



# DESIGN PATTERN

TP Les créateurs

## TABLE DES MATIERES

ExerciceS – Design Pattern Factory .....	2
Exercice 1.....	2
Exercice 1.1.....	2
Exercice 1.2.....	2
Exercice 1.3.....	2
Exercice 2.....	2
Exercices – Design Pattern Singleton .....	2
Exercice 3.....	2
Exercice 4.....	3
Exercices – Design Pattern Builder .....	3
Exercice 5.....	3
Exercice 5.1.....	3
Exercice 5.2.....	3
Exercice 5.3.....	3
Exercice 5.4.....	3
Exercices – Design Pattern Abstract Factory .....	3
Exercice 6.....	3

## EXERCICES – DESIGN PATTERN FACTORY

### EXERCICE 1

Le but est de créer une application contenant les différents types d'animaux et de les instancier à partir du Design Pattern Factory.

Pour cela, vous devrez prendre en compte 3 types d'animaux possibles :

- Le chat,
- Le chien
- Le cheval

Chaque animal doit avoir un cri particulier.

---

#### EXERCICE 1.1

Lister les différentes classes / interfaces nécessaires pour la réalisation de ce pattern

---

#### EXERCICE 1.2

Créer le diagramme de classe nécessaire

---

#### EXERCICE 1.3

Implémenter le diagramme de classe dans le langage de programmation de votre choix

### EXERCICE 2

Définissez une interface Vehicule avec des méthodes comme start() et stop().

Créez des classes pour différents types de véhicules (voiture, moto, vélo) qui implémentent cette interface.

Concevez une fabrique qui crée des instances de véhicules, en prenant en compte des caractéristiques spécifiques à chaque type de véhicule.

## EXERCICES – DESIGN PATTERN SINGLETON

### EXERCICE 3

Créer une variable statique à l'intérieure d'une classe contenant une instance d'elle-même

Assurez vous que la classe ne peut être instanciée depuis l'extérieure

Utilisez cette classe.

#### EXERCICE 4

Reprendre l'exercice 1 et transformer votre Factory en un Singleton sans perdre l'intérêt de la Factory

### EXERCICES – DESIGN PATTERN BUILDER

#### EXERCICE 5

Soit un objet de type Maison.

Cet objet peut être composé de différents éléments.

---

##### EXERCICE 5.1

Décomposer les différents éléments et les lister

---

##### EXERCICE 5.2

Créer un diagramme de classes UML pour définir la structure de vos classes selon le Pattern Builder

---

##### EXERCICE 5.3

Ecrire le code en programmation orientée objet

---

##### EXERCICE 5.4

Intégrer la possibilité de créer des immeubles à partir de ce pattern

### EXERCICES – DESIGN PATTERN ABSTRACT FACTORY

#### EXERCICE 6

Créez une interface `AbstractFactory` avec des méthodes pour créer différents produits liés (par exemple, `createButton()` et `createCheckbox()`).

Implémentez des classes concrètes de fabriques abstraites pour différents thèmes d'interface utilisateur (par exemple, `LightThemeFactory` et `DarkThemeFactory`), chacune produisant un ensemble cohérent de produits.