2024

CSE 222 – HW1 REPORT

ELIFNUR KABALCI
1801042617

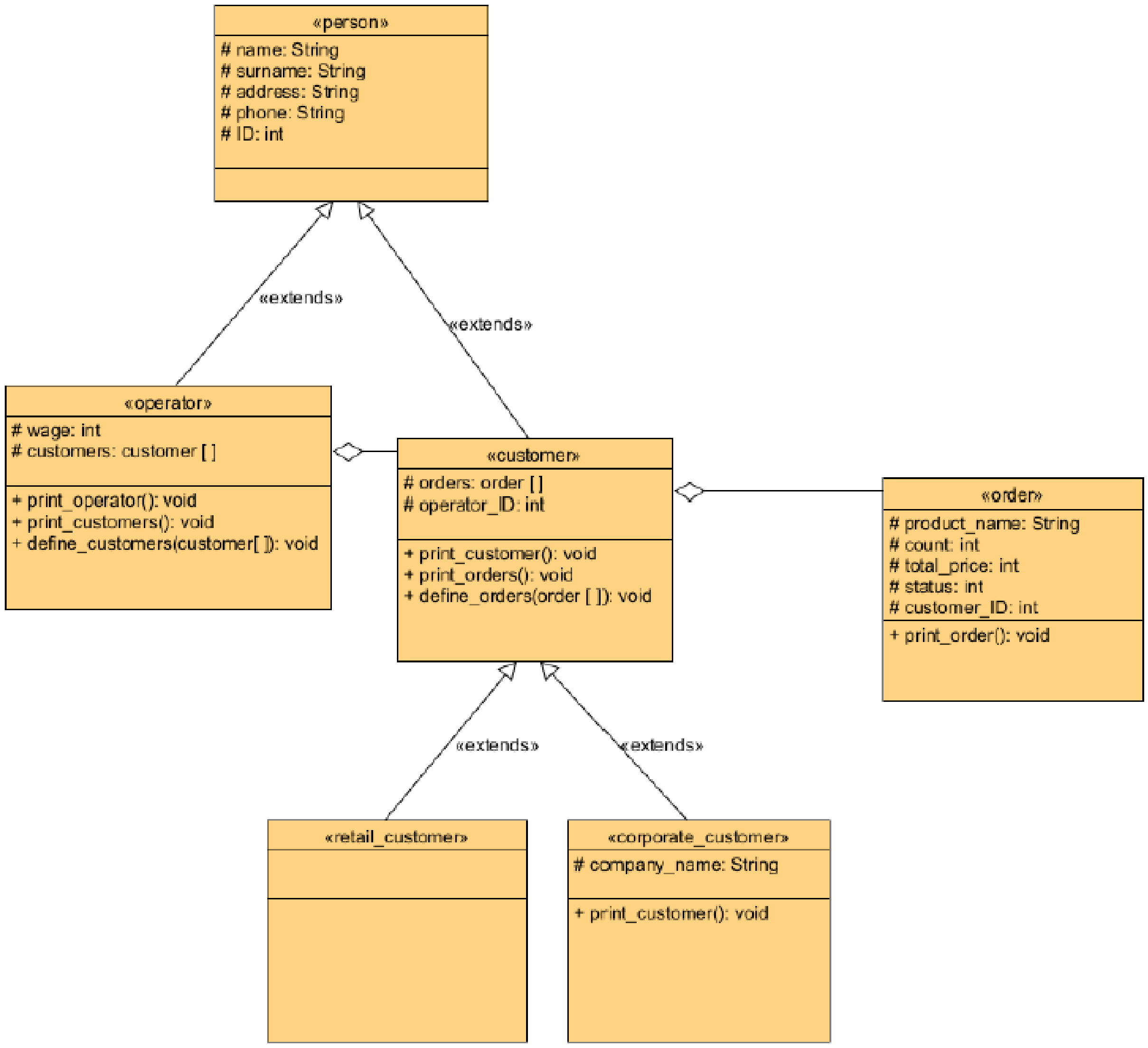
HW1 | cse222

Main Idea

We need to define the project shown with the class diagram and define the values read from content.txt to these classes.

We aim to use the input data received from the user as the customer or operator ID and print the information on the screen.

Class Diagram



Content.txt

Here are the values that will be used to fill the elements in the class diagram above. We extract the definitions of order, retail customer, corporate customer and operators and all the values that will help us identify them from this file.

