



UNIVERSITÉ DE MONTPELLIER

ÉVOLUTION ET RESTRUCTURATION DE LOGICIEL

TP3 : Compréhension des programmes

<https://github.com/aminamaiga/tp3SoopCluster.git>



Rendu par
Aminata MAIGA
Thierno BARRY

Année : 2021 — 2022

Exercice 1 : Graphe de couplage entre classes

1) Calcul d'une métrique de couplage entre deux classe A et B.

Le couplage est défini comme étant le degré d'interdépendance des composants logiciels. Cette interdépendance se manifeste à travers l'accès aux variables, les appels de méthodes et les liens d'héritage et d'interface.

Si la classe A utilise la classe B et inversement à un nombre de **X** fois, le couplage est égal à nbfois / X ; X est le nombre total de couple.

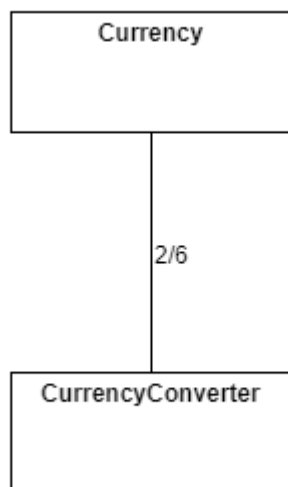
Application dans un exemple :

Si nous avons une classe Personne et 10 autres classes avec des méthodes qui crée une instance Personne et définit les valeurs des propriétés. La classe Personne accède à deux méthodes appartenant à deux de ces classes.

Donc les métriques du code sont égales à la valeur **2 / 11**. En effet, le couplage de classe ne mesure chaque classe qu'une seule fois pour cette mesure.

2) Génération d'un graphe de couplage pondéré entre les classes d'une application donnée.

Nous avons 5 classes et 6 appels entre les différentes classes. La classe Currency utilisé une propriété et une méthode déclarée dans la classe CurrencyConverter. Donc la métrique est égale à 2 sur le nombre total de dépendances entre les différentes classes de l'application.



Exercice 2 : Identification de Modules

github <https://github.com/aminamaiga/tp3SoopCluster.git>

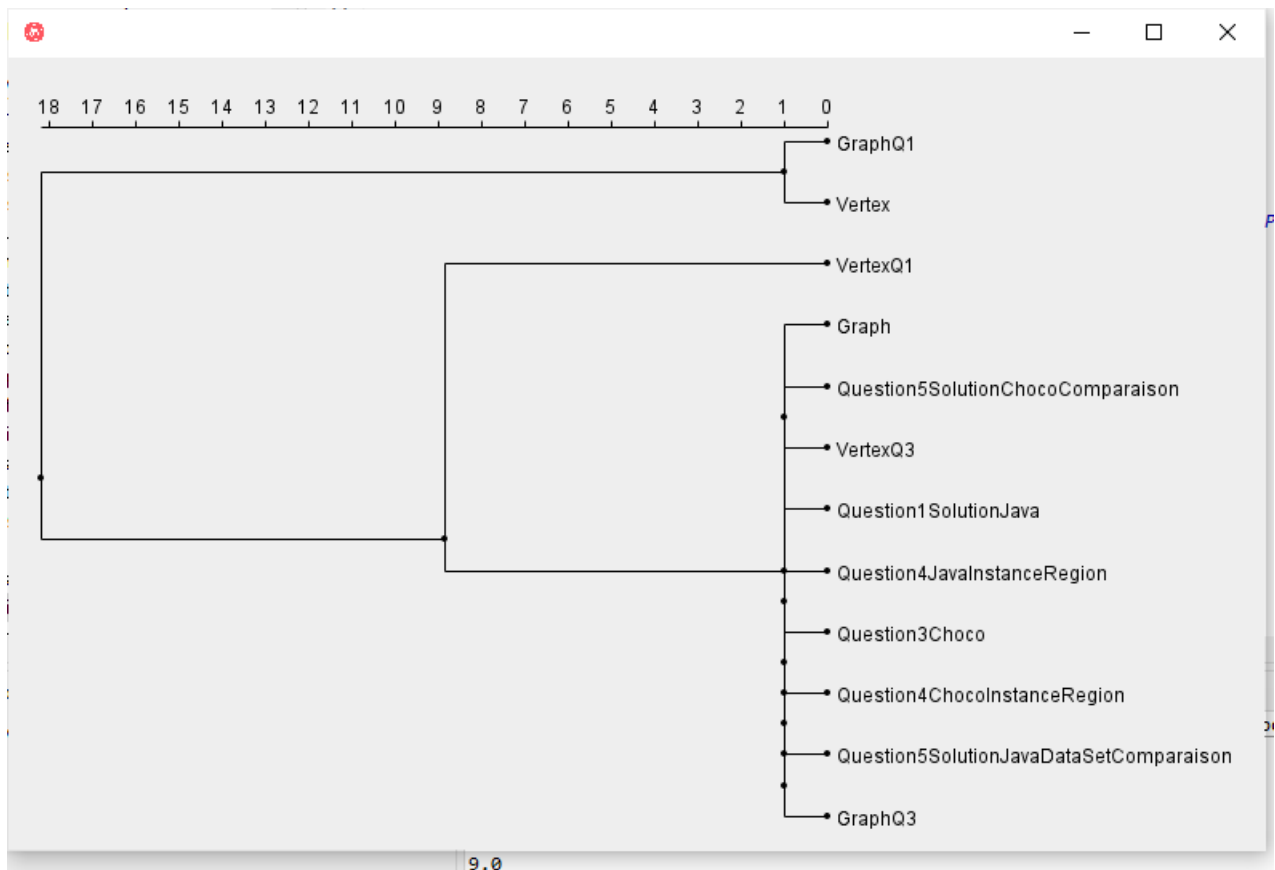
Exemple de sortie des couples d'une application.

