

Project one:

• الاسم: مي محمد زكي عبداللطيف

• Id:23011574

• Group:Monday 10.30

• Under Supervision of Eng/Sara Said

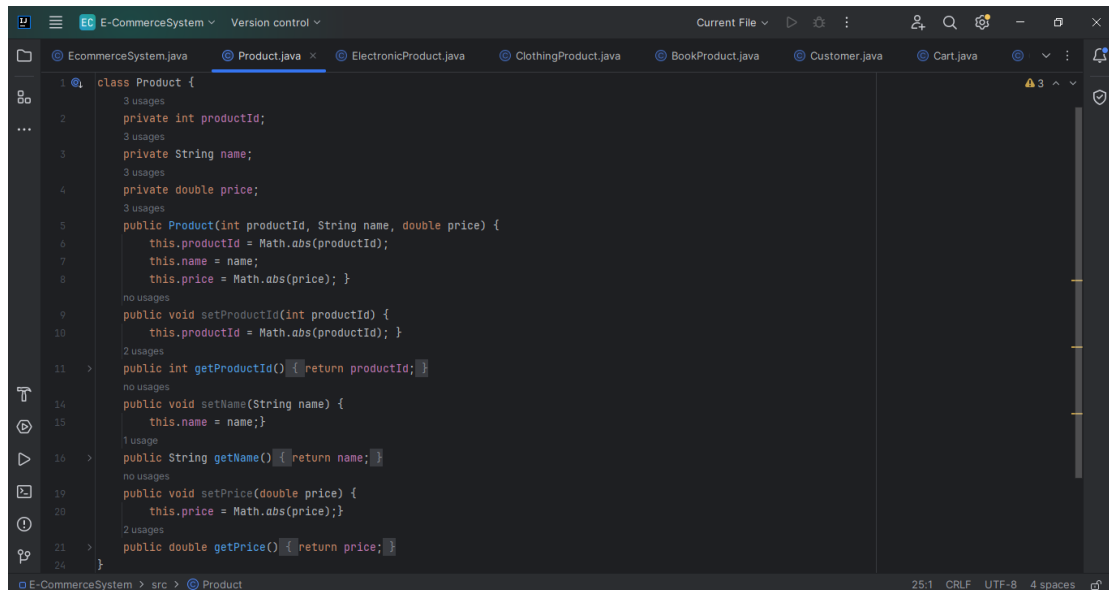
Code:

The Main:

```
1 import javax.swing.JOptionPane;
2 public class EcommerceSystem {
3     public static void main(String[] args) {
4         JOptionPane.showMessageDialog(parentComponent: null, message: "Welcome to the E-Commerce System!");
5         int customerId = Integer.parseInt(JOptionPane.showInputDialog("Please enter your ID:"));
6         String name = JOptionPane.showInputDialog("Please enter your name:");
7         String address = JOptionPane.showInputDialog("Please enter your address:");
8         Customer customer1 = new Customer(customerId, name, address);
9         int NumOfProducts = Integer.parseInt(JOptionPane.showInputDialog("How many products would you like to add to your cart?"));
10        Cart cart1 = new Cart(customerId, NumOfProducts);
11        for (int i = 0; i < NumOfProducts; i++) {
12            int choice = Integer.parseInt(JOptionPane.showInputDialog("Which product would you like to add? 1- Smartphone 2- T-Shirt 3- OOP"));
13            Product product;
14            if (choice == 1) {
15                product = new ElectronicProduct(name: "Smartphone", productid: 1, price: 599.99, brand: "Samsung", warrantyPeriod: 1);
16            } else if (choice == 2) {
17                product = new ClothingProduct(name: "T-shirt", productid: 2, price: 19.99, size: "Medium", fabric: "Cotton");
18            } else if (choice == 3) {
19                product = new BookProduct(name: "OOP", productid: 3, price: 39.99, author: "O'Reilly", publisher: "X Publications");
20            } else {
21                JOptionPane.showMessageDialog(parentComponent: null, message: "Invalid choice, Please try again.");
22                i--;
23                continue;
24            }
25            cart1.addProduct(product);
26        }
27        double totalPrice = cart1.calculatePrice();
```

