MINISTRY OF EDUCATION AND RESEARCH

TECHNICAL UNIVERSITY
OF CLUJ-NAPOCA, ROMANIA

FUNDAMENTAL PROGRAMMING TECHNIQUES

ASSIGNMENT 3

ORDER MANAGEMENT

1. Requirements

Consider an application **OrderManagement** for processing customer orders for a warehouse. Relational databases are used to store the products, the clients and the orders. Furthermore, the application should be structured in packages using a layered architecture presented in the support material (Assignment_3_Indications.pdf) and should use (minimally) the following classes:

- **Model classes** - the data models of the application
- **Business Logic classes –** implement the application logic
- **Presentation classes –** implement the user input/output
- **Data access classes -** implement the access to the database

The application should allow processing commands from a text file given as argument, perform the requested operations, save the data in the database, and generate reports in pdf format. Other classes and packages can be added to implement the full functionality of the application.

Implement a parser to read commands in the Presentation layer (instead of the standard graphical user interface), and a pdf file generator to generate reports, according to the following examples:

Table 1. Description of the basic commands to be considered for the application

Command name	Command Syntax	Description
Add client to the database	Insert client: Ion Popescu, Bucuresti	Insert in the database a new client with name Ion Popescu and address Bucuresti
Delete Client from the database	Delete client: Ion Popescu	Delete from database the client with name Ion Popescu
Add product to the database	Insert product: apple, 20, 1	Add product apple with quantity 20 and price 1
Delete product from the database	Delete product: apple	Delete product apple from database
Create order for client	Order: Ion Popescu, apple, 5	Create order for Ion Popescu, with apple quantity 5. Also update the apple stock to 15. Generate a bill in pdf format with the order and total price of 5
Generate reports	Report client Report order Report product	Generate pdf reports with all clients/orders/products displayed in a tabular form. The reports should contain the information corresponding to the entity for which reports are asked (client, order or product) returned from the database by a SELECT * query, displayed in a table in a PDF file.

2. Deliverables

- A **solution description document** (minimum 2000 words, Times New Roman, 10pt, Single Spacing) with the structure specified in the **Lab Description** document.
- **Source files**
- **JavaDoc files**
- **jar file** required for executing the application
- **PDF files** generated as a result of running the commands specified in Figure 1 (see page 4).
- **SQL dump file** containing the SQL statements for (1) creating the database and the tables, (2) populating the tables with the corresponding data.

The deliverables will be **submitted** as follows:

- Create a repository on **gitlab** with the name: ***PT2020_Group_LastName_FirstName_Assignment_3***
- Push the following: **source code (push the code not an archive with the code), jar file, PDF reports, SQL dump file, documentation**
- Share the repository with the user **utcn_dsrl.**

3. Evaluation

The assignment will be graded as follows:

Requirement	Grading

Minimum to pass
• Object-oriented programming design
• Classes with maximum 300 lines
• Methods with maximum 30 lines
• Java naming conventions
• Basic documentation
• Use javadoc for documenting classes and generate the corresponding JavaDoc files.
• Use relational databases for storing the data for the application, minimum three tables: Client, Product and Order.
• File parser for the basic commands presented in Figure 1 (see page 4).
• For the create order command:
  ○ Create a bill for each order as.pdf file
  ○ The product stock will be decremented after the order is finalized.
  ○ In case that there are not enough products, the order will not be created and the PDF document representing the bill will not be generated. In this case, instead of the bill, a PDF document will be generated in which an under-stock message will be written.
	5 points

• Reports as .pdf files, generated for running the text file with commands from Figure 1 (see page 4) • jar file - the application should permit to be run with the following command: **java -jar PT2020_Group_FirstName_LastName_Assignment_3.jar commands.txt**	
Quality of the Documentation	1 point
Layered Architecture (the application will contain at least four packages: **dataAccessLayer**, **businessLayer**, **model** and **presentation**)	2 points
Database Structure (more than three tables)	1 point
Use **reflection techniques** to create a generic class that contains the methods for accessing the DB: create object, edit object, delete object and find object. The queries for accessing the DB for a specific object that corresponds to a table will be generated dynamically through reflection.	1 point

The figure below (i.e. Figure 1) presents the content of the text file with the commands that should be parsed and processed:

```
Insert client: Ion Popescu, Bucuresti
Insert client: Luca George, Bucuresti
Report client
Insert client: Sandu Vasile, Cluj-Napoca
Report client
Delete client: Ion Popescu, Bucuresti
Report client
Insert product: apple, 20, 1
Insert product: peach, 50, 2
Insert product: apple, 20, 1
Report product
Delete Product: peach
Insert product: orange, 40, 1.5
Insert product: lemon, 70, 2
Report product
Order: Luca George, apple, 5
Order: Luca George, lemon, 5
Order: Sandu Vasile, apple, 100
Report client
Report order
Report product
```

Figure 1. Example of Text file with commands for application

4. Bibliography

- **Connect to MySql from a Java application**
  - http://www.mkyong.com/jdbc/how-to-connect-to-mysql-with-jdbc-driver-java/
  - http://theopentutorials.com/tutorials/java/jdbc/jdbc-mysql-create-database-example/
- **Layered architectures**
  - https://dzone.com/articles/layers-standard-enterprise
- **Reflection in Java**
  - http://tutorials.jenkov.com/java-reflection/index.html
- **Creating PDF files in Java**
  - https://www.baeldung.com/java-pdf-creation
- **JAVADOC**
  - https://www.baeldung.com/javadoc
- **SQL dump file generation**
  - https://dev.mysql.com/doc/workbench/en/wb-admin-export-import-management.html
  - https://dev.mysql.com/doc/refman/5.7/en/using-mysqldump.html