TP - Agence de location

On s'inspire du sujet du TD sur les agences de location de voiture (toujours disponible sur le portail) en y apportant les modifications et extensions suivantes :

- l'attribut `voitures` de la classe `Agence` est du type `java.util.List<Voiture>`
- on ajoute à la classe `Agence` la méthode :
  `public List<Voiture> selectionne(Critere c)`
  dont le résultat est un itérateur sur la collection des voitures qui satisfont le critère `c` passé en paramètre.
  Vous modifierez la méthode `afficheSelection` afin qu'elle utilise cette méthode.
- le critère intersection de la question **Q6** peut maintenant regrouper un nombre quelconque de critères (et plus seulement deux) qui sont ajoutés via la méthode `addCritere`. Son diagramme UML est le suivant :

InterCritere
-lesCriteres : List<Critere>
+InterCritere() +addCritere(c: Critere) +estSatisfaitPar(v : Voiture)

- les questions Q5 et Q7 sont à placer dans un `main` grâce auquel vous effectuerez quelques tests en créant quelques objets voitures et en affichant les résultats de sélections.
- On ajoute à la classe `Agence` la gestion des locations des voitures. Un client ne peut louer qu'un véhicule à la fois. On créera une classe `Client` simple où les clients sont simplement modélisés par un attribut correspondant à leur nom qui sera une chaîne de caractères[1].
  On décide de gérer ces locations par une table (`java.util.Map`) qui associe les clients (clés) avec les voitures (valeurs) qu'ils ont louées. Un client n'est présent dans cette table que si il loue actuellement une voiture.
  Modifiez la classe `Agence` pour lui ajouter les méthodes ci-dessous. N'oubliez pas les tests !
  - `public void loueVoiture(Client client, Voiture v) throws java.util.MissingResourceException`
    permet au client `client` de louer la voiture `v`.
    L'exception est levée soit si la voiture n'existe pas dans l'agence soit si elle est déjà louée (consultez la documentation pour les paramètres du constructeurs de cette exception).
  - `public boolean estLoueur(Client client)`
    renvoie `true` ssi `client` est un client qui loue actuellement une voiture.
  - `public boolean estLoue(Voiture v)`
    renvoie `true` ssi la voiture est actuellement louée.
  - `public void rendVoiture(Client client)`
    le client `client` rend la voiture qu'il a louée. Il ne se passe rien si il n'avait pas loué de voiture.
  - `public Collection<Voiture> lesVoituresLouees()`
    qui renvoie la collection des voitures de l'agence qui sont actuellement louées.

Vous pouvez récupérer sur le portail des fichiers préparés et à compléter. Les versions "`AgenceTD`" et "`InterCritereTD`" correspondent aux réponses au sujet du TD.

[1] Attention donc à ne pas confondre les objets clients et leurs noms !