

## Factory

### Contexte

Vous avez été embauché par une organisation secrète spécialisée dans la gestion et l'étude des créatures mythiques. Vos nouvelles responsabilités incluent la conception et le développement d'une application qui gère différentes créatures mythiques en utilisant le pattern Factory.

Pour cette tâche, vous devez créer un système qui peut générer différentes créatures mythiques sur la base de demandes spécifiques. Chaque créature a des caractéristiques propres, et de nouvelles créatures peuvent être ajoutées au système sans modification majeure du code existant.

### Objectif

Implémenter une application en utilisant le pattern Factory pour créer et gérer des créatures mythiques.

### Exigence

#### 1. Classe de Base

- Créez une classe `Creature` qui servira de classe de base pour toutes les créatures mythiques. Cette classe doit avoir une méthode abstraite ou virtuelle `get_description()` qui sera implémentée par les classes dérivées.

#### 2. Classes de Créatures

- Implémentez au moins trois classes de créatures différentes : `Dragon`, `Phoenix`, `Griffon`.
- Chaque classe doit hériter de la classe `Creature` et implémenter la méthode `get_description()`, qui retourne une description unique de la créature.

#### 3. Classe Factory

- Créez une classe `CreatureFactory` qui a une méthode `create_creature(type_creature : str)` pour générer une instance d'une créature en fonction du type spécifié.
- La méthode `create_creature` doit retourner un objet de la classe appropriée (`Dragon`, `Phoenix`, `Griffon`) en fonction du type demandé.

#### 4. Interface Utilisateur

- Créez une simple interface utilisateur (console ou graphique) où l'utilisateur peut entrer le type de créature qu'il souhaite créer.
- Affichez la description de la créature générée.

#### 5. Extensions

- Ajoutez la capacité d'intégrer de nouvelles créatures en tant que nouvelles classes sans modifier la structure de votre `CreatureFactory`.
- Documentez comment vous ajouteriez une nouvelle créature, par exemple un `Cerberus`.

#### 6. Langage

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