Name Surname: Ahmet Furkan Akdamar

Number: 190316068

Education Type: Evening Education

E-Commerce Application Project Report

I wrote the desired e-commerce application in accordance with the instructions below.

- There must be a User class that holds the username, name, surname, date of birth, password, email address, home and work addresses, products ordered, favorite products, and credit card objects belonging to the user. In this class, there should also be two methods that perform the user's product ordering and product favorites.
- There must be a CreditCard class that contains the credit card number, credit card user, security code and expiration date of the credit cards.
- There must be a Product class that holds the product name, product color, product category, product stock information, product weight, product description information. In addition, when a product is ordered by users, a method should be written that reduces the number of stocks as much as the number of products purchased and controls the stock number.
- For order transactions, there must be an Order class that holds the ordering user object, the ordered product object, and the credit card objects to which the payment is made. In addition, a method must be used here that accesses the method that controls the product stock information of the Product class and accesses the purchasing method of the User class.
- You are also expected to develop a test class that checks all the operations you do. (For adding users, adding credit cards, adding products, purchasing products, favoring products)

- Getter and setter methods of variables in all classes should be written..

I can explain the e-commerce application I wrote as follows:

users Class

In the Users class, I first wrote the required Attributes. I got the type of username, name, surname, date of birth, password, email address, home and work addresses attributes as String. I wrote an attributes named productCount that holds the number of products to be ordered in integer type. I wrote 3 Array Lists. I wrote a product type ArrayList named “productsOrdered” where the ordered products are kept. I wrote a product type ArrayList named “favoriteProducts” where favorite products are kept. I also wrote an ArrayList of creditCard type named “savedCards”, where the saved cards are kept.

```
private String userName;
private String name;
private String surName;
private String dateOfBirth;
private String password;
private String eMail;
private String homeAddress;
private String workAddress;
ArrayList <product> productsOrdered;
ArrayList <product> favoriteProducts;
ArrayList <creditCard> savedCards;
private int productCount; //The product count to be purchased
```

After, I wrote the Constructors of these Attributes.

```
public users (String userName, String name, String surName, String dateOfBirth,
              String password, String eMail, String homeAddress, String workAddress, int productCount){
    this.userName = userName;
    this.name = name;
    this.surName = surName;
    this.dateOfBirth = dateOfBirth;
    this.password = password;
    this.eMail = eMail;
    this.homeAddress = homeAddress;
    this.workAddress = workAddress;
    this.productsOrdered = new ArrayList<product>();
    this.favoriteProducts = new ArrayList<product>();
    this.savedCards = new ArrayList<creditCard>();
    this.productCount = productCount;
}
```

Then I wrote a method that prints the attributes in the User class.

```
public void writeAttributesUser(){
    System.out.println("\n" + "Name: " + this.name + "\n" + "Surname: " + this.surName + "\n" + "Username: " + this.userName+ "\n" +
            "Password: " + this.password + "\n" + "Email: " + this.eMail + "\n" + "Birthday: " + this.dateOfBirth + "\n" +
            "Home Address: " + this.homeAddress + "\n" + "Work Address: " + this.workAddress + "\n");
}
```

Then I wrote Getter and Setter methods.

```
public String getName() {

    return this.name;
}
public void setName(String name) {

    this.name = name;
}
```

Later I wrote a method that adds the card entered as a parameter to the saved cards array list. With this method, I save the selected card.

```
public void addCards (creditCard c){
    savedCards.add(c);
    System.out.println("\nThe card you selected has been added to Saved Cards.\n");
}
```

After I wrote a method that adds the product entered as a parameter to “favoriteProducts” array list. We can also call this method a kind of add to favorites button.

```
public void addProductFavorites (product p){
    favoriteProducts.add(p);
    System.out.println("\nThe product you selected has been added to your favorites.\n");

}
```

