

Examen Programmation par objets (Java) - 2h

Nous étudions quelques éléments d'un logiciel de gestion pour une entreprise qui propose des coffrets-cadeaux tels que des coffrets-séjours, des coffrets-bien-être, des coffrets aventure, des coffrets-culture ou des coffrets-gastronomie. Un coffret-cadeau contient une ou plusieurs prestations telles que : étape d'un séjour, soin dans un institut, expérience sportive, visite dans un espace culturel, repas dans un restaurant.

1 Prestation

Question 1 Définissez en Java plusieurs interfaces pour représenter les prestations :

- Une interface *Iprestation*. Une prestation se caractérise par des méthodes pour connaître le prix, et pour connaître et modifier respectivement la durée et un descriptif.
- Une interface *IetapeSejour*. Une étape d'un séjour est une prestation qui a de plus des méthodes pour connaître et modifier la longueur (en km).
- Une interface *IbienEtre*. Une prestation bien-être est une prestation qui a de plus des méthodes pour connaître et modifier respectivement le soin effectué ("massage", "soin visage", "hammam", etc.) et si la prestation utilise ou non un bassin.

Utilisez obligatoirement des noms de méthodes construits sur le principe des noms des accesseurs (get, set).

2 Etape d'un séjour

Une étape est décrite par une durée, un descriptif, une longueur et un prix d'hébergement en demi-pension pour une personne. Vous trouverez ci-dessous les premiers éléments de la classe décrivant des étapes de séjour.

```
public class EtapeSejour {
    private double duree;
    private String descriptif;
    private double longueur;
    private double prixHebergementDemiPension;

    public double getDuree() {return duree;}
    public void setDuree(double duree) {this.duree = duree;}
    public String getDescriptif() {return descriptif;}
    public void setDescriptif(String descriptif) {this.descriptif = descriptif;}
    public double getLongueur() {return longueur;}
    public void setLongueur(double longueur) {this.longueur = longueur;}
    public double getPrixHebergementDemiPension() {return prixHebergementDemiPension;}
    public void setPrixHebergementDemiPension(double prixHebergementDemiPension) {
        this.prixHebergementDemiPension = prixHebergementDemiPension;
    }
}
```

Question 2 Aucun constructeur n'a été placé dans cette classe. Peut-on écrire dans un programme `EtapeSejour e = new EtapeSejour();` ? Justifiez votre réponse.

Question 3 Que devez-vous ajouter au code actuel de la classe *EtapeSejour* pour qu'elle implémente l'interface *IetapeSejour* ? Pour information, le prix d'une étape correspond au prix de l'hébergement pour une personne en demi-pension auquel on ajoute 2 euros de frais.

3 Etape gastronomique

Lors d'une étape gastronomique, le repas du midi est pris dans un restaurant réputé situé sur le chemin. Elle est, en plus des informations d'une étape, décrite par le nom d'un restaurant et le prix du repas proposé pour une personne.

Question 4 *Ecrivez les éléments suivants pour une classe **EtapeGastronomique**, sous-classe de **Etape** : la première ligne correspondant à la déclaration et le(s) attribut(s). N'écrivez pas les accesseurs, mais ils sont supposés exister pour la suite.*

Question 5 *Redéfinissez dans la classe **EtapeGastronomique** la méthode donnant le prix d'une étape de manière à ce que le prix d'une étape gastronomique soit le prix de l'hébergement en demi-pension et les 2 euros de frais (comme dans la super-classe) augmenté par le prix du repas en restaurant.*

Question 6 *Peut-on écrire dans un programme (**main**) les expressions suivantes (justifiez) :*

```
EtapeSejour e1; e1= new EtapeGastronomique();  
EtapeGastronomique e2; e2= new EtapeGastronomique();
```

Question 7 *On prévoit qu'une étape gastronomique doit durer plus de 3h. En accord avec cette contrainte, on redéfinit la méthode **setDuree** de la manière suivante dans la sous-classe **EtapeGastronomique**.*

```
public void setDuree(double duree) {  
    if (this.getDuree() >=3) super.setDuree(duree);  
    else System.out.println("la durée d'une étape gastronomique doit dépasser 3h");  
}
```

*Le principe de substitution de Liskov est-il vérifié du point de vue comportemental par cette spécialisation de **EtapeSejour** par **EtapeGastronomique** ? Justifiez votre réponse par des arguments détaillés et/ou un programme mettant en situation une substitution d'une étape de séjour par une étape gastronomique qui fonctionne/ne fonctionne pas.*

Question 8 *Dans la méthode **setDuree** de **EtapeGastronomique**, remplacez le signalement de l'erreur, actuellement effectué par un message affiché sur la console, par un signalement d'exception. Ecrivez tout le code nécessaire (classe d'exception, déclaration, signalement). Faut-il modifier les entêtes des méthodes **setDuree** dans les super-types ? Si oui, dites comment et écrivez le code correspondant, sinon justifiez.*

4 Coffret cadeau

Question 9 *Ecrivez une classe générique **CoffretCadeau**. Un coffret cadeau contient une collection de prestations ; la classe est paramétrée par un type d'éléments conformes à l'interface décrivant les prestations. On peut ajouter une prestation si elle n'est pas déjà dans la liste. On peut calculer le prix du coffret cadeau en faisant la somme des prix des prestations incluses.*

Question 10 *Ecrivez l'essentiel d'une classe **CoffretSejour**. Un coffret séjour contient seulement des étapes de séjour. Ecrivez l'essentiel d'une classe **CoffretBienEtre**. Un coffret bien-être contient seulement des prestations bien-être. Ecrivez un programme créant un coffret séjour et y mettant deux séjours, dont un gastronomique.*

5 Annotations relatives à la substitution comportementale

Question 11 *Ecrivez une annotation, portant sur les méthodes et sur la phase d'exécution, et permettant d'indiquer si une méthode respecte la substitution comportementale.*

Question 12 *Utilisez cette annotation sur deux méthodes de la classe `EtapeGastronomique`.*

Question 13 *Ecrivez une méthode statique prenant comme paramètre une chaîne de caractères représentant le nom d'une classe et affichant sur le flot de sortie standard les noms des méthodes de cette classe ne respectant pas la substitution comportementale.*