

## ITU COMPUTER ENGINEERING DEPARTMENT

### BLG 233E DATA STRUCTURES

### HOMEWORK -2 : Banking Application



**Due Date : 4<sup>rd</sup> October, 2015**

In this assignment, you will implement a lineup of bank customers application by using the multilist data structure.

There are two types of customers, Premium (P) and Normal (N) respectively. The P type customers enter their own bank account number to the automat to get their number in the line. The N type customers do not enter any information, they just press to the button and get their number in the line. Within the waiting line, P type customers have higher priority than N type customers, namely P type customers' place comes in front of the N type customers. Within the groups, there is no additional precedence, for example, a P type customer is not superior to any other P type customer, and that applies to N type customers too.

Every customer waiting in the line may have multiple transactions. The transactions can be either a counter transaction where a customer deposits/withdraws money or a credit application transaction. For P type customers, the line number applies only for a single transaction. With this reason, if a P type customer has multiple transactions, the customer gets a new number in the line for his/her next transaction (i.e. goes to the end of the P line) after his/her first transaction was processed,. For N type customers the line number applies for all of his/her transactions, so N type customers get line numbers only for once (i.e. don't lose their order until all of their transactions are processed).

Customers are served from the beginning of the line one by one. When all of the transactions of a customer are processed, then the customer is removed from the line. However, a customer may decide to leave the bank before processing his/her transactions. The banking application in this case removes the customer from the waiting line and all his/her transactions.

A customer should have the following properties: name, surname, customer type, number of transaction, list of transactions.

A transaction should have the following property: transaction name.

Your program must include these operations:

- (A) Add a new customer **(20 pts)**
- (V) Process the next customer's transaction **(25 pts)**
- (S) Search a customer by name and surname **(10 pts)**
- (R) Remove a customer from the list **(15 pts)**
- (P) Print the customer list **(10 pts)**

You will briefly explain the main structure of your homework in your **REPORT**. Also remember to add the screenshots of console outputs to the report as well as a graphical demonstration of your data structure design. **(20 pts)**

**Commands:**

**A:** Adds a new customer to the line. The program requests the name of the customer, the customer type, and the transactions.

**S:** Searches a customer by name and surname. The user enters the name and the surname of the customer and the program should print the record of the related customer.

**V:** Processes the transaction(s) of the next customer according to the customer type and updates the line.

**R:** Removes a customer from the line who decides to leave the bank. The user enters the name, surname and type of the customer and then the program deletes this customer from the line.

**P:** Prints all of the customers actively waiting in the line.

**E:** Exits the program

Your program will take “**input\_file.txt**” as an input. It will process the commands within the input file one by one and will print the outputs to the “**output\_file.txt**”. A sample input file and the corresponding output file are given in the attachments. During the evaluation of your homework, your program will take a different input file which has the same structure with the sample input file and will be expected to create the correct output file.

The input file has a semi-colon separated structure. The first character represents the command and the rest of the parameters after the first semicolon represents the corresponding inputs for the command. For example

**A;Ali;Aksu;N;2;deposit;withdraw**

CommandName;CustomerName;CustomerSurname;CustomerType;NumberofTransactions;Transaction1;Transaction2;....

**R;Ali;Aksu;N**

CommandName;CustomerName;CustomerSurname;CustomerType

**S;Ali;Aksu**

CommandName;CustomerName;CustomerSurname

**Note:** If you have any question about the homework, please contact to the research assistant Mahiye ULUYAĞMUR- ÖZTÜRK via email ([muluyagmur@itu.edu.tr](mailto:muluyagmur@itu.edu.tr)).

### Submission Procedure:

1. Make sure to write your name and number to all of the files of your project in the following format:

```
/*  
* @Author  
* Student Name: !! enter here !!  
* Student ID : !! enter here !!  
* Date:  
*/
```

2. Briefly explain the main structure of your homework in your REPORT. Also remember to add the screenshots of console outputs to the report.

3. Use comments wherever necessary in your code to explain your aim.

4. Compile the code in SSH before you send your homework.

5. After you make sure that everything is compiled smoothly, archive all files (codes, report and other necessary files) into a zip file. Submit this file through [www.ninova.itu.edu.tr](http://www.ninova.itu.edu.tr). Ninova enables you to change your submission before the submission deadline. Do not miss submission deadline, the homeworks sent via e-mail will not be graded.

**Academic dishonesty including but not limited to cheating, plagiarism and collaboration is unacceptable and subject to disciplinary actions. Any student found guilty will receive F as his/her final grade for the course.**