Finally, I wrote a method called "orderingProduct" that performs the ordering process by calling the controlStock method.

```
public void orderingProduct (product p, users u){
    System.out.println("\nThe total number of products to be purchased is " + productCount + ".\n");
    if (p.controlStock(u)) {
        System.out.println("We make your purchase...");
        for (int i = 0; i < productCount; i++) {
            productsOrdered.add(p);

        }
        System.out.println("Your purchase is complete. The number of products remaining in the seller is " + p.getProdStockInf() + ". Thank you for choosing us.\n");
    }
    else
        System.out.println("Please try again when the product you want to order is in stock.");

}
```

creditCard Class

Firstly, I wrote the desired attributes in the credit card class. I got the type of credit card number, credit card user and expiration date of the credit cards attributes as String. Since the number of security code is not a very large number, I took it in integer type.

```
private String creditCardNumber;
private String creditCardUser;
private int securityCode;
private String expirationDateOfCC; //Expiration Date Of The Credit Cards
```

After, I wrote the Constructors of these Attributes.

```
public creditCard(String creditCardNumber, String creditCardUser, int securityCode, String expirationDateOfCC){
    this.creditCardNumber = creditCardNumber;
    this.creditCardUser = creditCardUser;
    this.securityCode = securityCode;
    this.expirationDateOfCC = expirationDateOfCC;

}
```

Then I wrote a method that prints the attributes in the creditCard class.

```
public void writeAttributesCard(){
    System.out.println("\n" + "Credit Card Number: " + this.creditCardNumber + "\n" + "Credit Card User: " + this.creditCardUser + "\n" +
            "Security Code: " + this.securityCode + "\n" + "Expiration Date Of The Credit Cards: " + this.expirationDateOfCC + "\n" );

}
```

Finally, I wrote the Getter and Setter methods.

```
public String getCreditCardNumber() {
    return this.creditCardNumber;
}
```

```
public void setCreditCardNumber(String creditCardNumber) {
    this.creditCardNumber = creditCardNumber;
}
```

product Class

Firstly, I wrote the desired attributes in the product class. I got the type of product name, product color, product category and product description information attributes as String. Since the product stock information should be integer, I wrote it in integer type. Since the product weight can be a decimal number, I preferred the double type.

```
private String productName;
private String productColor;
private String productCategory;
private int prodStockInf;     // Product Stock Information
private double productWeight;
private String prodDescriptionInf;      // Product Description Information
```

After, I wrote the Constructors of these Attributes.

```
public product (String productName, String productColor, String productCategory, int prodStockInf, double productWeight,
                String prodDescriptionInf){
    this.productName = productName;
    this.productColor = productColor;
    this.productCategory = productCategory;
    this.prodStockInf = prodStockInf;
    this.productWeight = productWeight;
    this.prodDescriptionInf = prodDescriptionInf;
}
```

Then I wrote a method that prints the attributes in the product class.

```
public void writeAttributesProduct(){
    System.out.println("\n" + "Product Name: " + this.productName + "\n" + "Product Color: " + this.productColor + "\n" +
            "Product Category: " + this.productCategory + "\n" + "Product Stock Information: " + this.prodStockInf + "\n" +
            "Product Weight: " + this.productWeight + "\n" + "Product Description Information: " + this.prodDescriptionInf + "\n" );


}
```

Later, I wrote Getter and Setter methods.

```
public String getProductName(){

    return this.productName;
}
```

```
public void setProductName(String productName){

    this.productName = productName;
}
```

Finally, I wrote a method called controlStock. This method is a method that controls whether there are products in stock as much as the number of products to be ordered and reduces the stock. At the same time, this method complements the purchasing method.

```
public boolean controlStock (users u){
    if (this.prodStockInf >= u.getProductCount()) {
        System.out.println("The number of products you want to order is available in our stock.");
        this.prodStockInf = this.prodStockInf - u.getProductCount();
        return true;
    }
    else {
        System.out.println("The number of products you want to order is not available in our stocks. The number of products in our stock is " +
                this.prodStockInf + " pieces.");
        return false;
    }
}
```

order Class

First of all, I wrote the attributes of the desired object type in the Order class.

```
private users orderingUser;
private product orderedProduct;
private creditCard paymentCard; // The credit card objects to which the payment is made.
```

After, I wrote the Constructors of these Attributes.

```
public order (users orderingUser, product orderedProduct, creditCard paymentCard ){
    this.orderingUser = orderingUser;
    this.orderedProduct = orderedProduct;
    this.paymentCard = paymentCard;

}
```

