

TD Location de voitures

(Les classes/interfaces de cet exercice seront à ranger dans un paquetage **agence**.)

Une agence de location de voitures offre à ses clients la possibilité de choisir la voiture louée en fonction de différents critères.

Les voitures sont définies par une marque, un nom de modèle, une année de production et un prix de location à la journée. Pour simplifier les deux premiers paramètres seront des objets de la classe **String** et les deux derniers seront des **int**. Deux voitures sont considérées égales si tous leurs attributs sont égaux.

Q 1 . Donner le diagramme UML de la classe **Voiture** pour laquelle on souhaite disposer des méthodes habituelles **equals** et **toString**.

Il est possible de sélectionner parmi les voitures à louer toutes les voitures satisfaisant un critère donné.

On définit l'interface **Critere** ainsi :

```
public interface Critere {
    /**
     * @param v la voiture dont on teste la conformité au critère
     * @return true si et seulement si la voiture est conforme au
     * critère (on dit que v satisfait le critère)
     */
    public boolean estSatisfaitPar(Voiture v);
}
```

Q 2 . On suppose une classe **Agence** définie (au minimum) ainsi :

agence::Agence
- voitures : Voiture[]
...
+ Agence(...)
+ afficheSelection(c : Critere)
...

Donnez le code de la méthode **afficheSelection** qui affiche parmi toutes les voitures de l'agence (contenues dans l'attribut **voitures**) celles qui satisfont le critère donné.

Q 3 . Donnez le code d'une classe **CriterePrix** qui est un critère satisfait par toutes les voitures dont le prix est inférieur à un prix fixé à la construction du critère.

Q 4 . Donnez le code d'une classe **CritereMarque** qui est un critère satisfait par toutes les voitures d'une marque donnée. La marque est précisée à la construction du critère.

Q 5 . En supposant que la référence **agence** est de type **Agence** et a été initialisée, donnez la ou les lignes de code permettant d'afficher toutes les voitures de cette agence dont le prix est inférieur à 100.

Q 6 . On peut naturellement souhaiter faire des intersections de critères, ce qui revient à appliquer le **et** logique. On obtient alors un nouveau critère satisfait si tous les critères qui le composent sont satisfaits.

Définissez une classe **InterCritere** qui permet de définir des critères par intersection de deux critères. Les critères dont on fait l'intersection sont précisés à la construction.

Q 7 . Donnez la ou les lignes de code permettant de créer un critère intersection d'un critère pour la marque "Timoléon" et d'un critère pour un prix inférieur à 100.