrêts qui ont été insérés dans le code (comme en VB6, avec un clic droit dans la première colonne.



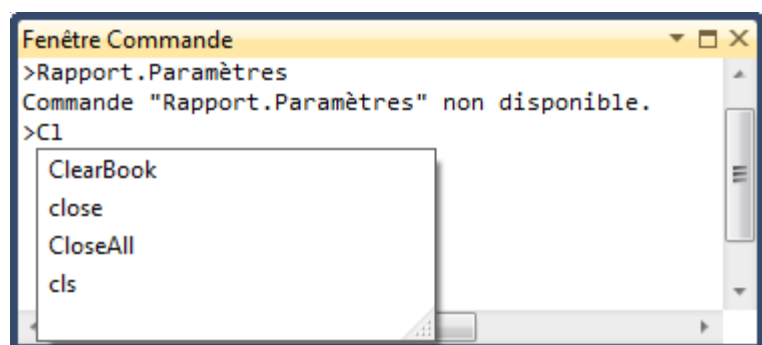
3.12) Fenêtre de sortie **Ctrl+Alt+O**

Permet d'afficher les messages envoyés par le compilateur au système d'exploitation.



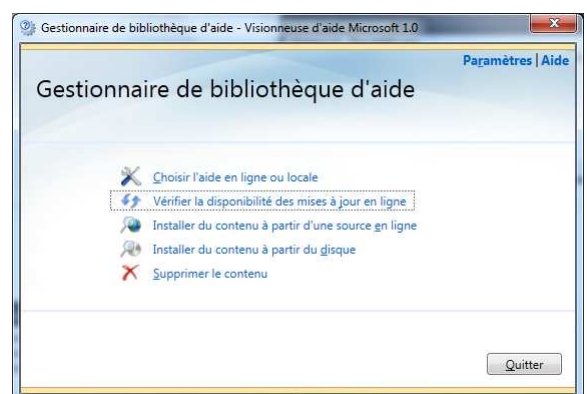
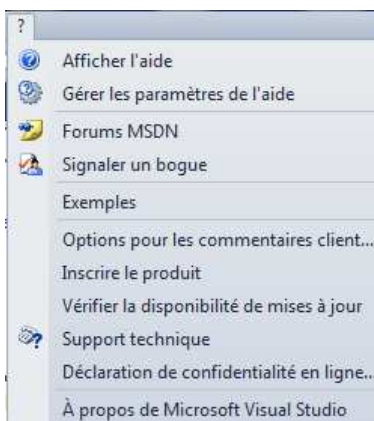
3.13) Fenêtre de commandes **Ctrl+Alt+A**

Permet d'exécuter une commande sur l'IDE



3.14) L'option aide " ? "










L'aide dynamique **F1**



La fenêtre de l'aide dynamique permet un affichage rapide des rubriques d'aides, celle-ci change en fonction des éléments sélectionnés par l'utilisateur.



4 - Nomenclature des objets visuels de VB.NET

btn	Button	
lst	ListBox	
chk	CheckBox	
cbo	ComboBox	
grp	GroupBox	
frm	Form	
lbl	Label	
txt	Textbox	
rb	Radiobutton	

5 - Déclaration de variable

Exemples : `Dim intX As Integer`
`Dim resources As System.Resources.ResourceManager = New`
`Private components As System.ComponentModel.IContainer`

Dim : Utilisé au niveau du module, de la classe, de la structure, de la procédure ou du bloc pour déclarer et allouer un espace de stockage pour les variables.

Le mot clé **Private** est un modificateur de visibilité qui s'applique à un champ (propriété ou méthode) d'une classe. On dit alors que le champ est privé. Un champ privé n'est visible que depuis sa propre classe. Il n'est visible nulle part ailleurs et en particulier pas dans les sous-classes.

Lorsque le mot clé **Public** s'applique à une classe, une interface ou à un champ (attribut ou méthode) d'une classe, on dit alors que la classe, l'interface ou le champ est public. Une classe ou une interface publique est visible de partout.

Les variables déclarées avec le mot clé **Static** continuent à exister et conservent leurs valeurs les plus récentes après l'arrêt de la procédure dans laquelle elles sont déclarées.

Les variables déclarées avec le mot clé **Protected** ont un accès protégé. Elles sont accessibles seulement à partir de leur propre classe ou d'une classe dérivée.

New permet la création immédiate d'un objet. Si on utilise **New** pour déclarer la variable objet, une nouvelle **instance** de l'objet est créée lors de l'exécution de l'instruction **Dim**.

Les membres **Private** ne sont accessibles que par les méthodes de la même classe.

Il est plus fréquent de rencontrer des propriétés **Private** que des méthodes **Private**.