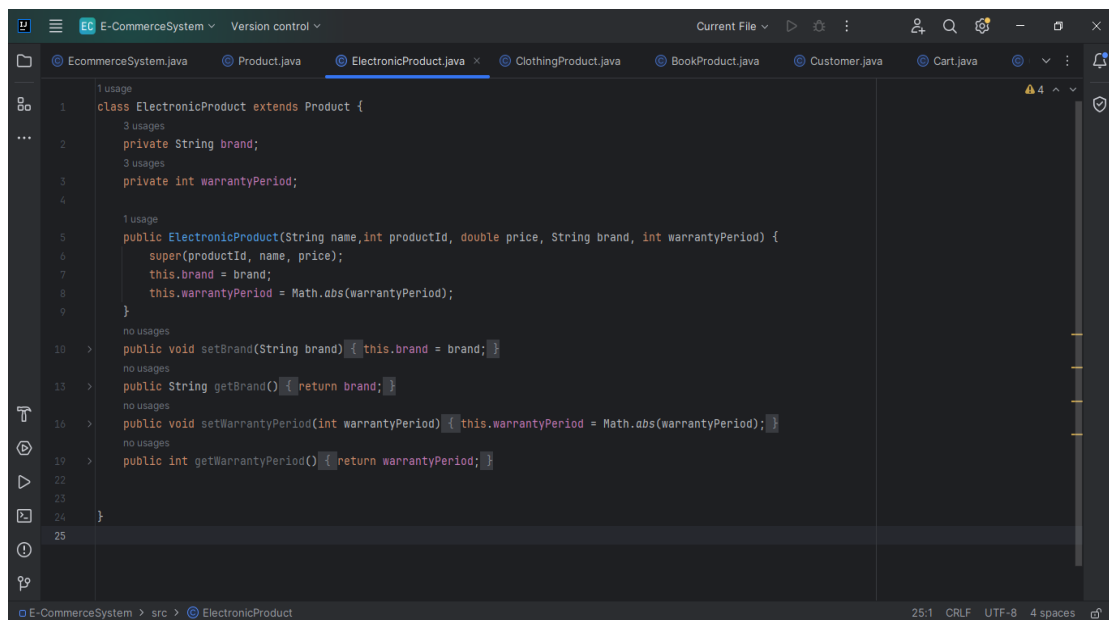
```
19        } else if (choice == 3) {
20            product = new BookProduct(name: "OOP", productid: 3, price: 39.99, author: "O'Reilly", publisher: "X Publications");
21        } else {
22            JOptionPane.showMessageDialog(parentComponent: null, message: "Invalid choice, Please try again.");
23            i--;
24            continue;
25        }
26        cart1.addProduct(product);
27    }
28    double totalPrice = cart1.calculatePrice();
29    int orderChoice = Integer.parseInt(JOptionPane.showInputDialog("Your total is $" + totalPrice + ", Would you like to place an order?"));
30    if (orderChoice == 1) {
31        cart1.placeOrder();
32    }
33    else {
34        JOptionPane.showMessageDialog(parentComponent: null, message: "Your order placement request has been cancelled.");
35    }
36    JOptionPane.showMessageDialog(parentComponent: null, message: "Thank you, " + customer1.getName() + ", for registering with us!");
37 }
38 }
39 }
40 }
```

Product Class:



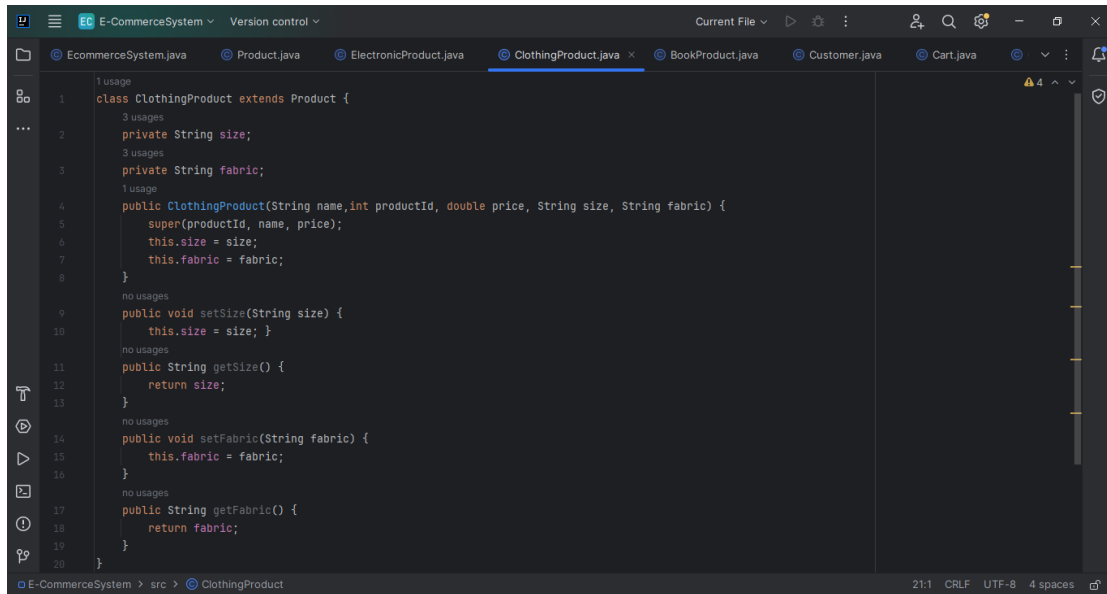
```
1 class Product {
2     private int productId;
3     private String name;
4     private double price;
5     public Product(int productId, String name, double price) {
6         this.productId = Math.abs(productId);
7         this.name = name;
8         this.price = Math.abs(price);
9     }
10    public void setProductId(int productId) {
11        this.productId = Math.abs(productId);
12    }
13    public int getProductId() { return productId; }
14    public void setName(String name) {
15        this.name = name;
16    }
17    public String getName() { return name; }
18    public void setPrice(double price) {
19        this.price = Math.abs(price);
20    }
21    public double getPrice() { return price; }
22 }
```

