

README – TP2 : Analyse et restructuration du couplage entre classes Java

Ouezzani Rahma

Année universitaire 2025–2026

Liens du projet

- **Dépôt GitHub** : <https://github.com/Rahma121-crtl/TP2-Evolution-et-restructuration-logiciel.git>
- **Vidéo de démonstration** : https://docs.google.com/videos/d/1Uk6QV3rmN_BFFzI4G3VYjH0sharing

Installation et préparation du projet

1. Dézipper le dossier du projet : `Partie2TPEvo.zip`
2. Ouvrir le projet dans **Eclipse IDE** (ou IntelliJ).
3. Vérifier que le JDK 17 (ou supérieur) est bien configuré.
4. Faire un **clic droit** > **Maven** > **Update Project**.
5. Vérifier que le dossier `src/main/java` contient les packages : `coupling`, `graph`, `tp2demo`, `utils`, `Analysis`.

Prérequis

- Java 17 ou supérieur
- Apache Maven
- Eclipse IDE
- Graphviz (pour visualiser les graphes)

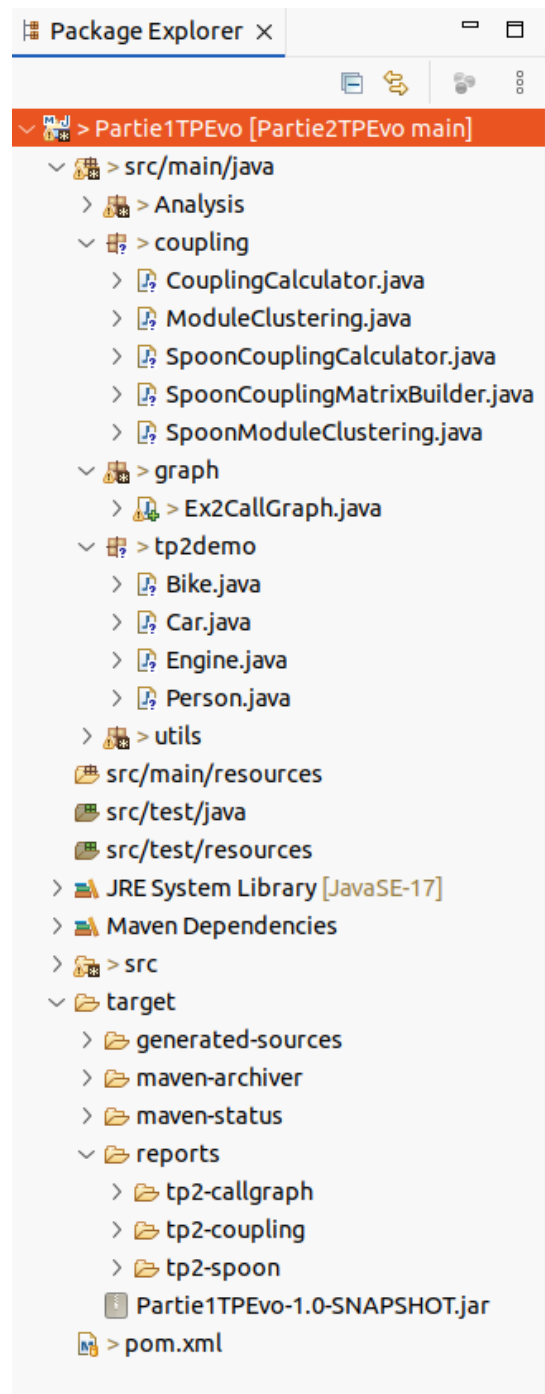


FIGURE 1 – Structure du projet Partie2TPEvo dans Eclipse

Exécution et vérification des exercices

Exercice 1 – Couplage entre classes (JDT)

- Exécuter la classe : `coupling.CouplingCalculator`.
- Vérifier la console :

Arêtes chargées: 3

Couplage(`tp2demo.Person`,`tp2demo.Car`) = 0.3333333333

Fichiers générés dans `reports/tp2-coupling/`

- Les fichiers générés sont :
 - `target/reports/tp2-coupling/coupling-matrix.csv`
 - `target/reports/tp2-coupling/coupling.dot`
- Pour générer et visualiser le graphe :

```
cd target/reports/tp2-coupling/
dot -Tpng coupling.dot -o coupling.png
xdg-open coupling.png      # (Linux)
start coupling.png         # (Windows)
```

- Le graphe `coupling.png` doit montrer les liens entre `Person`, `Car`, `Engine` et `Bike`.

Exercice 2 – Clustering hiérarchique

- Exécuter la classe : `coupling.ModuleClustering`.
- Résultat console attendu :

```
Matrice de couplage chargée (4 classes)
M=4, CP=0,010, maxClusters=2
Meilleur couplage trouvé : Car  Engine = 1.0000
Meilleur couplage trouvé : Person Bike = 1.0000
Clustering terminé. Nombre final de modules : 2
Modules :
- Car+Engine
- Person+Bike
```

- Vérifier le fichier produit : `target/reports/tp2-coupling/modules.csv`
- Chaque module regroupe bien les classes les plus couplées.

Exercice 3 – Analyse automatisée avec Spoon

- Exécuter les classes dans cet ordre :
 1. `SpoonCouplingMatrixBuilder` – construction de la matrice;
 2. `SpoonModuleClustering` – clustering automatique.
- Résultat console attendu :

```
=== Graphe de couplage via Spoon (filtré) ===
tp2demo.Person → [tp2demo.Car, tp2demo.Bike]
tp2demo.Car → [tp2demo.Engine]
Analyse Spoon terminée : 4 classes
Modules exportés : spoon-modules.csv
```

- Fichiers produits :
 - `spoon-coupling.csv`
 - `spoon-coupling-matrix.csv`
 - `spoon-modules.csv`
 - `spoon-coupling.dot`

- Pour visualiser le graphe généré :

```
cd target/reports/tp2-spoon/  
dot -Tpng spoon-coupling.dot -o spoon-coupling.png  
xdg-open spoon-coupling.png      # (Linux)  
start spoon-coupling.png         # (Windows)
```

Validation et résultats attendus

- Les graphes (.png) sont générés sans erreur et visualisent correctement les dépendances.
- Les matrices de couplage contiennent les 4 classes principales : **Person**, **Car**, **Bike**, **Engine**.
- Les modules finaux sont :
 - **Car+Engine**
 - **Person+Bike**