



**MANISA CELAL BAYAR UNIVERSITY**  
**FACULTY OF ENGINEERING**  
**COMPUTER ENGINEERING DEPARTMENT**

**OBJECT ORIENTED PROGRAMMING**  
**2020-2021 SPRING SEMESTER**

**HOMEWORK REPORT**  
**(E-COMMERCE APPLICATION PROJECT)**

**DELIVERY DATE**

**27/05/2021**

**PREPARED BY**

190315071, Umut AZAZİ

Ingredients

- 1) CLASSIFICATION USING .....3
  - 1.a User Class.....3
    - 1.a.1 Codes .....3
    - 1.a.2 Explanation.....5
  - 1.b Product Class .....6
    - 1.b.1 Codes .....6
    - 1.b.2 Explanation .....7
  - 1.c CreditCard Class.....7
    - 1.c.1 Codes.....7
    - 1.c.2 Explanation.....8
  - 1.d Order Class.....9
    - 1.d.1 Codes .....9
    - 1.d.2 Explanation .....9
- 2)TESTS .....10
  - 2.1 Codes .....10
  - 2.2 Explanation .....11
  - 2.3 Screenshots.....11

# 1) CLASSIFICATION USING

Used NetBeans IDE 8.2.

## 1.a User Class

### 1.a.1 Codes

```
1
2 package oop.homework;
3
4 /* Umut AZAZI
5
6     190315071
7
8 */
9 import java.util.ArrayList;
10
11
12
13 public class User {
14
15     private String username;
16     private String name;
17     private String surname;
18     private String birthday;
19     private String password;
20     private String email;
21     private String homeAddress;
22     private String workAddress;
23     private ArrayList<Order> productOrdered = new ArrayList<Order>();
24     private ArrayList<Product> favProduct = new ArrayList<Product>();
25     private ArrayList<CreditCard> creditCard = new ArrayList<CreditCard>();
26
27 //Constructors
28
29     public User(String username, String name, String surname, String birthday, String password, String email, String
homeAddress, String workAddress, ArrayList<CreditCard> creditCard) {
30         this.username = username;
31         this.name = name;
32         this.surname = surname;
33         this.birthday = birthday;
34         this.password = password;
35         this.email = email;
36         this.homeAddress = homeAddress;
37         this.workAddress = workAddress;
38         this.creditCard = creditCard;
39     }
40     public User(String username, String name, String surname, String birthday, String password, String email, String
homeAddress, String workAddress, ArrayList<CreditCard> creditCard, ArrayList<Order> productOrdered,
ArrayList<Product> favProduct) {
41         this.username = username;
42         this.name = name;
43         this.surname = surname;
44         this.birthday = birthday;
45         this.password = password;
46         this.email = email;
47         this.homeAddress = homeAddress;
48         this.workAddress = workAddress;
49         this.creditCard = creditCard;
50         this.productOrdered = productOrdered;
51         this.favProduct = favProduct;
52     }
```

```

53
54 public User(String username, String name, String surname, String birthday, String password, String email, String
homeAddress, String workAddress, CreditCard creditCard)
55 {
56     this.username = username;
57     this.name = name;
58     this.surname = surname;
59     this.birthday = birthday;
60     this.password = password;
61     this.email = email;
62     this.homeAddress = homeAddress;
63     this.workAddress = workAddress;
64     this.creditCard.add(creditCard);
65 }
66
67
68
69 //Methods
70 public void orderProduct(Product product, CreditCard creditCard, int amountOfProduct ){
71     Order order = new Order(this,creditCard,product,mountOfProduct);
72     if(order.stockInfo()) {
73         this.productOrdered.add(order);
74         product.reduceStock(amountOfProduct);
75     }
76     else{
77
78         System.out.println("Your amount is under stockamount please try again ! Stock of your product is :
"+product.getProductStockInfo());
79     }
80
81 }
82 public void addToFav(Product product){
83     this.favProduct.add(product);
84     System.out.println(product.getProductName()+ " is added to favorites");
85 }
86
87 //Getter and Setters
88 public String getUsername() {
89     return username;
90 }
91
92 public void setUsername(String username) {
93     this.username = username;
94 }
95
96 public String getName() {
97     return name;
98 }
99
100 public void setName(String name) {
101     this.name = name;
102 }
103
104 public String getSurname() {
105     return surname;
106 }
107
108 public void setSurname(String surname) {
109     this.surname = surname;
110 }
111
112 public String getBirthday() {
113     return birthday;
114 }
115
116 public void setBirthday(String birthday) {

```

```

117     this.birthday = birthday;
118 }
119
120 public String getPassword() {
121     return password;
122 }
123
124 public void setPassword(String password) {
125     this.password = password;
126 }
127
128 public String getEmail() {
129     return email;
130 }
131
132 public void setEmail(String email) {
133     this.email = email;
134 }
135
136 public String getHomeAddress() {
137     return homeAddress;
138 }
139
140 public void setHomeAddress(String homeAddress) {
141     this.homeAddress = homeAddress;
142 }
143
144 public String getWorkAddress() {
145     return workAddress;
146 }
147
148 public void setWorkAddress(String workAddress) {
149     this.workAddress = workAddress;
150 }
151 }
152

```

### 1.a.2 Explanation

In this class, variables that keep information such as the user's contact, order and favorite product are defined.