ElectronicProduct Class:



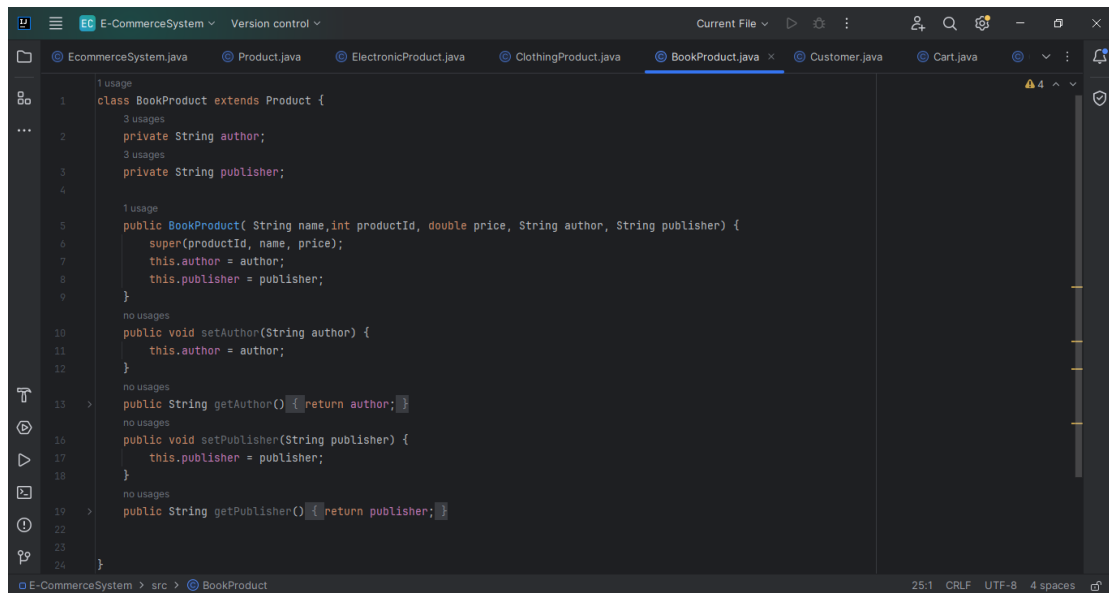
```
1 class ElectronicProduct extends Product {
2     private String brand;
3     private int warrantyPeriod;
4
5     public ElectronicProduct(String name, int productId, double price, String brand, int warrantyPeriod) {
6         super(productId, name, price);
7         this.brand = brand;
8         this.warrantyPeriod = Math.abs(warrantyPeriod);
9     }
10    public void setBrand(String brand) { this.brand = brand; }
11    public String getBrand() { return brand; }
12    public void setWarrantyPeriod(int warrantyPeriod) { this.warrantyPeriod = Math.abs(warrantyPeriod); }
13    public int getWarrantyPeriod() { return warrantyPeriod; }
14 }
```

ClothingProduct Class:



