

## Cours 5 : Génération de code (Règles de correspondance UML vers Java)

1. À toute **classe** UML doit correspondre une **classe** Java portant le même nom que la classe UML.
2. À toute **interface** UML doit correspondre une **interface** Java portant le même nom que l'interface UML.
3. À toute **propriété** d'une classe UML doit correspondre un **attribut** appartenant à la classe Java correspondant à la classe UML. Le nom de l'attribut doit être le même que le nom de la propriété. Le type de l'attribut doit être une correspondance Java du type de la propriété UML. Si le nombre maximal de valeurs pouvant être portées par la propriété est supérieur à 1, l'attribut Java est un tableau.
4. À toute **opération** d'une classe UML doit correspondre une **méthode** appartenant à la classe Java correspondant à la classe UML. Les noms des opérations doivent être les mêmes. Étant donné que Java ne supporte que les directions in et return, si l'opération contient des paramètres de direction *out* ou *inout*, nous considérons qu'il n'est pas possible de générer du code Java. Sinon, pour chaque paramètre de l'opération UML dont la direction est in doit correspondre un paramètre de l'opération Java. Les noms des paramètres doivent être les mêmes. Les types des paramètres doivent être une correspondance Java des types des paramètres UML. Si l'opération UML contient un paramètre de direction return, l'opération Java doit définir un retour qui lui correspond. Si l'opération UML ne contient pas de paramètre de direction return, l'opération Java retourne *void*.
5. Si une classe UML A est **associée** à une classe UML B et que l'association soit **navigable**, il doit correspondre un attribut dans la classe Java correspondant à la classe UML A. Le nom de l'attribut doit correspondre au nom du rôle de l'association. Le type de l'attribut doit être une correspondance Java de la classe UML B associée. Si l'association spécifie que le nombre maximal d'objets pouvant être reliés est supérieur à 1, l'attribut Java est un tableau. **Si l'association n'est pas navigable, nous considérons qu'il n'est pas possible de générer du code Java.**
6. Si une classe UML **hérite** d'une autre classe UML, il doit correspondre une relation **d'héritage** (*extends* en Java) entre les classes Java correspondantes. Comme Java ne supporte pas l'héritage multiple, si une classe UML hérite de plusieurs autres classes UML, nous considérons qu'il n'est pas possible de générer du code Java.
7. Si une classe UML **réalise** une ou plusieurs interfaces UML, il doit correspondre **une relation de réalisation entre la classe et les interfaces** Java correspondantes (*implements* en java).
8. Si une classe UML est **contenue** dans un package, la classe Java correspondante doit déclarer qu'elle **appartient à un package** Java. Le nom du package Java doit être le même que le nom du package UML.
9. Si un package UML **importe** un autre package UML, toutes les classes Java correspondant aux classes UML incluses dans le package UML doivent **déclarer un import Java vers toutes les classes** Java correspondant aux classes incluses dans le package UML importé.
10. Si des **notes de code** Java sont associées aux opérations des classes UML, ce code est **recopié** dans les méthodes Java correspondantes.

### Règle 5-modifiée :

S'il y a une **association navigable** entre une classe A vers une classe UML B, il aura dans la classe Java correspondant à A un attribut dont le nom est le rôle de l'extrémité navigable de l'association.

Si la **multiplicité de l'extrémité navigable est inférieur ou égale à 1**, le type de l'attribut est la classe Java correspondant à B.

- Si la **multiplicité de l'extrémité navigable est supérieure à 1**, deux cas de figures sont possibles :
  - Il y a une **collection définie** sur l'association dans une note UML, le type de l'attribut sera cette **collection spécialisée par la classe Java** correspondant à B.
  - Il n y a pas de **collection définie** pour la génération sur l'association, le type de l'attribut sera *ArrayList<B>*