

## Travaux pratique du chapitre 2 : Patterns de construction d'objets (créateur)

### Singleton

#### Contexte

Vous êtes engagé(e) comme développeur(euse) pour construire un système de gestion pour un zoo virtuel. La direction de ce zoo souhaite avoir une application qui suit l'état général du zoo et qui permet de prendre des décisions en temps réel. Pour cela, le système devra avoir un point d'entrée unique qui centralise toutes les informations globales à propos du zoo, tel qu'un contrôleur principal. Ce contrôleur permettra de garantir que les informations sont cohérentes et à jour.

#### Mission

Votre mission est de concevoir et d'implémenter un pattern Singleton pour créer un contrôleur principal qui gère l'accès aux informations centralisées du zoo. Vous devez garantir qu'il n'existe qu'une seule instance de ce contrôleur à tout moment pour éviter des conflits de données et assurer la bonne gestion des ressources du zoo.

#### Exigence

##### 1. Création du Contrôleur du Zoo :

- Implémenter une classe `ZooController` en utilisant le pattern Singleton.
- Assurez-vous que cette classe ne peut être instanciée qu'une seule fois.

##### 2. Fonctionnalités à Implémenter :

- Affichage de l'État du Zoo : Le contrôleur doit pouvoir afficher les statistiques globales du zoo, telles que le nombre total d'animaux, les visiteurs actuels, etc.
- Mise à Jour des Informations : Ajouter des fonctions pour mettre à jour ces informations de manière centralisée.

##### 3. Démonstration d'Utilisation :

- Implanter un petit script qui démontre l'utilisation de l'instance unique du `ZooController`.
- Montrez qu'il est impossible de créer plusieurs instances de `ZooController`.

##### 4. Langage :

- Le projet doit être codé en Java. Vous êtes encouragé(e) à ajouter des commentaires expliquant chaque étape de votre implémentation.