

## Consignes de rendu du projet

### Consignes générales

- Le projet est à développer en binôme.
- Les rendus doivent passer par la plateforme gitlab: <https://gitlab-etu.fil.univ-lille.fr/>  
Vous devez donc donner accès à votre projet à votre enseignant. Ce dépôt doit être utilisé pendant toute la durée du projet. Vous devez mettre en pratique le processus TDD vu en cours et utiliser le dépôt régulièrement (et non pas uniquement la veille du rendu pour un ‘push’ final). Une bonne utilisation de git fait partie des compétences que vous devez acquérir. Attention à ne pas placer les .class (ou autres fichiers inutiles - par exemple .project de eclipse, etc.) sur le dépôt, i.e. utilisez un fichier .gitignore.<sup>1</sup>

### Exigences

- Votre dépôt possédera un fichier README.md sur la branche main.
- Votre README sera correctement formaté (syntaxe markdown fortement recommandée) et structuré ainsi :
  1. Noms et prénoms du binôme, rapide introduction sur le sujet du projet.
  2. Rubrique ‘HowTo’, avec une **liste de commandes précises** (`makefile` autorisé) permettant de :
    - \* Récupérer les sources du projet depuis votre dépôt;
    - \* Générer la documentation (précisez la localisation de cette documentation);
    - \* Compiler et exécuter les sources. N’oubliez pas de préciser où se placer pour exécuter ces commandes;
    - \* Compiler et exécuter les tests (voir plus bas);
    - \* Générer et exécuter l’archive (.jar) du projet;
  3. Présentation d’éléments de code saillants :
    - \* Eléments de conception que vous jugez intéressants/importants;
    - \* Présentation des principes de conception mis en œuvre;
    - \* Tout point que vous souhaitez mettre en avant pour valoriser votre projet;
- Joignez à votre projet les diagrammes UML des différentes parties de conception au format JPEG, PNG ou PDF. Vous pouvez par exemple utiliser yUML (<http://yuml.me>) ou Mermaid (<https://mermaid.js.org/syntax/classDiagram.html>) qui permettent entre autres de facilement concevoir des diagrammes de classes. Ces diagrammes doivent être intégrés dans votre README.
- Technologies à utiliser: Java 8 minimum, JUnit 5 (voir plus bas).

### Evaluation

L’évaluation de vos rendus de projet s’appuiera sur les critères (non ordonnés) suivants:

- Respect des consignes ci-dessus (gestion du gitlab, qualité du README, etc.).
- “Propreté” de votre dépôt GIT : il ne doit pas contenir de fichiers inutiles (.class, .project, fichiers backup, etc.).
- Qualité de la documentation du code (en anglais, tout comme les classes).
- Qualité des tests.
- Conception (respects des principes, application des design patterns, UML fourni, etc.).
- Fonctionnement global du projet.
- Avancement régulier du projet.
- Présence et participation en TD/TP.

---

<sup>1</sup><https://www.toptal.com/developers/gitignore>

**Aucun IDE ne sera utilisé pour corriger votre projet.** La compilation et l'exécution du projet et des tests sera effectuée uniquement en ligne de commandes, en suivant scrupuleusement les indications fournies dans votre README.

#### **JUnit 5 : compilation et exécution**

Afin de compiler et d'exécuter vos classes de test, vous devez spécifier la bibliothèque JUnit utilisée. Pour cela, vous devez d'abord la télécharger (par exemple ici : <https://search.maven.org/search?q=g:org.junit.platform>) puis la rendre disponible en l'ajoutant via le paramètre `-classpath` de votre commande. Un tutoriel expliquant l'utilisation de la bibliothèque JUnit en ligne de commande peut être trouvé ici :  
<https://www.baeldung.com/junit-run-from-command-line>.