Later, I wrote Getter and Setter methods

```
public users getOrderingUser() {

    return this.orderingUser;
}
```

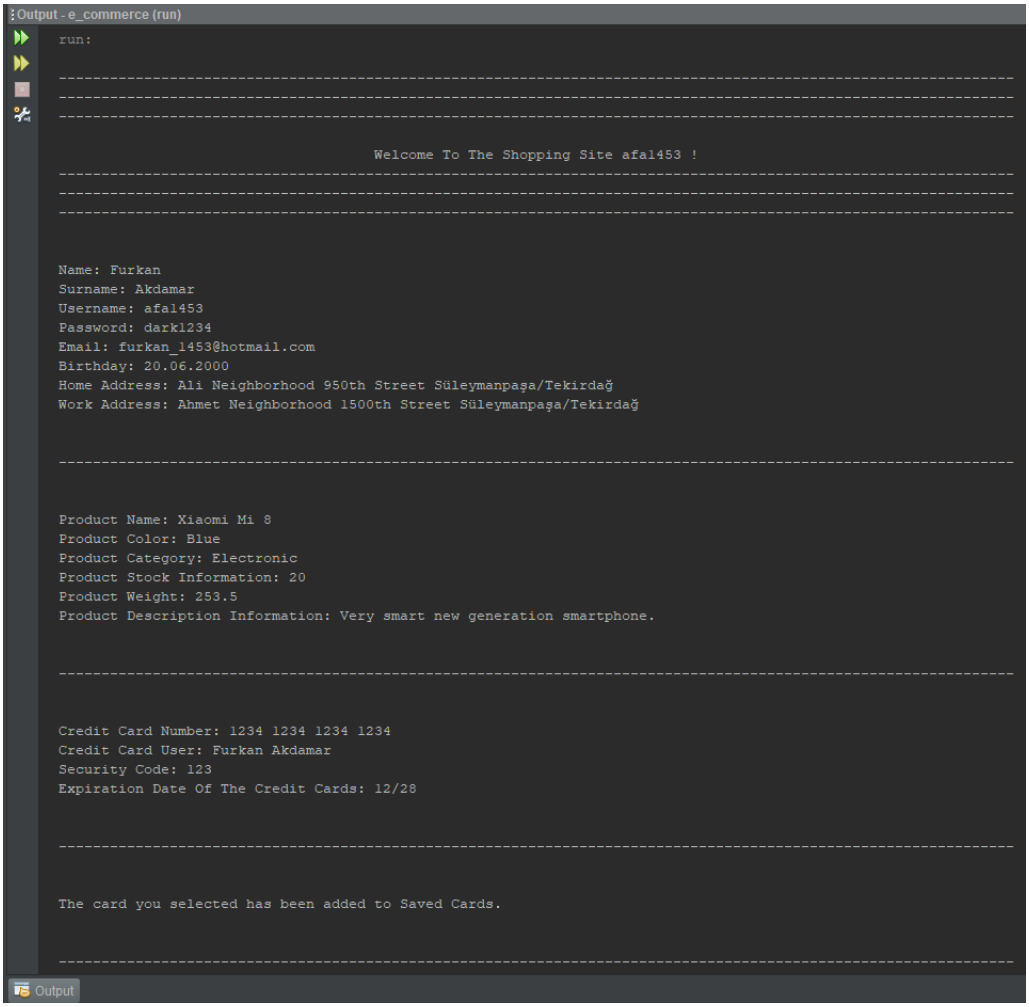
```java
public void setOrderingUser(users orderingUser) {

    this.orderingUser = orderingUser;
}
```

