

Projet Java IV 2024-2025

HELBHotel

Introduction :

La société HELBHotel désire moderniser la gestion des réservations pour sa chaîne d'hôtels. Pour cela, elle vous demande de créer un logiciel test dédié aux opérateurs présents sur place.

Description du problème :

Un hôtel est composé de plusieurs chambres réparties sur plusieurs étages, numérotés de 1 à n. Chaque étage possède la même configuration, définie dans un fichier « .hconfig », décrit plus loin dans ce document.

L'hôtel propose plusieurs types de chambres :

- Economique
- Business
- Luxe

Les réservations sont effectuées à l'avance sur le site internet de l'hôtel. Elles sont ensuite agrégées dans un fichier « reservation.csv » mis à jour quotidiennement. Les réservations valides apparaissent dans un espace dédié de l'interface.

Les réservations contiennent les informations suivantes :

- A) Nom, prénom
- B) Nombre de personnes (1 à 4)
- C) Fumeur/non-fumeur
- D) Motif du séjour : Tourisme/Affaire/Autre
- E) Nombre d'enfants présents

Une réservation est valide si tous ses champs sont renseignés correctement. Elle n'est pas valide si l'une des informations (autre que le code de réduction) manque, si le nombre de personnes n'est pas compris entre 1 et 4 ou si le nombre d'enfants spécifié n'est pas cohérent avec le nombre de personnes renseignées (sachant qu'on ne peut réserver une chambre sans, au moins, un adulte).

Une chambre est assignée automatiquement par le programme en fonction des caractéristiques de la réservation et d'autres critères définis dans la suite de ce document. Si la chambre assignée ne convient pas au client, il peut demander une autre chambre. L'opérateur peut alors soit demander au programme de lui proposer une autre chambre, toujours en fonction des caractéristiques de la réservation, soit choisir manuellement une chambre pour le client.

Lorsque le client quitte l'hôtel, il se présente chez l'opérateur. Ce dernier doit alors libérer manuellement la chambre, puis demander au client de noter son expérience sur une échelle de 1 à 5 étoiles.

L'hôtel propose un système de loterie afin d'encourager les clients à revenir passer une nuit. Une fois la chambre libérée, un ticket de loterie est généré pour le client. En fonction des caractéristiques de la chambre réservée et de l'évaluation du client, un ticket de loterie-jeu est attribué.

Il existe actuellement trois types de tickets de loterie-jeu, et tous permettent de gagner une réduction pour la prochaine visite :

- Le ticket Gold
- Le ticket Silver
- Le ticket Bronze

Le ticket Bronze propose le jeu suivant : Deux portes sont disponibles. Si le client choisit la bonne porte, il gagne une réduction de 25 % lors de sa prochaine visite.

Le ticket Silver propose le jeu suivant : Un mot est choisi au hasard et est présenté avec ses lettres dans le désordre. Si le client parvient à retrouver le mot original, en un seul essai, il gagne une réduction de 50 % pour sa prochaine visite.

Le ticket Gold propose le jeu suivant : Le client se voit proposer un nombre de portes équivalent au numéro de l'étage dans lequel il a résidé, augmenté de 2. Par exemple, si le client résidait au 3^e étage, il se voit proposer 5 portes. Parmi celles-ci, une est la porte gagnante. Le client joue alors le jeu du "Monty Hall", c'est-à-dire qu'à chaque tour, il décide s'il garde sa porte ou s'il en change, et une porte perdante est éliminée, jusqu'à ce qu'il ne reste que deux portes, et que la partie se termine sur la victoire ou l'échec du joueur. Si le client gagne, il obtient 100 % de réduction pour sa prochaine visite.

Il est demandé qu'au plus la chambre est luxueuse (Luxe > Business > Economique) et au plus l'évaluation est positive, au plus les chances d'obtenir un ticket Gold sont grandes.

Les réductions sont des codes uniques que le client peut présenter lors de sa visite à l'opérateur et qui lui octroient la réduction gagnée.

Concernant les codes de réduction, pour des raisons de sécurité, l'hôtel ne souhaite pas conserver les codes générés par l'application et la réduction associée. On vous demande donc de pouvoir générer des codes de réduction respectant les contraintes suivantes : Le code doit être composé d'une combinaison de 10 caractères.

