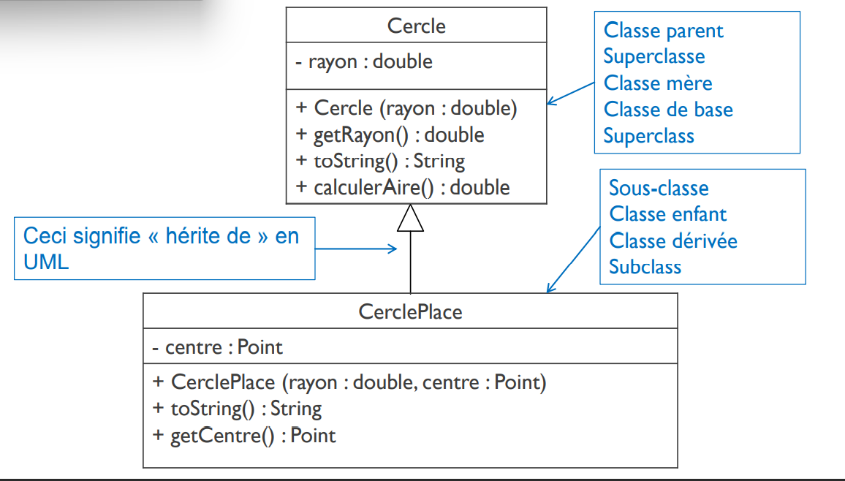
**Héritage**

**L’héritage** : mécanisme permettant de définir une nouvelle classe à partir d’une classe existante, la nouvelle classe héritera de tout les attributs et méthode de la classe parents. On l’utilise quand une classe possède les même attributs et méthode qu’une autre et afin d’éviter la duplication.



**UML** : L’héritage se caractérise par une flèche avec un triangle au bout allant vers la classe parents

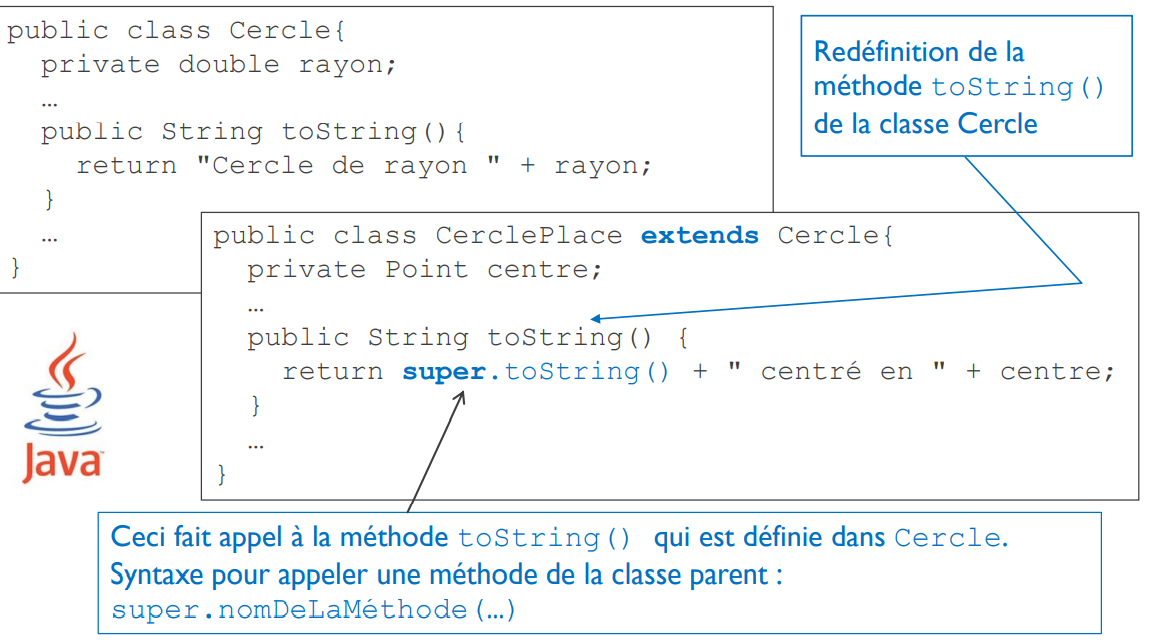
**JAVA** : on place extends ‘nom de la classe parents’ après le nom de la classe.



