

Java DUT 1 Feuille TP4
Université Paris-Est Marne-la-Vallée

Exercice 1.—

On souhaite modéliser un zoo composé d'animaux herbivore ou carnivores.

- a) Créer une classe **Animal** qui représente un animal avec un champs booléen **carnivore** qui indique si l'animal est carnivore ou non et un champs **weight** de type **double** qui correspond au poids de l'animal en kg. Écrire le constructeur permettant d'initialiser les deux champs.
- b) Écrire une méthode **toString** pour afficher un **Animal**.
- c) Écrire une méthode **main** dans une classe **AnimalTest**. Créer un **Animal** carnivore et un autre herbivore.
- d) Écrire une classe **Zoo** contenant des animaux. On utilisera un champs **zoo** de type **ArrayList<Animal>** pour stocker les animaux.
- e) Écrire une méthode **add** qui permet d'ajouter un **Animal** dans le **Zoo**. Vérifier que la méthode n'ajoute pas une référence null dans le **Zoo**.
- f) Écrire une méthode **toString** pour pouvoir afficher tout le **Zoo**.
- g) Écrire une méthode **numberOfCarnivore** dans **Zoo** qui renvoie le nombre d'animaux carnivores du **Zoo**.
- h) Sachant qu'un carnivore à besoin du quart de son poids en viande par semaine, écrire une méthode **getMeatForAWeek** dans la classe **Animal** qui renvoie la quantité de viande nécessaire pour le nourrir. Pour un herbivore, cette quantité sera nulle.
- i) Écrire une méthode **getMeatForAWeek** dans **Zoo** qui renvoie la quantité de viande nécessaire pour nourrir les animaux du **Zoo** pour une semaine.

Exercice 2.—

Pour attirer des visiteurs, les Zoos ont des animaux stars qui, contrairement aux animaux normaux, ont un nom de type **String**.

- a) Écrire une classe **StarAnimal** qui dérive de **Animal** avec un constructeur.
- b) Les carnivores qui sont des **StarAnimal** ont un régime alimentaire nécessitant la moitié de leur poids en viande. Que doit ajouter ou modifier et dans quelles classes pour que la méthode **getMeatForAWeek** de **Zoo** fonctionne correctement ?
- c) Créer un petit **Zoo** avec au moins un **StarAnimal** et tester **getMeatForAWeek**.
- g) Écrire une méthode **getStarAnimals** dans **Zoo** qui renvoie une liste des animaux stars. Le type de retour sera **List<StarAnimal>**.

Exercice 3.—

- a) Écrire une méthode `equals` dans `StarAnimal` permettant de tester de dire que deux `StarAnimal` sont égaux s'ils ont le même nom.
- b) Redéfinissez la méthode `hashCode` dans `StarAnimal` pour que deux `StarAnimal` égaux aient le même `hashCode`.
- c) Écrire une méthode `setOfStarAnimals` dans `Zoo` qui renvoie un ensemble des animaux stars. Le type de retour sera `Set<StarAnimal>`. Les `Set` en Java sont comme les ensembles de Python. On complètera le code ci-dessous.

```
public Set<StarAnimal> setOfStarAnimals() {  
    Set<StarAnimal> set = new HashSet<StarAnimal>();  
    ....  
}
```

- d) Testez `setOfStarAnimals` dans la classe `Test`. Qu'observe-t-on si la méthode `hashCode` n'est pas redéfinie dans `StarAnimal` ?