- Le code et son contenu ne doivent pas être compréhensibles pour le client (par exemple, éviter des formats comme REDUCTIO50).
- Chaque code généré devrait être unique, ou les chances de générer deux codes identiques raisonnablement faible et facilement paramétrable dans le code source.
- Le code doit inclure le pourcentage de réduction octroyé, de manière qu'il soit déchiffrable uniquement par l'application.

L'hôtel permet d'assigner automatiquement une chambre en fonction des caractéristiques de la réservation et du mode de réservation choisi par l'opérateur de l'hôtel. L'interface de l'opérateur devrait proposer les modes de réservation suivants :

- Random Assignment : Une chambre est choisie au hasard parmi toutes les chambres disponibles.
- Quiet Zone : La chambre est assignée de sorte que les fumeurs aient toujours accès à une fenêtre et que les clients avec enfants ne soient pas directement à côté d'une chambre sans enfants.
- Stay Purpose : La chambre est assignée en fonction du motif du séjour du client. Si le motif est "Affaire", la chambre assignée sera de type Business. Si le motif est "Tourisme" ou "Autre", la chambre sera déterminée selon les critères suivants : si le client est non-fumeur et sans enfants, il se verra attribuer une chambre Luxe. Dans le cas contraire, une chambre Économique sera assignée.
- Sequential Assignment : Afin d'aider à la logistique de nettoyage, on applique la même logique que pour Stay Purpose, mais en "collant" autant que possible toutes les chambres, en partant du 1er étage.

Les chambres sont numérotées de la manière suivante :

- Une lettre de A à Z pour indiquer l'étage.
- Un chiffre représentant le numéro de la chambre à cet étage.
- Un caractère parmi {E, B, L} pour désigner le type de chambre (Économique, Business, Luxe).

Par exemple, C2B désigne une chambre située au 3e étage, chambre n°2, de type Business.

Interface utilisateur :

L'interface principale destinée à l'opérateur de l'hôtel est schématisée dans les maquettes suivantes. Elle se divise en deux parties principales.

La partie gauche permet de visualiser les chambres d'un étage donné. Cette représentation reprend la structure du fichier « .hconfig » mentionné précédemment. Un menu déroulant (dropdown menu) indique l'étage affiché en affichant successivement la lettre et le numéro de l'étage associé. Ce menu permet également de changer l'étage à afficher. Les chambres sont colorées en fonction de leur catégorie, et une légende rappelle visuellement à l'opérateur le code couleur associé à chaque catégorie. Enfin, lorsqu'une chambre est occupée, elle apparaît en rouge sur l'interface.

La partie droite affiche les réservations du jour. Pour chaque réservation, on y trouve la première lettre du prénom suivie du nom de famille, ainsi que la proposition d'assignation de chambre. La couleur de chaque réservation respecte

le code couleur des chambres décrit précédemment. Les réservations peuvent être triées soit par nom de famille, soit par ordre de chambre proposée en partant du 1er étage. Si de nombreuses réservations sont présentes, une barre de défilement permet d'afficher les réservations suivantes. Cette section permet également de modifier le mode de réservation des chambres, ce qui met à jour l'affichage des réservations.

De plus, un bouton de rafraîchissement situé à droite de chaque réservation permet d'actualiser la proposition d'assignation de chambre, en fonction du mode d'assignation sélectionné.