An example content txt looks like this:

```
order;tv;2;2000;0;1500
order;computer;4;8000;1;1500
order;computer;2;5000;2;1501
order;smartphone;0;5000;2;1501
order;smartphone;2;3000;3;1502
retail_customer;yakup;genc;kocaeli;+902626052201;1500;500
corporate_customer;ibrahim;sogukpinar;kocaeli;+902626052202;1501;500;gebze technical
university
retail_customer;yusuf sinan;akgul;kocaeli;+902626052203;1502;501
operator;gokhan;kaya;istanbul;+902626050004;500;2000
operator;burcu;yilmaz;kocaeli;+902626050005;501;1900
operator;didem;gozupek kocaman;istanbul;+902626050006;502;2100
```

-- We have the following helpers for the equivalents of the numbers in the file:

order:

order;product_name;count;total_price;status;customer_id

retail_customer:

retail_customer;name;surname;address;phone;ID;operator_ID

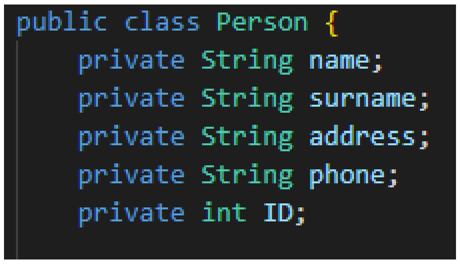
corporate_customer:

corporate_customer;name;surname;address;phone;ID;operator_ID;company_name

operator:

operator;name;surname;address;phone;ID;wage

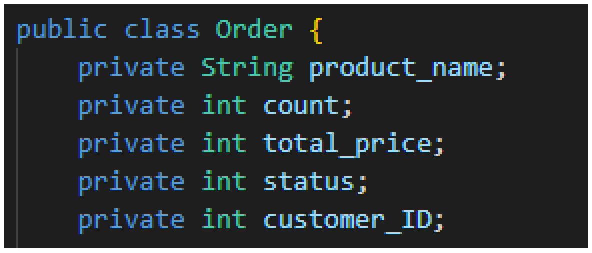
Person



```
public class Person {
    private String name;
    private String surname;
    private String address;
    private String phone;
    private int ID;
```

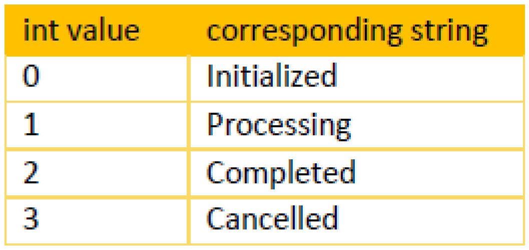
There is no action in this class. Here, there are only getter-setters of the values in the image. Since it is extended to Customer and Operator classes, its usage areas are there. Person class I is the top part of the class diagram.

Order



```
public class Order {
    private String product_name;
    private int count;
    private int total_price;
    private int status;
    private int customer_ID;
```

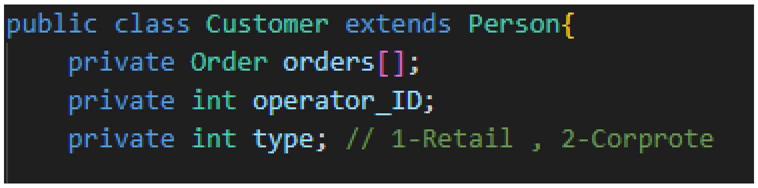
Order class is the least used structure after the person class. Here, too, there are mostly getter-setters of the values in the image. There is also a constructor here where we use set methods.



int value	corresponding string
0	Initialized
1	Processing
2	Completed
3	Cancelled

Additionally, the status of the order is given as an integer value in the txt file and we need to convert it to String. This process is done here with the status_turn method. This class also includes a print_order method to print the values it contains to the screen.

Customer



```
public class Customer extends Person{
    private Order orders[];
    private int operator_ID;
    private int type; // 1-Retail , 2-Corprote
```

Customer class is a class that extends from Person and extends to Retail and Corporate Customer classes. Every customer's orders are kept here. There is also an Operator who takes care of each customer. Its operator_id is kept as its identification information.

Type, which is a value not included in the class diagram, was used in the print method. Since corporate customer has more value than retail and both classes use the print_customer method within this class, I had to make conditions.

There is also a print_order method here to print the orders belonging to the customer to the screen and it calls the method with the same name in the order class.

Since there is an order array in the parameter given to us as a constant in the define order method, I wanted to use this method by calling it with a filled array defined elsewhere. However, while doing this, I got errors while assigning, and I realized that I had not set the initial definition of the orders array anywhere, and that is why I used this method.

