

Examen TP de Programmation Orientée Objet : Groupe 2

Durée : 1h30

Vous êtes en charge de la gestion d'un zoo. Ce zoo possède plusieurs types d'animaux, des cages où ces animaux sont placés et un système pour gérer les animaux, leur alimentation, ainsi que leur comportement dans le zoo.

Interface ComportementAnimal (1 pt)

Créez une interface ComportementAnimal qui définit les comportements communs à tous les animaux. Cette interface doit comporter les méthodes suivantes :

- Manger() : chaque animal doit avoir une méthode pour manger.
- SeDeplacer() : Cette méthode représente l'action de déplacement de l'animal. Le mode de déplacement est différent d'un animal à l'autre. Chaque animal devra implémenter cette méthode pour décrire son mode de déplacement.

Classe Animal : Classe non instanciable (4 pts)

Créez une classe Animal qui représente un animal générique dans le zoo. Cette classe implémente l'interface ComportementAnimal. Cette classe doit être une classe de base pour tous les types d'animaux du zoo. Chaque animal est identifié par un nom (String), type (String) : "Carnivore", "Herbivore" ou "Omnivore". Domestique (booléen) : true si l'animal est domestique et false si l'animal est sauvage.

Méthodes :

- Un constructeur pour initialiser les attributs nom, type et Domestique.
- toString() : renvoie une chaîne décrivant un animal.
- Les méthodes abstraites : Manger() et SeDeplacer() qui seront redéfinies dans les sous classes spécifiques. Ajouter dans le code de la classe Animal les signatures des deux méthodes comme suit : *public abstract void Manger() ; public abstract void SeDeplacer() ;*

Classe Lion (4 pts)

Créez une classe Lion. Cette classe représente un animal spécifique. La classe a un attribut supplémentaire : habitat (Lions des savanes africaines, Lions des forêts tropicales ...). Un Lion est un animal sauvage et carnivore. La classe doit redéfinir les méthodes Manger() et SeDeplacer() de manière spécifique au lion.

Méthodes :

- Un constructeur pour initialiser tous les attributs y compris l'habitat. Un Lion est un animal carnivore et sauvage.
- Manger() : Affiche le message *"nom chasse et mange une proie !" .*

- SeDeplacer() : Affiche le message *"nom marche dans la savane. Lorsqu'il chasse, il court rapidement pour attraper sa proie !"*
- toString()...

Classe Oiseau (4pts)

Créez la classe oiseau. Cette classe représente l'animal oiseau avec un attribut supplémentaire PeutVoler (booléen), true si l'animal peut voler, false sinon. Les oiseaux peuvent être domestique ou sauvage en fonction de leurs races. La classe redéfinit les méthodes abstraites de la classe Animal.

Méthodes :

- Un constructeur pour initialiser tous les attributs y compris l'attribut domestique et peutVoler . Un oiseau est un animal omnivore.
- Manger() : Affiche le message *" L'oiseau est omnivore. Il peut manger des graines, des fruits, des insectes et parfois de petits animaux."* .
- SeDeplacer() : Si l'oiseau peut voler, la méthode affichera un message *"nom vole dans le ciel"*. Si l'oiseau ne peut pas voler (comme une autruche), la méthode affiche *"nom marche sur le sol."*.
- toString()...

Classe Cage (4pts)

Créez une classe Cage qui représente une cage dans le zoo.

Une cage est décrite par nomCage (String),capaciteMaximale (int) : la capacité maximale d'animaux dans cette cage et TabAnimaux : un tableau polymorphe d'animaux dans la cage avec une taille égale capaciteMaximale.

Remarque : Ajoutez à la classe Animal un nouvel attribut cage de type Cage qui représentera la cage dans laquelle l'animal est placé. Cet attribut permettra de savoir dans quelle cage se trouve l'animal à un moment donné.

Méthodes de la classe cage :

- Void AjouterAnimal(Animal animal) : permet d'ajouter un animal dans la cage si la capacité maximale n'est pas atteinte . Si la cage a de la place disponible, l'animal est ajouté et l'attribut Cage de l'animal est mis à jour pour l'associer à la cage. Si la capacité maximale est atteinte, un message d'erreur est affiché *"La cage ... est pleine. Impossible d'ajouter ..."*.
- Void RetirerDeLaCage(Animal animal) : Cette méthode permet de retirer l'animal animal s'il se trouve de sa cage actuelle, en supprimant l'animal du tableau TabAnimaux .
- String toString() : Pour retourner une chaîne décrivant une cage, la chaîne contient les informations détaillées de la cage, telle que son numéro, sa capacité et la liste des animaux qu'elle contient.

Classe ZOO : (3pts)

Créez la classe ZOO. La classe contient la méthode main. Créez les cage et les animaux suivants :

Création des cages

- Créer une cage pour les carnivores (capacité maximale de 2 animaux). Cette cage pourra accueillir des lions. Nom de la cage : Cage Carnivores, Capacité maximale : 2 animaux.
- Créer une cage pour les oiseaux (capacité maximale de 3 animaux). Cette cage pourra accueillir des oiseaux comme un perroquet et une autruche. Nom de la cage : Cage Oiseaux. Capacité maximale : 3 animaux.

Création des animaux

Créer deux Lions et deux oiseaux avec des attributs différents.

- Lion 1 : Nom : Simba, Habitat : Savane Africaine
- Lion 2 : Nom : Mufasa, Habitat : Forêts tropicales
- Oiseau 1 : Nom : Perroquet, Peut voler : true, Domestique
- Oiseau 2 : Nom : Autruche, Peut voler : false, Sauvage

Ajouter les animaux dans leurs cages

- Simba et Mufasa doivent être ajoutés à la cage des carnivores.
- Le perroquet et l'autruche sont ajoutés à la cage des oiseaux.
- Appeler les méthode manger() et SeDeplacer () pour tous les animaux.

Tester les erreurs

- Ajouter un troisième lion dans la cage Carnivores (capacité maximale atteinte) :
 - Créez un nouveau lion appelé "Scar" et tentez de l'ajouter à la cage des carnivores.

Affichage des états des animaux et des cages

- Afficher les animaux dans la cage Carnivores et Oiseaux.