

**Hacettepe University**  
**Computer Science and Engineering Department**

**Name and Surname** : Necati Berk Özgür  
**Identity Number** : 21785229  
**Course** : BBM104  
**Experiment** : Programming Assignment 3  
**Subject** : Pizza Shop Order Control System  
**Data Due** : 27.04.2018  
**Advisors** : Pelin Canbay  
**e-mail** : [b21785229@cs.hacettepe.edu.tr](mailto:b21785229@cs.hacettepe.edu.tr)

**IMPORTANT WARNING: THIS PROGRAM TAKES THREE COMMAND LINE ARGUMENTS: input.txt order.txt customer.txt IN FOLLOWING ORDER.**

## **2. Software Using Documentation**

### **2.1. Software Usage**

By this software, user is able to manage the customers and orders of a pizzeria. **To conduct this task, user should provide three inputs “input.txt” “order.txt” “customer.txt” as command line arguments by following order.**

### **2.2. Provided Possibilities**

There is no extra facilities of this software than stated in the assignment paper.

### **2.3 Error Messages**

This program can handle some sort of errors:

- Error1 :           Error occurred while reading file! Check ... file!  
                  If program is faced with a undefined command or pizza/topping in .txt files, this error occurs.  
                  Check input files and try again.
- Error2 :           Maximum 3 toppings can be added!  
                  If user tries to add more than 3 toppings, this error is prompted.  
                  Check the number of toppings and try again.

### 3. Software Design Notes

#### 3.1. Description of the program

##### 3.1.1. Problem

Managing the orders and customers of a pizzeria.

##### 3.1.2. Solution

Storing orders and customers firstly. Then shaping these stored fields according to user's needs again (adding/removing/updating orders and customers)

#### 3.2. System Chart

INPUT	PROGRAMS	OUTPUT
Computer	Main.java	orderafter.txt
Order.txt	FileProcessor.java	customerafter.txt
Input.txt		output.txt
Command.txt		

#### 3.3. Main Data Structures

interface InterfaceTopping which is implemented by topping classes of pizza.  
class Pizza, class Order, class Customer.

#### 3.4. Algorithm

1. Make initialisation.
  - 1.1. Open input files.
  - 1.2. Assign attributes of customer and order files to their storages (arrays).
  - 1.3 Start reading input file.
2. For every line in input file, complete the tasks given by keeping storage elements up-to date.
3. Create updated customer and order files and output file.
4. Close files

#### 3.5. Special Design Properties

In this design, facility property of java language is used for two purposes.

First, to implement similar type of attributes. (InterfaceTopping implements Soudjouk, Salami, Onion and HotPepper.)

Second, to access database (arrays for this project) while hiding it from end user.

Class diagram which shows these interfaces is attached to report.

### 4. Software Testing Notes

#### 4.1 Bugs and Software Reliability

This program is not protected for miscreation of orders (for instance, creating an order to a non-existing customer ID).

