Due Date: 31.03.2019, 23:55

CENG112 – Data Structures
Homework #2

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

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

Assume that you are a Computer Manufacturer called "*Iztech PC Factory*", that manufactures computer parts. However, you immediately manufacture computer part 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. The "*Marketing Analyst"* randomly generates what to be produced and sends a request to the "*Iztech PC Factory*". Your "*Iztech PC Factory"* manufactures five different computer parts, where all of them must implement the **IProduct interface**;

1. RAM (Random Access Memory)
2. CPU (Central Processing Unit)
3. Graphics Card
4. Motherboard
5. Cache

The methods for interface IProduct is given below:

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

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

When a customer comes to buy a computer part, that computer part is decided randomly, and a customer request is formed accordingly. Based on the requested computer part, 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 computer part 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 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. **<u>You should only ask the user to enter a single input that is the number of requests for the simulation.</u>** After all the requests are completed, you should print a report that shows the number of computer parts that in the factory, in the warehouse and that are sold.

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

r1 chooses among Marketing Analyst, Storage Chief and Customer.

r2 chooses among RAM, CPU, Graphics Card, Motherboard, Cache.

```
Input number of requests
products[] ← {RAM, CPU, Graphics Card, Motherboard, Cache}
from 1 to number of requests
      select a random number r1 between 0-2
      if r1 = 0
            select a random number r2 between 0-4
            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 0-4
            product = products[r2]
            trigger Customer for product
      endif
      product = null
end
```

An example output is given below.

```
Enter the number of random request cycles: 13
1.    Customer buying Motherboard, FAIL, Motherboard warehouse empty
2.    Marketing Analyst requesting RAM, SUCCESS, RAM manufactured
3.    Marketing Analyst requesting RAM, SUCCESS, RAM manufactured
4.    Customer buying RAM, FAIL, RAM warehouse empty
5.    Marketing Analyst requesting CPU, SUCCESS, CPU manufactured
6.    Storage Chief storing RAM, SUCCESS, RAM stored in RAM warehouse
7.    Storage Chief storing RAM, SUCCESS, RAM stored in RAM warehouse
```

```
8.     Customer buying RAM, SUCCESS, Customer bought RAM

9.     Storage Chief storing CPU, SUCCESS, CPU stored in CPU warehouse

10.    Marketing Analyst requesting Cache, SUCCESS, Cache manufactured

11.    Storage Chief storing Cache, SUCCESS, Cache stored in Cache warehouse

12.    Customer buying Cache, SUCCESS, Customer bought Cache

13.    Marketing Analyst requesting CPU, SUCCESS, CPU manufactured


REPORT:

Amount of RAM in Factory Line: 0
Amount of CPU in Factory Line: 1
Amount of Graphics Card in Factory Line: 0
Amount of Motherboard in Factory Line: 0
Amount of Cache in Factory Line: 0

Amount of RAM in RAM Warehouse: 1
Amount of CPU in CPU Warehouse: 1
Amount of Graphics Card in Graphics Card Warehouse: 0
Amount of Motherboard in Motherboard Warehouse: 0
Amount of Cache in Cache Warehouse: 0

Amount of RAM Sold: 1
Amount of CPU Sold: 0
Amount of Graphics Card Sold: 0
Amount of Motherboard Sold: 0
Amount of Cache Sold: 1
```

NOTE: While implementing your program please make sure that **your program is user friendly**. Try to **make your user inputs simpler**. For example, in a selection process rather asking the user to write or type a long string, make the selections with numbers. Thereby, it would be easier for us to evaluate your homework, but also easy for you to test your program.

Assignment Rules

1. In this lecture's homework, there is **no cheating allowed**. If any cheating has been detected, they **will be graded as 0** and there will be no further discussion on this.
2. You are expected to submit your homework in groups. Therefore, **only one of you** will be sufficient to submit your homework.
3. Make sure you export your homework as an **Eclipse project**. You can use other IDEs as well, however, you must test if it is supported by Eclipse.
4. Make sure that your ".txt" files (if there is any) are in your project after you exported it.
5. Please submit your homework through CMS.
6. You are **not allowed to use Collections Framework**. You should implement the data structures on your own.

7. **Late submissions are strictly not allowed!** Thereby, do not send us email to allow your lately submitted homework.
8. Please be informed that your submissions may be anonymously used in software testing and maintenance research studies. Your names and student IDs will be replaced with non-identifying strings. If you do not want your submissions to be used in research studies, please inform the instructor (Dr. Tuglular) via e-mail.
9. Please export your Java Project as the given format with your assigned group ID. **If you do not follow the given format you will lose points from your homework**. This format is necessary for us to write and run our tests on your homework.

   Example:
   Project Name: G2_CENG112_HW2
   Zipped Project Name: G2_CENG112_HW2.zip