UFR ST - Besançon- L2 Info - Année 2015/16

Programmation par Objets

TP 7 - Facturation de véhicules dans un ferry

Objectif : Héritage, classes abstraites, polymorphisme, ArrayList

On s'intéresse dans ce problème à un ferry, qui transporte des véhicules de trois types : des vélos, des motos et des voitures. Les véhicules embarqués sont facturés selon le principe décrit ci-dessous.

Chaque type de véhicule possède un tarif de base. Il est de 5 € pour un vélo, de 20 € pour une moto et de 60 € pour une voiture. Mais enregistrer un véhicule coûte de plus en plus cher au fur et à mesure que le ferry se remplit, selon le principe que « plus on réserve tard, plus c'est cher ». Pour cela, chaque ajout d'un véhicule (quelque soit son type) dans le ferry augmente de 5% le tarif des véhicules.

Par exemple, si c'est un vélo qui s'enregistre en premier, il est facturé 5 €. Puis si une voiture s'enregistre ensuite, elle est facturée 63 €, qui correspondent à 60 (tarif de base d'une voiture), plus 3 (*une* fois 5% de 60). Puis si ensuite c'est à nouveau une voiture qui s'enregistre, elle est facturée 66 €, correspondant à 60 plus *deux* fois 5% de 60. Puis si c'est un vélo qui s'enregistre ensuite, il est facturé 5,75 €, c'est à dire 5 € de base plus *trois* fois 5% de 5 €. Puis si c'est ensuite une moto qui s'enregistre, elle est facturée 24 €, soit 20 € (tarif de base d'une moto) plus *quatre* fois 5% de 20 €, et ainsi de suite...

La compagnie qui gère le ferry vous demande de coder une application qui lui permette :
- d'enregistrer des véhicules un à un, en indiquant à chaque fois le montant facturé ;
- de faire le total de la recette (la somme des montants facturés) une fois que tous les véhicules sont enregistrés ;
- de connaître le nombre de véhicules de chaque type (vélo, moto ou voiture) qui se sont enregistrés.

Ce problème est à résoudre en codant une classe abstraite Vehicule, puis en codant trois sous-classes concrètes de Vehicule : Velo, Moto et Voiture. Le ferry sera lui codé par une classe Ferry qui gère une collection de véhicules (une *ArrayList* de véhicules).

Question 1. Codez en Java une classe abstraite Vehicule.

Indications. Une possibilité pour gérer l'augmentation des tarifs au fur et à mesure de l'enregistrement de nouveaux véhicules, consiste à compter au niveau de cette classe abstraite le nombre d'instances de Vehicule. À la première instance créée, l'augmentation est de 0 fois 5% ; à la deuxième, elle est de 1 fois 5% ; à la troisième, de 2 fois 5% ; etc. Une méthode nommée getAugmentation() permet alors de renvoyer cette augmentation. D'autre part, la méthode qui donne le *tarif* d'un véhicule est nécessairement abstraite, car elle dépend d'un tarif de base qui varie dans chaque sous-classe concrète.

Question 2. Codez les classes Velo, Moto et Voiture comme des sous-classes concrètes de Vehicule.

Indication. Chacune de ces classes dispose d'un attribut constant qui indique le tarif de base du véhicule, et d'une méthode qui retourne le tarif réel du véhicule : c'est le tarif de base majoré de l'augmentation indiquée par la méthode getAugmentation() de la question précédente.

Nous allons maintenant coder la classe Ferry. Ces attributs sont une collection (de type *ArrayList*) de Vehicule), ainsi que le nombre de véhicules enregistrés. Elle dispose
- d'une méthode qui permet d'enregistrer un véhicule (en l'ajoutant dans la collection), et qui retourne un réel correspondant au tarif appliqué pour cet enregistrement ;
- d'une méthode qui permet de calculer la recette (en parcourant la collection et en sommant tous les tarifs des véhicules enregistrés) ;
- de trois méthodes qui indiquent respectivement : le nombre de vélos enregistrés, le nombre de motos, le nombre de voitures.

Question 3. Codez la classe Ferry.

Question 4. Le programme principal par exemple
- enregistre successivement dans le ferry : un vélo, puis une voiture, puis une moto, puis une voiture, puis un vélo ;
- calcule et affiche le montant de la recette ;
- calcule et affiche le nombre de véhicules de chaque type.