



Product class

Start Page × Product.java × ElectronicProduct.java × ClothingProduct.java × BookProduct.java × Customer.java × Cart.java × Order.java × EcommerceSystem.java × EcommerceSystemGUI.java ×

Source History

1 // ElectronicProduct class, subclass of Product

2 class **ElectronicProduct** extends Product {

3 private String brand;

4 private int warrantyPeriod;

5

6 public **ElectronicProduct**(int productId, String name, float price, String brand, int warrantyPeriod) {

7 super(productId, name, price);

8 this.brand = brand;

9 this.warrantyPeriod = Math.abs(warrantyPeriod);

10 }

11

12 // Setters and getters for attributes

13 public String **getBrand**() {

14 return brand;

15 }

16

17 public void **setBrand**(String brand) {

18 this.brand = brand;

19 }

20

21 public int **getWarrantyPeriod**() {

22 return warrantyPeriod;

23 }

24

25 public void **setWarrantyPeriod**(int warrantyPeriod) {

26 this.warrantyPeriod = Math.abs(warrantyPeriod);

27 }

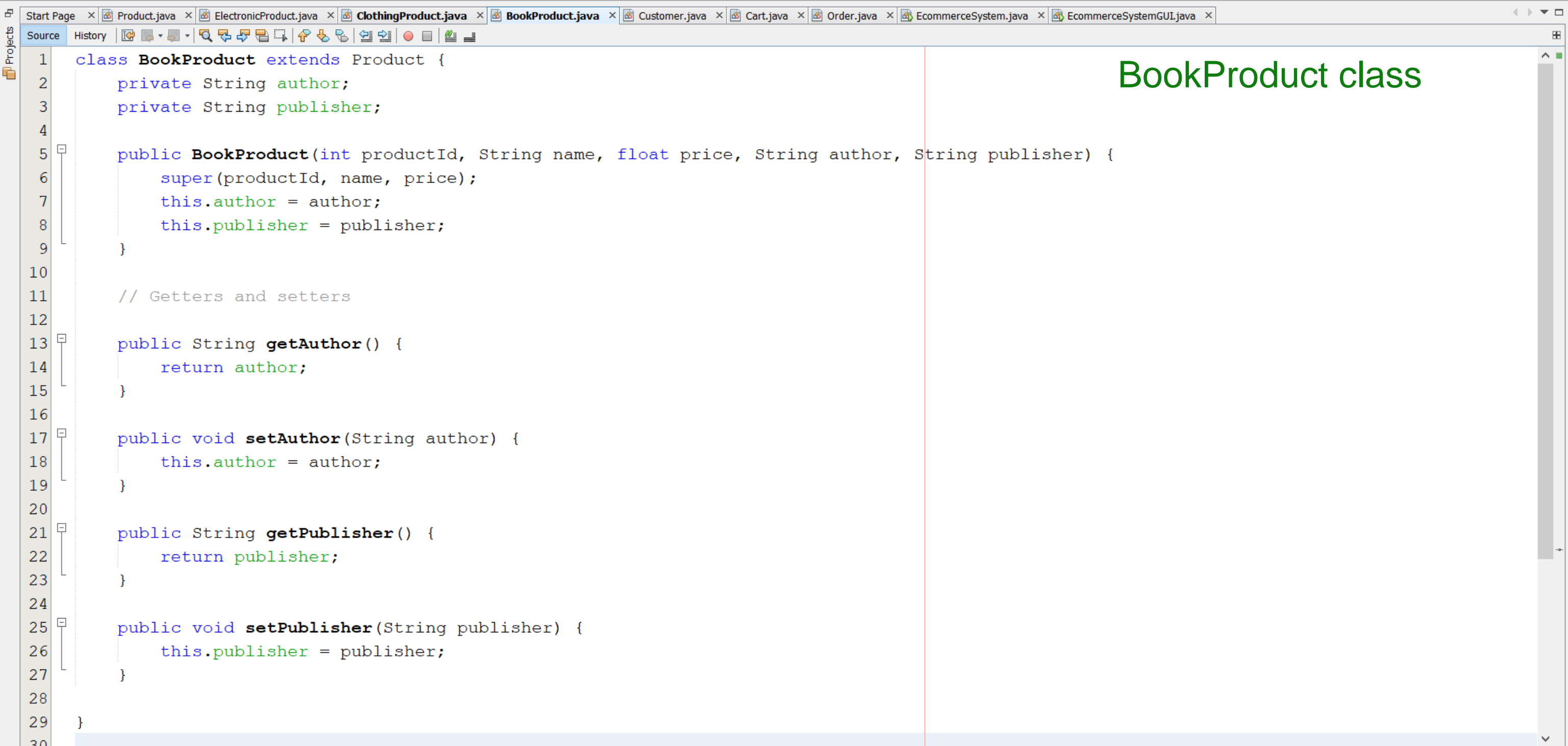
28 }

29

ElectronicProduct class

29

ClothingProduct class



Start Page × Product.java × ElectronicProduct.java × ClothingProduct.java × BookProduct.java × Customer.java × Cart.java × Order.java × EcommerceSystem.java × EcommerceSystemGUI.java ×

