INSTITUT SUPERIEUR POLYTECHNIQUE DE MADAGASCAR

Fahaizana-Fampandrosoana-Fihavanana

*** Code Sujet :**	**018-JAVA-01**
Matière :	**Java**
Niveau :	**L2**
Classe(s):	**ESIIA2-IGGLIA2- IMTICIA2-ISAIA2**

EXERCICE 1 – Application de la classe File de Priorité

1) Ecrire une classe `Voiture` (implements `Comparable<Voiture>`) :
   a. Avec, au moins, les attributs suivants :
      - Marque
      - Vitesse Max
      - Poids
      - Nombre de places
      - Annee de sortie
   b. Avec deux méthodes de votre choix (autres que les `get`..)
   c. Avec une redéfinition pertinente de la méthode `public String toString()`.
   d. On veut utiliser la classe `PriorityQueue<Voiture>`. Redéfinir la méthode `public int compareTo(Voiture autre)` héritée de l'interface `Comparable<Voiture>` . On veut donner la priorité aux voitures plus récentes.
      Pour indication :

```
//on compare this à autre,
//on retourne une valeur <0 si this est prioritaire par rapport à autre
//on retourne une valeur >0 si autre est prioritaire par rapport à this
//on retourne la valeur 0 sinon
@Override
public int compareTo(Voiture autre){
      return …
}
```

2) Dresser le diagramme de classes de la classe `Voiture`.
3) Si on voulait changer le critère de priorité, comment réécrire `compareTo()` pour chacune des règles suivantes :
   a. Ordre alphabétique des noms des constructeurs.
   b. Priorité aux voitures ayant le plus grand nombre de places. Si deux voitures ont le même nombre de places, donner la priorité à la voiture la plus rapide.
   c. Même règle que b. Mais si les voitures ont la même vitesse max et le même nombre de places, donner la priorité à la voiture la plus récente.

EXERCICE 2 – Indexation par une chaine de caractères

Partie A – Retrouver une note associée à un nom

On veut écrire une classe `Classe` qui contient les pairs (Nom, Moyenne) pour chaque étudiant. On veut pouvoir retrouver rapidement la note d'un étudiant à partir de son nom.

1) Quelle est la principale différence entre *`Set`* et *`Map`* (par exemple dans `HashSet<>` et `HashMap<>`) ?
2) Créer un tableau comparatif entre *`TreeMap`* et *`HashMap`*.
3) Est-ce qu'on pourrait utiliser un `ArrayList<>` ou un `LinkedList<>` ? Expliquer en une phrase.
4) Expliquer pourquoi un `HashMap<String,Integer>` serait idéal. Expliquer en une phrase.
5) Ecrire une méthode qui permet d'ajouter une paire `(Nom, Moyenne)` dans la classe.
6) Ecrire une méthode qui permet de retrouver rapidement la moyenne d'un étudiant donné.
7) Ecrire une méthode qui calcule la moyenne de la classe.

Partie B – On ajoute une information supplémentaire

Dans la même classe, on veut également stocker l'âge de l'étudiant.

8) Puisqu'on ne peut pas écrire `HashMap<String,Integer,Integer>`, proposer et implémenter une solution possible.
9) Proposer, sans implémenter une autre solution.

Notes :

- Si vous vous chronométrez, vous devriez terminer en deux heures
- N'oubliez pas de regarder régulièrement https://github.com/AndryRAB/JAVA-L2. Je vais y mettre notre projet sur l'infographie et la géométrie analytique et progressivement d'autres projets. Vous êtes encouragés à faire votre propre « *fork* ». Vous pourriez faire également des « *pull requests* » si vous avez des modifications à proposer.