



JAVA

Super-classes Object

Super-classe Object

En Java, toute classe définie hérite de la classe `Object`. Une variable de type `Object` peut être utilisée pour référencer un objet de type quelconque.

Exemple

```
Point p = new Point(1,3);  
Object o;  
o = p;
```

```
((Point)o).affiche();  
Point p1 = (Point) o;  
p1.affiche();
```

Attention. L'instruction `o.affiche()`; provoque une erreur car il n'y a pas de méthode `affiche` définie dans la classe `Object`. Il faut alors faire une conversion (cast) explicite comme le montre l'exemple ci-dessus.

Super-classe Object

La classe `Object` possède quelques méthodes, dont notamment :

- la méthode `public String toString()` qui renvoie, par défaut, une chaîne de caractères contenant le nom de la classe et l'adresse mémoire de l'objet (adresse précédée de `@`).

```
Point a = new Point(1,2);
System.out.println(a.toString()); //affiche Point@fc1aedf
System.out.println(a); //équivalent
```

La méthode `toString` définie par défaut le chaîne à afficher lorsque l'on veut afficher un objet. Plutôt que de définir une méthode d'affichage on redéfinit la méthode `toString` dans la classe souhaitée puis on affiche directement l'objet.

```
public class Point
{
    private int x,y;
    public Point(...){...}
    ...
    public String toString()
    {
        return "Je suis un point de coordonnées "+x+" "+y;
    }
}
```

...

```
Point p = new Point(0,2);
...
System.out.println(p);
```

Super-classe Object

- La méthode `public boolean equals(Object o)` se content de comparer les adresses mémoire des objets référencés.

```
Object o1 = new Point(1,2);
Object o2 = new Point(1,2);
o1.equals(o2); //renvoie false comme réponse
o1.equals(o1); //renvoie true comme réponse
```

Il est également possible de redéfinir/surcharger la méthode `equals` dans ses propres classes :

```
public class Point
{
    private int x,y;

    public Point(...){...}

    ...

    public boolean equals(Point p)
    {
        return ((p.getX()==x) && (p.getY()==y));
    }
}
```

Tester



eclipse