

Classes et objets

Exercice 1 *Classe et instanciation, premier programme java*

Soit la classe Chat dont un modèle de classe est donné en UML à la figure 1.

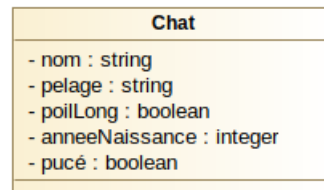


FIGURE 1 – Diagramme de classes pour la classe Chat

Question 1. Modélisez sous forme de diagramme d'objets 2 instances de la classe Chat. L'une d'entre elle représentera le chat nommé "Azraël", de pelage "roux et blanc", qui n'a pas les poils longs, né en 2017 et non pucé, et l'autre un autre chat nommé "Grosminet" avec les valeurs de votre choix.

Code 1 – Classe Chat en Java, première version

```
public class Chat {
    private String nom;
    private String pelage;
    private boolean poilLong;
    private int anneeNaissance;
    private boolean pucé;

    public Chat(String nom, String pelage, boolean poilLong, int
    ↪ anneeNaissance, boolean pucé) {
        this.nom = nom;
        this.pelage = pelage;
        this.poilLong = poilLong;
        this.anneeNaissance = anneeNaissance;
        this.pucé = pucé;
    }

    public String toString() {
        return "Chat [nom=" + nom + ", pelage=" + pelage + ", poilLong=" +
        ↪ poilLong + ", anneeNaissance="
            + anneeNaissance + ", pucé=" + pucé + "];"
    }
}
```

Question 2. Une traduction en Java de la classe Chat vous est donnée au listing 1. En plus des attributs présents dans le diagramme de classes, on a ajouté : un constructeur paramétré, et une méthode permettant d'obtenir pour toute instance de la classe Chat une chaîne de caractère la représentant. On vous donne également la classe ManipulationsChats, qui se compose simplement d'une méthode principale vide.

- a- Dans la méthode principale, créez 2 instances de la classe Chat, correspondant à Azraël et Grosminet. Affichez ensuite ces instances.
- b- Compilez ces 2 classes : dans le répertoire exo1, `javac *.java`. Vous obtenez alors 2 fichiers suffixés par `".class"`.
- c- Exécutez la méthode principale : dans le répertoire exo1, `java ManipulationsChats`.

Exercice 2 *Attributs de type non simples, énumérations*

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 `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 3 *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.

Exercice 4 *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.