Three constructors have been created. One of these was used to add the entered credit card to the previously specified credit card list.

The orderProduct and addToFav methods were created for the desired order operation and add favorites operation. The orderProduct method take parameters from Product and CreditCard classes for Order object. It's basically check stock info with help of stockInfo method from Order Class and add order to productOrdered. Also, reduce the amount with reduceStock method. If the the stock info is under that order amount it prints error text.

The addToFav method is basically add to favorites that product.

Getter's and setter's created to reach private variables.

## 1.b Product Class

### 1.b.1 Codes

```
1 package oop.homework;
2
3 /* Umut AZAZİ
4
5     190315071
6
7 */
8 public class Product {
9
10     private String productName;
11     private String productColor;
12     private String productCategory;
13     private int productStockInfo;
14     private int productWeight;
15     private String productDescriptionInfo;
16
17     //Constructors
18     public Product(String productName, String productColor, String productCategory, int productStockInfo, int
productWeight, String productDescriptionInfo) {
19         this.productName = productName;
20         this.productColor = productColor;
21         this.productCategory = productCategory;
22         this.productStockInfo = productStockInfo;
23         this.productWeight = productWeight;
24         this.productDescriptionInfo = productDescriptionInfo;
25     }
26
27     //Methods
28     public void reduceStock(int amountOfProduct) {
29
30         this.productStockInfo -= amountOfProduct;
31         System.out.println("From " + this.productName + "\t" + "is" + " " + this.productStockInfo + " Left");
32
33     }
34
35     //Getter and Setter
36     public String getProductName() {
37         return productName;
38     }
39
40     public void setProductName(String productName) {
41         this.productName = productName;
42     }
43
44     public String getProductColor() {
45         return productColor;
46     }
47
48     public void setProductColor(String productColor) {
49         this.productColor = productColor;
50     }
51
52     public String getProductCategory() {
53         return productCategory;
54     }
55
56     public void setProductCategory(String productCategory) {
57         this.productCategory = productCategory;
58     }
59 }
```

```

59
60 public int getProductStockInfo() {
61     return productStockInfo;
62 }
63
64 public void setProductStockInfo(int productStockInfo) {
65     this.productStockInfo = productStockInfo;
66 }
67
68 public int getProductWeight() {
69     return productWeight;
70 }
71
72 public void setProductWeight(int productWeight) {
73     this.productWeight = productWeight;
74 }
75
76 public String getProductDescriptionInfo() {
77     return productDescriptionInfo;
78 }
79
80 public void setProductDescriptionInfo(String productDescriptionInfo) {
81     this.productDescriptionInfo = productDescriptionInfo;
82 }
83 }
84

```

## 1.b.2 Explanation

In this class, variables that keep information such as the product's name, product stock info , color and category are defined.

Constructor have been created. Takes previously defined variables as parameters.(productName, productColor etc.)

The reduceStock method is takes integer as parameter. This method basicly reduce amount of product that holds in amountOfProduct and prints how many left after order that product.

Getter's and setter's created to reach private variables.

## 1.c CreditCard Class

### 1.c.1 Codes

```

1 package oop.homework;
2
3 /* Umut AZAZI
4
5     190315071
6
7 */
8 public class CreditCard {
9
10     private int cardNumber;

```

```

11 private String cardUser;
12 private int securityCode;
13 private String exDate;
14
15 //Constructors
16 public CreditCard(int cardNumber, String cardUser, int securityCode, String exDate) {
17     this.cardNumber = cardNumber;
18     this.cardUser = cardUser;
19     this.securityCode = securityCode;
20     this.exDate = exDate;
21 }
22
23 //Getter and Setters
24 public int getCardNumber() {
25     return cardNumber;
26 }
27
28 public void setCardNumber(int cardNumber) {
29     this.cardNumber = cardNumber;
30 }
31
32 public String getCardUser() {
33     return cardUser;
34 }
35
36 public void setCardUser(String cardUser) {
37     this.cardUser = cardUser;
38 }
39
40 public int getSecurityCode() {
41     return securityCode;
42 }
43
44 public void setSecurityCode(int securityCode) {
45     this.securityCode = securityCode;
46 }
47
48 public String getExDate() {
49     return exDate;
50 }
51
52 public void setExDate(String exDate) {
53     this.exDate = exDate;
54 }
55 }
56

```

### 1.c.2 Explanation

In this class, variables that keep information such as card number, card user, security code and expiration date are defined.

