UNIVERSITÉ DE REIMS CHAMPAGNE-ARDENNE

Licence 3 INFO
INFO0503

TP n°1

Modélisation et programmation *Java*

Le but de ce TP est de réaliser une modélisation et revoir les bases en Java en développant une application complète.

1 Constructeur automobile

Un constructeur automobile souhaite développer une application permettant de gérer sa production de voitures, ainsi que les commandes de clients. Une voiture est caractérisée par un numéro d'identification unique (une suite de lettres et de chiffres), son moteur, sa date de fabrication et la couleur de sa carrosserie. Elle peut posséder des options : vitres électriques, radar de recul, GPS, etc. Le constructeur produit plusieurs modèles de voitures. Par exemple, Renaud produit les voitures suivantes : Twingo, Clio, Captur, etc. Les moteurs des voitures sont caractérisés par un numéro unique, une carburation (essence, diesel, GPL, hybride) et une puissance.

Les voitures sont produites dans différentes usines, chaque usine possédant des parkings permettant de stocker temporairement les voitures. Les parkings sont numérotés, puis chaque place est identifiée par une rangée (une lettre) et un numéro dans la rangée (un entier). Pour gérer les stocks, le constructeur a besoin de connaître l'emplacement exact des voitures dans les différents parkings des différentes usines.

Nous devons pouvoir gérer aussi des commandes de clients (qui correspondent à des concessionnaires). Un client est identifié uniquement par un nom. Les commandes correspondent à des besoins en voitures (modèles, couleurs, moteur et options). Les voitures sont choisies dans un catalogue qui correspond à la liste des voitures proposées avec les options, couleurs, moteurs possibles. Elle est datée.

L'application que vous devez écrire doit permettre à l'utilisateur d'ajouter ou de supprimer des modèles de voitures et de gérer les parkings : ajouter, rechercher, déplacer ou supprimer des voitures. L'ensemble des données doit être sauvegardé dans un fichier (binaire), ce qui permet de conserver les données entre deux exécutions de l'application.

Questions

1. Proposez un diagramme de classes complet de l'application
2. Développez les différentes classes en *Java*
3. Réalisez l'application (en ligne de commande) demandée.

Le diagramme de classes complet sera à rendre avant le début de la prochaine séance.

2 Les collections en *Java*

Dans l'exercice précédent, nous avions besoin d'utiliser des ensembles (au sens large) de voitures. Il est possible d'exploiter les tableaux ou des collections proposées dans l'API *Java*.

Questions

1. En vous aidant de la *Javadoc*, déterminez les différents types de collection proposés dans l'API Java.
2. À l'aide d'un petit programme, générez des voitures aléatoirement (plusieurs milliers). En utilisant plusieurs types de collections :
   - Ajoutez les voitures dans la collection
   - Affichez toutes les voitures
   - Recherchez une voiture spécifique
   - Supprimez une voiture
3. Reprenez la question précédente et comparez l'utilisation des itérateurs et des accès "directs" proposés par les collections.
4. Comment est-il possible de comparer les différents types de collection ?
5. À partir de vos connaissances en algorithmique, puis de l'énoncé précédent, cherchez quelles collections sont les plus appropriées aux besoins.