Applications réflexives

TD 2 introspection (java.lang.reflect)

javadoc – conformité d'une classe

Introduction

Deux exercices pour la prise en main des fonctionnalités liées à l'introspection au sein de la bibliothèque java.lang.reflect : un embryon de javadoc et la validation structurelle d'un Java Bean. Le fichier **énoncé.jar** contient les 2 exercices à compléter.

Exercice 1 : un embryon de javadoc

L'application AfficheEnTete affiche l'entête d'une classe (ou interface) à partir de l'objet Class. Import, annotations et paramètres de type seront omis dans le corrigé pour simplifier. Une comparaison avec l'en-tête qu'affiche la javadoc permettra de valider votre affichage.

Pour la classe ***java.util.Vector*** par exemple, vous devez afficher (hors indentation)

```
package java.util
public class Vector extends AbstractList
      implements List,RandomAccess,Cloneable,Serializable
```

Pour l'interface ***java.util.List***

```
package java.util
public interface List extends Collection
```

Exercice 2 : validation structurelle d'un javabean

L'application TestBean teste si une classe est conforme à la structure standard d'un javabean (voir cours). En cas de non-conformité, un message explicite devra être porté par l'exception. Vous pouvez prendre la classe Personne du cours pour tester les différents cas d'erreurs. Par exemple, en retirant le setter de porteMonnaie, on devra afficher

```
« il manque le setter de l'attribut porteMonnaie »
```

En option :

Vous pouvez approfondir les méthodes d'introspection de paramétrage de type pour afficher dans l'exercice 1

```
package java.util
public class Vector<E>
       extends AbstractList<E>
       implements List<E>, RandomAccess, Cloneable, Serializable
et la liste des attributs
```