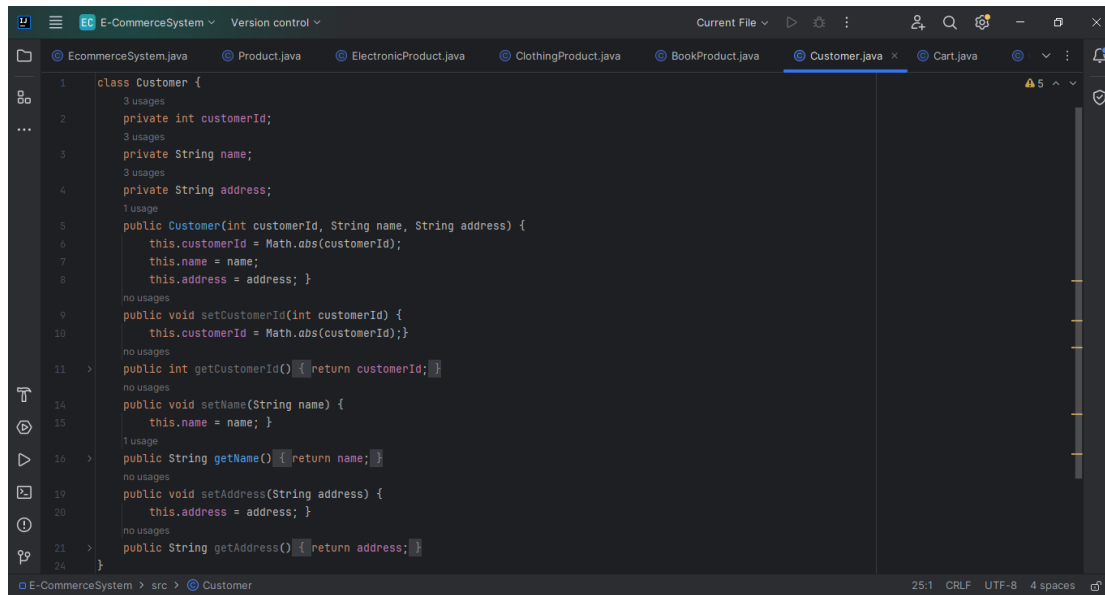
```
1 1 usage
2 class ClothingProduct extends Product {
3     3 usages
4     private String size;
5     3 usages
6     private String fabric;
7     1 usage
8     public ClothingProduct(String name,int productId, double price, String size, String fabric) {
9         super(productId, name, price);
10        this.size = size;
11        this.fabric = fabric;
12    }
13
14    no usages
15    public void setSize(String size) {
16        this.size = size; }
17
18    no usages
19    public String getSize() {
20        return size;
21    }
22
23    no usages
24    public void setFabric(String fabric) {
25        this.fabric = fabric;
26    }
27
28    no usages
29    public String getFabric() {
30        return fabric;
31    }
32 }
```

BookProduct Class:



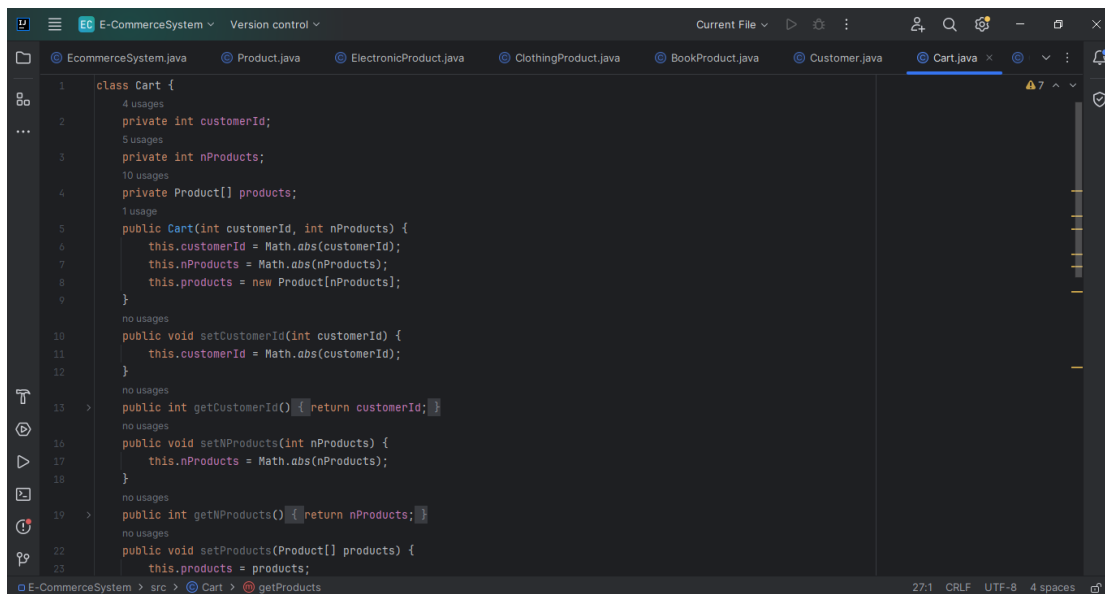
```
1 1 usage
2 class BookProduct extends Product {
3     3 usages
4     private String author;
5     3 usages
6     private String publisher;
7
8     1 usage
9     public BookProduct( String name,int productId, double price, String author, String publisher) {
10        super(productId, name, price);
11        this.author = author;
12        this.publisher = publisher;
13    }
14
15    no usages
16    public void setAuthor(String author) {
17        this.author = author;
18    }
19
20    no usages
21    public String getAuthor() {return author;}
22
23    no usages
24    public void setPublisher(String publisher) {
25        this.publisher = publisher;
26    }
27
28    no usages
29    public String getPublisher() {return publisher;}
30 }
```