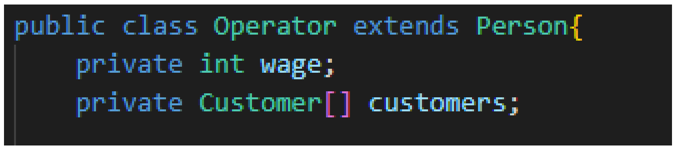
Retail Customer

There is no value stored in this class structure. There are no methods other than Constructor. Here, I just defined and performed the operations by calling the methods in the upper classes.

Corporate Customer

Here, in addition to the retail customer class, the company name value is kept. This is the additional feature that we press on the screen. In addition to the getter-setter I, constructor and customer class of this value, there is an additional method print_customer here. Because we need to make some guidance here. Since we cannot extract the company name from the customer class, we have to print it on the screen.

Operator


```
public class Operator extends Person{
    private int wage;
    private Customer[] customers;
```

Burada sadece operator ın maaşı ve her bir operator ın müşterilerinin listeleri bulunuyor. Bunlarla ilgili getter-setter lar var. Burada tanımlamaları yaptığım bir constructor ım var. Constructor ın içinde ona ait olan customer ları tanımlamak için alanlar bulunuyor. Bu class da da print_operator methodu var. Bu method customer bilgileri için kendi bünyesinde bulunan print_customers methodunu çağrıyor. Bu method da her bir customer için Customer methodundaki print methodunu çağıırıyor. O da Her bir müşterinin siparişi için order ın içindeki print methodunu çağrıyordu.

Main

In this class, I first read the content.txt file and saved it into a 2d String array.

Then, I passed this information through error checking based on string and integer. Thus, I have a 2D allinformation array that is free from errors. I went to the object creation method with this array.

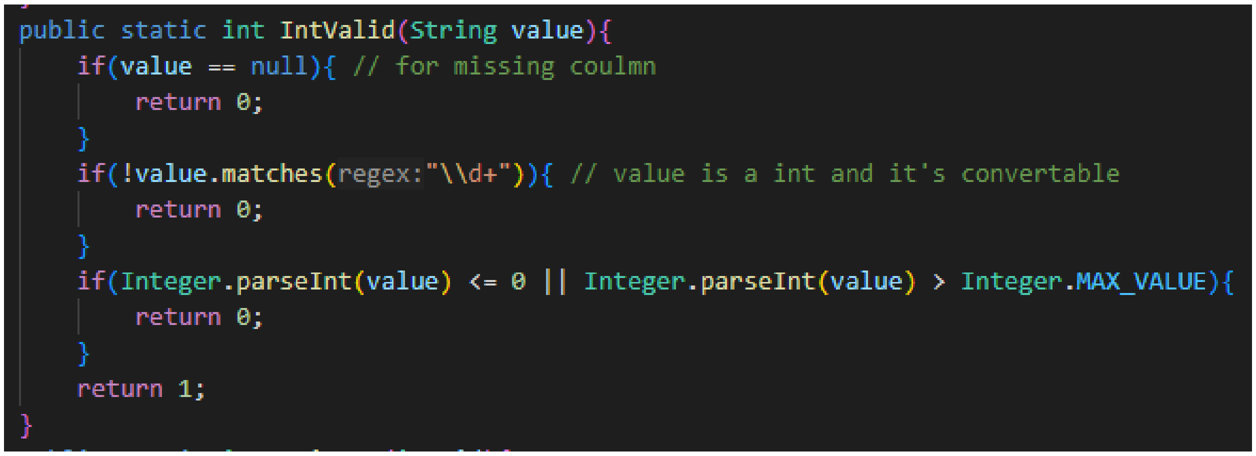
Here I focused on customer and operator types. While defining objects to classes belonging to these types, I sent all the data that came before it in the allinformation class. Thus, the operator would be able to find his own customers. Customers also place their orders.

After all the data was entered, the objects were created, and the created objects were kept in arrays of their own types, I started receiving data from the user.

I received an integer value from the user with the help of Console. I passed this through integer error checking and started to examine which id was the entered input. First, I examined the operators. If they didn't have the relevant id, I looked at the customers. If they don't have it either, I printed a warning that there is no one belonging to this id.

