Hacettepe University
Department of Computer Science & Engineering
BiL138 Programming/Software Laboratory
Experiment 1.2

Subject : Introduction to Java
Submission Date : 13.03.2012
Deadline : 20.03.2012 - 17.00
Programming Language : Java
Development Environment : Eclipse IDE for Java Developers
Advisors : Dr. Ebru SEZER, R.A. Cumhur Yiğit ÖZCAN, R.A. Gültekin IŞIK

Introduction
In this experiment, you are expected to implement a simple restaurant management system using Java programming language and get familiar with object oriented programming.

Problem
The software that you will implement is going to fulfill some basic operations that are included in most of today's restaurant management systems. User interactions will be performed via console. Main menu of the system is going to provide user with 5 options which are :

```
Welcome to Restaurant Management System
Please choose an operation :

1 - Create New Table Order
2 - Add New Order to a Table
3 - Checkout a Table
4 - End of Day Report
5 - Exit
```

1. Creating a new table order
When new customers arrive and they make their initial order, this order will be entered to the system via this command. When user (usually a waiter) chooses this option, program will ask for a table number. After the table number is entered, program will display the restaurant's menu - with a number for each item- for user to add an item of the menu to the order. This way the user adds the desired item to the current order. Then, program will ask the user if he/she wants to add another item to the current order. If the answer is yes (Y or y), user will be asked for another item number but the menu will not be printed again. When the order is completed (by entering N or n), user will be able to see the main menu back again.

```
Please enter table number :
3
1 - Coca-Cola           2.5
2 - Water               1.0
3 - Salad               3.0
4 - French Fries                2.0
5 - Sandwich (Cheese)           4.0
6 - French Toast (Cheese)               4.5
7 - Pizza (Meat)                7.0
Please enter an item number
5
Item added. Do you want to add another item? (Y/N)
```

2. Adding a new order to a table
This option is nearly same as the above. User will use this option to add another order for a table that has already made an initial order. Naturally, this option could be used only for tables that has at least one order before and hasn't checked out yet. Apart from this, everything will be similar to scenario above.

3. Checkout a table
When customers want to checkout, user will use this option to calculate total price of items that are ordered and display it to the user. Table number

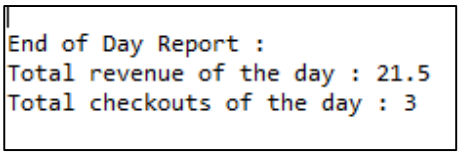
```
Please enter table number :
3
Total : 11.0

Welcome to Restaurant Management System
Please choose an operation :
```

which will checkout should be supplied by the user. Same as other options, main menu should be displayed back again after completing this operation.

4. End of day report

At the end of the day, user will get a report showing number of total checkouts and total revenue. After this option is operated, system should automatically shut itself down.



5. Exit

If user wishes to exit, system should ask if he/she wants to get an end of day report. According to the user's answer, end of day report would be printed to the console before exiting the program.

Menu

Restaurants menu will be supplied via a text file. You should check ftp server for a sample. Path of this file will be given as a command line argument. It will be the first and only argument of the program. Each time the system starts, it should read the menu from this file.

Tables

There are always 30 tables in the systems and they are numbered from 1 to 30. These numbers are unique identifiers for each table.

Limitations

* Number of items in a menu file is limited to 100
* Number of items in one order is limited to 20
* Number of orders for one table -before checkout- is limited to 10
* Number of checkouts for the whole system -for one day- is limited 200

Design Notes

* You may not use ArrayList class or any other class under java.util.Collections. You are expected to use arrays for this experiment.
* You don't have to check for inaccurate inputs. However if you do, explain your error messages in report.
* You need to use Scanner class for IO operations over the console. You may use any method you want to read from the menu file.
* You should stick with the examples given in this sheet in means of messages to be displayed on the console. Still, you are free to create your own error messages as soon as you explain them in your report.
* Class that contains your main method should be named "Main.java".

Report

You should write these topics on the report:
1. Cover Page,
2. Software Usage,
3. Software Design Notes: **Problem, Solution.** In the solution part, give a list of your classes and explain why you needed them (separately for each one).

Notes and Restrictions

1) Do not submit any file via e-mail. You should upload your files via “Online Experiment Submission System” which is at http://submit.cs.hacettepe.edu.tr

2) Submission Format:

<student_number>.zip

```
|---  src
        |--- Main.java
        |--- *.java
|--- report
        |--- report.pdf
```

3) You should submit only project files and source files; not compiled class files.
4) Your work should obey general software engineering principles and conventions. (Design, Comments, Indentation)
http://java.sun.com/docs/codeconv/html/CodeConvTOC.doc.html
http://java.sun.com/j2se/javadoc/writingdoccomments/

5) Save all your work until the assignment is graded.
6) This is not a team project. You are expected to provide an **INDIVIDUAL** work.
7) You should follow announcements on “Dersler.Bil138” newsgroup which is located in http://news.cs.hacettepe.edu.tr
8) No submission will be accepted after 20.03.2012 at 17.00