Source History

Projects

1 class Customer {
2 private int customerId;
3 private String name;
4 private String address;
5
6 public Customer(int customerId, String name, String address) {
7 this.customerId = Math.abs(customerId);
8 this.name = name;
9 this.address = address;
10 }
11
12 // Getters and setters
13
14 public int getCustomerId() {
15 return customerId;
16 }
17
18 public void setCustomerId(int customerId) {
19 this.customerId = customerId;
20 }
21
22 public String getName() {
23 return name;
24 }
25
26 public void setName(String name) {
27 this.name = name;
28 }
29
30 public String getAddress() {
31 return address;
32 }
33
34 public void setAddress(String address) {
35 this.address = address;
36 }
37 }

Customer class

37:1 INS

Start Page × Product.java × ElectronicProduct.java × ClothingProduct.java × BookProduct.java × Customer.java × Cart.java × Order.java × EcommerceSystem.java × EcommerceSystemGUI.java ×

Source History

1 class Cart {

2 private int customerId;

3 private int nProducts;

4 private Product[] products;

5

6 public Cart(int customerId, int nProducts) {

7 this.customerId = Math.abs(customerId);

8 this.nProducts = Math.abs(nProducts);

9 this.products = new Product[this.nProducts];

10 }

11 // Getters and setters

12 public int getCustomerId() {

13 return customerId;

14 }

15 public void setCustomerId(int customerId) {

16 this.customerId = customerId;

17 }

18 public int getnProducts() {

19 return nProducts;

20 }

21 public void setnProducts(int nProducts) {

22 this.nProducts = nProducts;

23 }

24 public Product[] getProducts() {

25 return products;

26 }

27 public void setProducts(Product[] products) {

28 this.products = products;

29 }

30 //Add products to cart

Cart class

57:1 INS



Start Page × Product.java × ElectronicProduct.java × ClothingProduct.java × BookProduct.java × Customer.java × Cart.java × Order.java × EcommerceSystem.java × EcommerceSystemGUI.java ×

Source History

Projects

1

class Order {

2

private int customerId;

3

private int orderId;

4

private Product[] products;

5

private float totalPrice;

6

7

public Order(int customerId, int orderId, Product[] products) {

8

this.customerId = Math.abs(customerId);

9

this.orderId = Math.abs(orderId);

10

this.products = products;

11

this.totalPrice = calculateTotalPrice();

12

}

13

// Calculate total price of products in order

14

private float calculateTotalPrice() {

15

float totalPrice = 0;

16

for (Product product : products) {

17

if (product != null) {

18

totalPrice += product.getPrice();

19

}}

20

return totalPrice;

21

}

22

// Print order information

23

public void printOrderInfo() {

24

System.out.println("Order ID: " + orderId);

25

System.out.println("Customer ID: " + customerId);

26

System.out.println("Products:");

27

for (Product product : products) {

28

if (product != null) {

29

System.out.println("- " + product.getName() + ": \$" + product.getPrice());

30

}}

31

System.out.println("Total Price: \$" + totalPrice);

32

}}

Order class


```
1 import java.util.Scanner;
2 // EcommerceSystem class
3 public class EcommerceSystem {
4     public static void main(String[] args) {
5         // Create electronic product
6         ElectronicProduct electronicProduct = new ElectronicProduct(1, "smartphone", 599.9f, "Samsung", 1);
7         // Create clothing product
8         ClothingProduct clothingProduct = new ClothingProduct(2, "T-shirt", 19.99f, "Medium", "Cotton");
9         // Create book product
10        BookProduct bookProduct = new BookProduct(3, "OOP", 39.99f, "O'Reilly", "X Publications");
11        // Create scanner object for user input
12        Scanner scanner = new Scanner(System.in);
13        // Prompt user for customer details
14        System.out.println("Welcome to the E-Commerce system !");
15        System.out.println("Please Enter your ID:");
16        int customerId = scanner.nextInt();
17        scanner.nextLine(); // Consume newline
18        System.out.println("Please Enter your name:");
19        String customerName = scanner.nextLine();
20        System.out.println("Please enter your address:");
21        String customerAddress = scanner.nextLine();
22        // Create customer
23        Customer customer = new Customer(customerId, customerName, customerAddress);
24        // Prompt user for number of products
25        System.out.println("How many products do you want to add to your cart?");
26        int numProducts = scanner.nextInt();
27        // Create cart for customer
28        Cart cart = new Cart(customer.getCustomerId(), numProducts);
29        // Prompt user for product details and add them to cart
```