Error Checking

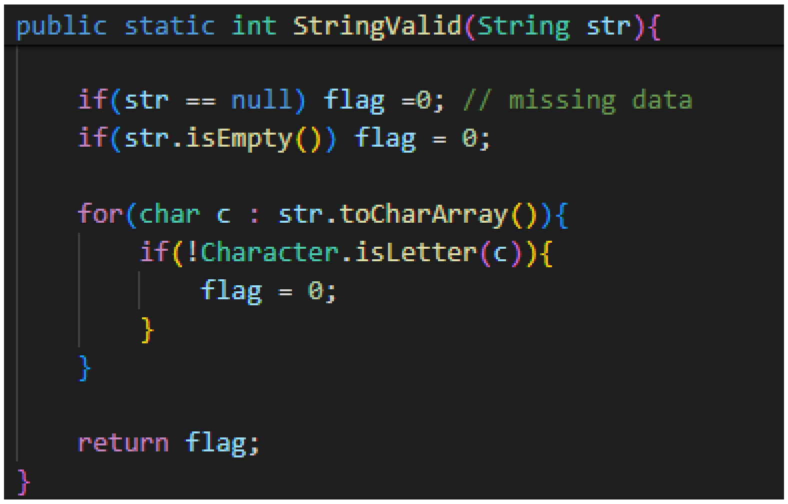
I tried to do error checking in each area of the project, but since some of them were repeated very frequently, I methodized them.



```java
public static int IntValid(String value){
    if(value == null){ // for missing coulmn
        return 0;
    }
    if(!value.matches(regex:"\\d+")){ // value is a int and it's convertable
        return 0;
    }
    if(Integer.parseInt(value) <= 0 || Integer.parseInt(value) > Integer.MAX_VALUE){
        return 0;
    }
    return 1;
}
```

This was written to check the accuracy of the integer value received from the user with the help of the console and all integer values read from the file.

The values must be positive and in the integer value field, they must be convertible, and if empty data is processed, it also causes the data to be deleted.



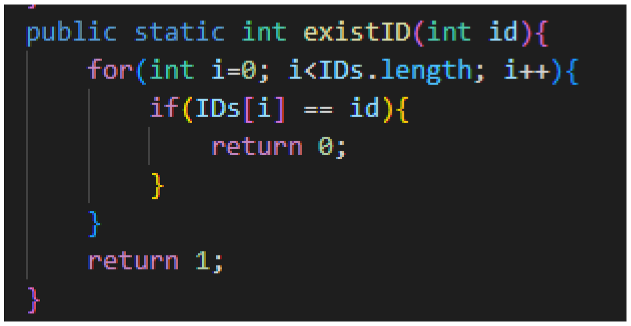
```java
public static int StringValid(String str){

    if(str == null) flag =0; // missing data
    if(str.isEmpty()) flag = 0;

    for(char c : str.toCharArray()){
        if(!Character.isLetter(c)){
            flag = 0;
        }
    }

    return flag;
}
```

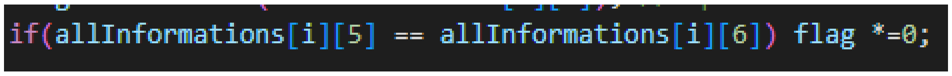
Here again, there is a missing data check, and then we examine whether all the characters in the string are letters.



```java
public static int existID(int id){
    for(int i=0; i<IDs.length; i++){
        if(IDs[i] == id){
            return 0;
        }
    }
    return 1;
}
```

In this method, the purpose of this method is to prevent us from adding a new id since it has been added before.

I used flags where all of these were called. I wanted it to return 0 if there is a problem, 1 otherwise.

