

DEVOIR DE TP

ECUE : **Programmation Orientée Objets**

Classe(s) : **L2-DSI EAD**

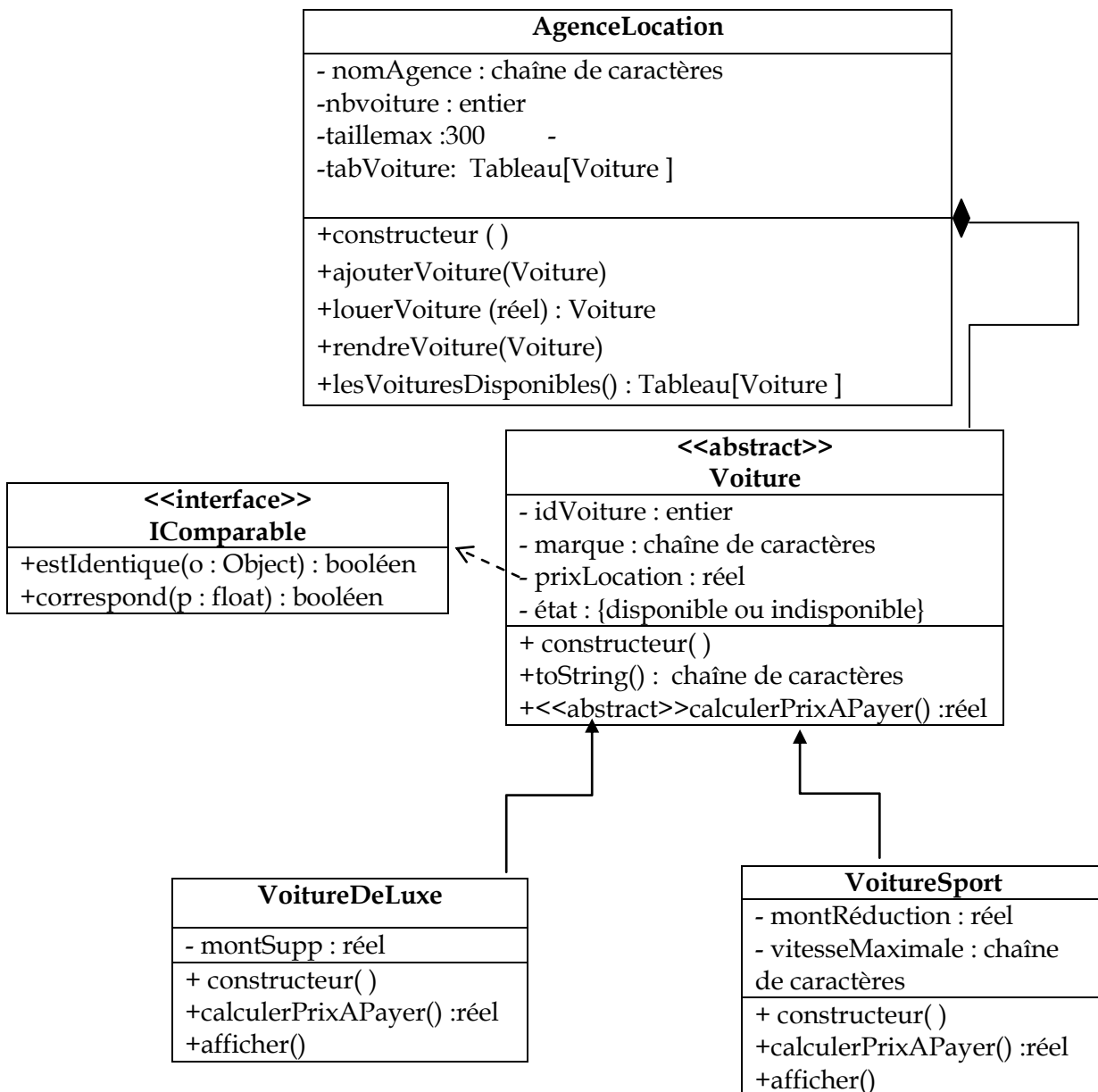
Date : 15/01/2023

Durée : **1h 30mn**

Nombre de pages : **2**

Problème :

On désire programmer une application de gestion des locations de voitures. Soit le diagramme de classes suivant :



Avec : L'interface **IComparable** est l'interface que doit implémenter la classe **Voiture** pour vérifier si deux voitures sont identiques cad ont le même idVoiture et aussi pour vérifier si une voiture a un prix à payer inférieur ou égal à un prix donné et qu'elle est aussi disponible.

Partie n°1 :

1. Écrire le code javade l'interface **IComparable**. (1 Point)

2. Écrire le code java de la classe **Voitures** sachant que : (4 points)

- L'idVoiture reflète l'ordre d'achat de la voiture (ajouter un attribut à la classe Voiture si c'est nécessaire)

- **constructeur(...)** qui initialise tous les attributs de la classe par des valeurs passées en paramètre sauf l'idVoiture qui doit être calculable et l'état.

Ecrire alors le code de ces méthodes :

- **toString()** : retourne une description des attributs d'une voiture (idVoiture, marque, état, le prix à payer) séparés par un espace.
- **calculerPrixAPayer ()** : calcule le prix à payer pour chaque type de voiture comme suit :
 - Le prix à payer pour la Voiture de luxe égal au prix de location auquel on ajoute un montant supplémentaire montSupp.
 - Le prix à payer pour la Voiture sport égal au prix de location auquel on déduit un montant de réduction montRéduction.

3. Écrire le code de la classe **VoitureDeLuxe** et **VoitureSport** .(3 Points)

Les méthodes à définir sont :

- **constructeur(...)** : initialise tous les attributs par des valeurs passées en paramètre.
- **calculerPrixAPayer()** : calcule le prix à payer pour une voiture de luxe (respectivement de sport) .
- **afficher ()** : affiche toutes les informations d'une voiture de luxe (respectivement de sport) .

4. Ecrire le code de la classe **AgenceLocation** caractérisée par son nomAgence et tabVoiture qui est un ArrayList et qui mémorise toutes les voitures de l'agence.

Les méthodes à définir sont :

- **constructeur()** : initialise l'attribut nomAgence par une valeur passée en paramètre. (1 Point)
- **ajouterVoiture(...)** : ajoute une voiture donnée à tabVoiture. (1 Point)
- **louerVoiture(...)** : retourne la première voiture qui a un prix inférieur ou égal à celui donné en paramètre tout en vérifiant qu'elle est disponible mais si aucune voiture ne répond à ces deux critères la méthode lève et propage l'exception **PasVoitureDispoException**. Le code de la classe **PasVoitureDispoException** est à implémenter (2 Points)
- **rendreVoiture(...)** : permet de changer l'état de disponibilité d'une voiture bien déterminée tout en vérifiant qu'elle existe déjà dans le tabVoiture. (1.5 Points)
- **lesVoituresDisponibles()** : retourne un ArrayList qui contient toutes les voitures disponibles. (1.5 Points)

Partie n°2 : (5 Points)

. Écrire le code de la classe Test réduite à une méthode main qui permet de :

- Créer une agence de location nommée EliteCar et ajouter les deux voitures suivantes :
 - v1 (1, "Ferrari", 600, disponible, 100, "300km/h")
 - V2 (2, "Jaguar ", 500, indisponible, 100)
 (1.5 Point)
- Louer une voiture dont le prix ne dépasse pas 500 dinars. On doit traiter l'exception **PasVoitureDispoException** par un simple affichage de l'exception. (1.5 Point)
- Afficher alors toutes les voitures disponibles (2 Point).

Bon travail