

CENG 112 - Data Structures

Assignment 2: Furniture Manufacturing and Ordering System

In this homework, you are expected to implement a “Furniture Manufacturing and Ordering System” application using Java. This homework will cover the topics given below;

1. Strings, Arrays, Generics
2. Queue ADT (Abstract Data Type)
3. Stack ADT (Abstract Data Type)

Assume that you are a Furniture Manufacturer called “*IZTECHEA*”, that manufactures different types of furniture. However, you immediately manufacture furniture according to the order that comes from your **Marketing Analyst** and you store it in **one factory line** that is based on a First-In-First-Out (FIFO) structure. **Marketing Analyst** randomly generates what to be produced and sends the request to the “*IZTECHEA*”. Your factory manufactures 6 different types of products, where all of them must implement the **IProduct interface**;

1. Sofa
2. Bed
3. Chair
4. Dresser
5. Table
6. Bookcase

The methods for interface IProduct is given below:

```
public interface IProduct {  
    public boolean isManufactured();  
    public boolean isStored();  
    public boolean isSold();  
}
```

When you manufacture a product, it is temporarily stored in the factory (production line) until the **Storage Chief** moves (i.e., “stores”) it into a warehouse. Note that, at each request, the **Storage Chief takes** one product from the factory line, i.e. queue, and stores each product in its own warehouse. Thereby, you will have **six warehouses**. The warehouses are designed to retrieve the first product that has been recently added to your warehouse. In other words, furniture is stored in Last-In-First-Out (LIFO) order.

When a **customer** comes to buy any furniture, that furniture is decided randomly, and a **customer** request is formed accordingly. Based on the requested furniture, you find the corresponding warehouse and remove/pop it from the warehouse to sell it to the customer. If you don't have the requested product in the respective warehouse, you should return a *FAIL* message. If you have it, you should return a *SUCCESS* message. *FAIL* and *SUCCESS* messages work in the same way for the marketing analyst and storage.

Write a simulation that randomly receives requests from **Marketing Analyst**, **Storage Chief**, and **Customer**. There is only one object representing each role. You should only ask the user to enter a single input that is the number of requests for the simulation. After all the requests are completed, you should print a report that shows the number of products in the factory, in the warehouse, and that is sold.

The main structure of the program is given as a pseudocode:

r1 chooses among Marketing Analyst, Storage Chief and Customer.

r2 chooses among Sofa, Bed, Chair, Dresser, Table, Bookcase.

```
Input number of requests
products[] {Sofa, Bed, Chair, Dresser, Table, Bookcase}
from 1 to number of requests
    select a random number r1 in [0,2] # 3 roles (i.e. activities)
    if r1 = 0
        select a random number r2 in [0,5] # 6 types of furniture
        product = products[r2]
        trigger Marketing Analyst for product
    endif
    if r1 = 1
        trigger Storage Chief
    endif
    if r1 = 2
        select a random number r2 between 0-5 # 6 types of furniture
        product = products[r2]
        trigger Customer for product
    endif
    product = null
end
```

An example input/output is given below:

Enter the number of random request cycles: 13

1. **Customer** buying Table, **FAIL**, Table warehouse empty
2. **Marketing Analyst** requesting Bed, **SUCCESS**, Bed manufactured
3. **Marketing Analyst** requesting Bed, **SUCCESS**, Bed manufactured
4. **Customer** buying Bed, **FAIL**, Bed warehouse empty
5. **Marketing Analyst** requesting Sofa, **SUCCESS**, Sofa manufactured
6. **Storage Chief** storing Bed, **SUCCESS**, Bed stored in Bed warehouse
7. **Storage Chief** storing Bed, **SUCCESS**, Bed stored in Bed warehouse
8. **Customer** buying Bed, **SUCCESS**, Customer bought Bed
9. **Storage Chief** storing Sofa, **SUCCESS**, Sofa stored in Sofa warehouse
10. **Marketing Analyst** requesting Chair, **SUCCESS**, Chair manufactured
11. **Storage Chief** storing Chair, **SUCCESS**, Chair stored in Chair warehouse
12. **Customer** buying Chair, **SUCCESS**, Customer bought Chair
13. **Marketing Analyst** requesting Sofa, **SUCCESS**, Sofa manufactured




REPORT:

Amount of Bed in Factory Line: 0
Amount of Sofa in Factory Line: 1
Amount of Dresser in Factory Line: 0
Amount of Table in Factory Line: 0
Amount of Chair in Factory Line: 0
Amount of Bookcase in Factory Line: 0

Amount of Bed in Bed Warehouse: 1
Amount of Sofa in Sofa Warehouse: 1
Amount of Dresser in Dresser Warehouse: 0
Amount of Table in Table Warehouse: 0
Amount of Chair in Chair Warehouse: 0
Amount of Bookcase in Bookcase Warehouse: 0

Amount of Bed Sold: 1
Amount of Sofa Sold: 0
Amount of Dresser Sold: 0
Amount of Table Sold: 0
Amount of Chair Sold: 1
Amount of Bookcase Sold: 0

Assignment Rules

- This is a group assignment (2 students). However, inter-group collaboration is not allowed!
- All assignments are subject to plagiarism detection and the suspected violations (the solutions derived from or inspired by the solution of other groups) cause to be graded as zero.
- It is not allowed to use Java Collections Framework.
- Your code should be easy to read and test:
 - Keep your code clean. Avoid duplication and redundancy. 
 - Follow Java Naming Conventions. 
 - Use relative paths instead of absolute ones. 

Submission Rules

All submissions must:

- be performed via **Microsoft Teams** by only one of the group members,
- be performed before the deadline,
- be exported as an Eclipse Project and saved in ZIP format,
- include all necessary data files (TXT, CSV, JSON, etc.) in the right directory,
- follow a specific naming convention such that CENG112_HW2_groupID.

Eclipse Project: CENG112_HW2_G5

Exported Archive File: CENG112_HW2_G5.zip

Submissions that do not comply with the rules above are penalized.

NOTE: Those who want to change groups can send their requests on Microsoft Teams.