

TP3 Héritage simple

1 Exercice 1

1.1 Voiture

On déclare la classe Voiture avec les attributs suivants :

```
public String marque;  
public int nbchevaux;  
public String[] options;
```

Compléter cette classe avec :

- un constructeur qui prend en paramètre la marque de la voiture, un tableau de String représentant les options, et un entier donnant le nombre de chevaux. Le constructeur devra créer un nouveau tableau pour les options, et recopier le tableau correspondant donné en argument.
- une méthode **affiche** qui affiche la voiture dans la console, en annonçant d'abord le marque et le nombre de chevaux, puis sur une deuxième ligne on trouvera toutes les options (attention, on veut toutes les options sur la même ligne, séparée par un espace).
- une méthode **demarrer** qui affiche dans la console "**Demarrage OK**".

1.2 Voiture de course

Il s'agit maintenant de créer la classe **VoitureCourse** qui hérite de Voiture et qui a un attribut supplémentaire, qui sera vrai si la voiture de course a un aileron, et faux sinon.

```
public boolean aileron;
```

Compléter cette classe avec :

- un constructeur qui prend en argument la marque de la voiture, un tableau de **String** représentant les options, et un booléen indiquant si la voiture dispose d'un aileron. Le nombre de chevaux sera automatiquement réglé à 250.

- une méthode **affiche** qui affiche la description pour une voiture normale suivi de "**Je suis une voiture de course avec aileron**" ou "**Je suis une voiture de course sans aileron**".
- une méthode **demarrer** qui affiche dans la console le message de démarrage pour une voiture normale suivi du bruit de la voiture de course : "**Vrouuum**".

Attention, si l'on désire changer le message d'affichage ou de démarrage pour une voiture normale, il faut que la classe **VoitureCourse** puisse récupérer ces nouveaux messages sans faire de modifications.

1.3 Concession

On crée la classe **Concession** avec les attributs suivants :

```
Voiture[] stock;
```

Compléter cette classe avec :

- un constructeur qui prend comme argument une Voiture et qui crée un stock constitué seulement de cette voiture.
- une méthode **ajoutVoiture** qui prend comme argument une Voiture, et l'ajoute au stock. Attention, il faudra créer un nouveau tableau ayant une case de plus, et recopier l'ancien.
- une méthode **affiche** qui affiche les voitures du stock, avec pour chaque voiture son numéro dans le stock.
- une méthode **rechercheOptions** qui prend en argument le nom d'une option, affiche dans la console toutes les voitures du stock disposant de cette option.

Vous testerez votre programme sur le main suivant :

```
public static void main(String[] args){
String[] op1={"ABS", "GPS"};
String[] op2={"ABS", "autoradio", "jantes alu"};
String[] op3={"GPS", "sieges cuir"};
Voiture v1=new Voiture("Audi", 120, op1);
Voiture v2=new Voiture("Peugeot", 80, op2);
VoitureCourse v3=new VoitureCourse("BMW", op3, true);
v1.demarrer();
v3.demarrer();
Concession c=new Concession(v1);
c.ajoutVoiture(v2);
c.ajoutVoiture(v3);
c.affiche();
c.rechercheOptions("GPS");
}
```

Il doit s'afficher dans la console :

```
Démarrage OK
Démarrage OK
Vrouuuuum
Le stock:
Voiture num 0
Marque: Audi CV: 120
ABS GPS
Voiture num 1
Marque: Peugeot CV: 80
ABS autoradio jantes alu
Voiture num 2
Marque: BMW CV: 250
GPS sieges cuir
Je suis une voiture de course avec aileron
Voitures en stock avec l'option GPS
Marque: Audi CV: 120
ABS GPS
Marque: BMW CV: 250
GPS sieges cuir
Je suis une voiture de course avec aileron
```

2 Exercice 2

2.1 Classe Personne

Créer une classe publique, nommée `Personne`, contenant 3 champs :

- un champ `nom` de type `String`,
- un champ `prenom` de type `String`,
- un champ `age` de type primitif `int`.
- Créer un constructeur initialisant tous les champs, de signature `Personne(String leNom, String leprenom, int lage)` initialisant les champs de la classe.
- Créer un constructeur de copie, de signature `Personne(Personne p)` qui initialise les champs de la classe à ceux de `p`.
- Créer la méthode `afficher()` qui affiche à l'écran le nom, le prenom et l'âge au format décrit ci-après :
Nom :
Prenom :
Age :

2.2 Classes Prof et étudiants (Héritage)

- On désire gérer des personnes de type prof qui vont avoir un attribut supplémentaire, la matière qu'elles enseignent. La méthode `afficher` sera redéfinie,

car elle doit permettre d'afficher en plus de tous les renseignements sur un prof, la matière qu'il enseigne.

- Créez une classe `Etudiant` qui doit hériter de `personne`, mais qui comporte comme attributs supplémentaires la section, le nombre de jour d'absence. La méthode `afficher` doit être redéfinie pour permettre l'affichage des deux attributs supplémentaires.

Tester les méthodes des différentes classes dans une classe Principale (main).