

Projet de modélisation - Patron MVC

Guillaume Santini

9 janvier 2025

1 Critères de réalisation et d'évaluation

1.1 Groupes de travail, modalités et date de remise

Les projets sont à réaliser par groupe de trois étudiants et seront à rendre avant le dimanche 2 février 2025 23h.59min. sur la plateforme moodle de l'université.

Les groupes sont définis et déclarés lors de la séance du mardi 14 janvier 2025.

Un seul membre de chaque groupe dépose le projet commun sur le moodle.

Le dépôt prend la forme d'un fichier archive au format ZIP dont le nom est de la forme `NOM1_Prenom1_NOM2_Prenom2_NOM3_Prenom3.zip`.

1.2 Éléments à remettre à la fin du projet

1.2.1 Fichiers constituant l'archive

L'archive contient un répertoire unique nommé `NOM1_Prenom1_NOM2_Prenom2_NOM3_Prenom3/` qui contient lui-même :

- `src/` un répertoire contenant tous les fichiers sources `.java` permettant de compiler votre code (merci de faire le ménage et ne pas m'envoyer les fichiers `.class`),
- `doc/` un répertoire contenant la Javadoc de votre projet,
- `rap/` un répertoire contenant :
 - `rapport.pdf` le fichier contenant le rapport de votre projet,
 - `mvc_general.pdf` un fichier contenant le diagramme de classe UML général reprenant l'ensemble des classes.

1.2.2 Modélisation

En plus du diagramme général, il vous est conseillé de proposer d'autres diagrammes plus simples dans votre rapport pour illustrer votre travail de modélisation. Les diagrammes simples doivent servir à expliquer plus en détail les aspects de votre modèle que vous souhaitez discuter.

Vous pouvez éventuellement joindre des diagrammes UML supplémentaire parmi les suivants :

- diagramme de cas d'utilisation,
- diagramme de classe,
- diagramme de séquence.

1.3 Implémentation

Le code est à produire en Java et doit être compilable et exécutable depuis un terminal.

Toutes les vues publiques doivent être documentées par une Javadoc, et il est recommandé de commenter raisonnablement le code.

Les tests unitaires ne sont pas requis.

2 Le sujet

L'objectif de ce travail est d'évaluer votre capacité à appliquer les principes SOLID et le patron de conception MVC dans le cadre de la production d'une application.

L'application que vous allez proposer est laissée à votre choix avec quelques exceptions et quelques contraintes :

- L'application devra proposer une interface homme-machine. Elle devra mettre en application les principes du patron de conception MVC. L'interface utilisateur sera un terminal de commandes.
- Dans certains cas il pourra être nécessaire d'intégrer d'autres patrons de conception vus en cours (Stratégie, Observateur, ...) sans que cela soit exigé. Notez sur ce point que si vous avez par exemple besoin d'intégrer une Stratégie, il faut l'intégrer et le discuter. Si ce n'est pas nécessaire, ne forcez pas les choses.
- L'application ne devra pas nécessiter l'installation de librairies autres que les librairies standards,
- Pas de sujet sur la gestion d'une bibliothèque, d'un thermomètre, ni sur des jeux de Morpion ou de Puissance 4 s'il vous plait.

Vue l'ouverture du sujet, je m'attends à ce que tous les groupes me rendent des choses assez différentes. Si vous avez un doute sur votre sujet, soumettez-moi vos idées en TD/TP, je pourrai vous orienter. Vous pouvez parfaitement reprendre un projet réalisé dans un autre contexte et l'adapter aux contraintes de cet exercice.

3 Les critères d'évaluation

Plusieurs critères seront considérés lors de l'évaluation de votre travail :

- Ambition et originalité du projet : il s'agit ici de récompenser votre investissement dans ce projet. Il est normal que les étudiants qui font des choses ambitieuses soient récompensés vis à vis de ceux qui fournissent un travail minimal ou insuffisant. Ce point pourra être discuté en TD/TP avec l'enseignant,
- Exécution du code : le code doit compiler et s'exécuter. Il doit être robuste (ne pas planter) et faire ce que l'on attend de lui.
- Correspondance du code et du modèle : le code ne sera pas passé en revue finement. Il sera cependant vérifié qu'il correspond bien au modèle proposé.
- Respect des bonnes pratiques : il faut respecter les principes de base de la programmation objet (organisation du code, nomage des classes, des méthodes et des variables, gestion des exceptions, encapsulation, documentation de la vue publique, ...),
- Respect des principes SOLID : ceux-ci devront être discutés dans une certaine mesure dans votre rapport, mais surtout être mis en pratique dans votre code et votre modélisation,
- Clarté des documents explicatifs : un diagramme UML sert à clarifier un point de modélisation ou de spécification. Il doit être discuté. Si le diagramme n'est pas clair il ne sera pas compris. Veillez également à ce que les images de votre rapport soient lisibles.

La taille du rapport ne doit pas dépasser 10 pages (hors page de titre et de table des matières, sans annexes), police de caractères de taille 10 non condensée, simple interligne. Il vous faudra donc être sélectif et convainquant dans la présentation de votre travail et le choix des diagrammes intégrés et discutés dans le document.