

Classes et objets

Exercice 1 *Classe et instantiation, premier programme java*

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

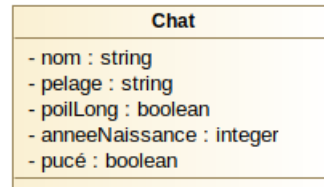


FIGURE 1 – Diagramme de classes pour la classe Chat

Question 1. Modélisez sous forme de diagramme d'objets deux 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`.

A partir de maintenant, vous pouvez utiliser un IDE (Integrated Development Environment). Nous vous suggérons d'utiliser IntelliJ IDEA. Vous pourrez également l'installer chez vous gratuitement. Vous trouverez sur moodle quelques éléments de base sur la prise en main d'IntelliJ IDEA. Vous pouvez utiliser un autre IDE (eclipse par exemple, ou visual studio code avec les plug ins adéquats) ou pas d'IDE du tout si vous le préférez.

Exercice 2 *Classe Point*

On s'intéresse ici à des points dans un repère orthonormé de dimension 2. Un point est représenté par son abscisse (coordonnée `x`) et son ordonnée (coordonnée `y`). L'origine du repère est le point de coordonnées `(0,0)`.

Question 1. Implémentez une classe `Point` que vous munirez des éléments suivants :

- des attributs de type `double` pour l'abscisse et l'ordonnée
- un constructeur paramétré permettant de positionner l'abscisse et l'ordonnée d'un point grâce aux valeurs des paramètres
- un constructeur non paramétré qui construit le point origine
- une méthode `toString` qui retourne une chaîne de caractères représentant le point receveur, sous la forme : `Point(x,y)` où `x` et `y` sont respectivement l'abscisse et l'ordonnée du receveur.
- une méthode `module` qui calcule le module du point receveur, c'est à dire sa distance à l'origine
- une méthode `distanceDe` qui calcule la distance du point receveur à un autre point pris en paramètre
- une méthode `symetrique` qui retourne le point symétrique du point receveur, par symétrie centrale par rapport à l'origine.

Question 2. Implémentez une méthode `main` que vous placerez dans la classe `ManipulationsPoint` qui réalise les traitements suivants :

- création de deux instances de la classe `Point` : une pour l'origine, et une pour le point `(3,4)`, que l'on placera dans des variables nommées respectivement `o` et `p1`.
- appel de la méthode `module` pour ces deux instances et affichage d'un message de type : `le module de (0,0) est 0 et le module de (3,4) est 5.`
- appel de la méthode `symetrique` sur `p1`, stockage de la référence d'objet obtenue dans une variable nommée `p2`, appel de la méthode `distanceDe` sur `p1` en passant `p2` en paramètre, vérification que le résultat obtenu est 10.

Exercice 3 *Se familiariser avec quelques éléments de Java et notamment la classique erreur `NullPointerException`*

On s'intéresse à une classe `SyntheseMeteo` qui synthétise les informations météorologiques mensuelles d'une ville : son nom est dans l'attribut `ville`, et ses 12 températures minimales (respectivement maximales) stockées dans le tableau `tMin` (respectivement `tMax`).

Question 1. Exécutez le `main` de la classe `ManipulationsSyntheseMeteo`. Quelle erreur obtenez-vous ? Pourquoi ? Corrigez la classe `SyntheseMeteo` de manière à ne plus avoir cette erreur.

Question 2. Regardez la méthode `temperatureMensuelleMax`. Le premier "tour" de la boucle `for` est inutile. Transformez la boucle `for` (qui est ici un *pour tout*) en une boucle `for` avec un indice de boucle `i` allant de 1 à la taille du tableau exclue.

Question 3. Ajoutez une méthode `moisLePlusChaud` qui retourne le mois le plus chaud sous forme d'un entier (1 pour janvier, 2 pour février, 12 pour décembre).

Question 4. Modifiez la méthode précédente pour retourner un littéral d'une énumération `Mois` plutôt que le numéro du mois. Vous créerez une énumération `Mois` dans un fichier `Mois.java`. Pour convertir un numéro de mois en un littéral de l'énumération `Mois`, vous utiliserez la méthode `numMoisVersMois` qui est commentée dans la classe `SyntheseMeteo`.

Question 5. Dans la méthode `main` de `ManipulationsSyntheseMeteo`, ajoutez en dernière ligne l'instruction `lille.setTmin(13, 10);` et exécutez de nouveau le `main`. Analysez l'erreur obtenue.

Question 6. Modifiez la méthode `setTmin` afin que :

- il n'y ait pas d'accès au tableau si le numéro de mois passé en paramètre n'est pas compris entre 1 et 12.
- la valeur ne soit pas reportée dans le tableau si elle est aberrante (inférieure à -100 ou supérieure à 100)

Question 7. Ajoutez une autre méthode `setTmin` qui prend en paramètre un littéral d'énumération `Mois` plutôt qu'un numéro de mois. Pour convertir un `Mois` en un numéro de mois, on utilisera la méthode `moisVersNumMois` qui est commentée dans la classe `SyntheseMeteo`.