Hacettepe University
Department of Computer Engineering
BBM104 Introduction to Programming Laboratory II
Programming Assignment 1

Submission Date	**: 27.03.2020**
Due Date	**:10.04.2020 (23:59)**
Programing Language	**: JAVA**
Title	**: Online Shopping Market**

INTRODUCTION

In this experiment you are expected to gain knowledge on basic JAVA programming. The program you are going to develop will deal with variables, loops, string operations and file read operations. Besides the programming task, you will also learn to comply with coding standards.

PROBLEM

In this experiment, you are expected to implement an Online Shopping Market System. You will read shopping list and price list from input files. For reading data from input files, you can use the code given in Appendix A. Your task is to calculate total cost of a shopping list based on the price list.

1. Shoppinglist.txt (Input)

Shopping list will be available as files in the format below:

[customer name] *space* **[surname]** *tab* **[type of membership]** *tab* **[shopping date]** *tab* **[product name]** *tab* **[quantity]** *tab* **[product name]** *tab* **[quantity]** *newline*

...

According to the shoppingList.txt file format, the shopping list file contains the **customer's name, type of membership** and **shopping date** in the first line. Type of membership can be either **Gold**, **Silver** or **Bronze**. There may be at least one or more item in the shopping list of each customer. Each item is defined with the **name of the item** and its **quantity**. Every item in the file separated with a **tab** character. There will be a **space** character between name and surname. Shopping list file can contain maximum of 10 users. A sample shopping list file is shown in Figure 1**.**

[customer name] *space* **[surname]** *tab* **[type of membership]** *tab* **[shopping date]** *tab* **[product name]** *tab* **[quantity]** *tab* **[product name]** *tab* **[quantity]** *newline …*

Hasan Uzun	gold	26.01.2020	Sweater	1	Jeans	2	
Ali Can	silver	20.02.2020	Sweater	2	Jeans	1	
Mehmet Karadeniz	bronze	19.01.2020	Jeans	2	Sweater	3	

Figure 1 shoppingList.txt Example

2. PriceList.txt File (Input)

This file contains the prices of each item. It also keeps the information about the expiry dates of the items. This file has the format of:

[product name] *tab* **[type of membership]** *tab* **[start date]** *tab* **[end date]** *tab* **[price]** ***newline***

[product name] ***tab*** **[type of membership]** ***tab*** **[start date]** ***tab*** **[end date]** ***tab*** **[price]** ***newline***

[product name] ***tab*** **[type of membership]** ***tab*** **[start date]** ***tab*** **[end date]** ***tab*** **[price]** ***newline***

…

The price of products varies according to the membership type and shopping date. Even if the product is purchased by the same type of member, the price may differ when purchased at different times. Every line in the file has the **product name, type of membership, start date, end date** and **price** (in Turkish Lira) columns separated by a tab character. Same product can appear in the file multiple times. You should pay attention to the type of membership and validity dates of products. Same products can occur in the file several times with different prices. An example file is shown in Figure 2**.**

Sweater	gold	01.01.2020	31.01.2020	55.5
Sweater	gold	01.02.2020	29.02.2020	62.5
Sweater	silver	01.02.2020	28.02.2020	58.5
Sweater	bronze	01.01.2020	31.01.2020	61.6
Jeans	silver	01.02.2020	28.02.2020	68.5
Jeans	gold	01.01.2020	31.01.2020	83.0
Jeans	bronze	01.01.2020	31.01.2020	103.0

Figure 2 priceList.txt Example

There is only one valid price for each item. In order to calculate the price of an item, you should take into account the following information: the name of the item (which is unique), the type of membership who wants to buy the item, the purchasing date.

3. Billing information

Once the shopping is completed and the right prices are calculated for every item in the shopping list, a bill is generated to reflect the total cost. The bills will be in the following format for each customer in the output screen:

[line][line][line][customer name] *space* **[surname][line][line][line]** *newline*
[product name] *tab* **[unit price]** *tab* **[quantity]** *tab* **[amount]** *newline*
Total *tab* **[Total amount]**
……

Items should be printed in the same order of the given shopping list file. An example output for the shopping file given in Figure 1 is seen in the **output screen** as shown in Figure 3.

```
---Hasan Uzun---
Sweater         55.5    1       55.5
Jeans   83.0    2       166.0
Total 221.5
---Ali Can---
Sweater         58.5    2       117.0
Jeans   68.5    1       68.5
Total 185.5
---Mehmet Karadeniz---
Jeans   103.0   2       206.0
Sweater         61.6    3       184.8
Total 390.8
```

Figure 3 Bill Example

Execution and Test

The input files is going to be given as program arguments. Shopping file is the **first argument** and the price list file is the **second argument**. In order to test your program, you should follow the following steps:

- Upload your java files to your server account (dev.cs.hacettepe.edu.tr)
- **Compile** your code (javac *.java)
- **Run** your program (java Main shoppingList.txt priceList.txt)
- Control your output data and format.

Submit Format

File hierarchy must be zipped before submitted (Not .rar, only .zip files are supported by the system)
<student id>.zip
- src.zip (Main.java, *.java)

Late Policy

You may use up to three extension days for the assignment. For each extension day, you will lose 10 points.

Grading Policy

Task	**Point**
Submitted	**1**
Compiled	**10**
Clean code	**15**
Coding standards	**5**
Output	**69**
Total	**100**

Notes and Restrictions

- Save all your work until the assignment is graded.
- Regardless of the length, use UNDERSTANDABLE names to your variables, classes and functions. The names of classes, attributes and methods should obey Java naming convention.
- Write READABLE SOURCE CODE block and comments.
- Name of all files are fixed (shoppingList.txt priceList.txt).
- All assignments must be original, individual work. Duplicate or very similar assignments are both going to be considered as cheating.
- Your programs will be executed in DEV machine, please make it work on dev before submitting.
- It is your duty to check the Piazza platform against any possible update about this assignment. If any instruction written by the TA violates any condition against this document, the new instruction(s) on Piazza is/are valid!

Appendix A.

You can use the following code to read your input files.

```
import java.io.IOException;
import java.nio.file.Files;
import java.nio.file.Paths;

public class ReadFromFile {

    public static String[] readFile(String path) {
        try {
        int i =0;
        int length= Files.readAllLines(Paths.get(path)).size();
        String[] results = new String[length];
            for (String line : Files.readAllLines(Paths.get(path))) {
                results[i++]= line;
            }
        return results;
        } catch (IOException e) {
            e.printStackTrace();
            return null;
        }
    }

    public static void main(String[] args) {
        String[] lines= readFile("testfile.txt");
        for (String line : lines) {
            System.out.println(line);
            }
    }
}
```