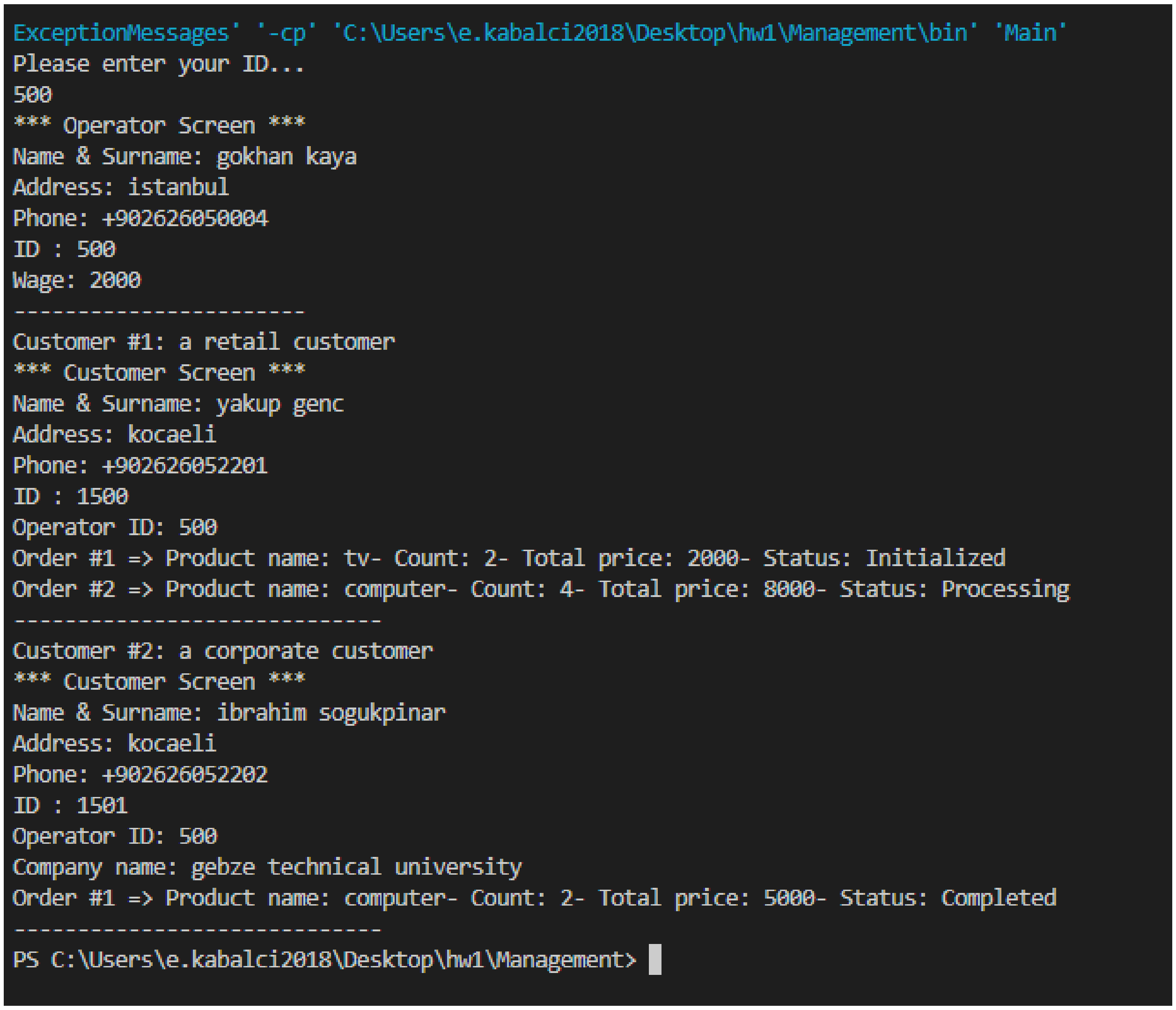


```
if(allInformations[i][5] == allInformations[i][6]) flag *=0;
```

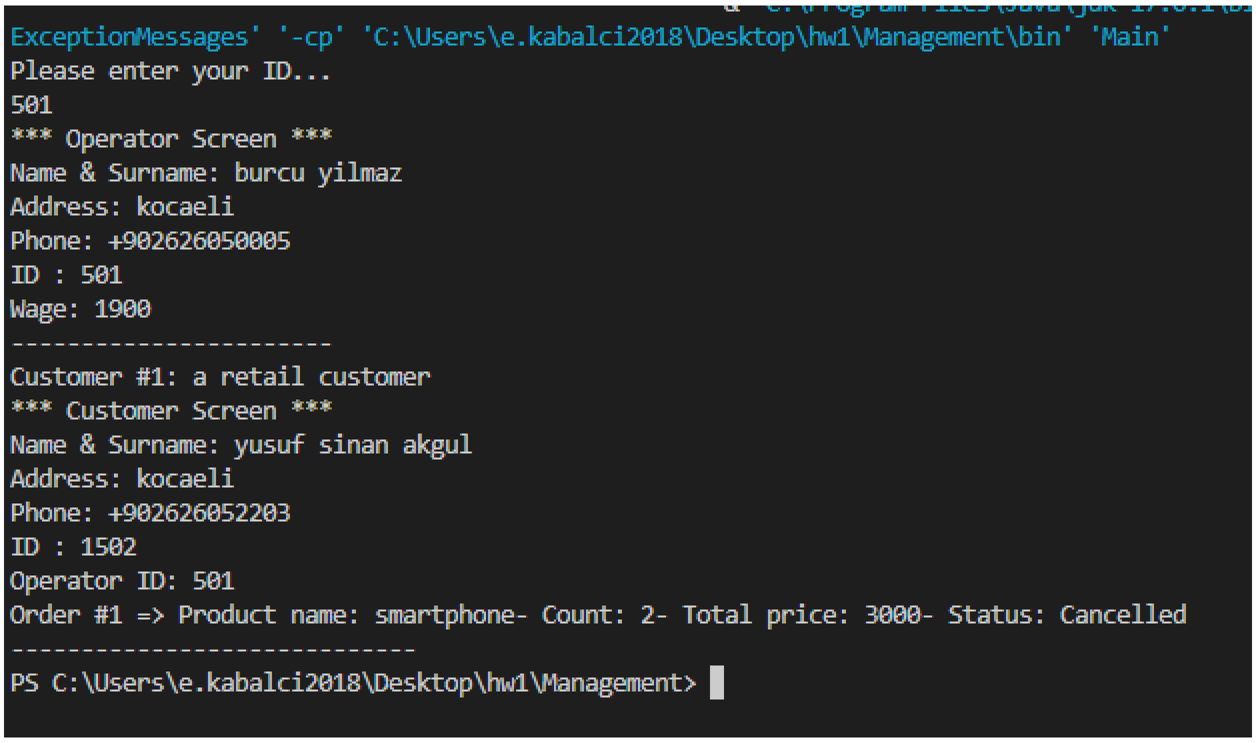
This was added to the controls of only customer types so that customer and operator ids are not the same.

