The friendly & not so intimidating Casumo Programming Test

Casumo

The friendly test

Objective

- Assess object oriented analysis and modelling skills, Java or Kotlin coding skills, code structuring and API design. Take your time on the task, but don't go too crazy. If you submit a solution that is in any way incomplete, the parts that you decided to focus on are relevant. Keeping the objective in mind, you are free to use whatever tools, libraries, frameworks are at your disposal.

- Please include a README in any format about decisions you made along the way, what you focused on, what you didn't focus on and why, as well as how to run and use the program.

The problem – Video rental store

For a video rental store we want to create a system for managing the rental administration. **We want two primary features:.**

1. Have an inventory of films
2. Calculate the price for rentals

Price

The price of rentals is based on the type of film rented and how many days the film is rented for. When renting, the customers say for how many days they want to rent for and pay up front. If the film is returned late, then rent for the extra days are charged when returning.

The store has three types of films.

1. New releases – For each day rented, the <premium price> is charged.
2. Regular films – A flat price of <basic price> is charged for the 3 days, and then for each additional day <basic price> is charged per day. .
3. Old film - A flat price of <basic price> is charged for the first 5 days, and then for each additional day <basic price> is charged per day. .

<premium price> is 40 SEK
<basic price> is 30 SEK

The program should expose a REST HTTP API.

The API should (at least) expose operations for

- Renting one or several films and calculating the price.
- Returning films and calculating possible surcharges.

Examples of price calculations

Matrix 11 (New release) 1 days 40 SEK

Spider Man (Regular rental) 5 days 90 SEK

Spider Man 2 (Regular rental) 2 days 30 SEK

Out of Africa (Old film) 7 days 90 SEK

Total price: 250 SEK

When returning films late:

Matrix 11 (New release) 2 extra days 80 SEK

Spider Man (Regular rental) 1 days 30 SEK

Total late charge: 110 SEK

Expectations

Since you made it that far, we already know that you can deliver a working solution but this task is not about it - we want to see how you structure the application, model the domain, the quality of your code, and how you test it.

Give us your best, the code you always wanted to work with but never had a chance to - show us how software should be made using Java or Kotlin, and any frameworks of your choice!

Keep in mind to not spend too much time on relatively *irrelevant* things like security, or excessive documentation.

Focus on structure, not on implementation details:

- Storing data in memory is fine, you don't need any particular SQL/noSQL engine underneath,
- We don't need full test coverage - show us how you structure tests, give meaningful examples,

- There are corners to cut, we know it. Don't forget to outline us which corners did you cut and why.

Remember that it's always better to sacrifice scope and not quality.

Let the open/close principle guide you through the design decisions.

Good luck!