

TP Visitor : Créez votre propre zoo virtuel

Contexte

Bienvenue au Zoo de JavaVille ! En tant qu'ingénieur logiciel, vous avez été chargé de développer un système pour suivre les différentes espèces d'animaux dans le zoo, ainsi que leurs comportements et caractéristiques. Afin de rendre le système extensible et maintenable, vous avez décidé d'utiliser le pattern Visitor.

Objectif

Votre mission est de modéliser un zoo avec plusieurs types d'animaux. Chaque animal acceptera un visiteur qui effectuera différentes opérations sur lui, comme l'affichage de ses informations, l'évaluation de son besoin en nourriture, et le calcul de l'espace requis dans le zoo.

Spécifications

1. Classes d'Animaux

- Créez une hiérarchie de classes pour les animaux du zoo. Par exemple, vous pourriez avoir une classe `Animal` abstraite ou une interface avec des classes concrètes comme `Lion`, `Elephant`, et `Monkey`.
- Chaque classe doit implémenter une méthode `accept` qui prend un visiteur en argument.

2. Interface Visitor

- Définissez une interface `Visitor` avec des méthodes spécifiques pour chaque type d'animal. Par exemple, `visitLion(Lion lion)`, `visitElephant(Elephant elephant)`, etc.

3. Implémentations de Visitor

- Implémentez plusieurs classes concrètes de visiteurs, par exemple :
- `DisplayVisitor` : affiche des informations sur chaque animal.
- `FoodRequirementVisitor` : calcule la quantité de nourriture nécessaire pour chaque animal.
- `SpaceRequirementVisitor` : détermine l'espace requis pour chaque animal dans le zoo.

4. Interaction avec le système

- Créez une classe `Zoo` qui contient une collection d'animaux.
- Ajoutez une méthode pour faire traverser un visiteur à travers tous les animaux du zoo, en appliquant les opérations du visiteur à chaque animal.

Exercice Supplémentaire

- Ajoutez un nouvel animal ou un nouveau visiteur. Comment cela affecte-t-il votre code ?
- Pensez à un moyen d'exporter les informations des animaux dans un format de fichier (par exemple, JSON ou XML) en utilisant un autre visiteur.