

Créer des objets en JAVA

Date de dernière mise à jour : 27/06/2007 à 19:36

Source : <http://www.vulgarisation-informatique.com/creer-objets-java.php>.

Distribution interdite sans accord écrit d'Anthony ROSSETTO (<http://www.vulgarisation-informatique.com/contact.php>)

Création d'un objet (instanciation d'une classe) :

Lorsque l'on crée un objet, étant donné que c'est toujours la même classe qui est utilisée pour le même type d'objets (si vous avez une classe rectangle et que vous souhaitez créer deux rectangles, vous n'utiliserez qu'une seule classe rectangle mais deux objets de type Rectangle), on parle d'instance. Une instance de classe correspond à un objet. Chaque objet doit être stocké dans une variable différente.

Pour créer un objet, il faut tout d'abord déclarer une variable du type de la classe. Si vous avez une classe qui se nomme **Test**, votre variable sera de type **Test**.

```
public
class
Test{
    public
    Test(
    )
    {
        System.out.println(
        &quot;L'objet a été créé&quot;
    )
    ; }
}
```

Nous allons maintenant voir comment créer un objet de type Test :

```
Test mon_objet =
new
Test(
)
;
```

Comme vous pouvez le voir, on utilise le mot clé **new** pour créer un nouvel objet.

Séparation entre la déclaration et la définition :

Vous pouvez déclarer un attribut de type objet et le définir ensuite, comme n'importe quel autre attribut. Voici ce que ça donne au niveau du code :

```
Test mon_objet;//ici n'importe quel code
mon_objet =
new
Test(
)
;
```

Vous pouvez donc créer une quantité d'objets infinie. Pour créer deux objets différents, on fera par exemple ceci :

```
Test mon_objet;Test mon_objet2;
```

```
//ici n'importe quel code
```

```
mon_objet =
```

```
new
```

```
Test()
```

```
;mon_objet2 =
```

```
new
```

```
Test()
```

```
;
```

Ensuite en fonction du code de votre classe, vous attribuerez des valeurs identiques ou différentes aux objets, le code de la classe restant le même.

Création de deux objets identiques :

Lorsque vous souhaitez créer deux variables qui pointent sur le même objet, c'est très simple. En effet, les variables d'objets ne sont que des pointeurs sur l'objet en lui-même. Pour faire simple, voici ce code :

```
Test mon_objet;//ici n'importe quel code
```

```
mon_objet =
```

```
new
```

```
Test()
```

```
;
```

La variable "mon_objet" ne contient pas l'objet en lui-même mais une adresse vers laquelle pointer pour modifier les propriétés de cet objet. C'est à dire que si nous écrivons ceci :

```
Test mon_objet;Test mon_objet2;
```

```
//ici n'importe quel code
```

```
mon_objet =
```

```
new
```

```
Test()
```

```
;mon_objet2 =
```

```
mon_objet;
```

On aura réalisé une duplication des deux variables, seulement comme la variable "mon_objet" pointe vers l'objet créé auparavant et que la variable "mon_objet2" est une copie de la précédente variable, la variable "mon_objet2" pointera en réalité sur l'objet créé auparavant. Vous n'aurez donc pas créé deux objets mais deux références qui pointent sur un seul objet. Pour créer deux objets identiques, on utilise une méthode spéciale appelée clone(). Le code devient le suivant :

```
Test mon_objet;Test mon_objet2;//ici n'importe quel code
```

```
mon_objet =
```

```
new
```

```
Test(
```

```
)
```

```
;mon_objet2 =
```

```
mon_objet.clone(
```

```
)
```

```
;
```

Cette fois-ci, si vous modifiez une propriété de "mon_objet2", "mon_objet" ne sera pas modifié, contrairement à ce qu'il se serait passé auparavant.

Source : <http://www.vulgarisation-informatique.com/creer-objets-java.php>.

Distribution interdite sans accord écrit d'Anthony ROSSETTO (<http://www.vulgarisation-informatique.com/contact.php>)