

# Projet Java avancé

Auteur : Mouelo Loan

## Description du concept :

L'idée de concept est une entreprise de location de véhicule (des voitures, des avions et des bateaux) qu'elle loue à des particuliers. L'entreprise possède un directeur, un manager et un commercial.

Le particulier peut louer un véhicule si celui-ci est disponible, le véhicule se voit donc attribuer le nom de la personne qui l'a loué.

On aura donc une classe parente « vehicle » que l'on étend aux différents modèles de véhicule.

Il y aura également une classe parente « person » pour gérer tous les types de personnes.

La finalité est de pouvoir avoir la liste de chaque véhicule avec toutes ses informations ainsi que la possibilité d'en récupérer 1 seul et également avoir la liste des particuliers.

```
public class main {  
  
    public static void main(String[] args) {  
        CarBusinessImpl carBusiness = new CarBusinessImpl();  
        Optional<Car> OneCar = carBusiness.get("AA-000-AA");  
        List<model.Car> AllCar = carBusiness.getList();  
        /**  
        BoatBusinessImpl boatBusiness = new BoatBusinessImpl();  
        Optional<Boat> OneBoat = boatBusiness.get("UYRGDHDH");  
        List<model.Boat> AllBoat = boatBusiness.getList();  
  
        AvionBusinessImpl avionBusiness = new AvionBusinessImpl();  
        Optional<Avion> OneAvion = avionBusiness.get("ZIEBIZENE");  
        List<model.Avion> AllAvion = avionBusiness.getList();  
  
        RentalCarBusinessImpl rentalCarBusiness = new RentalCarBusinessImpl();  
        Optional<RentalCar> OneRentalCar = rentalCarBusiness.get("Michel");  
        List<model.RentalCar> AllRentalCar = rentalCarBusiness.getList();  
        */  
        System.out.println(OneCar);  
        System.out.println(AllCar);  
        /**System.out.println(OneBoat);  
        System.out.println(AllBoat);  
        System.out.println(OneAvion);  
        System.out.println(AllAvion);  
        System.out.println(OneRentalCar);  
        System.out.println(AllRentalCar);  
        */  
    }  
}
```

Pour l'exemple suivant je récupère l'implémentation du business Car auquel je lui demande d'aller me récupérer la voiture immatriculée « AA-000-AA » puis la liste de toutes les voitures.

Voici le résultat obtenu dans la console grâce au DataMock mis en place pour les voitures :

```
Optional[Car [marques = BMW, registration = AA-000-AA, rentalTo = null, color = Black, NbPassenger = 4, RentalPeriod = 2, Availablity = YES, Price = 1500]]
```

```
[Car [marques = BMW, registration = AA-000-AA, rentalTo = null, color = Black, NbPassenger = 4, RentalPeriod = 2, Availablity = YES, Price = 1500], Car [marques = AUDI, registration = BB-111-BB, rentalTo = null, color = Red, NbPassenger = 2, RentalPeriod = 2, Availablity = NO, Price = 1000], Car [marques = MERCEDES, registration = CC-222-CC, rentalTo = null, color = Green, NbPassenger = 2, RentalPeriod = 1, Availablity = YES, Price = 2000], Car [marques = BMW, registration = DD-444-DD, rentalTo = null, color = Black, NbPassenger = 7, RentalPeriod = 2, Availablity = YES, Price = 2500]]
```

Le modèle défini pour une voiture est le suivant :

```
@Override
public String toString() {
    return "Car [marques = " + marques + ", registration = " + registration + ", rentalTo = " + rentalTo + ", color = " + getColor()
        + ", NbPassenger = " + getNbPassenger() + ", RentalPeriod = "
        + getRentalPeriod() + ", Availablity = " + getAvailablity() + ", Price = "
        + getPrice() + "];"
}
```

Les champs en bleu tels que « marques » ou « registration » sont spécifique à la classe « Car » tandis que les fonctions appelées en jaune font référence à la classe parente « Vehicle » via l'héritage mis en place.

La seule difficulté majeur à laquelle je me suis confronté est la mise en place du package GSON et son utilisation