

## Exercice : Système de Navigation de Véhicule avec le Pattern Strategy

### Contexte

Vous êtes chargé de développer un système de navigation pour un véhicule. Ce système doit offrir plusieurs stratégies de navigation, telles que la navigation routière, la navigation hors-route, et la navigation économique.

### Objectif de l'Exercice

Utiliser le design pattern Strategy pour permettre au système de navigation de changer dynamiquement son mode de navigation en fonction des préférences de l'utilisateur ou des conditions de conduite.

### Tâches

#### 1. Définir l'Interface Strategy :

- Créez une interface `NavigationStrategy` avec une méthode `navigate(String destination)`.

#### 2. Implémenter les Stratégies Concrètes :

- Développez des classes concrètes telles que `RoadStrategy`, `OffroadStrategy`, et `EconomicStrategy`, chacune implémentant `NavigationStrategy`. Chaque stratégie doit avoir une logique de navigation unique.

#### 3. Créer la Classe Contexte :

- Développez une classe `NavigationSystem` qui utilise une référence à `NavigationStrategy`. Cette classe doit permettre de changer la stratégie de navigation en cours d'utilisation.

#### 4. Tester le Système :

- Créez une instance de `NavigationSystem` et testez-la avec différentes stratégies. Par exemple, changez de `RoadStrategy` à `OffroadStrategy` en fonction de scénarios spécifiques.