

Énoncé Builder : Création d'un système de construction de véhicules modulaires

Contexte :

Vous travaillez pour une entreprise automobile fictive, **ModularCars**, qui propose un service de personnalisation complète de véhicules. Les clients peuvent choisir plusieurs éléments de leur voiture tels que :

- Le modèle (SUV, Berline, Sportive, etc.)
- Le type de moteur (Électrique, Hybride, Essence)
- La couleur
- Le nombre de sièges
- Les options supplémentaires (GPS, Toit ouvrant, etc.)

Vous devez concevoir une application qui permet de construire ces véhicules de manière modulaire en fonction des choix des clients. L'application doit aussi vérifier certaines **contraintes commerciales** pour assurer la cohérence des voitures créées.

Contraintes :

1. Modèles disponibles :

- SUV, Berline, Sport.

2. Types de moteur disponibles :

- Électrique, Hybride, Essence.

3. Validation des moteurs :

- Un modèle Sport ne peut pas avoir de moteur électrique (uniquement Hybride ou Essence).

4. Validation des sièges :

- Un modèle SUV doit obligatoirement avoir au moins 5 sièges.

5. Attributs obligatoires :

- Les voitures doivent avoir obligatoirement un modèle, un moteur et un nombre de sièges définis avant leur construction.

6. Options supplémentaires :

- Les clients peuvent ajouter des extras à leur voiture comme le GPS, le Toit ouvrant, ou d'autres équipements à choisir librement.

Spécifications techniques :

1. Vous devez utiliser le pattern **Builder** pour construire l'objet Car étape par étape.
2. La classe Car doit être immuable une fois construite.

3. Le **Builder** doit permettre de définir les attributs `model`, `engine`, `color`, `seats` et d'ajouter une liste d'extras (via `addExtra()`).
4. La méthode `build()` du **Builder** doit appliquer les **validations des contraintes** (par exemple, interdire un moteur électrique sur un modèle `Sport`).
5. Ajoutez une méthode `displayCar()` à la classe `Car` pour afficher les détails de la voiture construite.
6. Utilisez des valeurs par défaut pour les attributs non obligatoires (ex. `color = "undefined"`).

Exemple de scénario :

1. Construction réussie d'un SUV avec des extras :

- Modèle : `SUV`
- Moteur : `Hybride`
- Sièges : 7
- Couleur : `Bleu`
- Extras : `GPS`, `Toit ouvrant`

2. Erreur : Modèle Sport avec un moteur électrique.

3. Erreur : SUV avec moins de 5 sièges.