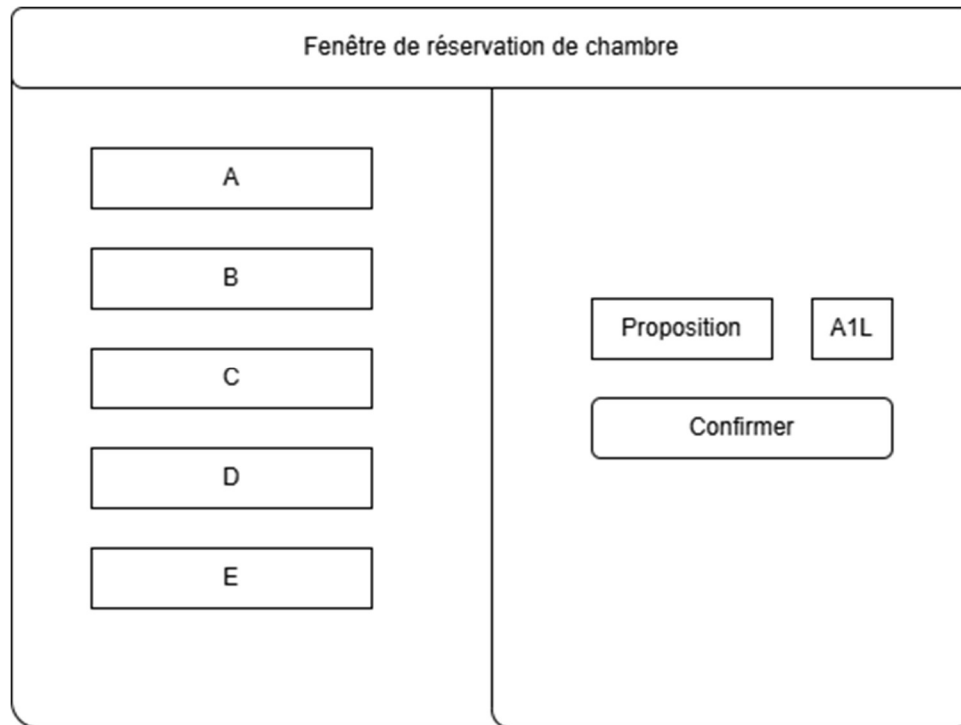
Lorsque l'opérateur confirme l'assignation d'une chambre dans la fenêtre de réservation, la réservation disparaît de cet espace dédié aux réservations.

The interface shows a reservation management system. On the left, a grid of room cards is displayed for 'Floor : A1'. The cards are color-coded by room type: Luxe (purple), Business (blue), and Economique (orange). The right panel contains controls for verifying codes, random assignment, and sorting by name. Below these controls, a list of reservations is shown, each with a refresh button (circular arrow icon) to update the room assignment based on the selected mode.

Room Type	Room Number
Luxe	A1L
Business	A2B
Economique	A3L
Economique	A4E
Economique	A5E
Economique	A6E
Economique	A7E
Business	A8B
Economique	A9B

Guest Name	Room Number
A.Anderson	A1L
B.Brown	A4E
C.Crow	C4E



Lorsque l'opérateur clique sur une réservation, cela affiche la fenêtre de réservation schématisée ci-dessous.



Le schéma représente une interface utilisateur pour la réservation d'une chambre. Elle est divisée en deux sections principales par une barre d'en-tête et une ligne de séparation. L'en-tête, qui s'étend sur toute la largeur, est intitulé "Fenêtre de réservation de chambre". La section de gauche, occupant la moitié gauche de l'écran, contient une liste verticale de cinq boutons rectangulaires, chacun contenant une lettre majuscule : "A", "B", "C", "D" et "E". La section de droite, occupant la moitié droite, est destinée à afficher les détails de la proposition. Elle contient deux boutons rectangulaires alignés horizontalement : "Proposition" à gauche et "A1L" à droite. En dessous de ces deux boutons, il y a un seul bouton rectangulaire centré, intitulé "Confirmer".

Dans cette interface, la partie gauche permet de visualiser les informations relatives à la réservation. La partie droite propose un espace éditable, prérempli par la proposition de réservation automatique. Si le client est satisfait de cette proposition, l'opérateur peut confirmer la réservation. En revanche, si ce n'est pas le cas, l'opérateur a la possibilité de modifier cet espace pour proposer manuellement une autre chambre.

Si la chambre proposée est une chambre existante et libre, l'opérateur peut confirmer la réservation en cliquant sur le bouton de confirmation. Si la chambre n'est pas disponible, le bouton sera grisé et il ne sera pas possible de le cliquer.

Si l'opérateur clique sur le bouton « Verify Code », situé en haut à droite de l'interface principale, une nouvelle fenêtre s'ouvre pour permettre la vérification des codes de réduction. Dans cette fenêtre, un label indique le nombre de caractères requis pour un code valide. L'opérateur peut alors saisir manuellement le code fourni par le client. Si le code est valide, un symbole «  » apparaît, accompagné du pourcentage de réduction correspondant. Dans le cas contraire, un symbole «  » s'affiche, avec un message précisant qu'aucun code valide n'a été saisi. Voici un schéma représentant cette fenêtre de vérification.

