



Exercice 1 : Graphe de couplage entre classes

Reprenez les résultats de l'exercice lié à construction du graphe d'appel.

1. Calculez une métrique de couplage entre deux classes A et B suivant la définition suivante :

Nombre de relations (relation = appel) entre les couples de méthodes appartenant respectivement aux deux classes en question ($A.m_i$ et $B.m_i$) / nombre de toutes les relations (binaire) entre les couples de méthodes appartenant respectivement à n'importe quelles deux classes de l'application analysée.

2. Générez un graphe de couple pondéré entre les classes de l'application RestSuite.

Rendu: Comme le TP précédent, veuillez rendre le projet avec les deux exercices ainsi que le fichier DOT contenant le graphe généré lors de l'analyse statique (ou le PNG généré par GraphViz).

Exercice 2 : Identification de Modules

Implémentez l'algorithme de clustering vu en cours développé en utilisant la métrique de l'exercice 1 :

- Définissez le code Java d'implémentation d'un algorithme de regroupement (clustering) hiérarchique des classes d'une application. A l'étape i, les deux clusters (e.g classes) à regrouper sont celles les plus couplés :

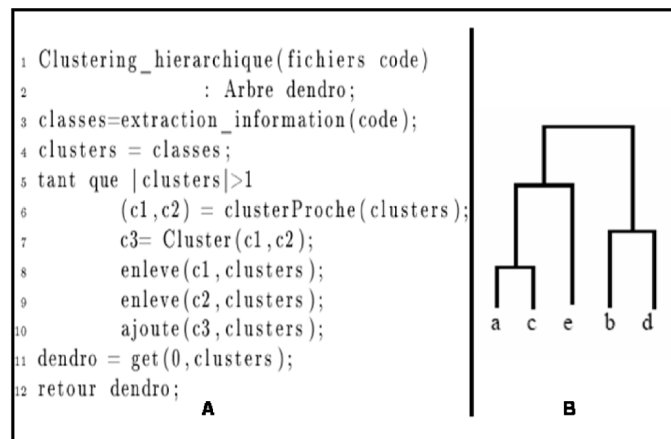


Figure 1: Hierarchical Clustering Algorithm.

- Définissez le code Java d'implémentation d'un algorithme d'identification des groupes de classes couplées (ces groupes peuvent être vus comme des services / microservices / modules / fonctionnalités) :

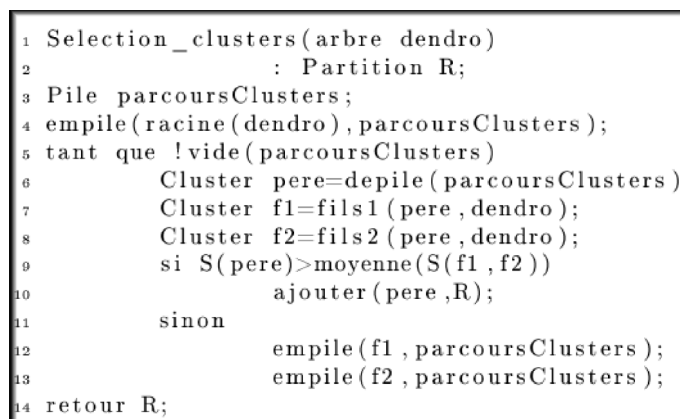


Figure 2: Algorithme de décomposition de votre arbre.