

TD3 : Le patron de conception Décorateur

On souhaite gérer des voitures dont le prix dépend de différentes options.

I UML

Créer un diagramme des classes en faisant apparaître les attributs et les opérations, ces dernières comprenant les accesseurs en lecture sur les attributs.

- 1° Une voiture est caractérisée par un nom de modèle (*String*), une masse (*float*) et un prix (*int*). Représenter la classe abstraite *Voiture* avec ces caractéristiques. La masse sera donnée par une méthode *getMasse()*, tandis que le prix sera calculé par une méthode abstraite *prix()*. On ajoutera une méthode *toString()* décrivant le modèle et le prix de la voiture.
- 2° Compléter le diagramme précédent avec une classe *Ferrari* et une classe *Porsche*, héritant de *Voiture* (une Ferrari coûte 300 000 euros, une Porsche 150 000 euros).
- 3° On souhaite maintenant munir les voitures d'un moteur. Une voiture motorisée a les caractéristiques suivantes :
 - sa masse et son prix sont augmentées de la masse et du prix du moteur (on modifiera la méthode *getMasse()* pour qu'elle renvoie la masse totale de la voiture motorisée) ;
 - une variable décrivant la force du moteur (*float*) ;
 - une méthode *float acceleration()* qui renvoie l'accélération de la voiture motorisée définie par le rapport entre la force du moteur et la masse totale de la voiture motorisée ;
 - une méthode *float distanceParcourue(float t)* qui renvoie la distance parcourue par la voiture motorisée en t secondes donnée par l'expression $0,5*a*t*t$ où a est l'accélération.Ajouter au diagramme une classe abstraite *Motorisation* modélisant ces caractéristiques.
- 4° Compléter le diagramme avec une classe concrète *Essence*, héritant de *Motorisation*. Le prix d'un moteur à essence sera de 1000 euros. Cette classe redéfinit aussi la méthode *toString()* pour renvoyer la description de la voiture et du type de motorisation ainsi que son prix total ;
- 5° Créer de la même façon une classe *Diesel* dont la motorisation coûte 1500 euros.
- 6° Compléter le diagramme avec une classe abstraite *Freinage* modélisant le type de freinage d'une voiture. On négligera la masse du système de freinage. Ajouter deux réalisations concrètes *Abs* (valeur 2000 euros) et *FreinageMagnetique* (valeur 1500 euros). Redéfinir de manière adéquate la méthode *toString()*.
- 7° Compléter le diagramme avec une classe *Course* associée à la classe *Motorisation*, qui permet de simuler une course de voitures motorisées. Représenter les opérations *inscrireVoitureMotorisee(Motorisation m)* et *simulerCourse(float t)* qui affiche la distance parcourue au temps t par chaque voiture inscrite dans la course.

II Java

Écrire chaque classe du diagramme UML et tester chaque méthode. Créer une classe *SimulationVoitures* effectuant les opérations suivantes :

- créer une Ferrari avec moteur à essence et un ABS,
- créer une Porsche hybride (un moteur essence et un moteur diesel) et un freinage magnétique,
- créer et simuler une course entre les deux voitures précédentes.