Java DUT 1 Feuille TD6
Université Paris-Est Marne-la-Vallée

Exercice 1.—

On souhaite modéliser la gestion d'un catalogue de livres. Un livre peut figurer ou non dans le catalogue et s'il figure il peut figurer en plusieurs exemplaires. Le libraire doit pouvoir ajouter et enlever **un exemplaire** d'un livre du catalogue et aussi consulter le nombre d'exemplaires présents en stock d'un livre donné.

a) On reprend la classe `Book` de la feuille TD4 pour réprésenter des livres. Chaque livre possède un titre (`title`) et un auteur (`author`) qui sont des chaînes de caractères, c'est-à-dire des objets `String`. Les champs seront `private` et `final`.

b) Écrire une interface `BookCatalog` comprenant des méthodes
   - `void add(Book b)`
   - `void remove(Book b)`
   - `int getNumber(Book b)`

c) Écrire dans la classe `Book` une méthode `public boolean equals(Object o)` renvoyant `true` si deux Book ont les même champs et `false` sinon (voir TD4). Écrire une méthode `public int hashCode()` renvoyant un entier. Deux Book égaux par equals devront avoir le même hashCode.

d) Écrire une classe `BookCatalogImplemementation` implémentant l'interface `BookCatalog`. Le Catalog sera représenté par une Map, un objet `catalog` de la classe `HashMap<Book,Integer>` (voir la documentation de l'interface `Map` et de la classe `HasMap`). Une telle Map est une table de correspondance indiquant pour chaque Book son nombre d'exemplaires.
   - Écrire un constructeur dans `BookCatalogImplementation`.
   - Écrire la méthode `void add(Book b)` qui permet d'ajouter un exemplaire d'un livre. Si le livre n'était pas dans la Map, il sera ajouté en un exemplaire.
   - Écrire la méthode `int getNumber(Book b)` qui renvoie le nombre d'exemplaire d'un livre. On renverra 0 si le livre n'est pas présent dans le catalogue.
   - Écrire la méthode `void remove(Book b)` qui permet de retirer un exemplaire d'un livre. Si le livre apparaît avec un nombre d'exemlaire 0, il sera retiré de la Map.
   - Écrire une méthode `toString` dans `BookCatalogImplentation`.

e) Écrire une classe `BookCatalalogTest` avec un méthode `main` et tester vos classes.

f) A quel moment sont utilisées les méthodes `equals` et `hashCode` de `Book` ? Pourquoi est-il important que les champs de Book soient `final` ?

g) Ajouter une méthode `size` renvoyant la taille d'un catalogue qui est le nombre total d'exemplaires présents dans le stock. Dans quelles classes faut-il la mettre ?

h) Ajouter dans la classe `Book` une gestion d'exception capturable pour que la création d'un livre avec un titre `null` lève une exception. Comment modifier les autres classes et méthodes ?

Exercice 2.—

Ecrire une classe `WordCount` contenant une méthode main qui prend en argument une suite de mots entrés au clavier et retourne la liste des mots avec leur nombre d'occurrences.

Par exemple, si on lance

```
java WordCount toto titi toto tutu tata titi titi
```

la sortie obtenue sera

```
tata 1
titi 3
tutu 1
toto 2
```