Customer Class:



```
1 class Customer {
2     private int customerId;
3     private String name;
4     private String address;
5     public Customer(int customerId, String name, String address) {
6         this.customerId = Math.abs(customerId);
7         this.name = name;
8         this.address = address;
9     }
10    public void setCustomerId(int customerId) {
11        this.customerId = Math.abs(customerId);
12    }
13    public int getCustomerId() { return customerId; }
14    public void setName(String name) {
15        this.name = name;
16    }
17    public String getName() { return name; }
18    public void setAddress(String address) {
19        this.address = address;
20    }
21    public String getAddress() { return address; }
22 }
```

Cart Class:



```
1 class Cart {
2     private int customerId;
3     private int nProducts;
4     private Product[] products;
5     public Cart(int customerId, int nProducts) {
6         this.customerId = Math.abs(customerId);
7         this.nProducts = Math.abs(nProducts);
8         this.products = new Product[nProducts];
9     }
10    public void setCustomerId(int customerId) {
11        this.customerId = Math.abs(customerId);
12    }
13    public int getCustomerId() { return customerId; }
14    public void setNProducts(int nProducts) {
15        this.nProducts = Math.abs(nProducts);
16    }
17    public int getNProducts() { return nProducts; }
18    public void setProducts(Product[] products) {
19        this.products = products;
20    }
21 }
```

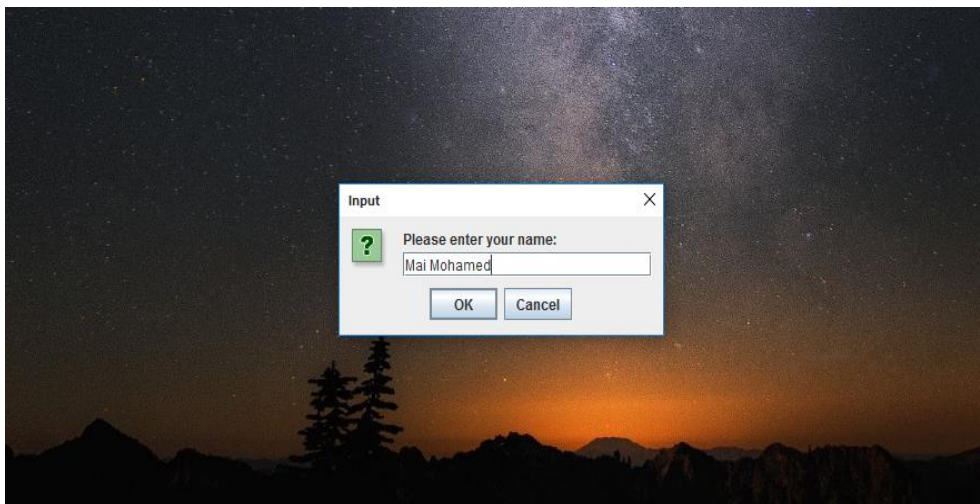
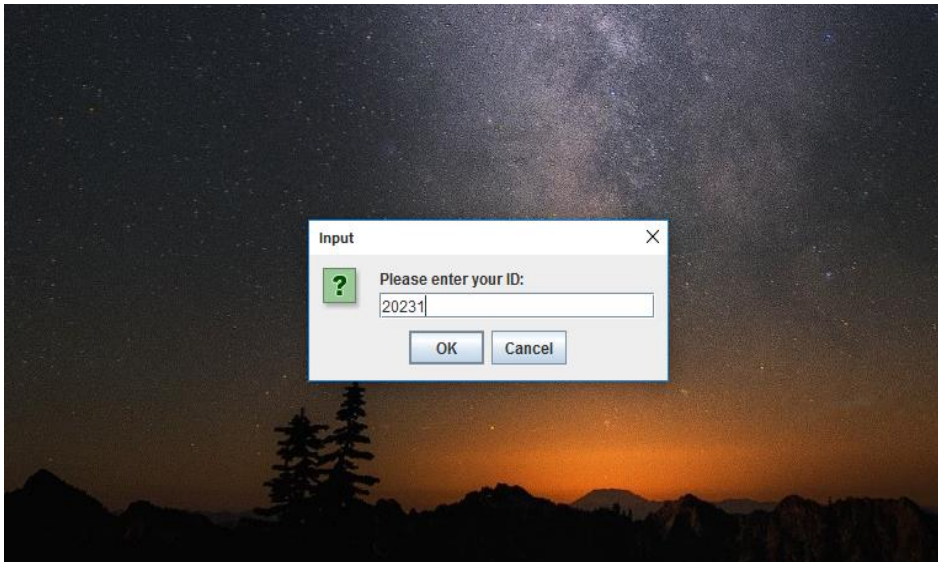
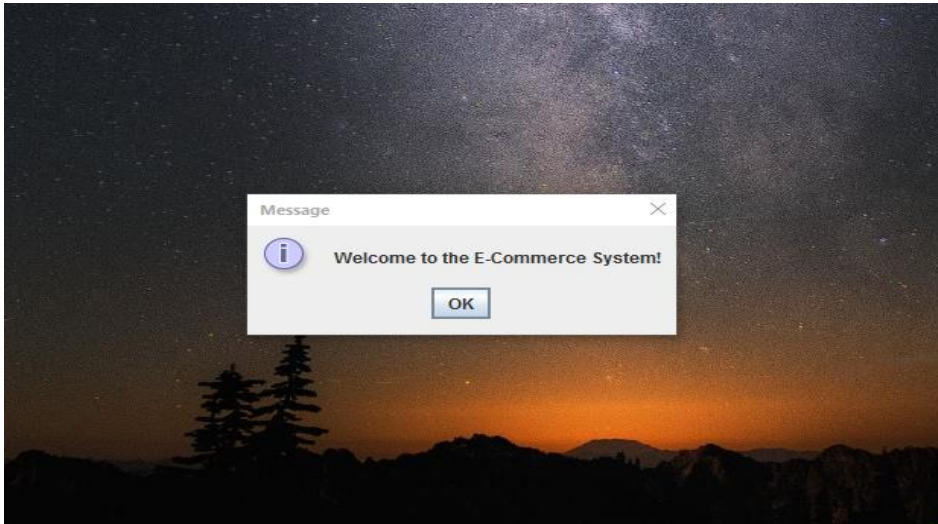
```
23     this.products = products;
24 }
25 no usages
26 public Product[] getProducts() {
27     return products;
28 }
29 1 usage
30 public void addProduct(Product product) {
31     for (int i = 0; i < nProducts; i++) {
32         if (products[i] == null) {
33             products[i] = product;
34             break; }
35     }
36 }
37 no usages
38 public void removeProduct(Product product) {
39     for (int i = 0; i < nProducts; i++) {
40         if (products[i] != null && products[i].getProductId() == product.getProductId()) {
41             products[i] = null;
42             break;
43         }
44     }
45 }
46 2 usages
47 public double calculatePrice() {
48     double totalPrice = 0.0;
49     for (Product product : products) {
50         if (product != null) {
```

