Due Date: 28.04.2019, 23:55

CENG112 – Data Structures
Homework #3

In this homework, you are expected to implement a "Bank Transaction Management System" application using Java. This homework will cover the topics given below;

1. Linked Lists
2. List ADT (Abstract Data Type)
3. Priority Queue ADT (Abstract Data Type)

Assume that you are designing a transaction management system for a bank branch that has only **one counter**. In this system there are different types of customers with different priorities. You can assume that all the customers come at the same time and they request a transaction immediately. The requests should be placed in a queue-based structure with **FIFO approach with priorities**.

There are three types of customers:

1. CORPORATE → Priority: 1
2. INDIVIDUAL → Priority: 2
3. NON-REGISTERED → Priority: 3

Those implement the ***ICustomer interface***, which should be including but not limited**:**

```
public interface ICustomer {
   public String getType();
   public int getPriority();
   public String toString();
   ...
}
```

In this system, each transaction has an occupation time that denotes how many minutes the bank counter will be occupied. You want to keep statistics about the estimated waiting time for each new-coming customer, and this time is equal to sum of occupation time of all prior transactions.

Accordingly, the ***Transaction*** class can be defined as:

```
public Transaction {
   private int id;                  // unique id in [1,1000]
   private ICustomer customer;      // the request owner
   private int occupation;          // needed time for transaction
   private int waiting;             // needed time for previous transactions
   private Transaction next;
   public String toString();
   … // Constructors, getter setter and other helper methods
}
```

Where the ***TransactionQueue*** class is defined as:

```
public class TransactionQueue {
   private Transaction head;
   private int queueLength;
   private int totalWaitingTime;
   private String date;
   private TransactionQueue next;
   public void insert(Transaction T) {…}
   public Transaction remove() {…}
   public boolean isEmpty() {…}
   public boolean clear() {…}
   public String toString() {…}
   … // Constructors, getter setter methods, and other helper methods
}
```

You are expected to read 5 days of transaction data from *CENG112_HW3_Transactions.txt* file where each line is formed as:

date, transaction_id, customer_type, occupation_time

It would be convenient that you define an additional structure (List ADT) to keep the transaction queues and corresponding dates. The structure can be formed as:

```
public TransactionQueueList {
   private TransactionQueue head;
   private int listLength;
   public TransactionQueue getTQ(String date) {…}
   public boolean updateTQ(String date, TransactionQueue TQ) {…}
   public void insertTo(TransactionQueue TQ) {…}
   public TransactionQueue removeTQ(String date) {…}
   public boolean isEmpty() {…}
   public boolean clear() {…}
   public String toString() {…}
   … // Constructors, getter setter methods, and other helper methods
}
```

The bank official periodically needs to print some statistics which include:

- Transaction queue for a specific date
- Total number of transactions for a specific date
- Total and average waiting time for a specific date
- Total number of transactions for a specific type of customer in a specific date
- Total and average waiting time for a specific type of customer in a specific date
- The date with the highest workload (the date with maximum average waiting time)

Summary of the Requirements

1. Read transactions file and create a “transaction” for each line of the file
2. Create a “transaction queue” for each date
3. Keep “transaction queues” in a list
4. Generate statistics using “the list of transaction queues”.

Example Output:

```
01.05.2019 COUNTER ← 101|COR|5|0 ← 102|IND|4|5 ← 100|NON|3|9 ← 103|NON|2|12
02.05.2019 COUNTER ← 201|COR|5|0 ← 204|COR|3|5 ← 200|NON|8|8 ← 202|NON|3|16 ← 203|NON|5|19
03.05.2019 COUNTER ← 303|COR|3|0 ← 301|IND|1|3 ← 302|IND|6|4 ← 300|NON|2|10
04.05.2019 COUNTER ← 401|IND|6|0 ← 402|IND|3|6 ← 400|NON|5|9 ← 403|NON|4|14
05.05.2019 COUNTER ← 501|COR|1|0 ← 502|COR|6|1 ← 500|NON|8|7 ← 503|NON|5|15 ← 504|NON|3|20

Total transaction count in 05.05.2019 → 5
Total waiting time in 05.05.2019 → 43.0
Average waiting time in 05.05.2019 → 8.6
Total transaction count for CORPORATE customer in 05.05.2019 → 2
Total waiting time for CORPORATE customer in 05.05.2019 → 1
Average waiting time for CORPORATE customer in 05.05.2019 → 0.5
The date with the highest workload (max total waiting time) → 02.05.2019
```

NOTE: To make the testing process easy please **DO NOT** design an interactive application. It would be enough that you just use the input file and generate a report for each date (console output) by considering the requirements.

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_HW3
   Zipped Project Name: G2_CENG112_HW3.zip