

**INF 1018**  
**Analyse de programmes**  
**Automne 2013**  
**Projet**

**Organisation générale**

- **Le projet sera réalisé en plusieurs étapes. Il se divise, principalement, en deux parties:**
  - *Thème (lecture)*: Vous avez reçu un document traitant de l' « aspect mining ». Je vous demande de bien l'étudier, et éventuellement d'effectuer des lectures complémentaires sur le sujet (deux ou trois autres documents – recherche personnelle sur le web). Par ailleurs, essayez de réfléchir à comment appliquer l'analyse de programmes dans ce contexte, d'une façon générale, et dans le contexte des techniques basées sur les chemins d'exécution en particulier.
  - *Réalisation (outil)*: Différentes options à implémenter
    - Extension des descriptions JavaCC pour implémenter plusieurs éléments d'analyse de programmes (différents objectifs – voir la suite du document).
    - Utilisation des AST (Eclipse) pour implémenter certaines options.
    - Analyse de programmes Java: expérimentation de l'outil que vous allez développer sur des programmes Java réels (considérez quelques programmes disponibles sur le web) - Présentation des résultats (compte rendu).
- **Biens livrables et évaluation:**
  - Validation de l'outil – *Démonstration au laboratoire*<sup>1</sup>: Il s'agit de faire une démonstration complète de l'outil développé sur des cas de programmes réels. Vous devez également remettre : une copie du code développé, une copie des résultats et votre discussion (maximum une page – voir suite du présent document pour les détails relatifs à ce point).

---

<sup>1</sup> La validation se fera selon un planning (voir document « planification »). Je vous rappelle que les moments des validations doivent être déterminés en accord avec Mr Daniel St-Yves (RV – par courriel).

## **Réalisation / JavaCC & AST (Eclipse)**

Les différents objectifs à atteindre sont donnés dans ce qui suit:<sup>2</sup>

- (1)- Analyse de la portée des attributs des classes: Pour chaque classe du programme analysé, donnez les statistiques descriptives suivantes :
  - % des attributs publics, % des attributs protégés et % des attributs privés.
- (2)- Analyse de la visibilité entre classes: Pour chaque classe du programme analysé, on s'intéresse au type de ses attributs à travers lequel nous pouvons avoir une idée sur la visibilité (permanente) entre objets :
  - % des attributs de type simple et % des attributs de référence
  - Donnez, pour chaque classe du programme, la liste des classes vis-à-vis desquelles elle a une visibilité permanente.
- (3)- Extraction du modèle objet: Modèle intégrant les classes et les relations
  - Héritage
  - Association
- (4)- Couplage entre classes: Nous nous intéressons au couplage entre les différentes classes du programme analysé relativement aux appels entre méthodes : nombre de fois qu'une classe Ci appelle (via ses méthodes) les méthodes d'une classe Cj.
- (5)- Graphe d'appels directs entre méthodes:
  - Pour chaque méthode d'une classe, donnez la liste des méthodes (et leur classe d'appartenance) qu'elle appelle directement (appels directs uniquement).

Suite à votre utilisation de JavaCC et des AST d'Eclipse, donnez votre avis sur les deux implémentations : facilité d'implémentation, avantages et inconvénients, impact d'une modification du langage Java sur ce que vous avez réalisé, etc. (maximum une page).

---

<sup>2</sup> Toutes les options doivent être implémentées en utilisant JavaCC. Par ailleurs, vous devez aussi implémenter les options 4 ou 5 (au choix) en utilisant les AST (Eclipse).