Fenêtre de vérification de code

Entrez le code (10 caractères)

Aucun code valide actuellement

Si l'opérateur clique sur une chambre occupée dans l'interface principale, il pourra également visualiser les informations de réservation. Cette fois, l'interface proposera de libérer la chambre.

Fenêtre de libération de chambre

A

B

C

D

E

Libérer la chambre

Lorsque l'opérateur clique sur le bouton de libération de la chambre, un pop-up de confirmation apparaît. Si l'opérateur confirme la libération de la chambre, l'interface propose alors de noter la qualité du séjour. Ensuite, un ticket de jeu est attribué, permettant au client de jouer à l'un des jeux mentionnés précédemment, dans l'espoir de gagner un code de réduction.

Pour chacune des fenêtres qu'il est possible d'ouvrir à partir de l'interface principale, l'ouverture de celle-ci « fige » l'interface principale, ce qui signifie qu'il

n'est pas possible d'interagir avec cette dernière tant que toutes les autres fenêtres ne sont pas fermées.

Fichiers ressources :

Le fichier « .hconfig » décrit la composition des étages de l'hôtel. Dans un souci d'uniformité, tous les étages de l'hôtel ont la même configuration, et il n'y a pas de chambre au rez-de-chaussée. Le fichier « .hconfig » se présente comme suit :

```
5
B, Z, E, Z
E, Z, Z, Z
E, Z, Z, B
E, Z, Z, L
```

La première ligne décrit le nombre d'étages de l'hôtel contenant des chambres. Les lignes suivantes schématisent la composition de l'hôtel sous la forme d'une matrice de n lignes et m colonnes, toutes complètes. Celle-ci reprend la nomenclature déjà utilisée, ainsi que la lettre « Z » pour indiquer un espace vide. Dans cet exemple, la première ligne « B,Z,E,Z » représente une chambre Business, suivie d'un espace vide, puis d'une chambre Economique, et enfin d'un autre espace vide.

Si le fichier « .hconfig » n'est pas valide, par exemple en cas de nombre d'étages invalide, de matrice incomplète ou remplie de caractères invalides, le programme ne doit pas s'exécuter. Une exception de type « IllegalArgumentException » doit être levée, accompagnée d'un message pertinent permettant de comprendre le problème rencontré dans le fichier.

Le fichier « reservation.csv » décrit les réservations journalières. Chaque ligne du fichier correspond à une réservation effectuée sur le site de l'hôtel pour ce jour-là. Lorsqu'une réservation est valide, elle est affichée dans l'interface.

Une fois qu'une réservation valide a été traitée par le logiciel, elle disparaît du fichier de réservation. Les réservations invalides devraient donc s'accumuler en début de fichier. Tout au long de la journée, il est possible qu'une nouvelle réservation soit effectuée. Le fichier de réservation est alors mis à jour avec une ou plusieurs nouvelles lignes, qui devront être traitées par le logiciel.

Note importante :

Les contraintes et fonctionnalités du projet sont susceptibles d'évoluer au cours du temps. Pensez donc à adapter une stratégie de développement adéquate. Certains points de la description ne sont pas précisés ou sont laissés volontairement vagues. Il revient à vous de faire certains choix d'interprétations. Veuillez toutefois à ce que votre approche soit logique et justifiée.

Travail demandé :

Pour ce projet, vous allez devoir implémenter **une application JavaFX** en appliquant les concepts vus en cours tel que **la conception pilotée par les responsabilités** et **l'application adéquate de design patterns**.

Le projet devra être **développé individuellement**.

