
Programmation Orientée Objet

TP3 : Classes abstraites et Interfaces

2022-2023

Exercice 1 :

Dans la continuité de l'exercice précédent, nous désirons adapter le programme pour transformer la classe « **Animal** » vers une classes abstraite.

Par la suite, toutes méthodes de la classe Animal doivent aussi être des méthodes abstraites.

- De ce fait, il faut redéfinir toutes les méthodes abstraites de Animal dans les classes filles.
- Ajoutez aussi une 3ème classes nommé « **Chien** ». Dans cette classe le diagnostique de l'âge et le poids sont respectivement : âge>10(vieux), poids>20(obèse).
- La méthode Diagnostique n'aura plus besoin d'utiliser l'opérateur **instanceof**.
- Définissez une interface « **Pet** » qui sera implémenter dans les classes Chat et Chien. Cette interface possède deux méthodes :
 - **CanBePet()** qui affiche que l'animal peut être un animal de compagnie.
 - **LifeSpan()** qui calcul l'espérance de vie d'un animal de compagnie qui a passé sa vie en captivité. Le calcul ajoute un pourcentage entre 30 à 40% à l'âge moyen de l'animal vivant dans la nature (par exemple : si un chat peut vivre jusqu'à 8 ans dans la nature, on lui ajoutera entre 30 à 40% s'il vie en captivité).
- Définissez une autre interface « **MakeSound** » contenant une méthode **Cri()** affichant à l'écran le cri de l'animal. Décidez-vous même où est ce qu'on peut implémenter cette interface.
- Maintenant essayer d'instancier une classe abstraite et une interface, est-ce que cela est possible ?

Exercice 2 :

Dans cet exercice on va reprendre l'exercice 3 de la série de TP2. On veut transformer la classe **Véhicule** vers une classe abstraite ainsi que ses méthodes calculPrix(), Afficher() et equals(). Bien-sûr il faut adapter ces méthodes dans les sous classes. Ajoutez après une autre classe appelé «**Hélicoptère**» comme une sous classe de Véhicule qui possède Heure_de_Vol comme attribut.

- Définissez une interface « **Flight** » qui sera implémenter dans les deux classes Avion et Hélicoptère. Cette interface possède deux méthodes :
 - o **CanFly()**, qui affiche que l'appareil peut voler et,
 - o **LongDistanceFlight()** qui teste si l'appareil peut voler pour des longues distances si ses heures de vol sont inférieur à 1000h.
- Définissez une autre interface appelé **Driving**, qui déclare une méthode **RoadDriving()** qui affiche que le véhicule a la possibilité de se déplacer sur des routes. Cette interface sera implémentée par la classe Voiture.
- Tester le programme en créant plusieurs objets et en faisant appel à aux différentes méthodes.
- Après essayez de déclarer un attribut dans l'une des interfaces, qu'est-ce que vous remarquez ?