Programming fundamentals II

Milestone 2 documentation

Carlos Córdoba Ruiz

Manuel Hurtado Lillo

Index:

Analysis of Requirements

The program will read three files, one with the information of the products (number of products, their code, their name, their price, the number of units, if the product is perishable, if the product is perishable it has a number of days, if it is not perishable it has an offer), this part is checked by salesmngmnt class, we have other file which contains information of the employees (number of employees, their name, their code, their password, the level of the employee, their turn (night or day), if it is day they have a retention and if it is night they have a plus) this part is managed by employemngmn class, and finally we have another text with the offers (number of offers, their id, the type and the percentage), that file is managed in offer class where we have another two child classes (OffertMult and OffertPercentage) .

To be able to use the program, the employee has to introduce his/her identification (login and password), the employee has 3 opportunities to introduce their identification, if the identification is not introduced correctly at the third time, the program will finish. If the identification is correct the program will show:

1. Place an order.
2. Modify product.
3. Change employee password.
4. Log out.

To **modify a product** we have to take into account that one product has different attributes, so the employee could change the name, the price, the code and the units of product, the employee will have to select the attribute he/she wants to change. The program has a requirement here, it verify that we cannot have two products with the same code, but we can have two products with the same name.

To **change the employee password**, the program will ask the employee to introduce the new password, and the password will be changed.

Log out will ask if we want to use a different user, if we don't the program will finish.

To **place an order,** the employee will be asked to introduce the number of products he/she wants to order for a single order, after that the system will show:

1.1 Add product
1.2 Visualize total price
1.3 Print out invoice
1.4 Finish placing order

Add product, the program will show the available products and the employee has to introduce the name of the product to introduce it, and the quantity of this product.

Visualize total price, the final price will be printed.

Print out invoice, that option will show the code, the name and the price of the products we have ordered and the name of the employee that has attended the order.

Finish placing order, the system will return to the previous options.

Design of the problem

The UML has been added in the file in order to see it better, the UML contains the elements and the relationships.

The program starts in the main (Shop), from the main we can use the rest of classes through the use of salesmngnm and employemngmn.

We can see that we have multiplicity between the main and salesmngmnt (1 to *), and the main and employemenmng (1 to *).

Salesmngmnt is related with products, where we have multiplicity, it is possible to have infinite products. Products is related to order with the relation 1 to *, it is possible to have an order composed by several products, and we have perishable and non perishable that are child classes of product, with heritance relation, we find a relation 1 to 1 between non-perishable and offer, and offer has two child classes (OfferPercentage and OfferMult), where we have heritance relation .

Finally we can see employemenmng related with employees (1 to *), employees has two child classes (Employeedayshift and Employeenightshift), where we have heritance relation.

User Manual

When we run the program, the first thing we will have to do is to introduce our username and password, so we need to be targeted in Employee.txt.

```
WELCOME TO THE PROGRAM, PLEASE, INTRODUCE YOUR USERNAME:
```

If the username introduced is not correct we will obtain the following message:

```
Error with the username
```

If the username introduced is correct the program will ask for the password.

```
INTRODUCE YOUR PASSWORD:
```

If the password introduced is not correct the following message will appear:

```
Error with the password
```

The user has three tries to introduce a correct username and a correct password, if it has been introduced not correctly three times, the program will finish showing:

```
End of the program
```

If a correct username and password are introduced, the user will go to the main menu, with the different functionalities:

```
Welcome to our program, please select an option:
1.Place an order.
2.Modify product.
3.Change employee password
4.Log out.
```

We will have to select one option, if a wrong option is introduced the following dialog will appear:

```
Please introduce again a valid option, from 1 to 4
```

1. If we select **place an order**, the number of products to order will be asked, and we will have to introduce it:

```
Write the number of products you want to order
```

Once we have introduced the number of products we want, we could see the following menu:

```
Select an option:
1.1 Add product.
1.2 Visualize total price.
1.3 Print out invoice .
1.4 Finish placing order
```

