

DM3 - Patrons de Conception

Dans les deux derniers devoirs, vous avez conçu un système de gestion de réservations pour les voyages par avion, train et bateau. Dans ce devoir, vous allez appliquer des patrons de conception au design de ce système. Le but de ce travail est d'augmenter la qualité du design pour mieux supporter des extensions ultérieures.

Les besoins fonctionnels du système restent les mêmes que dans le devoir 1. Vous pouvez vous baser sur votre solution au devoir 2, mais ce n'est pas nécessaire.

Fabrique

Les entités de voyage par avion (aéroports, compagnies, vols), par train (gares, lignes, trajets) et par bateau (ports, compagnies, itinéraires) doivent être gérées de façon transparente. La création des entités de voyage doit être implémentée à l'aide de fabriques.

Singleton

Les fabriques doivent être implémentées à l'aide de singletons.

État

Les différents états d'un siège (ou cabine) doivent être implémentées à l'aide du patron d'état.

Observateur

Les volets administratif et client dépendent d'un même modèle à l'interne (une abstraction d'une éventuelle base de données). Ces deux vues doivent rester cohérentes entre elles, ainsi qu'avec le modèle interne. Le modèle interne est une abstraction de l'information commune de laquelle dépendent les volets administratif et client. Cette architecture doit être implémentée avec le patron de l'observateur.

Commande

Toutes les opérations effectuées par l'administrateur (ex. gestion des entités de voyage, assignations de prix) peuvent être annulées (undo). Ces opérations doivent donc être implémentées suivant le patron de commande.

Visiteur

Lors de la consultation administrative, les vols, trajets ou itinéraires sont représentés de la manière suivante (voir devoir 1):

YUL-YYZ:[AIRCAN]AC481(2014.11.28:06.00-

2014.11.28:07:24)|PS(0/12)474.00|AM(5/16)355.50|EL(150/200)237.00 (*exemple pour les vols d'avion*). Cette donnée représente le vol AC481 par AIRCAN partant de YUL le 28/11/2014 à 6h arrivant à YYZ le même jour à 7h24 ayant une classe Première étroite avec 0 sièges réservés sur 12 chacun coûtant 474\$, une classe Affaire moyenne avec 5 sièges réservés sur 16 chacun coûtant 355.50\$ et une classe Économie large avec 150 sièges réservés sur 200 chacun coûtant 237\$.

Lors de la vérification des disponibilités par le client, ils sont représentés de la manière suivante (voir devoir 1):

YUL-YYZ:[AIRCAN]AC481(2014.11.28:06.00-2014.11.28:07:24)|237.00|E50 (*exemple pour les vols d'avion*). Cette donnée représente le vol AC481 par AIRCAN partant de YUL le 28/11/2014 à 6h arrivant à YYZ le même jour à 7h24 ayant 50 sièges disponibles en classe Économie au prix de 237\$.

Utiliser le patron du visiteur pour l'affichage de ces entités de voyage.

Tâches

1. Produisez le design du système de réservation en y incluant les patrons de conception mentionnés ci-haut. Présentez un diagramme de classe logiciel et un diagramme de séquence ou de collaboration par patron de conception. Identifiez-les clairement dans le rapport et précisez quelle variante du patron vous avez utilisée, si c'est le cas.
2. Votre design doit également inclure deux patrons de conception distincts en plus de ceux requis. Identifiez-les clairement dans le rapport et précisez quelle variante du patron vous avez utilisée, si c'est le cas. N'inclure que les diagrammes de classe.
3. Produisez un diagramme de paquets qui illustre ce design. Notez qu'il n'est pas nécessairement bon d'avoir un paquet par patron.
4. Discutez comment la qualité de votre conception a été améliorée par rapport au design du devoir 2. Cette tâche reste libre, mais soyez le plus rigoureux et complet dans votre argumentation. Vous pouvez vous appuyer sur les principes et techniques d'évaluation de conception.
5. Implémentez le système en Java. Vous devez générer le code à partir du design logiciel, puis compléter le code dans les méthodes à la main. **Assurez-vous que le design et le code sont cohérents entre eux.** Si vous le souhaitez, vous pouvez faire une rétro-ingénierie (reverse-engineering) dans Visual Paradigm pour que le code et les diagrammes soient synchronisés. Lors de l'exécution, le système comprendra au moins trois vols, trois trajets et trois itinéraires par défaut. Cette "base de données" peut être simplement codée en dur (string dans le code ou fichiers). Un vrai système de base de données serait hors du cadre du devoir. Si le code ne compile pas, vous aurez 0 à cette tâche. Incluez un fichier readme.txt qui explique comment utiliser votre application. Il n'est pas requis d'avoir une interface graphique.

