INFO 3 Université de Nantes

Programmation Objet Java 2013/2014

Examen n^o 1

Programmation Java

L'examen est à réaliser en 2h00 en monôme. Seuls vos projets Eclipse java (TP par exemple) et l'API de Java sont autorisés. Le TP doit être réalisé sur un poste fixe de la salle de TP.

Aucune question n'est acceptée. Si le cahier des charges vous semble flou, indiquez vos propres hypothèses en commentaire dans le code.

Les codes sources doivent appartenir à un *package* portant le nom du monôme, contenant lui-même un package pour chaque exercice (-1pt si ce n'est pas le cas) . Une archive *nomDuMonome.zip*, comprenant vos fichiers sources, votre *javadoc* et QuestionsDeCours.txt, est à déposer sur Madoc en fin de séance.

Exercice 1 : PolyBand [5pt]

L'entreprise *PolyBand* a besoin d'une application pour modéliser des groupes de musique. Voici le cahier des charges :

- un groupe est défini par les propriétés suivantes :
  - un style (Rock, Hard Rock, Jazz, ...)
  - un nom
  - des albums
  - des musiciens
  - un groupe ne peut contenir plus d'une fois le même musicien
  - un groupe ne peut contenir plus d'une fois le même album
- un groupe peut ne pas contenir d'album mais contient au moins un musicien
  - il faudra donc fournir plusieurs constructeurs pour respecter ces besoins (un groupe peut être créé en prenant des albums en paramètre ou bien sans prendre d'album en paramètre)
  - la création d'un groupe sans musicien doit lancer une exception "EmptyPolyBandException"
- un musicien est défini par les propriétés suivantes :
  - un nom
  - un prénom
  - un ou plusieurs instruments de musique (Guitare, Batterie, Basse, ...)
- un même musicien peut jouer dans plusieurs groupes
- un album est défini par les propriétés suivantes :
  - un nom
  - une année de parution
- un instrument de musique est simplement défini par son nom

Question :

1. Implémenter votre solution de modélisation d'un groupe de musique
2. Tester votre code via une classe interne. Le test consistera à instancier le groupe de rock Nirvana (Kurt Cobain-Guitare, Krist Novoselic-Basse, Dave Grohl-Batterie avec les Albums *NeverMind*(1991) et *In Utero*(1993)) et à afficher toutes les informations du groupe (sur le groupe, les albums et les musiciens). La création d'un groupe sans musicien doit aussi être testée.
3. Commenter votre code et générer la Javadoc

Exercice 2 : PolyEntreprise [5pt]

L'entreprise *PolyEntreprise* a l'organisation suivante :
- un employé est défini par :
  - un nom
  - un prénom
  - un âge
- un employé de l'entreprise est soit un développeur, soit un manager
- un manager est responsable d'un ensemble d'employés
- un employé a généralement un manager qui le dirige (seul le manager en chef, dirigeant de l'entreprise, n'a pas de manager)

Question :

1. Modéliser l'organisation des employés de *PolyEntreprise*
2. Tester votre code dans une classe `Test` en créant une hiérarchie d'employés ([manager1 [manager2 [employe1, employe2]], employe3, employe4]) et en l'affichant sur la console

Exercice 3 : PolyCocktail [6pt]

L'entreprise *PolyCocktail* veut une modélisation de ses cocktails.
- un cocktail est un mélange de plusieurs ingrédients
- les ingrédients possibles dans un cocktail sont : du jus d'abricot, du jus d'ananas, du jus de citron vert, du sirop de cerise, du sirop de grenadine et du soda à l'orange
- dans un cocktail, chaque ingrédient est associé à une dose. Par exemple un *abricot frappé* contient 4 doses de jus d'abricot, 3 de jus d'ananas, 1 de jus de citron vert et 1 de sirop de cerise. Un *indien* est composé de 1 dose de sirop de grenadine et de 10 doses de soda à l'orange

Question :

1. Modéliser les cocktails de *PolyCocktail*
2. Tester votre code dans une classe `Test` en créant un abricot frappé et en l'affichant sur la console
3. Proposer un moyen simple pour créer des instances de vos cocktails
4. Tester votre solution de création de cocktail
5. Proposer un moyen pour créer des copies (en profondeur) de vos cocktails
6. Tester votre solution de création de copie d'un cocktail (vérifier que la copie a les mêmes propriétés et que l'égalité sur les références, entre la copie et l'originale, est fausse)

Exercice 4 : Questions de cours [4pt]

Créer un fichier `QuestionsDeCours.txt` dans lequel vous noterez vos réponses.

Question :

1. En pratique, à quoi sert le concept objet de l'héritage ? (donner des exemples si nécessaires)
2. Quel patron de conception est utilisé pour implémenter les flux d'entrée/sortie en Java ?
3. Quel est l'intérêt des classes abstraites ?
4. Quelle est la principale différence entre les templates C++ et les generic Java ? Qu'en déduisez vous sur les contraintes d'utilisation des generic Java ?