



« Reflection »

CREATION : 2010/05/18

MISE A JOUR : 2010/09/27

Derrière le terme anglais de « reflection » se cachent en Français deux notions : l'**introspection** et l'**intercession**. Un programme JAVA s'exécute dans une machine virtuelle. Il est possible de consulter des informations sur les classes (attributs, méthodes et constructeurs) à l'exécution : cela s'appelle l'**introspection**. Cette notion se rapproche RTTI (Runtime Type Information) du C++, vu en ZZ3. Si le gestionnaire de sécurité le permet, on peut également « modifier » les classes dynamiquement, c'est l'**intercession**.

Pour presque tous les types d'éléments que l'on peut manipuler, il y a une classe correspondante : **Class**, **AccessibleObject**, **Constructor**, **Method** et **Field**...

Partie 1 : Mise en place de quelques classes

Nous allons reprendre la classe **Personne** écrite pour le TP Agenda. Créer une classe **Enfant** qui dérive de **Personne**. Le constructeur de **Personne** doit afficher le nom de la classe de l'objet créé. On peut le faire car **tout objet** connaît sa classe grâce à la méthode **getClass()**. Pour les types primitifs et **void**, il suffit d'utiliser le « champ » **class** pour avoir la même information.

- Charger la classe **Personne** en mémoire grâce à la méthode **forName()**.
- Créer une instance de **Personne** grâce à **newInstance()**. Vérifier que le constructeur sans argument est appelé.
- Vérifier lors de l'exécution que cette instance n'est pas une instance de la classe **Enfant**.
- Charger la classe **Enfant**. Créer un objet de cette classe et refaire les étapes précédentes.
- Afficher le nom de la classe mère de l'objet de classe **Enfant**
- Lister les constructeurs de la classe **Personne**.
- Pouvez-vous instancier un objet de la classe **Personne** avec **newInstance()** et des arguments ?

Partie 2 : Exploration de classes

- Lister les interfaces que la classe **JFrame** implémente.
- Lister les méthodes de la classe **JFrame**
- Lister les attributs/champs de la classe **JFrame**. Vérifier la visibilité des éléments.
- Si vous repérez une méthode ou une classe avec une annotation, vérifiez que l'information est accessible (ou pas).

La suite ?

Nous avons essentiellement fait de l'instrospection. En ce qui concerne l'intercession, la machine virtuelle Java permet de modifier la visibilité d'un attribut si le gestionnaire de sécurité le permet... La manipulation de bytecode n'est possible que par des outils tiers, mais c'est une autre histoire...