COURS WINDEV 8-1

15/02/2015

Les variables de classe et méthodes de classe

Création d'une classe.

Instanciation d'objets.

Variables dites statiques

Méthode de classe dite statique.

Cours WinDev 8-1

VERSION 19

Problématique.

Chaque instance de classe (chaque objet) possède son propre jeu de tous les champs définis dans la classe.

Prenons par exemple la classe employé définie ainsi :

```
classe Employé
        privé
            numéro : entier
            nom : chaîne
            Prenon : réel
        public
            Employé(numéro, nom, prenom) //Constructeur
            ….
fin classe
```

Créons 2 objets, instances de la classe Employé :

emp1 ← new Employé(100,'Baptiste','Jean-Luc')

emp2 ← new Employé(200,'Convenant','Jean-Claude')

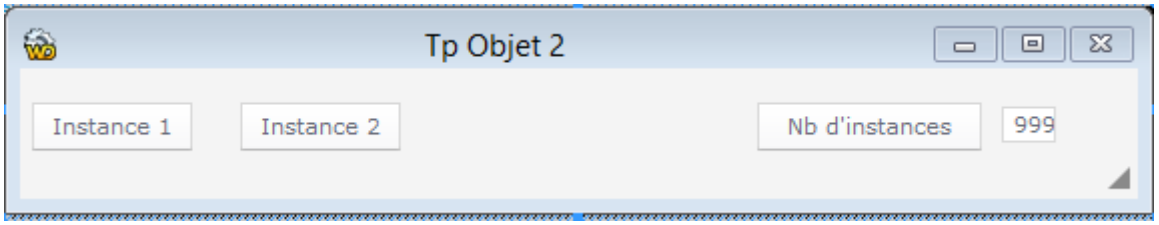
En mémoire, chacun des objets dispose de ses propres variables :

Emp1	Emp2
100 Baptiste Jean-Luc	200 Convenant Jean-Claude

La grande question est donc : Comment connaître le nombre d'instances de la classe employé ?

Illustrons la problématique avec WinDev.

Créez un nouveau projet nommé TpObjet2, sans analyse avec une fenêtre nommée départ identique à celle-ci :

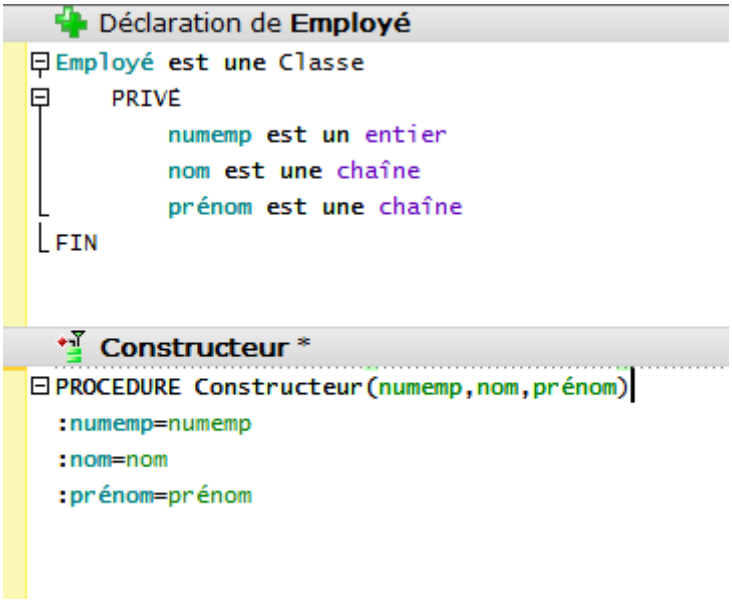


Contenu :

3 boutons "Binstance1", "Binstance2", "Bnbinstances"

1 champ de saisie nommé "Cnbinstances" de type numérique sans libellé.

