

Usage des Annotations Java pour Javanaise V2

Fabienne Boyer
Laboratoire LIG - UGA

Fabienne.Boyer@imag.fr



Principes

■ 3 types de définitions des annotations

- ◆ Définition de l'annotation (ex: classe Version)
- ◆ Définition des classes traitant l'annotation (ex: classe GetVersion)
- ◆ Définition des classes ou des interfaces annotées (ex: class MyClass)

Politiques de rétention

■ Source

Ces annotations sont censées être traitées avant/pendant la phase de compilation. Elles sont « perdues » après cette phase.

Java fournit un support pour traiter ce type d'annotations (APT, Annotation Processing Tool).

■ Class

Ces annotations sont présentes dans le fichier class, mais seront perdues lors de l'exécution (non présentes ni manipulables dans la VM).

■ Runtime

Ces annotations sont présentes dans les classes chargées, donc accessibles durant l'exécution.

Exemple de définition d' une annotation (Fichier: Version.java)

```
package annotation.example; ;
import java.lang.annotation.*;

// The annotation is available at execution time
@Retention(RetentionPolicy.RUNTIME)

// The annotation is associated with a type (Classe, interface)
@Target(ElementType.TYPE)

public @interface Version {
    String name();
}
```

Types d'éléments annotés

- **@Target=TYPE** classe, une interface ou enum
- **@Target=FIELD** attribut
- **@Target=METHOD**
- **@Target=PARAMETER**
- **@Target=CONSTRUCTOR**
- **@Target=LOCAL_VARIABLE**
- **@Target=PACKAGE**

Exemple de définition d' une classe annotée (Fichier: MyClass.java)

```
package annotation.example;

// Specify the annotation on the class
@Version ( name = "optimized" )

public class MyClass {
    ...
}
```

Exemple d' utilisation d' une annotation (Fichier: GetVersion.java)

```
package annotation.examples;
public class GetVersion {

public static void PrintVersion(Class c) {
// Is the annotation defined on the class?
if( c.isAnnotationPresent(Version.class) ) {
// Get a reference on the annotation
version = c.getAnnotation(Version.class);
// Get the annotation value
System.out.print("La classe " + c.getName() + " a pour
version " + version.name());
}
...
}
```

Principe d' utilisation dans Javanaise

- Le générateur de classes d' interposition *jvnc* interprête les annotations présentes dans les classes des objets Jvn
- On utilise donc des annotations dont la rétention est de type Class
- Ces annotations permettent de connaître le type des méthodes des objets Jvn (Read/Write)