**Universitatea Tehnica din Cluj-Napoca**
**Departament Calculatoare**

Programming Techniques, Undergraduate 2018/19
Prof. I. Salomie / Conf. T. Cioara / SL. C. Pop
{ioan.salomie, tudor.cioara, cristina.pop}@cs.utcluj.ro

**TP Lab – Homework 3**
**Grading**

## Mandatory requirements for accepting the assignment:

1. OOP paradigms
2. Classes of maximum 200 lines (except the UI classes)
3. Methods of maximum 30 lines
4. Java naming conventions https://google.github.io/styleguide/javaguide.html

## Minimal requirements for accepting the assignment (5 points):

- Graphical interface:
  - Window for client operations: add new client, edit client, delete client, view all clients in a table (JTable)
  - Window for product operations: add new product, edit product, delete product, view all product in a table (JTable)
- Create a product order for a client: the application 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 that there are not enough products, an **under stock message** will be displayed. After the order is finalized, the product stock is decremented.
- Use relational databases for storing the data for the application, minimum three tables: Client, Product and Order.
- 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
- Documentation

## Requirements for maximum grade (5 points):

| Requirement | Grading |
|---|---|
| Create a bill for each order as a text file or .pdf file | 1 point |
| Quality of the Documentation | 1 point |
| **Layered Architecture** (the application will contain at least four packages: **dataAccessLayer**, **businessLayer, model** and **presentation**) | 3 points |

**Requirements for extra points:**

| Requirement | Grading |
|---|---|
| Correct Database Structure (needs 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. | 2 points |

# Submission

- Share the source code and the documentation to the indicated bitbucket account.

# Tutorials

- 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/
- https://dzone.com/articles/layers-standard-enterprise
- Reflection: http://tutorials.jenkov.com/java-reflection/index.html