Java Developer - Java Challenge

Objective

Evaluate your OO analysis and modelling skills, developing skills, how you structure your code and design an API.
You should not need to spend more than 4 hours in the task, if the solution provided is incomplete, explain which parts you decided to focus on and why they are relevant.

Keep the objective in mind, feel free to use any tool, libraries, or frameworks, add a README.md file, adding about the decisions you made, what you focused on and what you did not focus and why, as well as how to run and use the program.

When you are ready

Upload the code to GitHub and send an email with the link to jose.segura@globalpay.com

Remember to add all external files if it required, for instance DML, Notes, … And don't forget to add a README.md file.

The problem - Car rental system

For a car rental system, we want to create a piece of software for managing the rental administration with three primaries functions:

- Have an inventory of cars
- Calculate the price for rental
- Keep the track of the customer loyalty points

Price

The price of rentals is based in the type of the car and how many days it has been rented for.
The users will say when renting the car for how many days they want to rent it and they will pay up front.
If the car is returned late, then rent for extra days will be charged when returning.

The system has three types of cars:

- Premium cars- Price is **premium price** times number of days rented.
  - Extra day: **premium price** + 20% of <premium price>
- SUV cars - Price is:
  - For the first 7 days: **SUV price** per day.
  - Between 7 and 30 days: 80% of **SUV price** per day.
  - More than 30 days: 50% of **SUV price** per day
  - Extra day: **SUV price** + 60% of **small price** per day
- Small cars - Price is:
  - For the first 7 days: **small price** per day.
  - More than 7 days: 60% of <**small price** per day
  - Extra day: **small price** + 30% of **small price**

Prices:

- **premium price**: 300€
- **SUV price**: 150€
- **small price**: 50€

The program should expose a rest-api HTTP API (you could use any framework, if you don't have a personal preference, you could use Spring Boot)

The API should (at least) expose the following operations:

- Rent one or several cars and calculate the price.
- Return a car and calculate surcharges (if exist)

Example of price calculations

- BMW 7 (Premium) 10 day -> 3000€
- Kia Sorento (SUV) 9 days -> 1290€
- Nissan Juke (SUV) 2 days -> 300€
- Seat Ibiza (small) 10 days -> 440€

When returning cars late:

- BMW 7 (Premium) 2 extra days -> 720€
- Nissan Juke (SUV) 1 day extra -> 180€

Loyalty points

Customers get loyalty points when renting a car, regardless of the time.

- Premium car: 5 points.
- SUV car: 3 points.
- Small car: 1 point.