

# TP 1 - Restructuration de logiciels

## 1 Installation

```
1      git clone https://github.com/MartinSanchezIut/TP1-  
      RestructurationLogiciels.git
```

Listing 1 – Clone du repository

Ensuite importer le projet dans IntelliJ.

## 2 Exercice 1

Pour cet exercice j'ai utilisé des visiteurs sur les Classes, les Methodes et les packages.

Ces visiteurs sont envoyés dans le Parseur. Le parseur va s'occuper de récupérer l'ensemble des fichiers renseignés par l'utilisateur dans la CLI pour les convertir en AST.

Ces AST accepteront le visiteur renseigné à la construction du parseur.

Le parseur contient les méthodes permettant de répondre à l'exercice.

### 2.1 Utilisation

Exécutez le fichier "Main" dans le package "me.martin.p2Exo1". Il contient un CLI qui vous proposera de visualiser l'ensemble des données de l'exercice. Vous pouvez aussi les voir une par une.

## 3 Exercice 2

Pour cet exercice j'ai utilisé un visiteur sur les méthodes et sur les appels de méthodes.

Le parseur se sert du visiteur de méthodes pour récupérer toutes les méthodes du projet que l'utilisateur a renseigné dans la CLI. Il vient ensuite appliquer sur ces méthodes

le visiteur d'appel de méthode. Avec cela on peut récupérer la méthode appelante et toutes les méthodes appelées. On formate tout ça dans un graphe, composé d'un ensemble de sommet et d'un ensemble d'arêtes.

Pour l'interface graphique, on se contente de créer une image du graphe. Pour cela, on utilise une librairie : `jgraphet` qui nous permet de créer un graphe à partir de notre objet. Ce graphe est au format DOT.

On fait ensuite appel à un api de Graphviz (`graphviz-java`) pour convertir notre fichier DOT en fichier PNG. Cette partie du code risque de ne pas fonctionner, c'est pourquoi le CLI vous donnera une URL qui génère l'image dans votre navigateur, au cas où je n'aurai pas réussi à faire fonctionner l'api Graphviz correctement.

### **3.1 Utilisation**

Exécutez le fichier `"Main"` dans le package `"me.martin.p2Exo2"`. Il contient un CLI qui vous proposera de renseigner le projet à analyser. Le programme créera une image qui sera stockée dans `"src/main/resources/img/"`.

## **4 Applications de test**

Deux applications de test sont disponibles dans le fichier `"src/main/resources/"`. Par défaut c'est `"PP.java"` qui sera utilisé.