

Cartouche du document

Année : ING 1
Matière : UML
Activité : Travail dirigé

Objectifs

Il s'agit dans ce travail dirigé d'introduire la conception orientée objet à travers

- le concept d'interface pour privilégier la programmation générique à la programmation spécifique
- la réutilisation : Héritage ou Composition
- une première approche des design patterns : méthode de fabrique

Sommaire des exercices

- 1 - Réutilisation : Confusion entre héritage sémantique et héritage fonctionnel
- 2 - Réutilisation : La bonne utilisation de l'héritage
- 3 - Interface : Passer de la programmation spécifique à la programmation générique

Corps des exercices

1 - Réutilisation : Confusion entre héritage sémantique et héritage fonctionnel

Énoncé :

On essaie dans cet exercice de montrer que l'héritage n'est pas toujours une bonne solution pour la réutilisation. Dans le package `java.util` la classe `Stack` a été implémentée en dérivant la classe `Vector`.

Question 1)

Énoncé de la question

Expliquer en quoi le code suivant :

```
Stack pile = new Stack();
pile.push("Bas de la pile");
pile.push("Haut de la pile");
pile.insertElementAt("Perdu", 0);
while (!pile.empty()) {
System.out.println(pile.pop());
}
```

viole le fonctionnement d'une pile.

Question 2)

Énoncé de la question

En quoi a-t-on fait un héritage fonctionnel et non pas un héritage sémantique.

Question 3)

Énoncé de la question

Pourquoi ne peut-on pas dire qu'une pile **est un** vecteur ?

Question 4)

Énoncé de la question

Proposer une implémentation de la classe Pile en utilisant la classe Vector par composition ?

2 - Réutilisation : La bonne utilisation de l'héritage

Énoncé :

On montre dans cet exercice que l'héritage est bon choix pour la réutilisation quand la relation entre de la forme **ObjetFille est un (ou est une sorte de) ObjetMere**. On suppose que la classe Vehicule a été implémentée. Définition : un véhicule est un moyen qui permet à un ou plusieurs être humains de se déplacer d'un endroit A à un endroit B.

Question 1)

Énoncé de la question

Quel sont les liens sémantiques entre un véhicule et

- une voiture, une moto, un vélo, un cheval, un éléphant, un train, un avion, un bateau et un radeau
- un chauffeur

Question 2)

Énoncé de la question

On suppose que dans la classe Vehicule on a implémenté l'opération **void deplaceToi()**. On constate un peu plus tard que le déplacement dépend du vent et que la signature devient **void deplaceToi(int vent)**.

Que faut-il recoder si

- 1) si on a réutilisé par héritage la classe Vehicule pour les classes Voiture, Moto, Velo, Cheval, Elephant, Train, Avion, Bateau et Radeau.
- 2) si on a réutilisé par composition la classe Vehicule pour les classes Voiture, Moto, Velo, Cheval, Elephant, Train, Avion, Bateau et Radeau.

Question 3)

Énoncé de la question

Qu'est ce qui nous assure dans ce cas que l'héritage est une bonne solution ?

Question 4)

Énoncé de la question

Reprenons le lien sémantique **un chauffeur utilise un véhicule**. Cela a pour conséquence qu'un véhicule peut ne pas avoir de chauffeur. On peut effectivement envisager qu'un véhicule se déplace automatiquement sans chauffeur.

Expliquer pourquoi la composition est un meilleur choix.

3 - Interface : Passer de la programmation spécifique à la programmation générique

Énoncé :

Dans le projet Mini Système Expert, on a le diagramme de classe suivant :

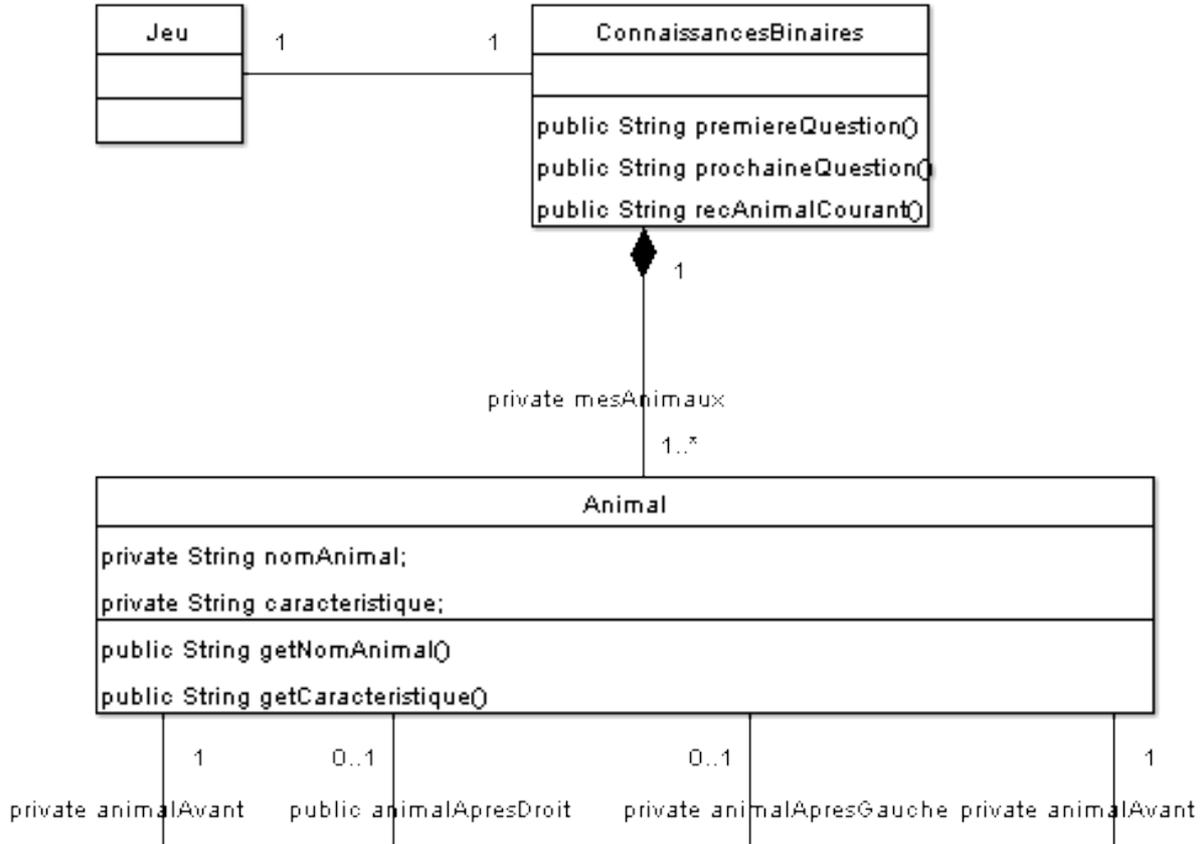


Diagramme de classes spécifique : Jeu, Connaissances et animaux

Question 1)

Énoncé de la question

Ce diagramme de classes est spécifique à certain mode de structuration des connaissances. On vous demande de définir un autre diagramme qui permet de découpler la classe `Jeu` et la classe `ConnaissancesBinaires` afin d'envisager un autre mode de structuration des connaissances.

Question 2)

Énoncé de la question

Que faut-il ajouter dans le diagramme de classes pour que l'on puisse créer l'objet de connaissances sans que le client `Jeu` connaisse le type de cet objet ?