Cours de programmation orientée-objet
Examen de seconde session 2015

Livres fermés. Durée : 3 heures 1/2.

Veuillez répondre à chaque question sur des feuilles séparées sur lesquelles figurent nom, prénom et section. Soyez bref et concis, mais précis.

1. On souhaite développer en Java une classe `DVDCatalog` permettant de gérer des catalogues de DVDs. Une instance de cette classe représente un ensemble (non ordonné) de DVDs, chacun de ceux-ci étant caractérisé par un titre de film, un réalisteur et un producteur (représentés par leur nom et prénom), et une année de sortie.

   La classe `DVDCatalog` doit pouvoir être instanciée sans fournir de paramètre, ce qui crée alors un catalogue vide. Elle doit aussi définir une méthode permettant d'ajouter un DVD à un catalogue, et une autre capable d'exporter l'ensemble des données d'un catalogue dans un format choisi par l'utilisateur.

   Afin de standardiser cette opération d'exportation, on impose que la classe `DVDCatalog` implémente l'interface suivante :

```
import java.io.IOException;
public interface DVDCatalogExporter
{
  void export(String filename, DVDCatalogFormat format) throws IOException;
}
```

   Dans cette interface, le paramètre `filename` spécifie le nom du fichier dans lequel la méthode `export()` écrit les données exportées. Toute erreur se produisant au cours de l'exécution de cette méthode doit déclencher une exception `IOException`. Le paramètre `format` est quant à lui destiné à spécifier le format selon lequel les données d'un catalogue doivent être exportées. Son type, `DVDCatalogFormat`, correspond à l'interface suivante :

```
public interface DVDCatalogFormat
{
  String beginDocument();
  String beginDVD();
  String outputTitle(String title);
  String outputDirector(String firstname, String lastname);
  String outputProducer(String firstname, String lastname);
  String outputYear(int year);
  String endDVD();
  String endDocument();
}
```

   Chacune des méthodes de cette interface a pour responsabilité de générer une chaîne de caractères représentant une donnée bien précise d'un catalogue (comme par exemple le titre d'un DVD, le nom et prénom du réalisateur, etc) dans le format choisi.

(a) Ecrire le code de la classe `DVDCatalog`.

(b) Ecrire le code d'une classe `BEncodeDVDCatalogFormat` implémentant valablement l'interface `DVDCatalogFormat`, et représentant le format *BEncode*. Une opération d'exportation dans ce format doit produire un fichier texte **d'une seule ligne** respectant les règles suivantes :

- Il existe quatre types d'*éléments* : entier, chaîne de caractères, liste et dictionnaire.
- Un nombre entier est précédé de « `i` » et suivi de « `e` ». Par exemple, « `42` » doit être formaté comme suit : « `i42e` ».
- Une chaîne de caractères est représentée par la concaténation de sa taille et de son contenu, séparés par « `:` ». Par exemple, « `test` » doit être formaté comme suit : « `4:test` ».
- Une *liste* est démarrée par « `l` » et terminée par « `e` ». Elle possède une énumération d'éléments de type quelconque placés les uns à la suite des autres.
- Un *dictionnaire* est démarré par « `d` » et terminé par « `e` ». Il contient une suite d'*entrées* définies comme la concaténation d'une *clé* sous forme de chaîne de caractères, et d'un *contenu* représenté par un élément de type quelconque. Les entrées doivent être énumérées par **ordre lexicographique** des clés.
- Le fichier contient une unique liste énumérant les DVDs les uns à la suite des autres.
- Un DVD doit être décrit sous la forme d'un dictionnaire à quatre entrées.
- Le titre d'un DVD constitue une entrée ayant « `5:title` » pour clé, et une chaîne de caractères comme contenu.
- Le réalisateur d'un DVD constitue une entrée ayant « `8:director` » pour clé, et un dictionnaire comme contenu. À l'intérieur de ce nouveau dictionnaire doivent se trouver le prénom (précédé de la clé « `5:first` ») et le nom (précédé de la clé « `4:last` »).
- Le producteur d'un DVD suit la même règle que le réalisateur, en remplaçant la clé « `8:director` » par la clé « `8:producer` ».
- L'année d'un DVD constitue une entrée ayant « `4:year` » pour clé, et un nombre entier comme contenu.