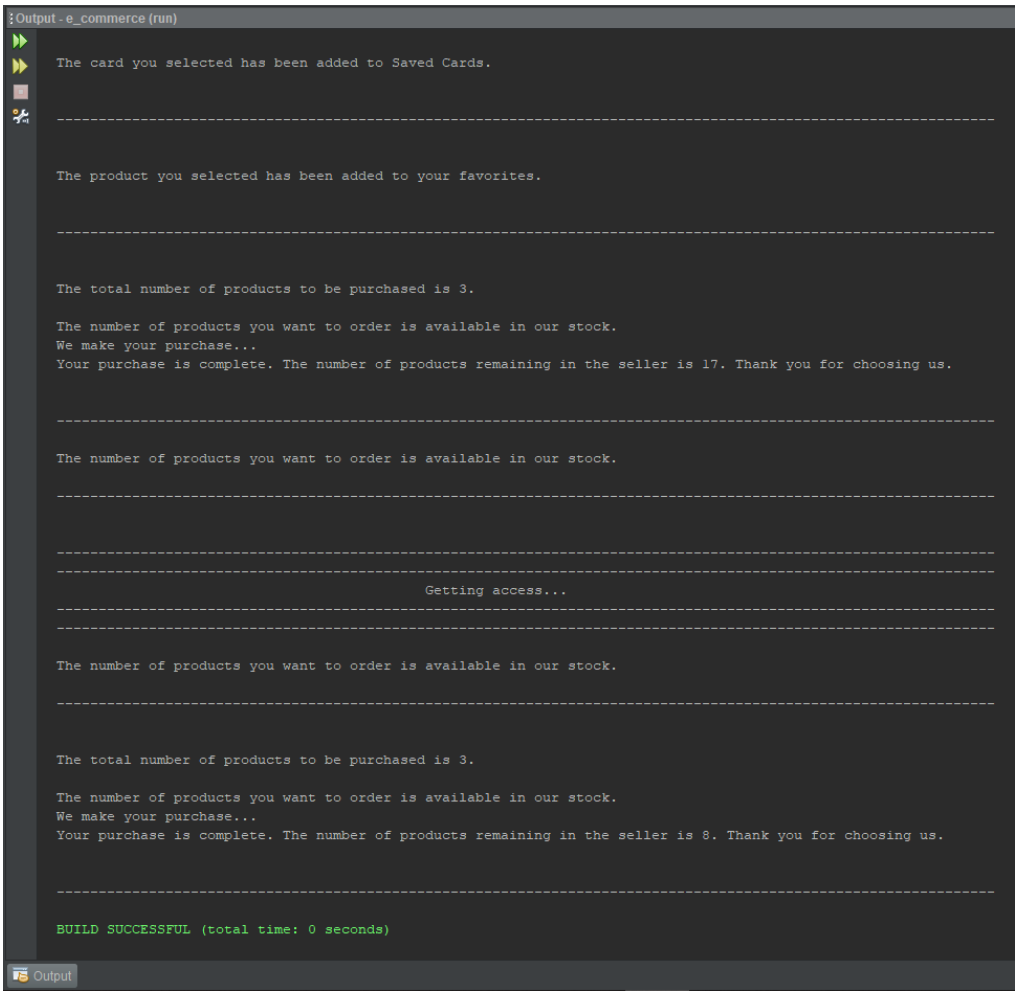
Finally, I wrote a method that accesses the method of checking the product stock information of the Product class and the purchasing method of the User class, as stated in the instructions. And I edited it to print.

```java
public void accessToMethods(users user1, product product1){
    System.out.println("\n-------------------------------------------------------------------------------------------------------------------");
    System.out.println("-------------------------------------------------------------------------------------------------------------------");
    System.out.println("                                                Getting access...");
    System.out.println("-------------------------------------------------------------------------------------------------------------------");
    System.out.println("-------------------------------------------------------------------------------------------------------------------\n");
    product1.controlStock(user1);
    System.out.println("\n-------------------------------------------------------------------------------------------------------------------\n");
    user1.orderingProduct(product1,user1);
}
```

test Class

I checked, edited and printed all the processes I did in the test class. Screenshots showing the outputs of the test class are as shown below:




Output - e_commerce (run)
The card you selected has been added to Saved Cards.
The product you selected has been added to your favorites.
The total number of products to be purchased is 3.
The number of products you want to order is available in our stock.
We make your purchase...
Your purchase is complete. The number of products remaining in the seller is 17. Thank you for choosing us.
The number of products you want to order is available in our stock.
Getting access...
The number of products you want to order is available in our stock.
The total number of products to be purchased is 3.
The number of products you want to order is available in our stock.
We make your purchase...
Your purchase is complete. The number of products remaining in the seller is 8. Thank you for choosing us.
BUILD SUCCESSFUL (total time: 0 seconds)
Output