Outputs:

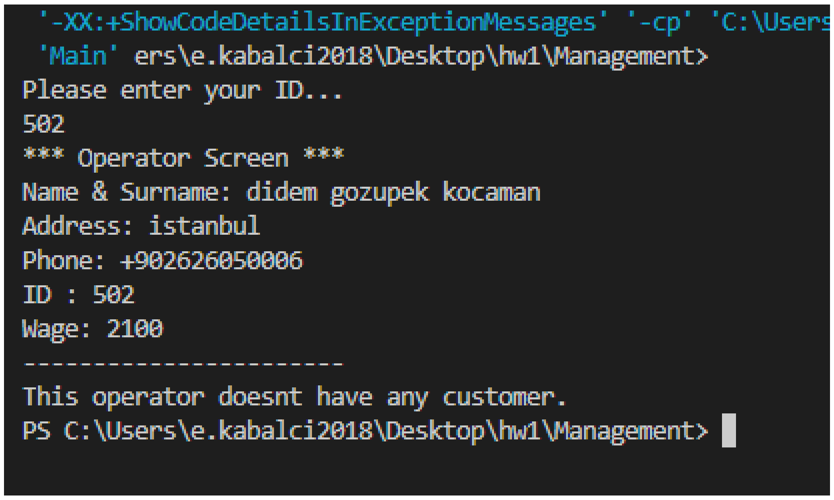
The questions that make up these outputs are the same as those in the homework PDF. It was produced using the content file given to us in the beginning.



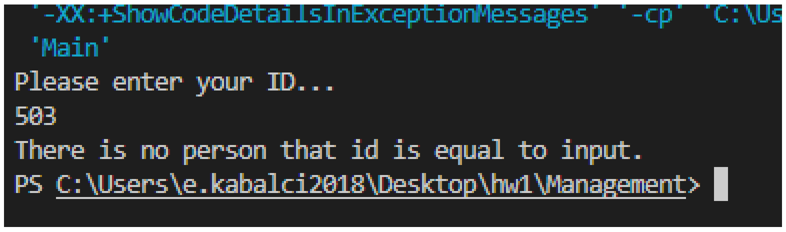
```
ExceptionMessages' '-cp' 'C:\Users\e.kabalci2018\Desktop\hw1\Management\bin' 'Main'
Please enter your ID...
500
*** Operator Screen ***
Name & Surname: gokhan kaya
Address: istanbul
Phone: +902626050004
ID : 500
Wage: 2000
-----------------------
Customer #1: a retail customer
*** Customer Screen ***
Name & Surname: yakup genc
Address: kocaeli
Phone: +902626052201
ID : 1500
Operator ID: 500
Order #1 => Product name: tv- Count: 2- Total price: 2000- Status: Initialized
Order #2 => Product name: computer- Count: 4- Total price: 8000- Status: Processing
-----------------------------
Customer #2: a corporate customer
*** Customer Screen ***
Name & Surname: ibrahim sogukpinar
Address: kocaeli
Phone: +902626052202
ID : 1501
Operator ID: 500
Company name: gebze technical university
Order #1 => Product name: computer- Count: 2- Total price: 5000- Status: Completed
-----------------------------
PS C:\Users\e.kabalci2018\Desktop\hw1\Management>
```