Créez la classe employé :



Maintenant passons au code du bouton **Binstance1**

```
empl1 est un Employé(100,"Baptiste", "Jean-Luc")
```

Maintenant passons au code du bouton **Binstance2**

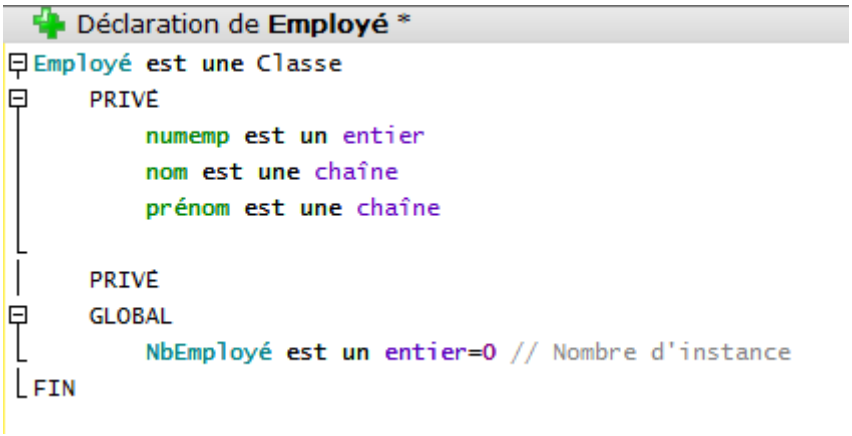
```
Empl2 est un Employé(200,"Convenant", "Jean-Claude")
```

Dans l'état actuel des choses si vous cliquez sur l'un des 2 boutons chargé de créer une nouvelle instance le fonctionnement est assez "silencieux" car rien ne s'affichera à l'écran. Les 2 objets empl1 et empl2 sont créés en mémoire, les constructeurs ont rempli leurs offices en affectant des données aux variables.

Maintenant, je vous rappelle la question, comment savoir nombre d'instances de ma classe en mémoire ? La résolution de ce problème passe par la création de ce que l'on nomme une variable de classe. Le principe est simple : une variable de classe est une variable globale accessible par les objets et indépendante d'eux.

On met en œuvre ?

Il nous faut donc modifier la déclaration d'employé pour créer cette nouvelle variable.



Nbemploye est privé à la classe, donc seulement accessible par un accesseur et global à toutes les instances. Pigé ? .

Un peu de définitions :

Variables de classe – définition :

Une classe va pouvoir comporter des variables d'instance, qui sont les variables propres à chaque objet et des variables de classe qui sont propres à la classe.

Les variables de classe existent alors **en un seul exemplaire.** Elles sont chargées en mémoire dès le chargement de la classe avant même qu'un objet ait été créé ! Une telle variable doit être précédée du mot clef **static** en C# ou en Java ou **Privé et globale** et WinDev

La protection des données membres.

L'un des aspects essentiels du concept « orienté objet » est l'encapsulation, qui consiste à définir des étiquettes pour les données membres et les fonctions membres afin de préciser si celles-ci sont accessibles à partir d'autres classes ou non...

De cette manière, des données membres portant l'étiquette *PRIVE* ne peuvent pas être manipulées directement par les fonctions membres des autres classes. Ainsi, pour pouvoir manipuler ces données

membres, le créateur de la classe (vous en l'occurrence) doit prévoir des fonctions membres spéciales portant l'étiquette *public*, permettant de manipuler ces données.

- Les fonctions membres permettant d'accéder aux données membres sont appelées **accesseurs,** parfois *getter* (appellation d'origine anglophone)
- Les fonctions membres permettant de modifier les données membres sont appelées **mutateurs,** parfois *setter* (appellation d'origine anglophone)

Le constructeur va être chargé d'incrémenter notre variable et il nous faudra aussi créer une méthode publique dite "Accesseur" qui elle seule aura le droit d'accéder à ce membre privé.

Modifions le code en conséquence.

```
Constructeur *
PROCEDURE Constructeur(numemp,nom,prénom)
 :numemp=numemp
 :nom=nom
 :prénom=prénom

 ::NbEmployé=::NbEmployé+1
```

Remarquez les "::" devant NbEmployé cela indique que vous accédez à une variable de classe.

Voici l'accesseur que vous allez créer :

```
Méthode GetNbemploye *
FONCTION GLOBAL Employé::GetNbemploye
 RENVOYER ::NbEmployé
```

En fait son rôle est de renvoyer NbEmploye. Pour bien montrer que c'est une variable de classe mettez le nom de la classe devant le nom de la fonction.

Maintenant, voyons le code du bouton **Bnbinstances**:

```
Cnbinstances=Employé::GetNbemploye()
```

Le champ cnbinstance va recevoir le renvoi de la méthode de classe GetNbemploye.

Allez-y cliquez sur le bouton instance1 puis instance2 puis enfin sur Nb d'instance logiquement votre champ Cnlnstance devrait contenir le chiffre 2. Si ce n'est pas le cas….Reprenez le support !

A bientôt pour le prochain support objet.