Interfaces:

Electric

+ **chargeBattery():** *void*

Method to charge the vehicle's battery

Combustion

+ **refuel():** *void*

Method to refuel the vehicle

Rentable

+ **rentOut(gallery:** *Gallery***):** *Rentable*

Method to rent out the vehicle

+ **returnVehicle(gallery:** *Gallery***):** *void*

Method to return the vehicle

Diesel extending ***Combustion*** and ***Rentable***

+ **cleanDieselFilter():** *void*

Method to clean the diesel filter

Vehicle abstract class

- **model**: *String*

The model of the vehicle

- **year**: *int*

The year of the vehicle

+ **Vehicle**(**model**: *String*, **year**: *int*)

Constructor to initialize the vehicle

+ **getModel():** *String*

Getter for the model

+ **getYear():** *int*

Getter for the year

+ **startEngine():** *void*

Abstract method to start the vehicle's engine

Aircraft abstract class, extending the **Vehicle** class

+ **Aircraft**(**model**: *String*, **year**: *int*)

Constructor to initialize the Aircraft

+ **fly()**: *void*

Abstract method to define flying behavior

Ship abstract class, extending the **Vehicle** class

+ **Ship**(**model**: *String*, **year**: *int*)

Constructor to initialize the Ship

+ **sail()**: *void*

Abstract method to define sailing behavior

Car abstract class, extending the **Vehicle** class and implementing ***Comparable*** with *Car*

- **horsepower**: *int*

The horsepower of the car

+ **Car**(**model**: *String*, **year**: *int*, **horsepower**: *int*)

Constructor to initialize the Car with horsepower

+ **getHorsepower()**: *int*

Getter for the horsepower

+ **compareTo**(**other**: *Car*): *int*

Method to compare cars based on horsepower

Class definition for **Tesla**, extending **Car** and implementing ***Electric*** and ***Rentable*** interfaces

+ **Tesla**(**model**: *String*, **year**: *int*, **horsepower**: *int*)

Constructor to initialize the Tesla

Class definition for **Ford**, extending **Car** and implementing ***Combustion*** interface

+ **Ford**(**model**: *String*, **year**: *int*, **horsepower**: *int*)

Constructor to initialize the Ford

Class definition for **Mercedes**, extending **Car** and implementing ***Electric*** and ***Diesel*** interfaces

+ **Mercedes**(**model**: *String*, **year**: *int*, **horsepower**: *int*)

Constructor to initialize the Mercedes

Class definition for **Gallery** to manage a collection of cars

- **combustionCars**: *ArrayList<Combustion>*

List to store combustion engine cars

- **electricCars**: *ArrayList<Electric>*

List to store electric cars

+ **Gallery()**

Constructor to initialize the Gallery

+ **addCar**(**car**: *Car*): *void*

Method to add a car to the appropriate list

+ **addCombustionCar**(**car**: *Combustion*): *void*

Method to add a combustion engine car to the list

+ **addElectricCar**(**car**: *Electric*): *void*

Method to add an electric car to the list

+ **displayRentableCars**(): *void*

Method to display rentable cars sorted by model