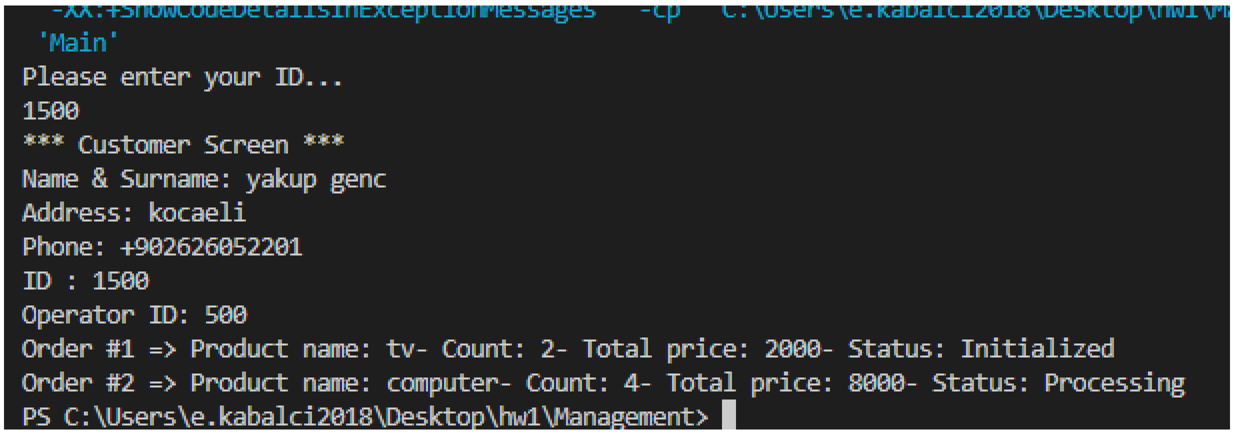
```
ExceptionMessages' '-cp' 'C:\Users\e.kabalci2018\Desktop\hw1\Management\bin' 'Main'
Please enter your ID...
501
*** Operator Screen ***
Name & Surname: burcu yilmaz
Address: kocaeli
Phone: +902626050005
ID : 501
Wage: 1900
-----------------------
Customer #1: a retail customer
*** Customer Screen ***
Name & Surname: yusuf sinan akgul
Address: kocaeli
Phone: +902626052203
ID : 1502
Operator ID: 501
Order #1 => Product name: smartphone- Count: 2- Total price: 3000- Status: Cancelled
------------------------------
PS C:\Users\e.kabalci2018\Desktop\hw1\Management>
```



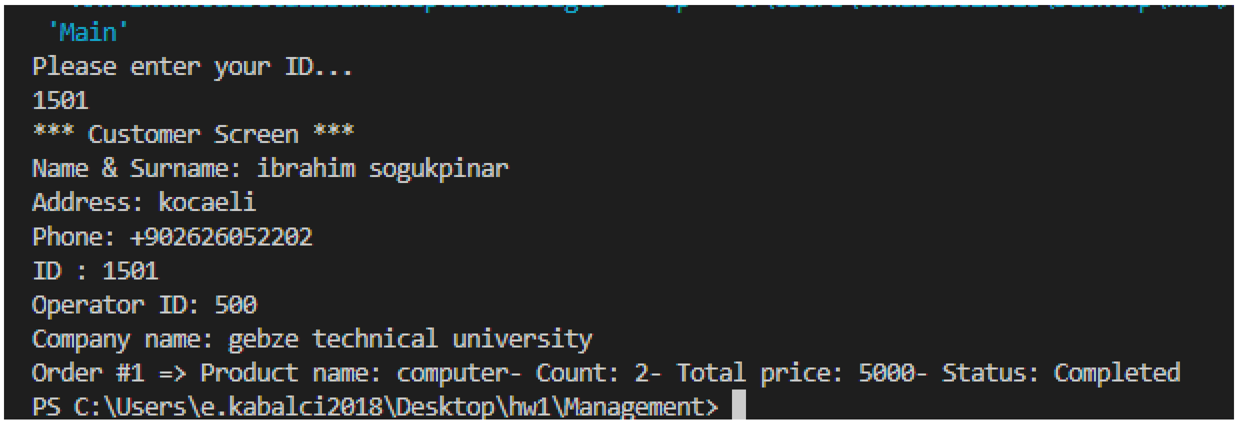
```
 '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users
 'Main' ers\e.kabalci2018\Desktop\hw1\Management>
Please enter your ID...
502
*** Operator Screen ***
Name & Surname: didem gozupek kocaman
Address: istanbul
Phone: +902626050006
ID : 502
Wage: 2100
-----------------------
This operator doesnt have any customer.
PS C:\Users\e.kabalci2018\Desktop\hw1\Management>
```



