

TD4 : L'héritage
(classes et méthodes abstraites)

Exercice 1/

On développe pour un zoo une application de réalité augmentée. Lorsque qu'un animal passe près d'une des caméras qui filment l'ensemble des enclos, son image est entourée sur un écran et son espèce ainsi que les informations qui y sont rattachées y sont affichées.

Question 1/ Créer une classe *Animal* puis les classes *Âne*, *Chouette* et *Lion* qui en héritent. Chaque animal sera identifié par un numéro

Pourquoi faut-il que la classe *Animal* soit abstraite ?

Question 2/ Destinée à des enfants, cette application affiche le type de cri que pousse l'animal, ainsi la chouette hulule, le lion rugit et l'âne brait. Au lieu de stocker cette information à l'aide d'une variable d'instance, on décide de définir une méthode d'instance abstraite `void cri()` au niveau de la classe *Animal* qui affiche sur la console le cri de chaque bête. Vous spécifierez par la suite son comportement dans chacune des sous-classes concrètes.

Exercice 2/

On désire modéliser différentes catégories de sociétés :

- les commerces,
- les commerces subventionnés,
- les sociétés d'assurance.

Un commerce se caractérise par un nom, un total des ventes, un total des dépenses au titre des salaires, un total des dépenses pour le matériel. Un commerce subventionné se caractérise par un nom, un total des ventes, un total des dépenses au titre des salaires, un total des dépenses pour le matériel et un total des subventions. Une société d'assurance se caractérise par un nom, un total des remboursement et un total des versements.

On souhaite également calculer les bénéfices de chaque société. Le bénéfice se calcule de la façon suivante :

- pour un commerce : ventes – salaires – matériel,
- pour un commerce subventionné : ventes – salaires – matériel + subventions,
- pour une société d'assurance : versement – remboursement.

L'affichage d'une société est également prévu. L'affichage écrit à l'écran les caractéristiques de la société (selon sa catégorie) et également ses bénéfices.

Question 1/ Proposez un arbre d'héritage pour décrire cette modélisation. Pour chaque classe vous indiquerez :

- si elle est abstraite ou concrète,
- les variables de classes (en indiquant qu'il s'agit de variables de classe),
- les variables d'instances (en indiquant qu'il s'agit de variables d'instance),
- les entêtes des méthodes abstraites (en indiquant qu'il s'agit de méthodes abstraites),
- les entêtes des méthodes d'instance (en indiquant qu'il s'agit de méthodes d'instance),
- les entêtes des méthodes de classe (en indiquant qu'il s'agit de méthodes de classe).

Attention, certaines de ces rubriques peuvent être vides (exemple : classe sans variables de classe).

Question 2 Codez en langage Java votre proposition.

Exercice 3/

On s'intéresse ici à définir des droites (en dimension 2) selon trois représentations :

- une équation cartésienne : $y = a x + b$,
- deux points du plan, (x_1, y_1) et (x_2, y_2) ,
- un point du plan (x, y) et un vecteur (u, v) .

On désire tester le parallélisme entre droites et l'égalité entre droites.

Quelques rappels :

Ces trois représentations sont équivalentes. Pour comparer deux droites, on doit pouvoir calculer le coefficient a et le coefficient b :

ils sont déjà définis pour une droite définie par $y=ax+b$,

pour une droite définie par deux points (x_1, y_1) et (x_2, y_2) , il faut utiliser :

$$a = (y_1 - y_2) / (x_1 - x_2)$$

$$b = y_1 - a * x_1$$

pour une droite définie par un point du plan (x, y) et un vecteur (u, v)

$$a = v/u$$

$$b = y - a * x$$

Deux droites sont parallèles si elles ont la même valeur pour a , elles sont égales si elles ont les mêmes valeurs pour a et b .

Question Proposez une solution reposant sur la classe abstraite `Droite`, chacune des représentations faisant l'objet d'une sous-classe.