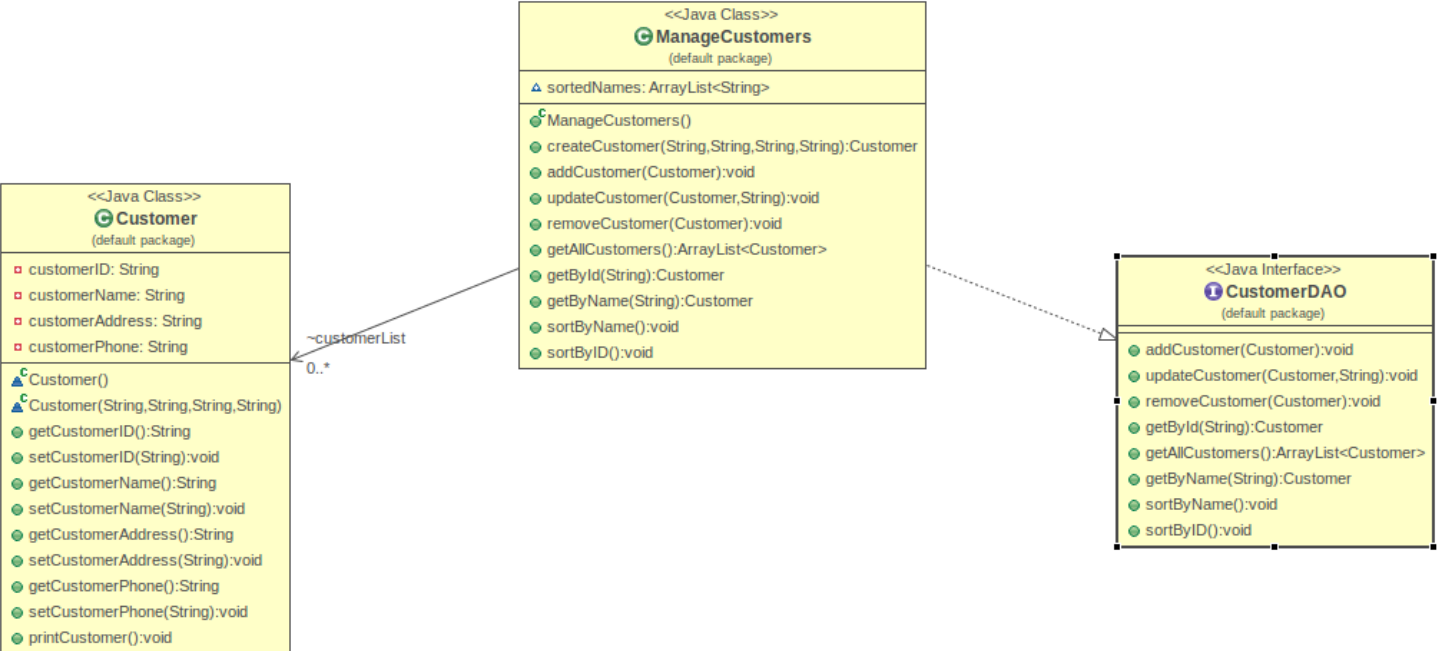
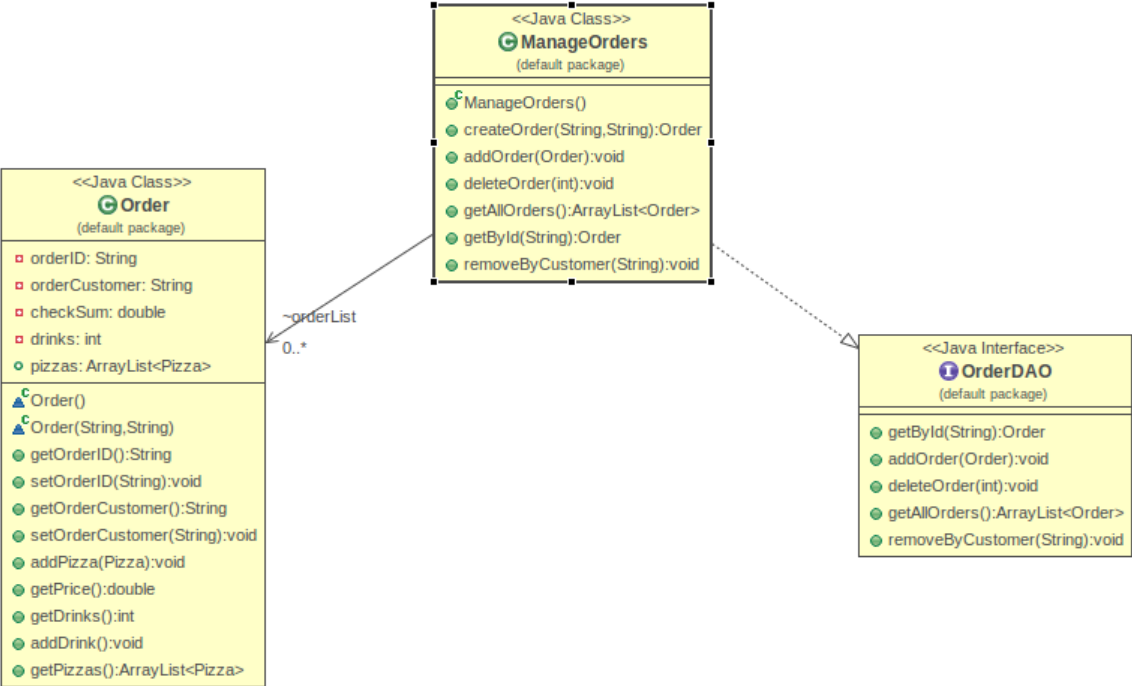
#### 4.2 Software Extendibility and Upgradability

By modular construction(interface, class etc.) of java language and the means of DAO, this code is easily upgradable.

## **REFERENCES**

Pressman, 1987 (Supplied by ftp of Hacettepe University Dep. of Computer Science and Engineering)

ATTACHMENTS



ATTACHMENT(cont.)