**Une classe ne peut hériter que d’une seule classe en java**

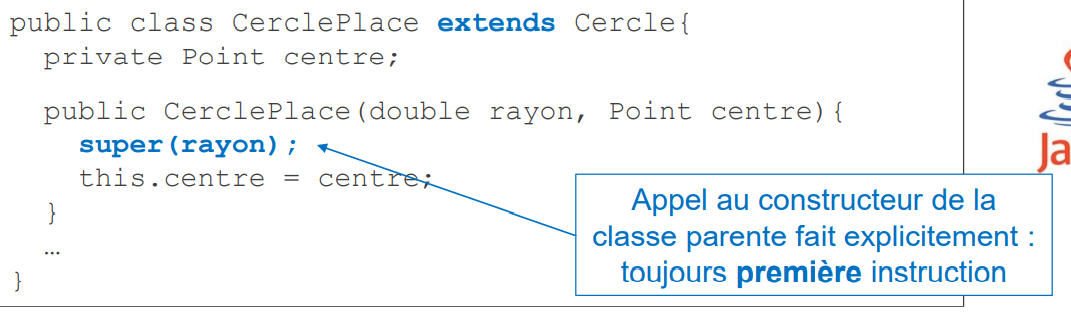
**Plusieurs classes peuvent hériter de la même classe**

**Redéfinir une méthode :** on peut redéfinir une méthode **que si** celle si est visible dans la classe enfant (public) et n’est pas static. **Pour redéfinir une méthode** il faut qu’elle ait la même signature et que le type de retour soir le même

****

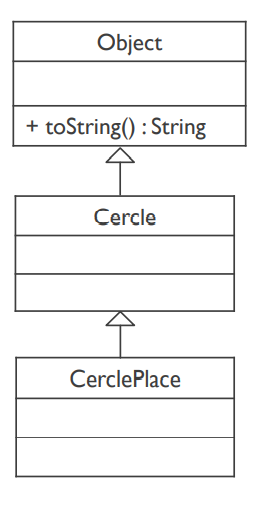
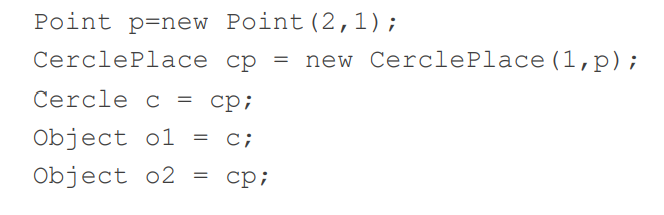
**Chaine de constructeur.**

Pour faire appel au constructeur de la classe parent on doit mettre en première place dans le constructeur **super(attributs du constructeur )** et bien sur les attributs doivent être place en paramètres.

****

**Polymorphisme**

* **Une instance peut être assigné a des variables de plusieurs type (classe).**
* **Seul les méthodes/attributs définis par le type sont utilisable**
* **On peut assigner une variable d’un certain type vers un type du même niveau ou plus haut dans la chaine.**

****

* Le type défini quel méthode/attribut seront disponible
* Lors de l’appel d’un attribut ce seront **celui de son type** qui seront appelé (ex-attributs dans la classe Objet si type objet)
* **Pour l’appel d’une méthode** java cherchera **d’abords dans la classe original avant d’aller chercher dans le type assigner** s’il ne trouve pas et s’il ne trouve dans aucun des 2 c’est une erreur. (ex ci-dessous. Java cherchera d’abord dans la classe CerclePlace et ensuite Cercle s’il ne trouve pas la méthode appelé)