EcommerceSystem class

Start Page × Product.java × ElectronicProduct.java × ClothingProduct.java × BookProduct.java × Customer.java × Cart.java × Order.java × EcommerceSystem.java × EcommerceSystemGUI.java ×

Source History

28 Cart cart = new Cart(customer.getCustomerId(), numProducts);

29 // Prompt user for product details and add them to cart

30 for (int i = 0; i < numProducts; i++) {

31 System.out.println("Which product would you like to add? " + (i + 1) + ":");

32 System.out.println("1- Smartphone\n2- T-shirt\n3- OOP");

33 int choice = scanner.nextInt();

34 scanner.nextLine(); // Consume newline

35 switch (choice) {

36 case 1:

37 cart.addProduct(electronicProduct, i);

38 break;

39 case 2:

40 cart.addProduct(clothingProduct, i);

41 break;

42 case 3:

43 cart.addProduct(bookProduct, i);

44 break;

45 default:

46 System.out.println("Invalid choice.");

47 }

48 }

49 float totalPrice = cart.calculatePrice();

50 System.out.println("Total price for your order: \$" + totalPrice);

51 // Ask user if they want to place an order

52 System.out.println("Would you like to place the order? 1-yes 2-No");

53 String placeOrderChoice = scanner.next();

54 if (placeOrderChoice.equalsIgnoreCase("1")) {

55 // Place order

56 cart.placeOrder();

57 Order order = new Order(customer.getCustomerId(), 1, cart.getProducts());

EcommerceSystem class

Start Page × Product.java × ElectronicProduct.java × ClothingProduct.java × BookProduct.java × Customer.java × Cart.java × Order.java × EcommerceSystem.java × EcommerceSystemGUI.java ×

Source History

Projects

48

}

49

float totalPrice = cart.calculatePrice();

50

System.out.println("Total price for your order: \$" + totalPrice);

51

// Ask user if they want to place an order

52

System.out.println("Would you like to place the order? 1-yes 2-No");

53

String placeOrderChoice = scanner.next();

54

if (placeOrderChoice.equalsIgnoreCase("1")) {

55

// Place order

56

cart.placeOrder();

57

Order order = new Order(customer.getCustomerId(), 1, cart.getProducts());

58

// Print order information

59

System.out.println("Here is your order's summary :");

60

order.printOrderInfo();

61

} else {

62

System.out.println("Order not placed.");