- Selecting **Add product** we will see the list of products that are available, and we will have to write the name of the first product to order, the system will show the products we have available with their characteristics:

```
Perishable [days=4, code=1524, name=champu, price=12.3, units=7]
NonPerishable [off=11, code=1523, name=gel, price=12.8, units=8]
Write the name of the product
```

If we introduce an incorrect name the program will keep asking for a correct product name:

```
Please introduce a correct product name
```

After introduce a correct name the program will show us the next state:

```
Please introduce the number of units you want
```

Then the user has to introduce the number of units he/she wants of that product, if the user insert a higher quantity than the quantity we have available of that product, the program shows:

```
Error with the number of units you want ,introduce it again please
```

The program will keep showing that message until the user introduces a correct number for that product. After introduce a correct quantity the program will ask if we want to introduce another product.

```
Do you want to introduce another product? 1.Yes 2.No
```

Choosing 1, we will be able to introduce another product, choosing 2 we will be back to the menu of option 1. If we have selected n number of products and we have already added n number of products, if we select 1 we will go to the menu of option 1 as well.

- Selecting **Visualize total price,** we could see the total price of the products we have introduced before, and we will be back in the menu of the option 1.

```
The final price is: 86.10
Select an option:
1.1 Add product.
1.2 Visualize total price.
1.3 Print out invoice .
1.4 Finish placing order
```

(In that example we selected two champu products)

- If we select **print out invoice** we could see the characteristics of the products we have ordered previously, and we will be back in the menu of the option 1.

```
Perishable [days=4, code=1524, name=champu, price=12.3, units=7]
Select an option:
1.1 Add product.
1.2 Visualize total price.
1.3 Print out invoice .
1.4 Finish placing order
```

(In that example we selected two products, champu and gel)

- Finally selecting **finish placing order** we can go back to the main menu.

2. **Modify product,** after choosing that option the system will show four different options:

```
Select an option:
2.1.Modify name.
2.2.Modify price.
2.3.Modify code.
2.4.Modify Units.

2.5.Return
```

- The option **modify name,** will show the following statement:

```
Introduce the name of the product you want to change its name
```

Now we have to introduce the name of the product to change the name, if we introduce a name of a product we don't have in products.txt the system will keep asking for a correct name.

```
Introduce a correct name
```

When a correct name is introduced, we will be asked to introduce the new name of the product.

```
Introduce the new name of the product
```

Then, we just have to write the new name we want and the name will be changed, we will be back at the menu of option 2.

- Selecting **modify price,** the program will ask for the name of the product we want to change its price, and like in the option 'modify name' if we introduce an incorrect name the program will keep asking for a correct product name until we write a name we have in products.txt.

```
Introduce the name of the product you want to change its price
```

```
Introduce a correct name
```

After introduce a correct name we could select the new price for that product, the price will be changed and we will go back at the menu of option 2.

```
Introduce the new price of the product
```

- **Modify code,** that option will work like 'modify name' and 'modify price', the system will ask the name of the product, the program will keep asking for a correct name if we have introduced an incorrect one, and once we introduce a correct name we could change the code of the product.

```
Introduce the new code of the product
```

After introduce the new code we will go back at the menu of option 2.

- **Modify Units,** the system will ask the name of the product, the program will keep asking for a correct name if we have introduced an incorrect one, and once we introduce a correct name we could change the number of the products we have selected.
  ```
  Introduce the number of products
  ```

After introduce the number of products for the product we selected, we will go back at the menu of option 2.

- **Return,** by selecting this option we will go back to the main menu.

3. **Change employee password,** selecting that option the system will ask us to introduce the new password:

   ```
   Introduce the new password:
   ```

   Here we just have to introduce our new password, our password will be changed and we will go back to the main menu.

4. **Log out,** the program will show:

   ```
   Do you want to continue with other user?, true or false
   ```

   If we write true we will go back to the very beginning and we will start, we could run the program with a different username, if we select false the program will finish.

   ```
   End of the program
   ```