Projet DCLL – Groupe Moustaki

# Github

L’url du projet sur Github :

<https://github.com/fabien-fac/DCLL>

[git@github.com:fabien-fac/DCLL.git](mailto:git@github.com:fabien-fac/DCLL.git)

# Contributeurs

Les participants du projet sont :

### Clastrier Laurent : [LeSouss](https://github.com/LeSouss)

### Debat Johan : [JohanD9](https://github.com/JohanD9)

### Michel Joris : [jorismi](https://github.com/jorismi)

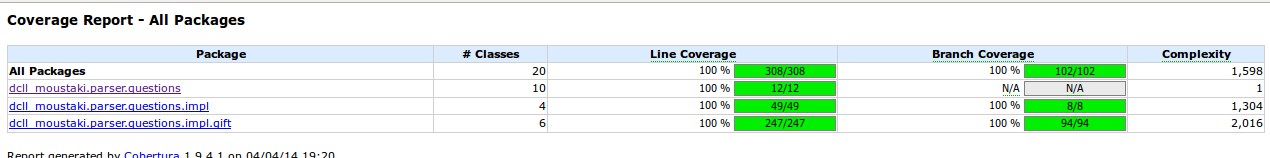
### Moreau Mathieu : [mathieu-moreau](https://github.com/mathieu-moreau)

### Rolfo Fabien : [fabien-fac](https://github.com/fabien-fac)

# Réalisation du projet

## Couverture du code

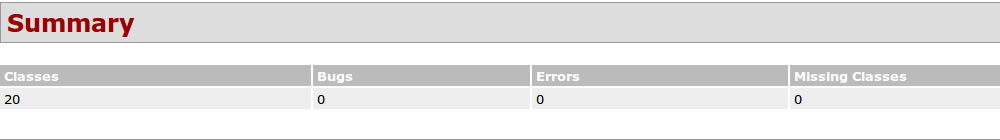
Le Framework de test JUnit a était utilisé pour réaliser la couverture du code du parseur. Pour garantir la couverture intégrale du code, nous avons utilisé le plugin cobertura.



Comme présenté sur l’image ci-dessus, qui est le résultat généré par cobertura, le code est couvert à 100%

## erreurs findbugs

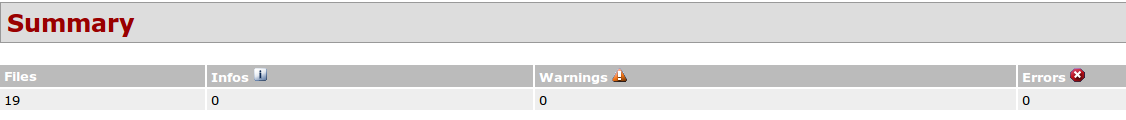
Pour trouver des bugs dans le programme Java, le plugin findbugs, qui identifie des patterns reconnus comme étant des bugs, a était utilisé.



Apres réglage de ce plugin (voir document d’acceuil du nouveau développeur) aucun bug n’a était trouvé.

## erreurs checkstyle

Pour garantir la qualité du code source et sa lisibilité, nous avons utilisé le plugin checkstyle.



Comme présenté dans le document d’accueil du nouveau développeur, nous avons dû régler ce plugin pour obtenir 0% d’erreur.

## Présentation du code

Nous avons utilisé le plugin jxr qui présente le code source avec des liens entre les classes, afin de pouvoir naviguer dans le code en dehors d’un IDE.

## Couverture fonctionnelle

Plusieurs jeux de test ont était réalisé pour garantir la couverture fonctionnelle. Le parseur peut donc, entre autre, traiter les questions de types :

* Multiple choice / single response
* Multiple choice / multiple response

# Architecture du projet

Pour la réalisation de ce projet nous avons utilisé l’architecture déjà fonctionnelle sur Git Hub. Nous avons ensuite allégé l’architecture en gardant les classes nécessaires à la réalisation de ce projet comme le décrit le diagramme de classe ci-dessous.