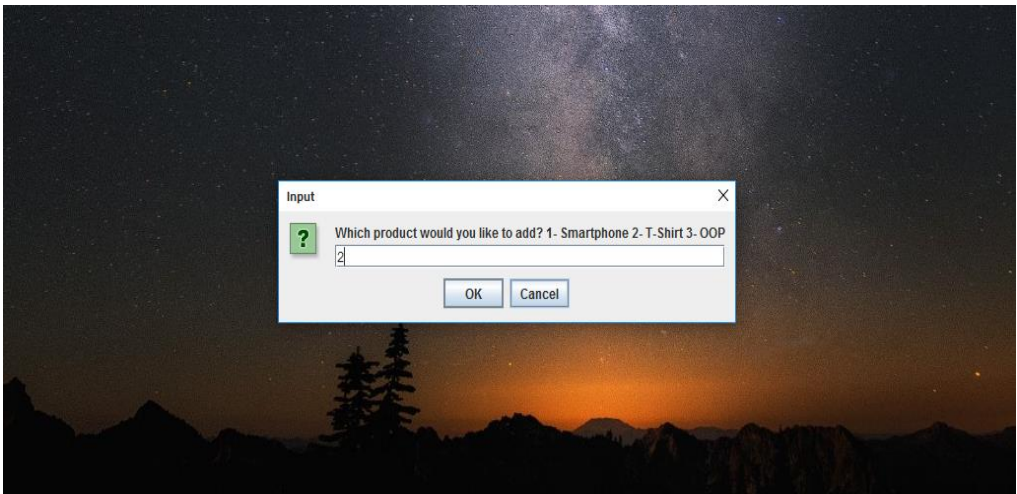
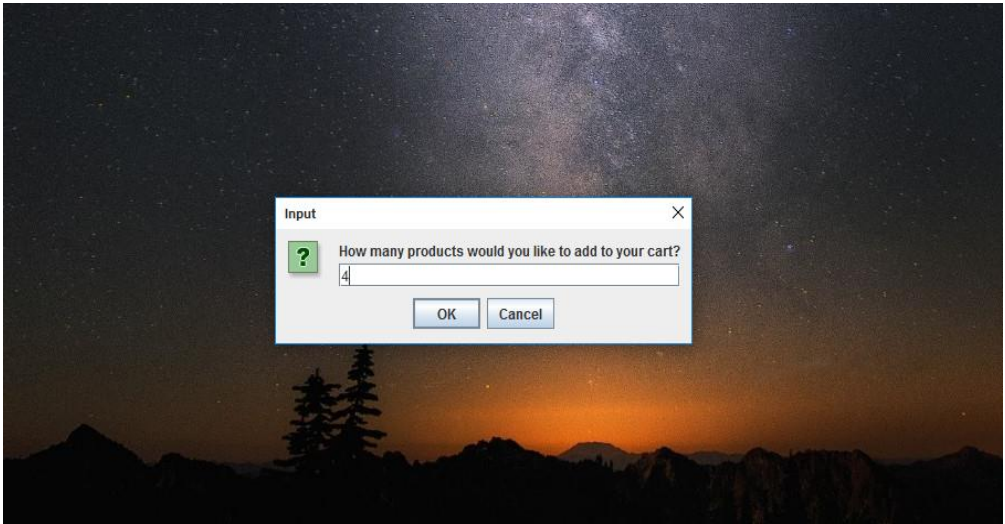
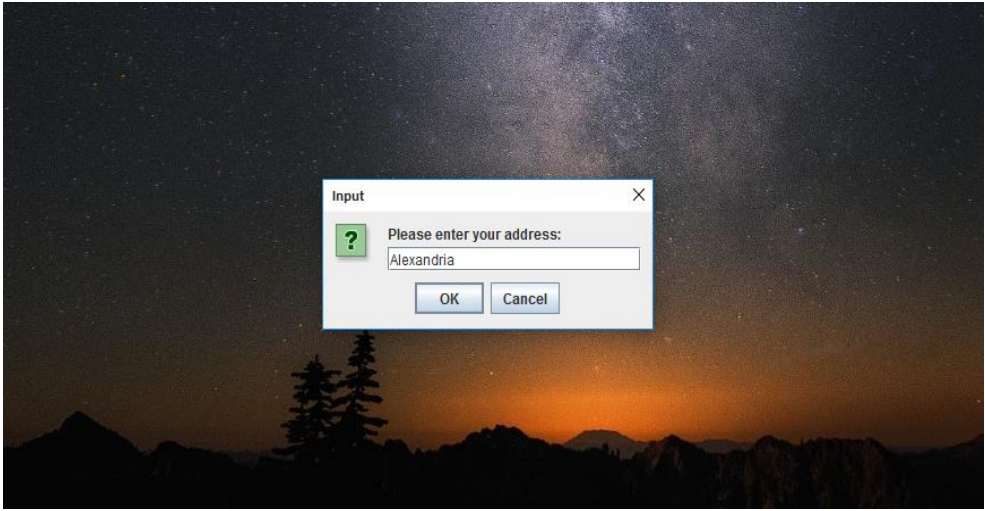
```
23     this.products = products;
24 }
25 no usages
26 public Product[] getProducts() {
27     return products;
28 }
29 1 usage
30 public void addProduct(Product product) {
31     for (int i = 0; i < nProducts; i++) {
32         if (products[i] == null) {
33             products[i] = product;
34             break; }
35     }
36 }
37 no usages
38 public void removeProduct(Product product) {
39     for (int i = 0; i < nProducts; i++) {
40         if (products[i] != null && products[i].getProductId() == product.getProductId()) {
41             products[i] = null;
42             break;
43         }
44     }
45 }
46 2 usages
47 public double calculatePrice() {
48     double totalPrice = 0.0;
49     for (Product product : products) {
50         if (product != null) {
```