Contrainte de développement :

- Votre programme devra être compilable et exécutable dans un environnement Linux Ubuntu tel qu'une des machines virtuelles utilisées en cours et qui sera disponible sur le SharePoint d'eCampus. Un script bash nommé « run.sh » devra permettre la compilation et l'exécution de l'application en utilisant seulement la commande suivante « bash run.sh » en terminale. Si le fichier n'est pas fourni, ou si la compilation et l'exécution ne fonctionnent pas dans l'environnement spécifié, le projet ne sera pas corrigé et sanctionné d'un zéro. **Testez donc avant que tout fonctionne !**
- Votre programme devra être développé en utilisant exclusivement la bibliothèque graphique JavaFX, compatible avec la version d'OpenJFX et du JDK installé sur la machine virtuelle.
- Toutes les ressources disponibles sur eCampus peuvent être mobilisées.
- Votre programme devra prendre en compte **le(s) fichier(s) de simulation**, fourni par l'enseignant.
- A l'exception des commentaires, l'entièreté de votre programme **devra être codé en anglais** (nom de variables/fonction/classes/etc...).
- Votre code devra respecter les principes de designs orientés objet comme vu au cours. Pensez donc à faire des choix logiques de design de classes afin de produire un **code propre et maintenable**.

- Votre code devra présenter **une structure correcte et maintenable**.
Notamment :
 - Evitez la duplication de code.
 - Evitez les constantes magiques.
 - Evitez le code mort.
 - Commentez intelligemment et suffisamment votre code
 - Nommez correctement vos variables, méthodes et classes et organisez correctement votre code.

Un code dont la qualité sera jugée insuffisante pour cause de non-consideration suffisante d'un de ces aspects **sera sanctionné d'un zéro**.

Il est fortement conseillé de consulter la partie du cours de Java Q3 sur les bonnes pratiques de développement.

- Votre code doit être suffisamment commenté. Pensez donc, pour chaque méthode « complexe », à au minimum, expliquer ce que représentent les paramètres, ce que fait la méthode et ce qui est retourné. Un code insuffisamment commenté ne sera pas évalué et **sera sanctionné d'un zéro**.

Rapport :

Il vous est demandé de rédiger un rapport décrivant votre projet. Votre rapport devra contenir au minimum les sections suivantes :

Introduction : Cette section devra introduire votre projet. Décrire ce qui a été réalisé et présenter brièvement la structure de votre rapport. (max 1 page)

Fonctionnalités de base : Cette section devra expliquer les fonctionnalités offertes par votre application d'un point de vue à la fois fonctionnel et technique. (max 5 pages)

Analyse et Application des Design Patterns : Cette section devra expliquer la structure de votre implémentation en utilisant les outils d'analyses déjà vus durant votre parcours. Attention tous les diagrammes doivent être commentés. Vous devrez également dans cette section décrire les patterns utilisés et expliquer comment et pourquoi vous les avez utilisés. Les diagrammes d'analyse pourront vous aider à illustrer vos propos. (max 5 pages)

Limitations : Les limites de votre application, par exemple : dans quels cas d'utilisation votre application pourrait ne pas fonctionner comme prévu ? Y a-t-il des aspects techniques qui n'ont pas été traité ? Concernant ces différentes limitations, comment pourraient-elles être traitées par la suite ? Plusieurs points de vue sont possibles, il revient à l'étudiant de choisir les points qu'il considère les plus pertinents pour réaliser son autocritique. (max 2 pages)

Conclusion : Votre conclusion sur le projet. Ce que vous avez réussi à faire ou non durant le projet et les apprentissages que vous en tirez. (max 1 page)

Il vous est demandé de respecter le nombre maximum de pages par sections. La taille de la police d'écriture devra obligatoirement être supérieur ou égale à 12 et inférieur ou égale à 14 pour le contenu et les titres. Une grille d'auto-évaluation permettant de mieux comprendre les attendus de l'enseignant concernant votre rapport vous sera communiquée ultérieurement.

Maitrise des productions :

L'évaluation du projet vise à attester de la bonne maitrise, des concepts liés au cours et au développement du projet, par l'étudiant. Toute réalisation pour laquelle l'étudiant ne peut pas démontrer une maîtrise suffisante lors de l'évaluation orale ne sera pas prise en considération et sera évaluée comme nulle.

Deadline et remise :

La date limite pour la remise du projet est fixée au **dimanche 18/05 à 23h59**. Le projet devra être déposé sur eCampus à l'intérieur d'un fichier .zip contenant toutes les sources de votre projet ainsi que le rapport **au format .DOCX**. Attention certaines activités obligatoires autour du projet pourraient être annoncées hors session.

