

TP3

HAI913I

Compréhension des programmes

Exercice 1 : Graphe de couplage entre classes

Reprenez les résultats de l'exercice lié à construction du graphe d'appel du TP N°2.

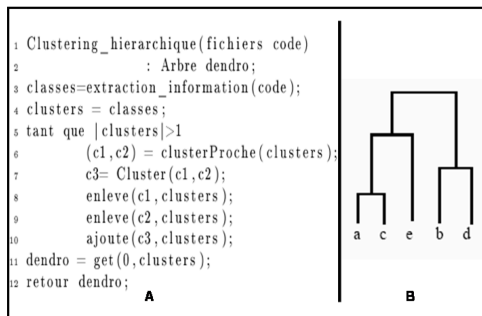
- 1) Calculez une métrique de couplage entre deux classes A et B suivant la définition suivante :

Nombre de relations (relation = appel) entre les couples de méthodes appartenant respectivement aux deux classes en question ($A.m_i$ et $B.m_j$) / 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 couplage pondéré entre les classes d'une application donnée.

Exercice 2 : Identification de Modules

- 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 :



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

```

1 Selection_clusters(arbre dendro)
2     : Partition R;
3 Pile parcoursClusters;
4 empile(racine(dendro),parcoursClusters);
5 tant que !vide(parcoursClusters)
6     Cluster pere=depile(parcoursClusters);
7     Cluster f1=fils1(pere,dendro);
8     Cluster f2=fils2(pere,dendro);
9     si S(pere)>moyenne(S(f1,f2))
10        ajouter(pere,R);
11     sinon
12        empile(f1,parcoursClusters);
13        empile(f2,parcoursClusters);
14 retour R;

```

Exercice 3 : Spoon pour l'identification de modules

Spoon (<http://spoon.gforge.inria.fr/>) est une bibliothèque open source qui vous permet de transformer et d'analyser un code source Java. Spoon fournit un méta modèle Java complet et détaillé dans lequel tout élément de programme (classes, méthodes, champs, instructions, expressions, etc.) est accessible à la fois pour la lecture et la modification. Spoon prend un code source en entrée et produit un code source transformé prêt à être compilé.

- Utilisez l'outil Spoon pour répondre aux mêmes questions de l'exercice 1 et 2 de ce TP (TP N°3).

Exercice 4 : Utilisation de Spoon pour l'analyse dynamique (Optionnel – Bonus)

- Utilisez spoon pour instrumenter le code (insérer des instructions « Sensors »).
- Exécutez quelques scénarios de votre code instrumenté et afficher les traces d'exécution. Une trace est une liste d'appels.
- Construisez le graphe d'appel dynamique.