

ANÁLISE E DESENHO DE SOFTWARE

PROJECTO 2017/20178

PARTE II

O objectivo desta segunda parte do projecto é a implementação dos casos de uso **Rent Items** e **Return Rented Items** detalhados na primeira parte.

Para tal vão acrescentar o vosso código ao projecto disponibilizado.

O código actual implementa os casos de uso **Process Sale** e **Filter Sales** realizados em iterações anteriores por uma outra equipa de desenvolvimento.

A implementação segue a estrutura típica por camadas:

- Na camada de domínio (pacote `business`) estão incluídas as classes que capturam o estado e comportamento dos conceitos analisados nos modelos de domínio e de desenho. Explore os padrões leccionados na disciplina e usem aqueles que considerarem vantajosos para estruturar ou resolver certas tarefas no projecto. Justifiquem as vossas decisões no relatório.
- Na camada de persistência (pacote `dataaccess`) encontram classes que armazenam e fazem *queries* a tabelas de dados de uma base de dados relacional. No final da execução do programa, as mudanças efectuadas devem ser armazenadas nas respectivas tabelas. Usamos a biblioteca Java Derby que permite este tipo de funcionamento sem estarmos a instalar *software* de bases de dados. Devem ler as classes implementadas neste pacote para perceber como se executam comandos SQL. Leiam também os *scripts* SQL da pasta `data/scripts/`. Nestes scripts podem incluir informação para criar tabelas e inserir dados que precisam de existir mas para os quais ainda não temos casos de uso implementados:
 - produtos disponíveis para aluguer
 - alugueres
 - produtos incluídos num certo aluguer
 - devoluções (de produtos alugados)
 - produtos incluídos numa certa devolução
- Na camada dos casos de uso (pacote `use_cases`) encontram as classes que servem de *handlers*, i.e., representantes dos casos de uso que contêm as tarefas necessárias à execução desse caso de uso (cf. padrão GRASP *Controller*).

- No pacote `client` encontram um exemplo de execução. Devem criar uma ou mais classes que demonstrem as funcionalidades que implementaram.
- No pacote `dbutils` encontram classes que apagam o conteúdo e recriam as tabelas da base de dados com a informação original definida nos *scripts* SQL.

O código deve vir acompanhado por *javadocs*, contratos e comentários adequados.

Devem igualmente incluir no pacote `src/test/business/` uma classe JUnit que efectue testes para as classes relacionadas com o caso de uso **Return Rented Items**.

A estrutura das classes e as suas relações devem estar de acordo com o modelo de desenho que tenham proposto.

Devem referir no relatório o conjunto de decisões mais importantes no que toca ao desenvolvimento e teste do vosso código.

Nota importante: quando importarem o projecto para o vosso Eclipse, façam-no via Maven para que tudo funcione correctamente! Quando o projecto estiver importado, a primeira coisa a fazer é executar a classe `CreateDatabase` do pacote `dbutils`. A partir daí, a classe `SimpleClient` já pode ser executada.

Entrega

O relatório correspondente a esta parte é a *continuação* do relatório descrito na primeira parte, estando contido no mesmo ficheiro pdf. Deve portanto manter a respectiva formatação. O projecto Eclipse deve ser compactado com a sua estrutura de pacotes intacta (não incluam os binários *.class*). O ficheiro zip deve ser entregue no moodle respeitando o mesmo prazo do relatório, ou seja, até às 23:00 do dia 1 de Junho de 2018.