Cet exemple illustre le couplage entre deux classes X et Y et la distance entre ces deux classes. Cette distance est appelée linkage.

Exemple 1 :

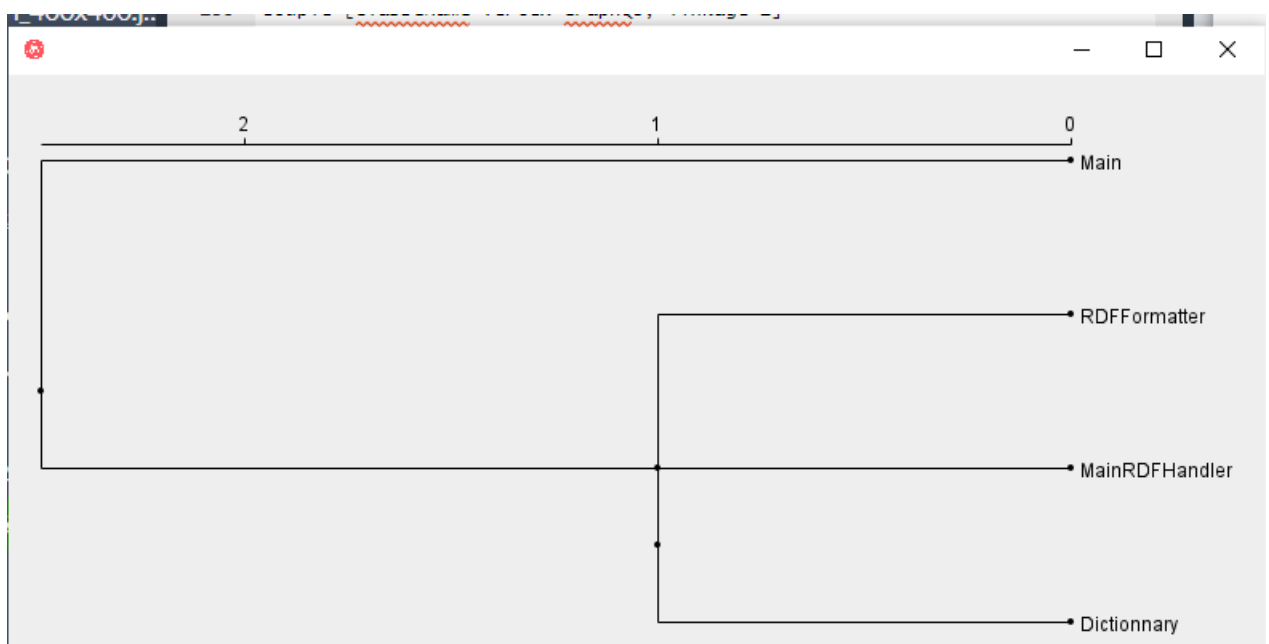
```
Couple [classeName=Vertex GraphQ3, linkage=1]
Couple [classeName=Vertex VertexQ3, linkage=1]
Couple [classeName=Vertex Question4ChocoInstanceRegion, linkage=1]
Couple [classeName=Vertex Question5SolutionJavaDataSetComparaison,
linkage=1]
Couple [classeName=Vertex VertexQ1, linkage=1]
Couple [classeName=Vertex GraphQ1, linkage=9]
Couple [classeName=Vertex Question5SolutionChocoComparaison,
linkage=1]
Couple [classeName=Vertex Graph, linkage=2]
Couple [classeName=VertexQ1 GraphQ1, linkage=8]
Couple [classeName=GraphQ1 Question1SolutionJava, linkage=9]
Couple [classeName=Graph Question4ChocoInstanceRegion, linkage=9]
Couple [classeName=Graph Question1SolutionJava, linkage=9]
Couple [classeName=Graph Question3Choco, linkage=9]
Couple [classeName=Graph GraphQ1, linkage=36]
Couple [classeName=Graph Question5SolutionJavaDataSetComparaison,
linkage=9]
Couple [classeName=Graph Question5SolutionChocoComparaison,
linkage=9]
Couple [classeName=Graph Question4JavaInstanceRegion, linkage=9]
```

