

INF 1018 Analyse de Programmes

Projet de la session – Automne 2023

Date limite de remise : 18 décembre 2023 à 23h55

Travail en équipe: entre 4 et 5 personnes

Pondération: 25% (incluant 2,5% d'évaluation des pairs).

Le projet sera réalisé en plusieurs étapes. Il se divise, principalement, en deux parties:

Thème spécifique (Recherche) – 30% : Cette partie se base sur le sujet de l'«aspect mining». Il y existe plusieurs approches permettant de **décoder**, par analyse de programmes, les «bouts» de code pouvant correspondre à des préoccupations **transverses**. Ces techniques sont étroitement liées aux techniques de Refactoring (**remaniement**) et d'**Aspect** (aspectualisation du code objet). Vous êtes demandé d'effectuer une recherche sur le sujet et 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. Présentez les résultats de votre recherche sous forme d'un **rapport** qui ne dépasse pas **15 pages**, au maximum, incluant la page de titre, la table des matières, l'introduction, la conclusion et la bibliographie.

Réalisation (outil) – 70% : 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 au début quelques programmes Java concrets pour tester votre programme – Une étude de cas particulière vous sera indiquée ultérieurement) - Présentation des résultats (compte rendu).

• **Biens livrables et évaluation :**

Validation de l'outil – *Démonstration sur Zoom par un représentant de l'équipe*: Il s'agit de faire une démonstration complète de l'outil développé sur un cas de programme réel. Vous devez également remettre:

1. Une copie du code développé ;
2. Une copie des résultats ;
3. Votre conclusion (dernier point de la page suivante).

Réalisation / JavaCC & AST (Eclipse)

Les différents objectifs à atteindre dans ce projet sont donnés dans ce qui suit :*

1. ***Analyse de la portée des attributs des classes d'un programme*** : 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 d'un programme*** : 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 :
 - a. % des attributs de type simple et % des attributs de référence (objet)
 - b. Donnez, pour chaque classe du programme, la liste des classes vis-à-vis desquelles elle a une visibilité permanente.
3. ***Reverse Engineering - Extraction du modèle objet (diagramme de classes)*** :
Il s'agira d'analyser le programme dans le but d'extraire, par reverse engineering, le diagramme de classes UML (modèle intégrant les classes et les relations entre classes). On se limitera aux deux relations : Héritage et Association (l'agrégation étant une forme particulière d'association).
4. ***Analyse de la dépendance entre classes d'un programme - Couplage entre classes*** : Nous nous intéressons dans ce contexte au couplage entre les différentes classes du programme analysé relativement aux appels entre méthodes : Il faudra compter le nombre de fois qu'une classe C_i appelle (via ses méthodes) les méthodes d'une classe C_j . Donnez pour chaque classe C_i du programme analysé la liste des classes qu'elle appelle (avec le nombre d'appels).
5. ***Extraction du graphe d'appels directs entre méthodes d'un programme*** :
Il faudra, pour chaque méthode d'une classe, donner la liste des méthodes (et leur classe d'appartenance) qu'elle appelle directement (appels directs uniquement – ne pas considérer les appels indirects).
6. À la suite de votre utilisation de JavaCC et des AST d'Eclipse, donnez votre avis sur les deux implémentations en fonction de critères tels que : facilité d'implémentation, avantages et inconvénients, impact d'une modification du langage Java sur ce que vous avez réalisé, etc. (maximum une page).

* Les trois premières options doivent être implémentées en utilisant JavaCC. Par ailleurs, vous devez implémenter les options 4 et/ou 5 (au choix) en utilisant les AST (Eclipse).