

TP Annotations

- 1) Créer une annotation nommée **@Info** permettant de documenter les classes et interfaces d'un projet en précisant son *auteur*, sa *date* et sa *version*
- 2) Proposer **deux solutions différentes** pour appliquer une liste d'informations (**@Info**) sur une classe ou une interface. Il s'agit de permettre de documenter une classe ou une interface avec la liste de toutes les annotations **@Info** qui décrivent les interventions de maintenance qui ont été effectuées.
- 3) Créer un mécanisme qui alimente les propriétés de type String annotées **@A_INITIALISER** des classes avec une chaîne définie au lancement du programme ou une valeur par défaut définie en attribut de l'annotation.
- 4) Créer un processeur d'annotation qui lève une erreur à la compilation s'il trouve du code (classe, ou méthode) annoté avec **@Dangerous**
- 5) Créer un détecteur d'obsolescence qui fouille une classe pour voir si elle fait référence à des champs de type « déprécié » (annotés **@Deprecated**). Chaque référence à un type déprécié donnera lieu à l'écriture d'un message dans la console.
- 6) Services avec paramètres contrôlés :
IL s'agit de créer un Container pour fournir des instances de Services avec des méthodes dont les valeurs des paramètres seront contrôlés.
Les annotations **@NOT_NULL**, **@NUMBER_RANGE(min, max)** etc. appliquées sur les paramètres des méthodes permettront de contrôler leur valeur au moment de l'exécution.

Classe à Créer :

```
public class ControlledServiceProvider
```

Avec comme Méthode :

```
public static <T> T getControlledService(  
    Class<T> serviceInterface,  
    Class<? extends T> serviceClass)  
    throws Exception
```

Solution Technique possible : Proxy dynamique et InvocationHandler