

## 1 Exercices

- ```
public void remove(Car car){
    cars.remove(car);
}
```
- ```
public void add(Car car){
    if(!cars.contains(car)) {
        cars.add(car);
    }
}
```
- ```
public List<Car> selectByBrand(String brand){
    List<Car> selectedCars = new ArrayList<>();
    for(int i=0; i<cars.size(); i++){
        Car car = cars.get(i);
        if(car.brand().equals(brand)){
            selectedCars.add(car);
        }
    }
    return selectedCars;
}
```
- ```
public List<Car> selectByMaxPrice(double maxPrice){
    List<Car> selectedCars = new ArrayList<>();
    for(Car car : cars){
        if(car.dailyRentalPrice()<=maxPrice){
            selectedCars.add(car);
        }
    }
    return selectedCars;
}
```

- Les deux méthodes ont un code quasi-identique. Les seules différences sont sur le paramètre et le code pour le test du `if`. On peut imaginer de nombreux critères : voitures datant de moins d'un certains nombre d'années, d'un certain modèle, coûtant au moins un certain prix, ...
- La partie différente entre les deux méthodes est le test à l'intérieur du `if`. On va donc créer une interface nous permettant de tester un critère sur une voiture. On a donc besoin d'une unique méthode ayant un paramètre de type `Car` et retournant un `boolean`.

```
public interface Criterion {
    boolean test(Car car);
}
```

- Si on souhaite tester un critère que n'importe quel type d'objet, il nous faut créer une interface générique. Cette interface existe en Java et s'appelle `Predicate<T>`. On peut définir l'interface avec le code suivnat :

```
public interface Predicate<T> {  
    boolean test(T t);  
}
```

8. `public class MaxPriceCriterion implements Predicate<Car> {`

```
    private final double maxPrice;
```

```
    public MaxPriceCriterion(double maxPrice) {  
        this.maxPrice = maxPrice;  
    }
```

```
    @Override
```

```
    public boolean test(Car car) {
```

```
        return car.dailyRentalPrice() <= maxPrice;
```

```
    }
```

```
}
```

9. `public List<Car> select(Predicate<Car> criterion){`

```
    List<Car> selectedCars = new ArrayList<>();
```

```
    for (Car c: cars){
```

```
        if (criterion.test(c)){
```

```
            selectedCars.add(c);
```

```
        }
```

```
    }
```

```
    return selectedCars;
```

```
}
```