Bonus: Implémentez le système en Java avec une interface utilisateur graphique (GUI). La fenêtre principale comporte deux boutons qui ouvrent chacun une nouvelle fenêtre à chaque fois qu'on clique dessus. Un bouton ouvre une fenêtre pour le GUI de l'administrateur et l'autre ouvre une fenêtre pour le GUI du client. On peut donc avoir plusieurs fenêtres d'administration et client ouvertes en même temps, mais elles n'ont pas de dépendance directe entre elles. Grâce au patron de l'observateur, chaque changement effectué par une des fenêtres sera reflétée dans toutes les autres fenêtres ouvertes. Par exemple, si l'administrateur ajoute un nouveau vol, il s'affichera en temps réel sur la fenêtre du client qui affichait les vols disponibles si l'aéroport d'origine et la date de départ concordent. À l'inverse, si un client réserve une cabine, l'administrateur verra le nombre de cabines réservées pour cet itinéraire augmenter.

Barème

- Design Logiciel [60%]:
 - Diagramme de classe pour le patron de fabriques [7%]
 - Diagramme de séquence ou de collaboration pour le patron de fabriques [3%]
 - Diagramme de classe pour le patron du singleton [2%]
 - Diagramme de classe pour le patron d'état [4%]
 - Diagramme de séquence ou de collaboration pour le patron d'état [2%]
 - Diagramme de classe pour le patron de l'observateur [7%]
 - Diagramme de séquence ou de collaboration pour le patron de l'observateur [3%]
 - Diagramme de classe pour le patron de commande [8%]
 - Diagramme de séquence ou de collaboration pour le patron de commande [3%]
 - Diagramme de classe pour le patron du visiteur [8%]
 - Diagramme de séquence ou de collaboration pour le patron du visiteur [4%]
 - Diagramme de classe pour les deux patrons de conception additionnels [6%]
 - Diagramme de paquets [3%]
- Discussion et justification de la qualité du design [10%]
- Génération de code et implémentation du système en Java [30%]:
 - Les six patrons de conception requis [15%]
 - Les deux patrons de conception additionnels [5%]
 - Le reste de l'application [10%]
 - Omettre le fichier readme.txt [-2%]
- Bonus - Interface graphique [10%]

Téléversez toutes les images, fichiers source, fichiers de données, projet vpp, et y faire référence dans le rapport. Le fichier ZIP doit comprendre deux dossiers : Design et Code.

Ressources supplémentaires

Vous devez utiliser Visual Paradigm (<http://www.visual-paradigm.com/download/>) pour tous les diagrammes UML. Vous pouvez le télécharger et l'installer. La clé pour la licence est disponible sur StudiUM.

Informations pratiques

Le devoir vaut 14% de la note finale.

Voir la date de remise sur StudiUM à 23h55. Tout retard engendrera une pénalité de 5% par jour pour un maximum de deux jours.

Le devoir est à faire en équipe d'au plus trois.

La remise du devoir est un fichier ZIP qui comprend un fichier HTML simple (rapport.htm) ainsi que tous les fichiers additionnels nécessaires (.jpg, .java, .txt, .vpp, tous les autres fichiers que vous voulez remettre). Le rapport doit faire référence de manière explicite à tous les fichiers et inclure toutes les images directement visibles sur la page. Vous devez inscrire dans l'entête du rapport : le nom de tous les membres de votre équipe, les quatre derniers chiffres de vos matricules, vos courriels et le temps mis par chaque membre sur le devoir (pour des raisons statistiques uniquement). Votre solution doit être incluse en entier dans le body du rapport. Inscrivez toutes vos hypothèses. Puis décrivez votre solution pour chaque tâche sous forme de rapport. Vous pouvez trouver un exemple du rapport ici (<http://www-ens.iro.umontreal.ca/~syriani/courses/rapport.htm>).

Omettre le rapport engendrera une pénalité de 5%.