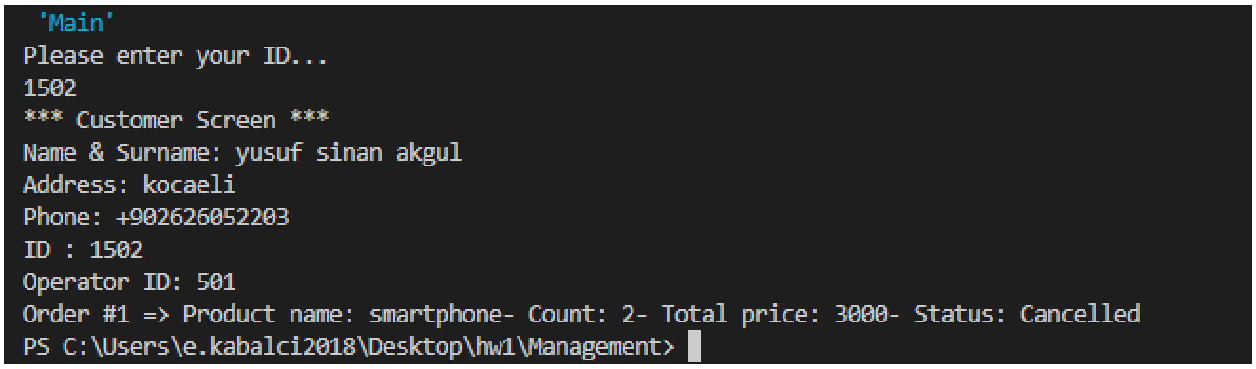
```
 '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Us
 'Main'
Please enter your ID...
503
There is no person that id is equal to input.
PS C:\Users\e.kabalci2018\Desktop\hw1\Management>
```



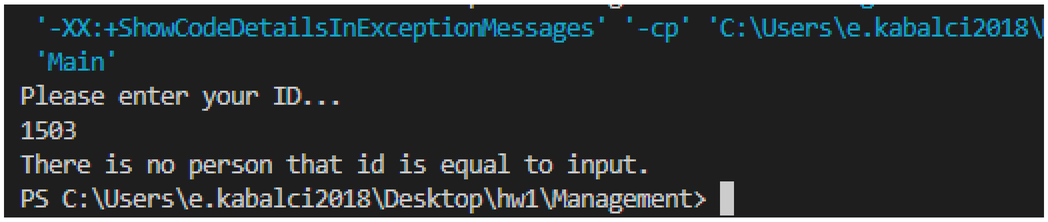
```
 -XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\e.kabalci2018\Desktop\hw1\M
 'Main'
Please enter your ID...
1500
*** Customer Screen ***
Name & Surname: yakup genc
Address: kocaeli
Phone: +902626052201
ID : 1500
Operator ID: 500
Order #1 => Product name: tv- Count: 2- Total price: 2000- Status: Initialized
Order #2 => Product name: computer- Count: 4- Total price: 8000- Status: Processing
PS C:\Users\e.kabalci2018\Desktop\hw1\Management>
```



```
 'Main'
Please enter your ID...
1501
*** Customer Screen ***
Name & Surname: ibrahim sogukpinar
Address: kocaeli
Phone: +902626052202
ID : 1501
Operator ID: 500
Company name: gebze technical university
Order #1 => Product name: computer- Count: 2- Total price: 5000- Status: Completed
PS C:\Users\e.kabalci2018\Desktop\hw1\Management>
```



```
 'Main'
Please enter your ID...
1502
*** Customer Screen ***
Name & Surname: yusuf sinan akgul
Address: kocaeli
Phone: +902626052203
ID : 1502
Operator ID: 501
Order #1 => Product name: smartphone- Count: 2- Total price: 3000- Status: Cancelled
PS C:\Users\e.kabalci2018\Desktop\hw1\Management>
```



```
 '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\e.kabalci2018\
 'Main'
Please enter your ID...
1503
There is no person that id is equal to input.
PS C:\Users\e.kabalci2018\Desktop\hw1\Management>
```