Attributs de classe vs d'instance, références d'objets

Exercice 1 *Références, énumérations, premiers attributs de classe*

On se base sur la classe Chat vue au TD précédent. On souhaite affiner la description du pelage des chats. Le pelage des chats sera décrit par :

- ses couleurs,
- le fait d'être multicolore,
- le fait d'être à poils longs,
- une description textuelle

Les couleurs possibles pour les pelages des chats sont les suivantes : noir, bleu (usuellement on dit gris), chocolat, lilas (beige rosé), cannelle, fauve, roux, blanc. Le pelage d'un chat peut être multicolore (si et seulement si le nombre de couleurs est strictement plus grand que 1).

Question 1. Proposez une classe PelageChat en UML.

Question 2. Modifiez la modélisation de la classe Chat pour prendre en compte cette nouvelle classe représentant le pelage.

Question 3. Reprenez votre diagramme d'objets pour Azraël. Ajoutez à votre diagramme un jumeau d'Azraël nommé Azbis. Vous envisagerez 2 solutions pour les pelages d'Azraël et Azbis.

Question 4. Dans un nouveau répertoire/projet exo2, placez :

- une nouvelle classe PelageChat.java que vous écrirez en vous inspirant de Chat.java
- une version modifiée de Chat.java.
- une version modifiée de ManipulationsChats.java, avec Azraël et Azbis. Vous ferez ici en sorte que le pelage d'Azraël et le pelage d'Azbis soient 2 objets distincts, que vous comparerez avec un == qui devra vous répondre faux bien que les 2 objets portent les mêmes valeurs.

Question 5. On veut rajouter à la classe Chat les informations suivantes :

- les chats sont carnivores
- Les chats ont une espérance de vie de 15 ans
- Les chats peuvent ronronner.

Ces informations vous paraissent-elles de même nature que celles déjà placées dans la classe Chat ? Proposez-en une modélisation.

Exercice 2 *Classe Compteur : attributs d'instances et attributs de classe*

On souhaite mettre en place une classe Compteur. Un compteur a une valeur, qui est un entier initialisé à 0. Un compteur connaît également le nombre de compteurs déjà créés, également initialisé à 0.

Question 1. Modélisez en UML la classe Compteur.

Question 2. Complétez le code Java partiel de la classe Compteur qui vous est donné avec la déclaration et l'initialisation des attributs.

Question 3. Exécutez le code de la classe ManipulationCompteurs (après compilation). Notez que les deux attributs se comportent différemment l'un de l'autre, et comprenez bien pourquoi.

Exercice 3 *Vecteurs de points*

Lors du TD précédent, on a créé une classe Point. Nous allons maintenant mettre en place une classe Vecteur.

Question 1. Dans un nouveau répertoire/projet, recopiez la classe Point (le fichier Point.java) et mettez en place une classe Vecteur vide.

Question 2. Un vecteur est ici défini par un point donnant la coordonnée en x et la coordonnée en y du vecteur, par rapport à l'origine. Ainsi, le vecteur de coordonnées $\binom{a}{b}$ peut être représenté par l'origine d'une part, et le point de coordonnées(a, b). Le vecteur $\overrightarrow{AB}$ avec A de coordonnées (xa, ya) et B de coordonnées (xb, yb) est le même que le vecteur $\overrightarrow{OC}$ avec O l'origine et C le point de coordonnées $(xb-xa, yb-ya)$. Ajoutez à la classe Vecteur un attribut matérialisant le point représentant le vecteur, et deux constructeurs : un constructeur prenant en paramètre les deux coordonnées du vecteur, et un constructeur prenant en paramètre 2 points.

Question 3. Ajoutez à la classe Vecteur les méthodes suivantes :
- méthode `sommeAvec` qui calcule la somme du vecteur receveur et du vecteur paramètre, et retourne le (nouveau) vecteur somme
- méthode `ajout` qui ajoute au vecteur receveur le vecteur pris en paramètre
- méthode de classe `somme` qui somme deux vecteurs pris en paramètre et retourne le (nouveau) vecteur somme [1].

Question 4. On donne les instuctions suivantes :

Programme 1 : *Instructions de manipulation de vecteurs*

```
Vecteur v1=new Vecteur(1, 2);
Vecteur v2=new Vecteur(2, 1);
Vecteur v3=new Vecteur(3, 3);
Vecteur v4=v1.sommeAvec(v2);
Vecteur v5=v3;
```

1. Combien d'objets ont été créés par ces instructions ?
2. A t-on v5==v3 ?
3. A t-on v3==v4 ?

1. On note au passage qu'il aurait été sympathique de pouvoir utiliser l'opérateur + pour sommer 2 vecteurs, mais la surcharge d'opérateur, qui est réalisable dans beaucoup de langages, ne l'est pas en Java

---

Exercice 4 *(Exercice supplémentaire de révision) Gestion des stages*

Question 1. Proposez une modélisation UML simple pour la gestion des stages décrits ci-après. Chaque stage a un sujet (court texte) et un employeur (l'employeur est une entreprise décrite simplement ici par son nom et son adresse). Un stage peut référencer un stagiaire (si le stage a été pourvu), un stagiaire est un utilisateur du système modélisé, décrit simplement ici par son nom, son prénom et son adresse électronique. Chaque stage référence le responsable des stages qui est aussi un utilisateur du système modélisé. Un stage a une thématique principale choisie parmi : web, IA, Données, algo, recherche.

Question 2. Proposez des implémentations Java pour les classes modélisées à la question précédente. En vous inspirant des exemples vus aux exercices précédents, placez des constructeurs et dans chaque classe une méthode permettant d'en représenter des instances sous forme de chaîne de caractères, et ajoutez une classe ManipulationStages où vous placerez de quoi créer quelques utilisateurs, quelques entreprises et quelques stages, et les afficher.