Le dendrogramme représente la sortie du programme 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 :



Exemple 2 : Calcul du couplage dans une petite applicaion

Couple [className=RDFFormatter MainRDFFHandler, linkage=1]
 Couple [className=Dictionnary MainRDFFHandler, linkage=4]
 Couple [className=Dictionnary Main, linkage=4]



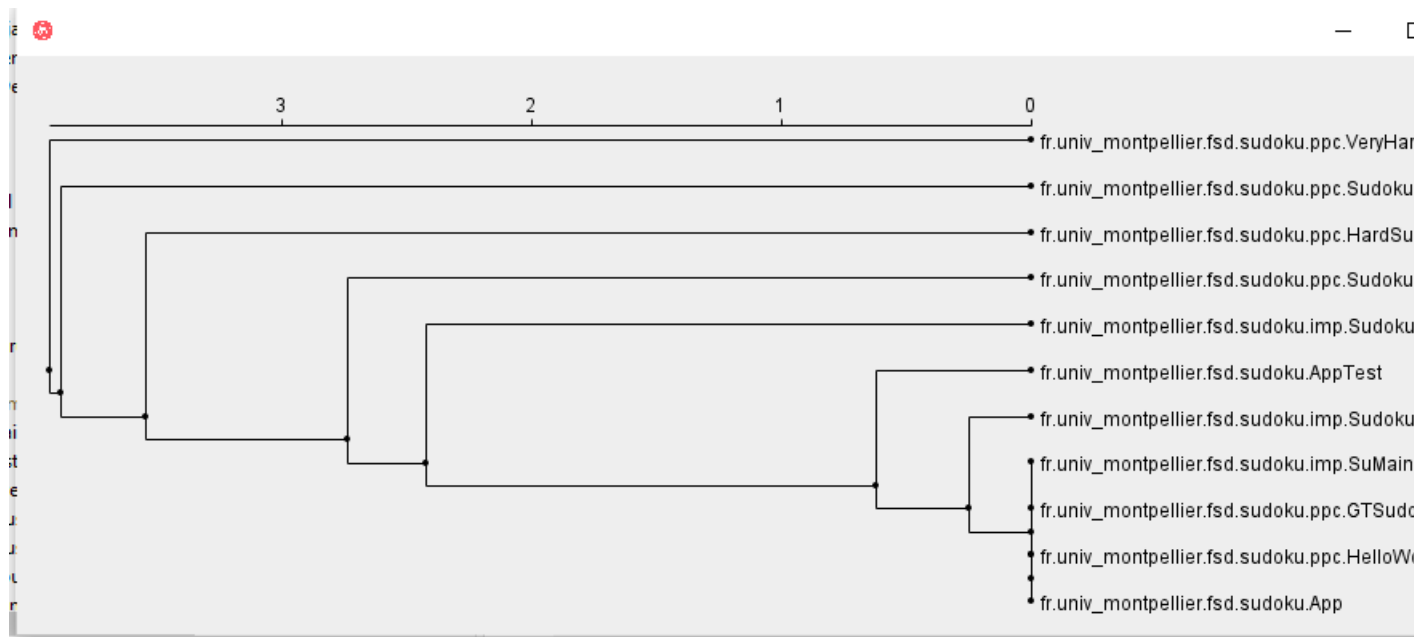
Exercice 3 : Spoon pour l'identification de modules (En cours) regroupement avec Soopn

[illegible]

```

invocations currencyConverter.MainWindow
invocations java.lang.Exception
invocations currencyConverter.JTextFieldLimit
invocations java.lang.String
invocations java.lang.String
invocations javax.swing.text.PlainDocument
invocations java.util.ArrayList
invocations javax.swing.JComboBox
invocations currencyConverter.Currency
invocations java.util.ArrayList
invocations java.util.ArrayList
invocations currencyConverter.Currency
invocations java.util.ArrayList
invocations currencyConverter.Currency
invocations java.util.ArrayList
invocations java.util.ArrayList
invocations currencyConverter.Currency
invocations java.util.ArrayList
invocations java.util.HashMap
invocations currencyConverter.Currency
invocations java.util.ArrayList
invocations void
currencyConverter.MainWindow
1.0
1.0
1.0
1.0
1.0
currencyConverter.JTextFieldLimit
1.0
1.0
1.0
1.0
1.0
currencyConverter.AboutWindow
1.0
1.0
1.0
1.0
1.0
currencyConverter.CurrencyConverter
1.0
1.0
1.0
1.0
1.0
currencyConverter.Currency
1.0
1.0
1.0
1.0
1.0

```



Lien github du projet en cours <https://github.com/aminamaiga/tp3SoopCluster>.
git