MINISTRY OF EDUCATION AND RESEARCH

TECHNICAL UNIVERSITY
OF CLUJ-NAPOCA, ROMANIA

FUNDAMENTAL PROGRAMMING TECHNIQUES

ASSIGNMENT 3

ORDERS MANAGEMENT

1. Requirements

Consider an application **Orders Management** for processing client orders for a warehouse. Relational databases should be used to store the products, the clients, and the orders. The application should be designed according to the layered architecture pattern and should use (minimally) the following classes:

- **Model classes** - represent the data models of the application
- **Business Logic classes** - contain the application logic
- **Presentation classes –** GUI related classes
- **Data access classes** - classes that contain the access to the database

Note: Other classes and packages can be added to implement the full functionality of the application.

2. Deliverables

- A **documentation** written in the template provided on the laboratory Web site.
- **Source files, JavaDoc files, SQL dump file** (containing the SQL statements for: creating the database and the tables, and populating the tables with the corresponding data) – will be uploaded on the personal ***Gitlab*** account created according to the instructions in the **Laboratory Resources** document, and following the steps:
  - Create a private repository on ***Gitlab*** named according to the following template:
    PT2024_Group_FirstName_LastName_Assignment_3
  - Push the source code, the documentation, the JavaDoc files and the SQL dump file (**!!!not an archive with the code!!!**).
  - Share the repository with the user **utcn_dsrl**

3. Evaluation

The assignment will be graded as follows:

Requirement	Grading

• Use an object-oriented programming design, classes with maximum 300 lines, methods with maximum 30 lines, Java naming conventions.
• 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.
• Create a graphical user interface including:
  • A window for client operations: add new client, edit client, delete client, view all clients in a table (JTable)
  • A window for product operations: add new product, edit product, delete product, view all product in a table (JTable)
  • A window for creating product orders - the user will be able to select an existing product, select an existing client, and insert a desired quantity for the product to create a valid order. In case there are not enough products,
	5 points

an under-stock message will be displayed. After the order is finalized, the product stock is decremented. • Use reflection techniques to create a method that receives a list of objects and generates the header of the table by extracting through reflection the object properties and then populates the table with the values of the elements from the list. • Good quality documentation covering the sections from the documentation template.	
Layered Architecture (the application will contain at least four packages: dataAccessLayer, businessLayer, model and presentation).	2 points
Define an immutable Bill class in the Model package using Java records. A Bill object will be generated for each order and will be stored in a Log table. The bills can only be inserted and read from the Log table; no updates are allowed.	1 point
Use reflection techniques to create a generic class that contains the methods for accessing the DB (all tables except Log): 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.	2 points

4. Bibliography

- **Connect to MySql from a Java application**
  - https://www.baeldung.com/java-jdbc
  - http://www.mkyong.com/jdbc/how-to-connect-to-mysql-with-jdbc-driver-java/
- **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