

Exercice Java : Gestion d'un Zoo Virtuel

Cet exercice met en œuvre tous les concepts clés de la Programmation Orientée Objet (POO) en Java, ainsi que l'utilisation des interfaces, des classes anonymes et d'un menu interactif via une console.

Concepts abordés

1. **Encapsulation, abstraction, polymorphisme, héritage** : Bases de la POO.
2. **Interfaces et classes anonymes** : Création de comportements dynamiques et flexibles.
3. **Gestion modulaire** : Menu interactif dans un package séparé.

Scénario

Vous devez créer un **système de gestion de zoo virtuel** qui permet :

- D'ajouter des animaux à différents enclos.
- De lister les animaux présents dans le zoo ou par enclos.
- De nourrir les animaux.

Spécifications

Classes et Interfaces

1. **Animal (classe abstraite)** :

- Champs : `id` (int), `name` (String), `species` (String).
- Méthodes : `String getDetails()`, `void eat()` (abstraite).

2. **Mammal, Bird (sous-classes)** :

- Héritent de `Animal` et implémentent leurs comportements spécifiques.

3. **Enclosure (classe)** :

- Champs : `id` (int), `name` (String), `List<Animal>`.
- Méthodes : Ajouter, supprimer, lister les animaux.

4. **ZooManager (classe principale)** :

- Gère les enclos et les interactions avec le zoo.

5. **ConsoleMenu (package séparé)** :

- Fournit une interface console pour interagir avec le système.

6. **ZooAction (interface)** :

- Utilisée pour définir des actions dynamiques (par exemple, nourrir tous les animaux).

Exemple de sortie

```
=== Menu Zoo ===  
1. Ajouter un animal  
2. Lister les animaux  
3. Déplacer un animal  
4. Nourrir les animaux  
5. Quitter  
Choix : 1  
Entrez le nom de l'animal : Simba  
Entrez l'espèce : Lion  
Choix de l'enclos : 1
```

```
=== Menu Zoo ===  
1. Ajouter un animal  
2. Lister les animaux  
3. Déplacer un animal  
4. Nourrir les animaux  
5. Quitter  
Choix : 2  
Enclos Savane : [Simba (Lion)]
```