Développement et Triche (1/2) :

- Tout acte de triche sera sanctionné par **une note de fraude au bulletin et sera notifié à la direction qui pourra possiblement décider de sanctions supplémentaires.**
- Des parties de code réutilisés d'un projet existant (d'un autre étudiant ou disponible sur le net) sans références dans votre rapport et sans mention de l'utilité du code utilisé est toujours considéré comme une fraude
- Pour ce projet, **vous ne pouvez pas reprendre des parties du code d'un autre étudiant, de cette année, ou d'une année précédente.**
- Pour ce projet **vous ne pouvez pas vous inspirer/servir d'un code disponible sur internet.**

Si vous avez un doute, contactez l'enseignant le plus tôt possible afin d'éviter du refactoring inutile, ou pire, **une note de zéro/fraude.**

Développement et Triche (2/2) – Utilisation de LLMs :

- Pour ce projet, le seul LLM qu'il vous est permis d'utiliser est ChatGPT dans sa version gratuite. Son utilisation reste toutefois soumise à la condition suivante : vous devez créer un compte dédié au projet avec l'adresse électronique suivante, à créer également : nom.prenom.java.q3.2425@gmail.com et dont le mot de passe sera fourni à l'enseignant. Tous les prompts effectués en rapport avec le projet (implémentation et rapport) devront être fait sur ce compte. À tout moment l'enseignant doit pouvoir avoir une vue **rapide** sur **l'historique des prompts réalisés**. En particulier le jour de l'évaluation finale du projet. Il revient à l'étudiant d'assurer l'accès à cet historique. Tout manquement à cette consigne pourra entraîner **une note de zéro/fraude**. Ces prompts devront rester disponibles pour toute la durée de la session et jusqu'à la délibération de celle-ci. Attention, bien que ChatGPT soit autorisé, il ne vous est pas permis d'inclure dans votre projet des éléments autre que les briques fonctionnelles déjà vues au cours et dans les codes ressources disponible sur eCampus. Si vous incluez dans votre implémentation des éléments proposés par ChatGPT qui vous feraient sortir du cadre fixé par les contraintes de cet énoncé **vous serez pénalisé**. Notez que pour votre propre apprentissage, l'enseignant vous recommande de ne pas y recourir, ou du moins, d'essayer dans un premier lieu de vous en passer. Il est extrêmement déconseillé d'intégrer un résultat donné par ChatGPT dans votre travail sans en avoir fait une **relecture critique**.

Conseil pratique :

Voici quelques conseils qui j'espère pourront vous aider.

- Veillez à ce que votre code ne contienne pas de constantes magique et/ou de duplication qui serait facilement évitable avec l'utilisation de méthodes.
- Veillez à ce que votre code soit suffisamment commenté.
- Veillez à effectivement implémenter les designs patterns quand cela est pertinent.
- Réfléchissez en terme d'« attribution de responsabilités » en accord avec les principes vu au cours (segmentation logique en classes).
- Ne négligez pas la théorie du cours (vous serez interrogés dessus).
- Ne négligez pas votre rapport. Tachez d'y expliquer/justifier explicitement vos choix d'implémentation. (Exemple : pourquoi un héritage ici ? pourquoi avoir choisi (ou non) d'implémenter ce design pattern ? quel avantage en termes de structure ? ...)
- Prenez le temps de bien comprendre tout l'énoncé avant de vous lancer (et lisez la FAQ).
- **Vérifiez que votre code compile et run effectivement via le script bash prévu sur la machine Ubuntu présente dans le SharePoint du cours.**

FAQ :

- **Puis je ajouter d'autres sections ou sous-sections dans le rapport ?**

Oui. La partie rapport de ce document donne seulement la structure minimum.

- **Puis je coder ou rendre mon rapport en anglais ?**

Oui. Pour ce qui est du code vous devez obligatoirement coder en anglais.

- **Puis je programmer sur Windows avec Eclipse ?**

Oui vous pouvez programmer comme vous le désirez mais vous devez respecter les contraintes de ce document, notamment : votre code doit être exécutable sur un environnement Linux Ubuntu via un script run.sh que vous devez fournir en même temps que vos sources (voir les contraintes de développement). Une machine préconfigurée sera disponible sur le SharePoint, celle-ci pourrait être utilisée pour la démo de votre application lors de la défense du projet.