Par exemple, dans ce format, un catalogue contenant les DVDs :

- *Blade Runner*, réalisé par Ridley Scott, produit par Michael Deeley, et sorti en 1982.
- *Total Recall*, réalisé par Paul Verhoeven, produit par Buzz Feitshans, et sorti en 1990.

s'exporte de la façon suivante :

```
ld8:directord5:first6:Ridley4:last5:Scotte8:producerd5:first7:Michael4:last6:Deeleye5:t
itle12:Blade Runner4:yeari1982eed8:directord5:first4:Paul4:last9:Verhoevene8:producerd5
:first4:Buzz4:last9:Feitshanse5:title12:Total Recall4:yeari1990eee
```

Notes :

— On demande que chaque instance d'un catalogue puisse être **clonée en profondeur**.

— Vous êtes libre de développer des classes supplémentaires nécessaires à votre solution.

— Veillez à signaler les situations d'erreur par des exceptions implémentées par vos soins (il **n'est pas nécessaire** de vérifier les doublons).

— Il **n'est pas nécessaire** de récupérer l'ensemble des exceptions déclenchées lors de l'exécution de votre code, **sauf** les erreurs pouvant se déclencher dans la méthode `export`.

— Il **n'est pas nécessaire** d'insérer vos classes dans un autre package que celui par défaut.

— Une documentation pouvant se révéler utile est fournie en annexe.

2. Soit l'extrait de code ci-dessous :

```
class A
{
  public A()
  {
    System.out.println("A");
  }
}

class B extends A
{
  public B()
  {
    super();
    System.out.println("B");
  }
}

public class C extends B
{
  public C()
  {
    super();
    System.out.println("C");
  }

  public static void main(String[] args) {
    new C();
  }
}
```

Répondre aux questions suivantes **en justifiant** :

(a) Qu'est-il affiché sur la console suite à l'exécution du code ci-dessus ?

(b) Nommer et définir le mécanisme illustré dans le code ci-dessus tout en expliquant l'utilité du mot-clé `super`.

(c) Aurait-il été possible de placer le mot-clé `super` après les instructions `System.out.println()` dans le code ci-dessus ?

(d) Votre réponse à la question (a) resterait-elle inchangée suite à la suppression des deux occurrences de l'instruction `super()` dans le code ci-dessus ?

(e) En modifiant le code de la classe `A` de la manière suivante :

```
class A
{
  public A(int i)
  {
    System.out.println("A: " + i);
  }
}
```

Le code des classes `B` et `C` reste t-il correct ?

3. Répondre aux questions suivantes **en justifiant**. En Java :

(a) Quelle forme donne-t-on aux méthodes destinées à tester l'équivalence de deux objets ? Quelles propriétés doivent-elles satisfaire ?

(b) Que produit l'exécution de ce fragment de code ?

```
int[] a, b;
a = new int[3]; a[0] = 1; a[1] = 2; a[2] = 3;
b = new int[3]; b[0] = 1; b[1] = 2; b[2] = 3;
System.out.println(a != b);
```

Documentation Java

Cette annexe fournit la documentation de quelques classes de la bibliothèque standard Java pouvant être utiles à la résolution des exercices.

Note : l'utilisation de ces classes et méthodes n'est pas obligatoire !

Classe `java.util.Vector<E>` **:** file de données (politique FIFO)

- `void add(E e)` : ajoute la référence `e` en fin de file.
- `E get(int i)` : retourne la référence stockée à l'indice `i` (le premier élément possédant un indice égale à 0).
- `E set(int i, E e)` : remplace la référence stockée à l'indice `i` par la référence `e` et retourne l'ancienne référence.
- `E remove(int i)` : supprime la référence stockée à l'indice `i` et retourne cette référence. Tous les éléments situés à droite de celui qui est supprimé sont décalés d'une position vers la gauche.
- `boolean isEmpty()` : indique si la file est vide.
- `int size()` : retourne la taille de la file.

Classe `java.io.FileWriter` **:** écriture dans un fichier

- `FileWriter(String f) throws IOException` : instancie un objet ouvrant en écriture le fichier nommé `f`.
- `void write(String s) throws IOException` : écrit le contenu de `s` dans le fichier.
- `void close() throws IOException` : ferme le fichier.

Classe `java.lang.String` **:** représentation et manipulation de chaînes de caractères

---

- `length()` : taille de la chaîne de caractères représentée.

Manipulation de tableaux

- La variable `length` d'un objet tableau contient la taille de celui-ci.