// Define the Vehicle class

class Vehicle {

// Attributes

String licensePlate;

String make;

String model;

int year;

Status status;

// Enum for vehicle status

enum Status { Available, Reserved, Rented, Maintenance, OutOfService }

// Lazy initialization for licensePlate and status

lazy String licensePlate = "";

lazy Status status = Status.Available;

// Abstract method

abstract String getInfo();

// Custom method to set license plate (avoid naming conflict with auto-generated setter)

void updateLicensePlate(String plate) {

licensePlate = plate.toUpperCase();

}

}

// Define the Car class (inherits from Vehicle)

class Car extends Vehicle {

int numSeats;

// Override getInfo()

@Override

String getInfo() {

return super.getInfo() + "\tSeats: " + numSeats;

}

}

// Define the Motorcycle class (inherits from Vehicle)

class Motorcycle extends Vehicle {

boolean hasSidecar;

// Override getInfo()

@Override

String getInfo() {

return super.getInfo() + "\tHas Sidecar: " + hasSidecar;

}

}

// Define the Truck class (inherits from Vehicle)

class Truck extends Vehicle {

double cargoCapacity;

// Override getInfo()

@Override

String getInfo() {

return super.getInfo() + "\tCargo Capacity: " + cargoCapacity;

}

}

// Define the SportCar class (inherits from Car, final)

final class SportCar extends Car {

int horsepower;

boolean hasTurbo;

// Override getInfo()

@Override

String getInfo() {

return super.getInfo() + "\tHorsepower: " + horsepower + "\tHas Turbo: " + hasTurbo;

}

}

// Define the Customer class

class Customer {

String customerId;

String name;

String phoneNumber;

// toString() method

@Override

String toString() {

return "Customer ID: " + customerId + "\tName: " + name + "\tPhone: " + phoneNumber;

}

}

// Define the RentalRecord class

class RentalRecord {

Vehicle vehicle;

Customer customer;

LocalDate recordDate;

double totalAmount;

String recordType;

// toString() method

@Override

String toString() {

return "Vehicle: " + vehicle.getLicensePlate() + "\tCustomer: " + customer.getName() +

"\tDate: " + recordDate + "\tAmount: " + totalAmount + "\tType: " + recordType;

}

}

// Define the RentalHistory class

class RentalHistory {

List<RentalRecord> rentalRecords;

// Methods

void addRecord(RentalRecord record) {

rentalRecords.add(record);

}

List<RentalRecord> getRentalHistory() {

return rentalRecords;

}

List<RentalRecord> getRentalRecordsByCustomer(String customerName) {

List<RentalRecord> result = new ArrayList<>();

for (RentalRecord record : rentalRecords) {

if (record.getCustomer().getName().equals(customerName)) {

result.add(record);

}

}

return result;

}

List<RentalRecord> getRentalRecordsByVehicle(String licensePlate) {

List<RentalRecord> result = new ArrayList<>();

for (RentalRecord record : rentalRecords) {

if (record.getVehicle().getLicensePlate().equals(licensePlate)) {

result.add(record);

}

}

return result;

}

}

// Define the RentalSystem class

class RentalSystem {

List<Vehicle> vehicles;

RentalHistory rentalHistory;

// Methods

boolean rentVehicle(Vehicle vehicle, Customer customer, LocalDate rentalDate, double rentalAmount) {

if (vehicle.getStatus() == Vehicle.Status.AVAILABLE) {

vehicle.setStatus(Vehicle.Status.RENTED);

RentalRecord record = new RentalRecord(vehicle, customer, rentalDate, rentalAmount, "RENT");

rentalHistory.addRecord(record);

return true;

}

return false;

}

boolean returnVehicle(Vehicle vehicle, Customer customer, LocalDate returnDate, double extraFees) {

if (vehicle.getStatus() == Vehicle.Status.RENTED) {

vehicle.setStatus(Vehicle.Status.AVAILABLE);

RentalRecord record = new RentalRecord(vehicle, customer, returnDate, extraFees, "RETURN");

rentalHistory.addRecord(record);

return true;

}

return false;

}

}