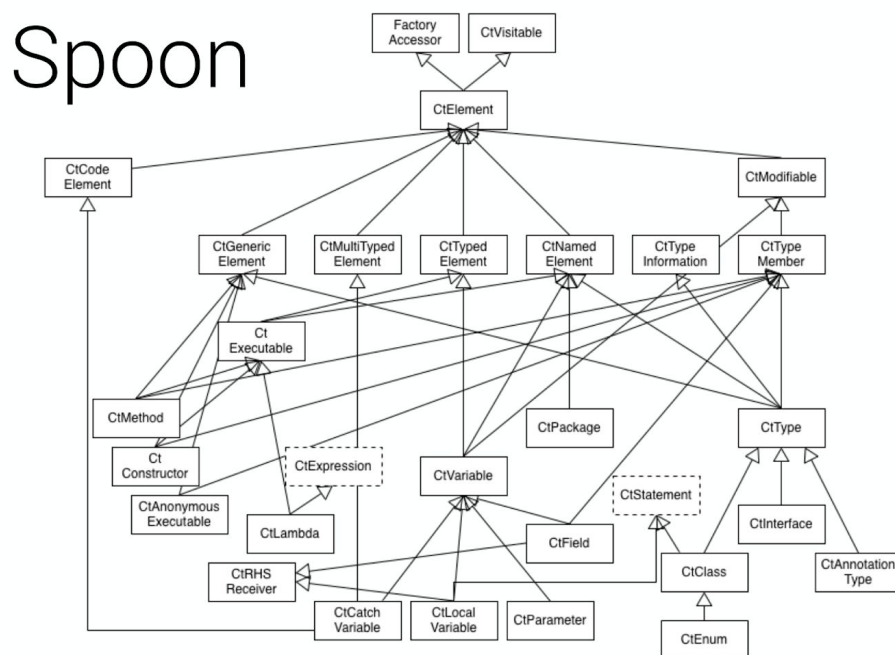


# Spoon

Thibaut ETIENNE, Fabien FERAUD, Florian NOVELLON

Spoon est une librairie en java créée en 2005 par l'INRIA

Cette librairie prend en entrée des fichiers java et va ensuite créer un modèle Java que l'on pourra analyser, modifier, etc, pour ensuite après l'avoir modifié, obtenir à nouveau des fichiers java prenant en compte nos modifications.



On va prendre comme exemple une classe java représentant une voiture, avec comme attribut “modèle”, “marque” et “couleur”, avec un constructor et une fonction toString.

On va donc apporter des transformations à cette classe java (pas directement à cette classe, nous aurons toujours la classe Vehicule.java dans notre package “original” mais nous aurons une nouvelle classe Vehicule.java dans le package “transform”). Notre objectif va être d’ajouter une année à notre véhicule, et de créer une nouvelle méthode, modifier le constructeur et le toString.

### Ajouter un attribut :

Pour ajouter un attribut cela se fait simplement, nous avons fait le choix de créer une classe "addField(nom, type)"

nom sera le nom de l'attribut, type sera son type.

pour la création de cet attribut il faudra donc utiliser CtField.

```
CtField<?> field = factory.Core().createField();
field.setSimpleName(nom);
field.setType(factory.Code().createCtTypeReference(class1));
field.addModifier(ModifierKind.PRIVATE);
ctClass.addFieldAtTop(field);
```

### Ajouter une méthode :

"addMethod(nom, contenu)"

le nom sera le nom de la méthode, et le contenu sera le corps de celle-ci.

```
CtMethod<Void> ctMethod = factory.Core().createMethod();
ctMethod.setSimpleName(nom);
ctMethod.setVisibility(ModifierKind.PUBLIC);

ctMethod.setType(factory.Type().createReference(void.class));
ctMethod.setBody(factory.Code().createCodeSnippetStatement(contenu));

ctClass.addMethod(ctMethod);
```

### Ajouter un commentaire :

L'ajout de commentaire sur n'importe quelle portion de la classe est assez simple, il suffit d'utiliser.

Ici par exemple nous allons vous montrer comment nous avons ajouté un commentaire au dessus d'une méthode.

```
CtComment com = factory.Core().createComment();
com.setContent("méthode " + nom + " ajouté avec SPOON");
ctMethod.addComment(com);
```

### Modifier un élément existant (constructeur, méthode) :

La modification d'un élément existant n'est pas réellement une modification puisque l'on ne modifie pas le code de la classe java de base. Ce que nous réalisons est une copie avec modification.

```
CtMethod<?> ctMethod = ctClass.getMethod("toString");

StringBuilder stringBuilder = new StringBuilder();

stringBuilder.append("return ");
stringBuilder.append "\"" + ctClass.getSimpleName() + " de l'"");

for(CtFieldReference<?> reference : ctClass.getAllFields()){
    stringBuilder.append("+ ");
    stringBuilder.append(reference.getSimpleName());
    stringBuilder.append(" : \" + ");
    stringBuilder.append(reference.getSimpleName());
    stringBuilder.append("\n");
}

List<CtStatement> ctStatement = new ArrayList<CtStatement>();

ctStatement.add(factory.Code().createCodeSnippetStatement(stringBuilder.
toString()));

ctMethod.getBody().setStatements(ctStatement);
```