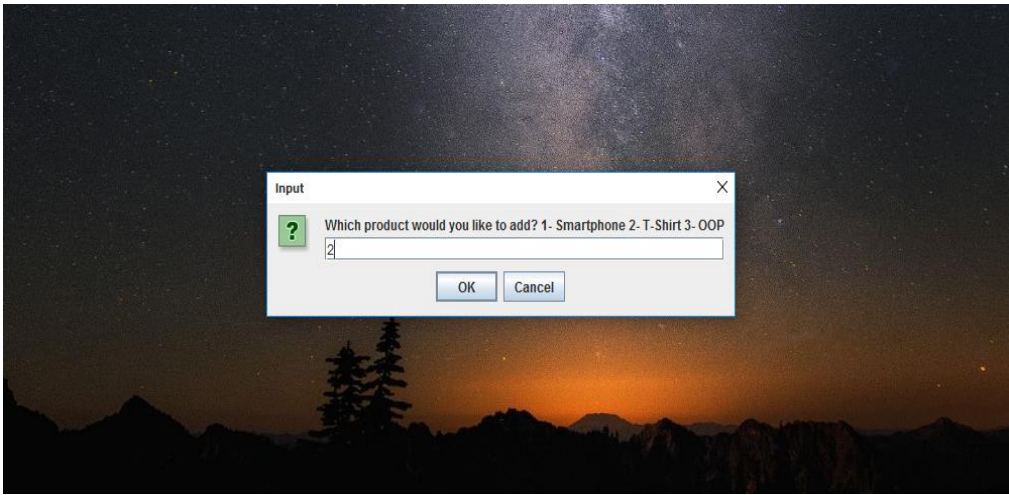
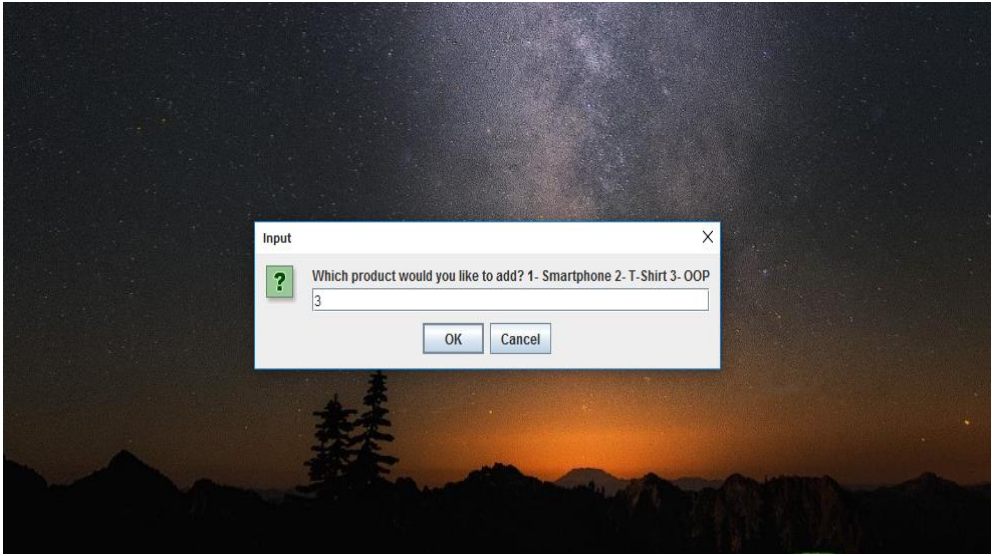
Order Class:

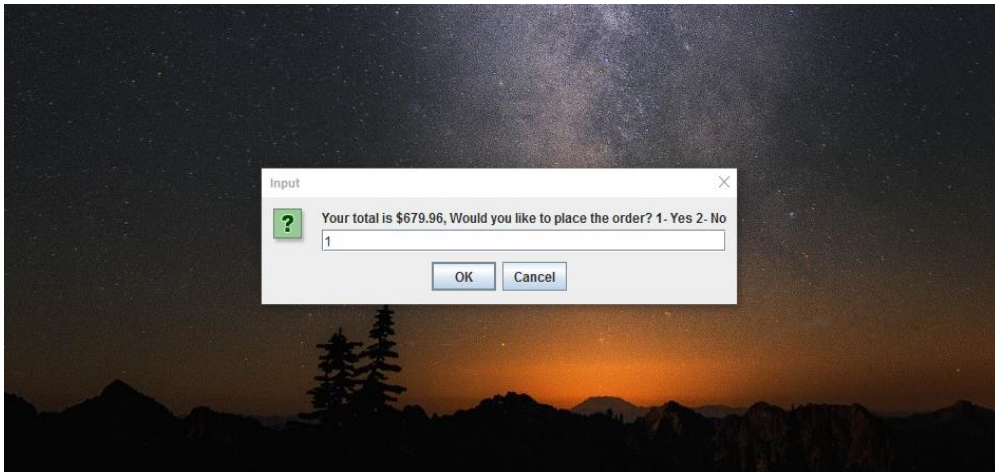
```
1 import javax.swing.*;
2 2 usages
3 class Order {
4     2 usages
5     int customerId;
6     2 usages
7     int orderId;
8     4 usages
9     Product[] products;
10    2 usages
11    double totalPrice;
12    1 usage
13    public Order(int customerId, int orderId, Product[] products, double totalPrice) {
14        this.customerId = Math.abs(customerId);
15        this.orderId = Math.abs(orderId);
16        this.products = products;
17        this.totalPrice = Math.abs(totalPrice); }
18    1 usage
19    public void printOrderInfo() {
20        String message = "Here's your order summary:\n" +
21            "Order ID: " + orderId + "\n" +
22            "Customer ID: " + customerId + "\n" +
23            "Products:\n";
24        for (int i = 0; i < products.length; i++) {
25            message += products[i].getName() + " - $" + products[i].getPrice() + "\n"; }
26        message += "\nTotal Price: $" + totalPrice;
27        JOptionPane.showMessageDialog(parentComponent, null, message);
28    }
29 }
```

The output:









Final Result:

