

Épreuve de conception logicielle

Durée : 2h – aucun document autorisé

Toute ambiguïté que vous pourriez rencontrer dans ce sujet devra être résolue en décrivant brièvement le choix que vous avez fait.

1. Petites questions (6 pts)

- a) Expliquez à quoi sert le patron de conception *Proxy* en donnant un exemple (simple) de mise en œuvre. (2 pts)
- b) Expliquez à quoi sert le patron de conception *State* en donnant un exemple (simple) de mise en œuvre. (2 pts)
- c) Expliquez les différences entre les patrons de conception *Decorator* et *Strategy*. (2 pts)

2. Conception UML – étude de cas (14 pts)

On considère une application pour gérer la vente de livres via Internet.

Elle permet à un utilisateur quelconque de se connecter au système, de parcourir un catalogue de livres proposés, et cela suivant différents critères de recherche. Des critiques, envoyées via Internet (au travers d'une interface dédiée à la saisie des critiques), sont proposées en plus des informations sur le livre (titre, auteur, résumé, etc.). L'internaute peut éventuellement commander un ou plusieurs livres. Dans ce cas, il doit remplir un bon de commande en ligne afin de s'identifier, de donner les informations concernant le mode de paiement ainsi que les livres & quantités commandées. On ne conserve pas les informations sur le moyen de paiement d'une commande sur l'autre. Les profils utilisateurs ne sont pas pris en compte non plus. Enfin, les livres sont toujours vendus au même prix ; il n'y a pas de remise.

L'entreprise gère également un stock (physique) de livres. Lorsqu'une commande est passée, si le livre est en stock, il est immédiatement envoyé au client. Dans le cas contraire, le livre doit d'abord être commandé chez le fournisseur. Il est ensuite envoyé au client dès qu'il est reçu. L'application permet également à tout client de suivre l'état de ses commandes (en cours, en attente du fournisseur, expédiée, arrivée).

On considère que l'entreprise ne gère pas complètement l'expédition des livres et utilise pour cela les services d'une entreprise spécialisée de transport. Le système envoie une demande de prise en charge au système de l'entreprise de transport en fournissant le nombre de livres et le poids total, il reçoit en retour un prix (interne, non facturé au client) et un numéro du colis. L'envoi est toujours facturé à 0,4 euros par livre, quel que soit leur prix ou leur nombre.

Pour simplifier l'étude, on considère que ce nouveau système ne gère pas la facturation avec cette entreprise, juste l'expédition.

Les questions a) et b) peuvent être traitées séparément.
La question c) dépend de la question b).

QUESTIONS :

- a) Donnez le diagramme de cas d'utilisation du système à développer (diagramme reliant les acteurs et les cas). *Une description détaillée de chaque cas d'utilisation n'est pas demandée.* Attention donc à bien choisir un nom *explicite* pour chaque cas d'utilisation et chaque acteur.
- b) Donnez le diagramme de classes du système, avec les classes, leurs attributs, les associations entre classes avec leur nom et cardinalité, les relations d'héritage, et les éventuelles classes d'association. *Les méthodes ne sont pas demandées.*
- c) Écrivez un scénario *Gherkin* (format d'entrée de *Cucumber*) correspondant à un scénario pertinent de l'achat d'un seul livre par un utilisateur possédant déjà un compte. Expliquez ensuite quelles actions vous effectueriez sur quels objets pour chaque phrase (c'est-à-dire chaque *stepdef Cucumber*) du background et du scénario *Gherkin* afin de l'implémenter. Vous pouvez utiliser du pseudo-code ou des explications en français pour cette deuxième partie de la question.