

## TD7 : Introspection

*Un programme écrit en Java a cette particularité fort intéressante de pouvoir dynamiquement analyser le code d'un objet Java inconnu. Cette propriété, appelée **introspection** ou **réflexivité**, a de nombreux usages. C'est notamment utilisé par Netbeans pour découvrir les propriétés d'un javabean déposé dans la palette.*

### VUE D'ENSEMBLE

---

Nous n'allons pas tout expliquer ici mais plutôt vous donnez une première vue d'ensemble après quoi nous vous renvoyons vers le tutoriel de SUN pour les détails.

- ✓ Il existe une classe nommée **Class** qui représente la classe d'un objet. On peut obtenir un objet de classe **Class** (Et oui !) de plusieurs façons :
  - ✓ via la méthode `getClass()` d'un objet quelconque (ex: `s.getClass()`);
  - ✓ via l'attribut 'class' de la classe (ex: `String.class`) ;
  - ✓ via la méthode 'forName' de la classe 'Class' à laquelle on donne une chaîne contenant le nom de la classe (ex: `Class.forName(« java.lang.String »)`)
- ✓ La classe `Class` fournit de nombreuses méthodes pour se renseigner sur la classe en question. On peut ainsi connaître son **nom**, ses **modificateurs**, sa **classe mère** et les **interfaces** implémentées.
- ✓ On peut aussi avoir le détail de tous les membres (**attributs** et **méthodes**) publics et connaître quels sont les **constructeurs** disponibles.
- ✓ Non seulement on peut se renseigner sur un objet inconnu lors de l'écriture du code mais on peut aussi **manipuler dynamiquement des objets**. On peut ainsi créer un objet d'une classe inconnue au moment de la rédaction du code, accéder et modifier les valeurs des champs de cet objet et appeler ses méthodes.

### APPRENTISSAGE

---

Au total, le sujet est assez simple mais il existe beaucoup de classes à mettre en oeuvre et des consultations fréquentes de l'API sont nécessaires. Nous vous proposons les lectures suivantes

- [http://www.idiom.com/~zilla/Computer/java\\_reflection.html](http://www.idiom.com/~zilla/Computer/java_reflection.html) : pour un petit exemple
- <http://java.sun.com/docs/books/tutorial/reflect/index.html> : le tutoriel de SUN, plus complet

### APPLICATION

---

Une classe possède la propriété « brol » de type T si elle offre les méthodes publiques

```
void setBrol(T) et T getBrol()
```

Écrire une méthode qui reçoit un objet inconnu en paramètre et retourne l'ensemble de ses propriétés (nom + type).