- **Le rapport est-il important ?**

Oui. Le rapport est une **pièce centrale de votre projet** et c'est le premier outil de communication qui me servira à juger de la bonne réalisation du projet, pas seulement du point de vue du code mais également de la méthodologie utilisée. Il convient pour le rapport d'être pertinent et complet quant aux informations contenues.

- **Que voulez-vous dire par « tous les diagrammes doivent être commentés ».**

Les diagrammes doivent servir à illustrer et appuyer vos explications sur la structure de votre implémentation. Ils ne remplacent aucunement un texte explicatif revenant sur les points d'attention.

- **Je n'ai pas réussi à tout réaliser. Est-ce que ça vaut la peine de vous rendre le projet ?**

Oui, veuillez toutefois à être claire sur les parties non implémentées. Il est très déconseiller de dissimuler ou d'« oublier » de mentionner qu'une partie n'a pas été réalisée. Veuillez toutefois à bien respecter les consignes. Par exemple, votre code doit pouvoir compiler avec le script bash demandé, la qualité du code doit être suffisante, etc...

- **Puis je réaliser le projet en groupe ?**

Non, le projet doit être réalisé individuellement.

- **Que voulez-vous dire par « Votre code devra respecter les principes de design orienté objet comme vu au cours. »**

L'orienté objet fait intervenir certains principes comme l'héritage ou les méthodes statiques. Il revient à vous de décider quand les mettre en œuvre ou non. Votre

approche devra toutefois être logique et justifiée. Cela implique notamment, de faire apparaître de l'héritage quand cela a du sens, de rendre une classe abstraite quand cela a du sens, d'utiliser intelligemment l'encapsulation etc...

- **Dois-je vraiment faire de l'orienté objet ? Mon programme peut fonctionner sans.**

Vous devez absolument mettre en œuvre l'orienté objet pour ce projet. Cela fait partie des contraintes du projet. Un non-respect de ces contraintes sera pénalisé. L'utilisation de l'orienté objet n'est pas une contrainte faible. Veillez donc à la respecter.

- **Doit-on faire de l'encapsulation ou de l'héritage et du polymorphisme ?**

Vous devez toujours en faire usage quand cela a du sens et ne pas le faire est généralement un signe de mauvaise conception.

- **Dois-je vraiment utiliser des design patterns ? Mon programme peut fonctionner sans.**

Vous devez absolument mettre en œuvre les design patterns pour ce projet. Cela fait partie des contraintes du projet. Un non-respect de ces contraintes sera pénalisé. L'utilisation des design patterns n'est pas une contrainte faible. Veillez donc à la respecter. Il ne s'agit toutefois pas d'utiliser les patterns sans réflexion quant à leur utilité dans le projet. Leur utilisation et choix ou non choix doit être pertinente et justifiée.

- **L'aspect graphique et l'ergonomie sont-ils des critères importants ?**

Ces critères seront susceptibles d'être pris en compte dans l'évaluation mais sont nettement moins importants que l'implémentation des fonctionnalités et le respect des contraintes.

Dis grossièrement : Mieux vaut un programme moche mais avec toutes les fonctionnalités implémentées qu'un programme magnifique mais avec des fonctionnalités manquantes et un design logique (architecture du code) désastreuse.

- **Puis je utiliser un outil comme Scene Builder pour la création d'interfaces graphique ?**

Oui tant que le code généré fasse partie de la bibliothèque d'interfaces utilisateur JavaFX.

- **Puis je utiliser des designs patterns qui n'ont pas été vu au cours ?**

Oui mais vous devez les définir, justifier leur utilisation et fournir des sources. Vous devez dans tous les cas justifier leur utilisation.

- **L'interface graphique doit-elle être exactement celle présentée dans l'énoncé ?**

Les informations et fonctionnalités présentes dans les maquettes doivent apparaître dans l'application. Vous ne devez pas ajouter de fonctionnalités, ne pas modifier la demande du client, mais vous êtes libres de parfaire l'esthétique et l'ergonomie.

- **Comment m'assurer que mes choix d'interprétation sont logiques et justifiés ?**

En demandant à l'enseignant, qui joue le rôle de client/chef de projet. Cela est d'ailleurs représentatif d'un processus de développement réaliste.