

Gestion de Projets des étudiants

Projet Java avancé M1 MIAAGE 2023/2024

1. Dates et informations importantes

- Envoi de l'énoncé du projet : 03/10/2023
- Envoi de la liste des groupes par e-mail : 10/10/2022
- Rendu du projet (contenu du rendu mentionné dans la section 5): 1/12/2023 à 23H
- Soutenances : la semaine du 04/12/2023 dans vos séances de TD (les détails : noms des binômes, horaire et salle vous seront communiqué ultérieurement)
- Travail par groupe de 2 personnes

2. Description du projet

Un(e) enseignant(e) souhaite construire une application permettant de gérer les projets des étudiants pour l'année scolaire courante. Un étudiant est identifié par un **numéro** unique et est caractérisé par un **nom** et un **prénom**. Un étudiant appartient à une **formation**.

Une **formation** est identifiée par un **numéro** unique et est caractérisée par un **nom** (ex. ID, MIAAGE-IF ou MIAAGE-SITN) et une **promotion** (ex. "Initiale", "En Alternance" ou "Formation Continue").

Un étudiant réalise, au cours de sa formation, plusieurs projets en binôme (i.e. en groupe de 2 personnes maximum). Un projet est identifié par un **numéro** unique et est caractérisé par un **nom de matière**, un **sujet** et une **date de remise** prévue du rapport.

Un binôme est identifié par un **numéro** relatif au projet (le binôme N°1 du projet d'Entrepôt de données peut différer du binôme N°1 du projet de J2EE).

A chaque projet correspond deux notes :

- (1) une **note de rapport** qui est commune aux deux membres du binôme.
- (2) une **note de soutenance** qui, quant à elle, est individuelle et peut donc être différente pour les membres d'un même binôme.

Pour chaque binôme et chaque projet, la base de données stocke la date de remise effective du rapport à l'enseignant. Cette date permet de calculer un nombre de points à déduire de la note finale du projet en fonction du nombre de jours de retards. La note finale du projet n'est pas stockée dans la base de données mais est calculée à partir des notes de rapport et de soutenance (qui sont stockées dans la base de données) et du nombre de points en moins par jour de retard calculé à partir de la date de remise du rapport).

3. Interface graphique

On demande la réalisation de toutes les interfaces graphiques nécessaires à fin de répondre à toutes les fonctionnalités demandées dans le cahier des charges. Vous pouvez utiliser aussi bien Swing que JavaFX (d'autres outils sont autorisés).

4. Gestion des données

Il faut évidemment alimenté le projet par des valeurs pour chaque type de données. Prenez soin de saisir des valeurs significatives pour la bonne compréhension de ce dernier lors de l'exécution. Par exemple pour le nom d'un étudiant pas de aaaa mais plutôt Maurice Deschamps.

Pour le stockage des données trois solutions vous sont proposées :

- Utiliser les structures prédéfinis JAVA comme ArrayList, etc.
- Les fichiers csv, XML ou Json.
- Les bases de données (BD): MySQL.

L'utilisation d'une BD est fortement recommandée. Si des projets répondent parfaitement à tous les critères définis dans ce rapport et n'utilisent pas les BD ceci n'influencera nullement la note finale.

5. Conditions du rendu

Le format de rendu est une archive au format **ZIP**. L'archive aura pour nom Nom1Nom2.zip, où Nom1 et Nom2 sont les noms des membres du binôme par ordre alphabétique. L'extraction de l'archive devra créer un dossier Nom1Nom2 contenant les éléments suivants :

- Un répertoire **src** avec les sources de votre implémentation java.
- Votre **jar exécutable** pouvant être lancés au moyen de la commande java -jar avec pour nom la concaténation du nom des membres du binôme contenant toutes les dépendances nécessaires à l'exécution du projet.
- Un fichier **README** contenant vos prénoms, noms ainsi que les commandes à taper pour compiler et exécuter votre projet.
- Un **Maven** ou un **ant** (ou un **makefile**) permettant de compiler et exécuter votre projet simplement en gérant les dépendances.
- Un répertoire **doc** contenant :
 - Une documentation pour l'utilisateur **user.pdf** décrivant à un utilisateur comment se servir de votre projet (interfaces, utilisation, options...).
 - Une documentation pour le développeur **dev.pdf**, devant justifier les choix effectués (héritage/généralisation, polymorphisme, etc.), présenter l'architecture choisie (diagramme de classes), indiquer quelles ont été les difficultés rencontrées au cours

du projet ainsi que la répartition du travail entre les membres du binôme. Ce rapport doit faire le point sur les fonctionnalités apportées, celles qui n'ont pas été faites (et expliquer pourquoi). Il ne doit pas paraphraser le code, mais doit rendre explicite ce que ne montre pas le code. Il doit montrer que le code produit a fait l'objet d'un travail réfléchi et minutieux (comment un bug a été résolu, comment la redondance dans le code a été évitée, comment telle difficulté technique a été contournée, quels ont été les choix, les pistes examinées ou abandonnées...). Ce rapport est le témoin de vos qualités scientifiques mais aussi littéraires (style, grammaire, orthographe, présentation).

- Un **rapport d'expérimentations** étudiant les choix algorithmiques et leurs effets (telle optimisation, telle règle, tel choix de pièces, telle utilisation des threads, etc.).

Votre projet doit pouvoir s'exécuter sans utiliser Eclipse ! Une fois que vous avez fait l'archive, vérifiez que tout fonctionne sur un autre ordinateur !

Pour créer un jar exécutable avec les dépendances, vous pouvez utiliser un Maven ou autre. Sinon, dans Eclipse, si vous avez bien ajouté en "external jar file" au class-path de votre projet les librairies externes (ici après avoir [téléchargé le jar de commons-cli](#)), vous pouvez utiliser export > Runnable JAR file, choisir la configuration du Main de votre projet et bien cocher l'inclusion des librairies dans le jar généré.

Il va sans dire que les différents points suivants doivent être pris en compte :

- Uniformité de la langue utilisée dans le code (anglais conseillé) et des conventions de nommage.
- Projet compilant sans erreur et fonctionnant sur les machines de l'université.
- Gestion propre des différentes exceptions.
- Le code devra être propre, les exceptions correctement gérées, les classes correctement organisées en packages. La visibilité des méthodes et champs doit être pertinente (privée ou non...).
- Le projet doit évidemment être propre à chaque groupe. **Attention** : un détecteur automatique de plagiat sera utilisé sur l'ensemble des projets (groupes confondus). Si du texte ou une portion de code a été empruntée (sur internet, chez un autre binôme), il faudra l'indiquer dans le rapport. Il existe certainement du code sur Internet lié à ce projet : il sera ajouté dans le détecteur de plagiat. Tout manque de sincérité ira **directement en conseil de discipline**.
- La documentation (rapports, commentaires...) compte dans la note finale. On préférera un projet qui fonctionne bien avec peu de fonctionnalités qu'un projet bancal avec plus de fonctionnalités.
- **Une soutenance** de 10/15 minutes aura lieu par binôme à la date précisée en début de document. Elle doit être préparée et menée par le binôme (i.e. fonctionnant parfaitement du premier coup sur les machines UNIX). Les projets seront

téléchargés (au moment de la deadline) sur les machines (donc vous n'avez pas besoin d'amener votre ordinateur et vous n'aurez pas besoin de vous loger). Pendant la soutenance, ne perdez pas de temps à nous expliquer le sujet : nous le connaissons puisque nous l'avons écrit. Essayez de montrer ce qui fonctionne et de nous convaincre que vous avez fait du bon travail.

Bon courage