

# TP1 : Analyse statique

## Évolution et restructuration des logiciels

Mohamad Satea Almallouhi - Tony Nguyen  
*M1 Génie Logiciel*  
Faculté des Sciences  
Université de Montpellier.

6 octobre 2024

### Résumé

*Rapport d'exercice sur l'analyse statique*

# Table des matières

<b>Introduction</b>	<b>2</b>
<b>Démonstration</b>	<b>2</b>
<b>Installation</b>	<b>2</b>
<b>1 UML !!!!</b>	<b>3</b>
<b>2 Graphes</b>	<b>3</b>
2.1 Algorithmes de création du graphe d'appel . . . . .	3
2.2 Le couplage . . . . .	3
2.3 Graphe de couplage inter-classes . . . . .	3
2.4 Montrer les visiteur . . . . .	3
<b>3 Clustering</b>	<b>3</b>
3.1 Algorithme de clustering . . . . .	3
<b>4 Spoon</b>	<b>3</b>

## Introduction

Dans le cadre de l'Unité d'Enseignement Évolution et restructuration des logiciels, nous allons analysés un programme en observant son code source de manière statique.

Tout d'abord, à partir du travail précédent, nous allons nous servir du graph d'appel des méthodes écrites dans les classes du projet analysé. Cela nous permettra de calculer le couplage entre les différentes classes.

Ainsi, grâce au graph de couplage, nous allons partitionné notre ensemble de classes en différent modules.

## Démonstration vidéo

En ligne sur Youtube, à l'adresse URL <https://youtu.be/fDvHlrK8IRg> une démonstration vidéo de notre travail.

## Installation

Vous trouverez les instructions dans le README.md

To Do

- add code picture
- add resultat picture
- diagram class visiteur
- diagram class de l'app
- definition du couplage avec écritures math
- tous les algo en latex stylé
- → !!! → vidéo ← !!! ←

Faire une vidéo, le rapport avec des screenshot des résultats et du code et enfin un read.md(instruction). En plus, pour le bonus, faire une belle application, des tests unitaires, faire le rapport en Latex.

## 1 UML!!!!

## 2 Graphes

Nous nous intéressons au couplage entre les classes de notre application. Il serait logique de réunir les classes fortement dépendantes les unes des autres. De la même façon, les classes qui n'ont aucun rapport entre elle, pour des raisons de clarté, peuvent être isolé.

### 2.1 Algorithmes de création du graphe d'appel

La construction de ce graph est la base de ce travail. Il nous permettra ensuite de calculer le couplage entre les classes ...

Les méthodes qui se surcharge entre elles (les méthodes ayant le même nom dans une classe mais avec une signature différente) sont confondues.

```

Data: programme
Result: graphe d'appel
g ← new Graph();
for each class c do
  for each method m implemented in c do
    for each method i in m do
      if !isExternalMethod(i) then
        ajoute une arte de m vers i;
      end
    end
  end
end
end

```

**Algorithm 2:** An algorithm to make call graph

### 2.2 Le couplage

**Définition** TODO INSERT EPIC MATH FORMULA HERE

### 2.3 Graphe de couplage inter-classes

UML TODO INSERT EPIC CLASS DIAGRAM HERE

### 2.4 Montrer les visiteur

???

## 3 Clustering

Nous allons maintenant voir comment nous avons rassemblé les classes entre elles.

### 3.1 Algorithme de clustering

## 4 Spoon

**Data:** *programme*

**Result:** *graphe d'appel*

$g \leftarrow \text{new Graph}();$

**Algorithm 3:** An algorithm to make coupling graph

**Data:** *programme*

**Result:** *graphe d'appel*

$g \leftarrow \text{new Graph}();$

**Algorithm 4:** Clustering algorithm