Constructor have been created. Takes previously defined variables as parameters.(cardUser, cardNumber etc.)

Getter's and setter's created to reach private variables.



## 1.d Order Class

### 1.d.1 Codes

```
1 package oop.homework;
2
3 /* Umut AZAZI
4
5     190315071
6
7 */
8 public class Order {
9
10     private User userForOrdering;
11     private CreditCard creditCardForOrdering;
12     private Product productForOrdering;
13     private int amountForOrdering;
14
15     //Constructors
16     Order(User orderingUser, CreditCard orderingCreditCard, Product orderingProduct, int amountOfOrder) {
17         this.userForOrdering = orderingUser;
18         this.creditCardForOrdering = orderingCreditCard;
19         this.productForOrdering = orderingProduct;
20         this.amountForOrdering = amountOfOrder;
21     }
22
23     //Getter and Setters
24     public Product getProductForOrdering() {
25
26         return this.productForOrdering;
27     }
28     //Methods
29
30     public boolean stockInfo() {
31         int remainingStock = productForOrdering.getProductStockInfo() - amountForOrdering;
32         return (remainingStock >= 0);
33     }
34 }
35
```

### 1.d.2 Explanation

In this class, variables that keep information such as orderer user, card that used to order, product that ordered and amount of product that ordered are defined. Some variables have specific types as User, CreditCard and Product.

Constructor have been created. Takes previously defined variables as parameters.(userForOrdering, productForOrdering etc.)

The stockInfo method basically subtract the amount of product that ordered from the stock of product. After that, checks that is it greater and equals than zero and returns as boolean.

Getter's created to reach private variables.

## 2) TESTS

Used NetBeans IDE 8.2.

### 2.1 Codes

```
1 package oop.homework;
2
3 public class Test {
4
5     /* Umut AZAZI
6
7     190315071
8
9     */
10    public static void main(String[] args) {
11        //CreditCard Objects
12        CreditCard cc1 = new CreditCard(3456000, "Umut", 314, "12.02.23");
13        CreditCard cc2 = new CreditCard(2345000, "John", 456, "03.02.25");
14        CreditCard cc3 = new CreditCard(1234566, "Umut", 543, "12.02.23");
15
16        //User Objects
17        User user1 = new User("umtazazi", "Umut", "Azazi", "19.02.2000", "12345uuu", "umutazazi3@gmail.com",
"Erzene mah..", "Erzene mah", cc1);
18        User user2 = new User("johnwick", "John", "Anderson", "20.09.2002", "whokilledmydog1243",
"wickjohn@gmail.com", "Usa NYC", "NYC Street 15", cc2);
19
20        //Product Objects
21        Product product1 = new Product("Nvidia 2080Ti", "Grey", "Graphic Card", 30, 1000, "New Factory");
22        Product product2 = new Product("Rampage Screen", "Black", "IPS Screen", 20, 1500, "New Factory");
23        Product product3 = new Product("James Donkey SSD", "Yellow", "SSD", 40, 100, "New Factory");
24        Product product4 = new Product("Intel Core i7", "Black", "CPU", 200, 50, "New Factory");
25
26        Product product5 = new Product("7.62", "Gold", "Bullet", 500, 50, "New Factory");
27        Product product6 = new Product("Deagle", "Silver", "Gun", 30, 1300, "New Factory");
28
29        //Using orderProduct method
30        user1.orderProduct(product1, cc3, 1);
31        user1.orderProduct(product2, cc1, 1);
32        user1.orderProduct(product3, cc1, 3);
33        user1.orderProduct(product4, cc3, 3);
34
35        user2.orderProduct(product5, cc2, 1000);
36        user2.orderProduct(product6, cc2, 25);
37
38        //Using addToFav method
39        user2.addToFav(product5);
40
41
42
43
44    }
45
46 }
47
```

## 2.2 Explanation

In this class, made some scenarios to test methods and objects. Firstly, created some CreditCard Objects to use in User Objects. Then created User Objects. There are two users Umut and John.

After that, created products to order. Orders have created realistic. Finally, methods were used to test it. As you can see, the product stock decreases after the product order is placed. If you order more than stock quantity, you will get an error message. Also, you can see the product added to favorites. Umut is ordering some computer parts. He order successfully. John wanted to order some bullets but his amount is much than stock amount so he get error message. After that, he add to favorites this product to order next days again . You can check Screenshots for these tests.

## 2.3 Screenshots

### Screenshot about Umut's process

```
From Nvidia 2080Ti    is 29 Left
From Rampage Screen  is 19 Left
From James Donkey SSD is 37 Left
From Intel Core i7   is 197 Left
```

### Screenshot about John's process

```
Your amount is under stock amount please try again ! Stock of your product is : 500
From Deagle    is 5 Left
7.62 is added to favorites
```

