**Car Rental System**

**📁 Structure:**

* Vehicle (Abstract class)
* Car (extends Vehicle)
* Customer
* Booking
* CarRentalSystem (Main class)

💻 **Java Code:**

import java.util.\*;

// Abstract class

abstract class Vehicle {

protected String plateNumber;

protected String model;

protected double rentPerDay;

protected boolean isAvailable = true;

public Vehicle(String plateNumber, String model, double rentPerDay) {

this.plateNumber = plateNumber;

this.model = model;

this.rentPerDay = rentPerDay;

}

public String getPlateNumber() {

return plateNumber;

}

public boolean isAvailable() {

return isAvailable;

}

public void setAvailable(boolean status) {

this.isAvailable = status;

}

public abstract void showInfo();

}

// Concrete class

class Car extends Vehicle {

private int seats;

public Car(String plateNumber, String model, double rentPerDay, int seats) {

super(plateNumber, model, rentPerDay);

this.seats = seats;

}

@Override

public void showInfo() {

System.out.println("Car: " + model + " | Plate: " + plateNumber + " | Rent/day: $" + rentPerDay + " | Seats: " + seats + " | Available: " + isAvailable);

}

}

class Customer {

private String name;

private String licenseNumber;

public Customer(String name, String licenseNumber) {

this.name = name;

this.licenseNumber = licenseNumber;

}

public String getName() {

return name;

}

}

class Booking {

private Customer customer;

private Vehicle vehicle;

private int days;

public Booking(Customer customer, Vehicle vehicle, int days) {

this.customer = customer;

this.vehicle = vehicle;

this.days = days;

}

public void showBooking() {

System.out.println(customer.getName() + " booked " + vehicle.getPlateNumber() + " for " + days + " days. Total: $" + (vehicle.rentPerDay \* days));

}

}

public class CarRentalSystem {

private static List<Vehicle> vehicles = new ArrayList<>();

private static List<Booking> bookings = new ArrayList<>();

private static Scanner scanner = new Scanner(System.in);

public static void main(String[] args) {

addSampleCars();

while (true) {

System.out.println("\n--- Car Rental System ---");

System.out.println("1. Show Available Cars");

System.out.println("2. Book a Car");

System.out.println("3. Show All Bookings");

System.out.println("4. Exit");

System.out.print("Choose an option: ");

int choice = scanner.nextInt();

scanner.nextLine(); // consume newline

switch (choice) {

case 1:

showAvailableCars();

break;

case 2:

bookCar();

break;

case 3:

showAllBookings();

break;

case 4:

System.out.println("Thanks for using the system!");

return;

default:

System.out.println("Invalid choice!");

}

}

}

private static void addSampleCars() {

vehicles.add(new Car("ABC-123", "Toyota Corolla", 50.0, 5));

vehicles.add(new Car("XYZ-789", "Honda Civic", 60.0, 5));

vehicles.add(new Car("LMN-456", "BMW X5", 120.0, 7));

}

private static void showAvailableCars() {

System.out.println("\nAvailable Cars:");

for (Vehicle v : vehicles) {

if (v.isAvailable()) {

v.showInfo();

}

}

}

private static void bookCar() {

System.out.print("\nEnter your name: ");

String name = scanner.nextLine();

System.out.print("Enter your license number: ");

String license = scanner.nextLine();

Customer customer = new Customer(name, license);

System.out.print("Enter plate number of car to book: ");

String plate = scanner.nextLine();

Vehicle vehicleToBook = null;

for (Vehicle v : vehicles) {

if (v.getPlateNumber().equalsIgnoreCase(plate) && v.isAvailable()) {

vehicleToBook = v;

break;

}

}

if (vehicleToBook != null) {

System.out.print("Enter number of days: ");

int days = scanner.nextInt();

scanner.nextLine(); // consume newline

vehicleToBook.setAvailable(false);

Booking booking = new Booking(customer, vehicleToBook, days);

bookings.add(booking);

System.out.println("Booking confirmed!");

} else {

System.out.println("Car not found or already booked.");

}

}

private static void showAllBookings() {

System.out.println("\nBookings:");

for (Booking b : bookings) {

b.showBooking();

}

}

}