63

}

64

// Close scanner

65

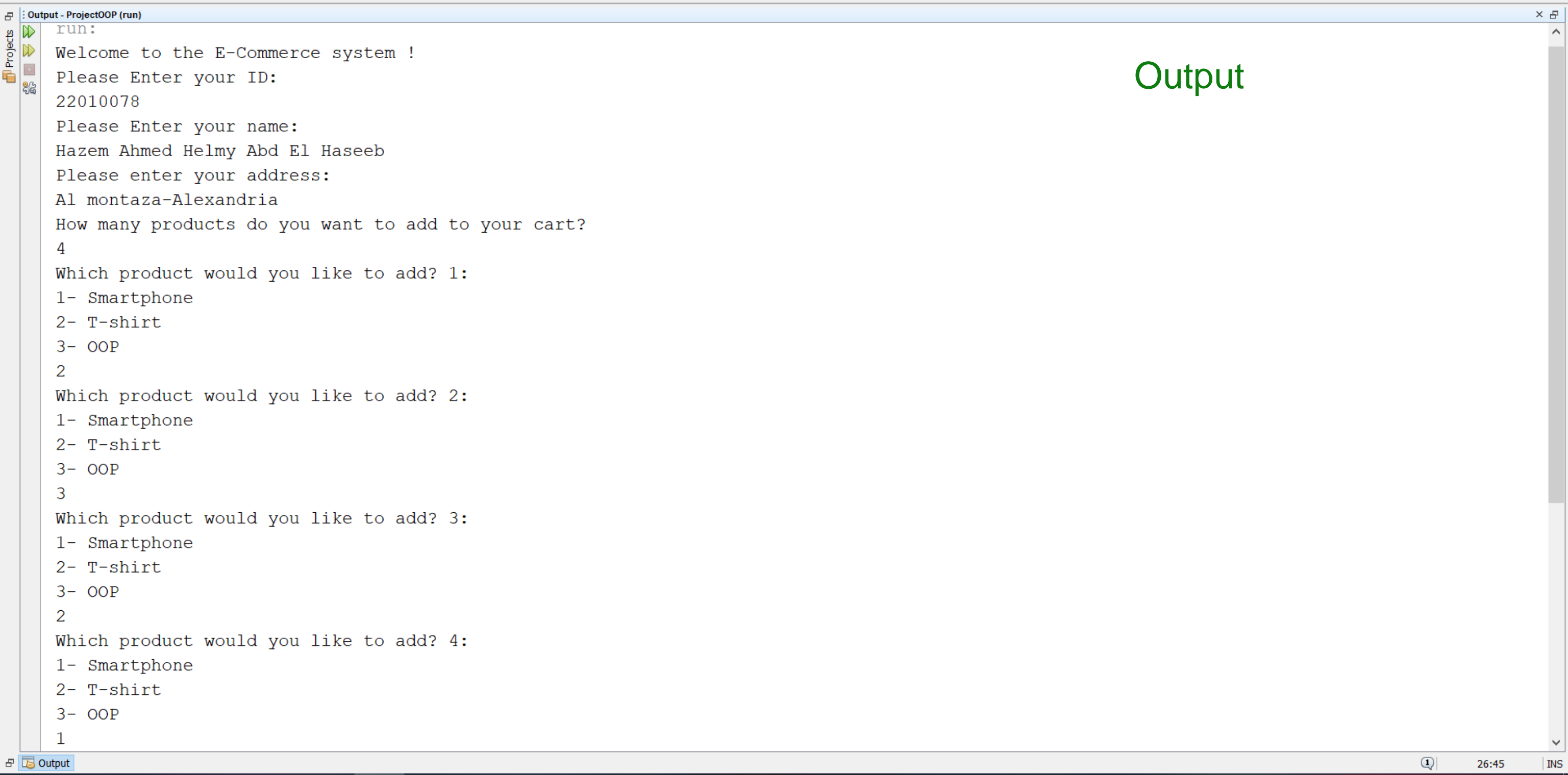
scanner.close();

66

}}

67

EcommerceSystem class



run:

Welcome to the E-Commerce system !

Please Enter your ID:

22010078

Please Enter your name:

Hazem Ahmed Helmy Abd El Haseeb

Please enter your address:

Al montaza-Alexandria

How many products do you want to add to your cart?

4

Which product would you like to add? 1:

1- Smartphone

2- T-shirt

3- OOP

2

Which product would you like to add? 2:

1- Smartphone

2- T-shirt

3- OOP

3

Which product would you like to add? 3:

1- Smartphone

2- T-shirt

3- OOP

2

Which product would you like to add? 4:

1- Smartphone

2- T-shirt

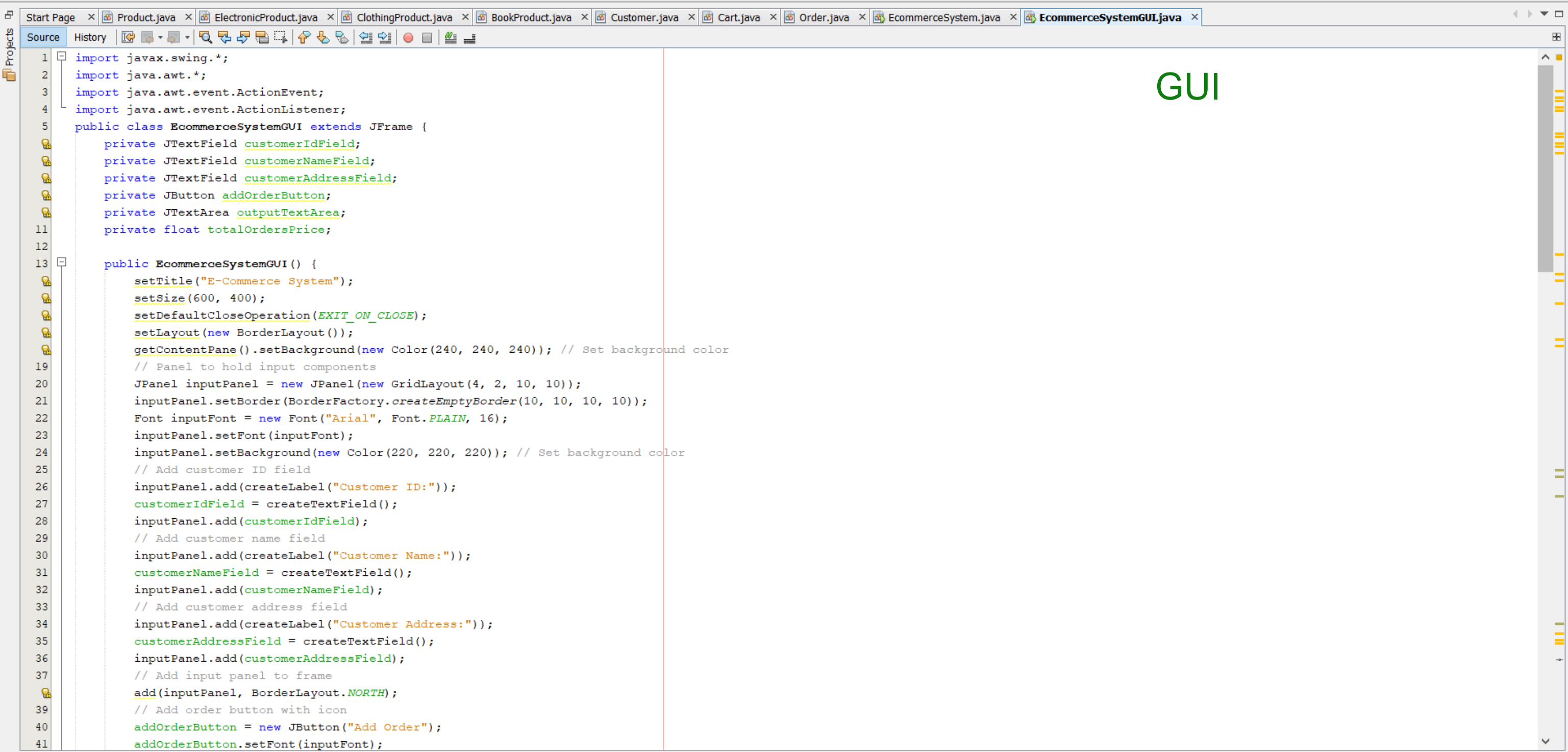
3- OOP

1

Output

Output

```
Total price for your order: $679.87
Would you like to place the order? 1=yes 2-No
1
Order placed.
Here is your order's summary :
Order ID: 1
Customer ID: 22010078
Products:
- T-shirt: $19.99
- OOP: $39.99
- T-shirt: $19.99
- smartphone: $599.9
Total Price: $679.87
BUILD SUCCESSFUL (total time: 1 minute 18 seconds)
```



```
41      addOrderButton.setFont(inputFont);
42      addOrderButton.addActionListener(new ActionListener() {
43          public void actionPerformed(ActionEvent e) {
44              int numOrders = getNumberOfOrders();
45              if (numOrders > 0) {
46                  addOrders(numOrders);
47              }
48          }
49      });
50      add(addOrderButton, BorderLayout.CENTER);
51      // Add output text area
52      outputTextArea = new JTextArea();
53      outputTextArea.setFont(inputFont);
54      outputTextArea.setBackground(new Color(255, 255, 255)); // Set background color
55      JScrollPane scrollPane = new JScrollPane(outputTextArea);
56      scrollPane.setVerticalScrollBarPolicy(ScrollPaneConstants.VERTICAL_SCROLLBAR_ALWAYS);
57      add(scrollPane, BorderLayout.SOUTH);
58      setVisible(true);
59  }
60  private JLabel createLabel(String text) {
61      JLabel label = new JLabel(text);
62      label.setFont(new Font("Arial", Font.BOLD, 16));
63      return label;
64  }
65  private JTextField createTextField() {
66      JTextField textField = new JTextField();
67      textField.setFont(new Font("Arial", Font.PLAIN, 16));
68      textField.setBackground(new Color(255, 255, 255)); // Set background color
69      return textField;
70  }
71  private int getNumberOfOrders() {
72      String numOrdersStr = JOptionPane.showInputDialog(this, "Enter the number of orders:", "Number of Orders", JOptionPane.QUESTION_MESSAGE);
73      try {
74          return Integer.parseInt(numOrdersStr);
75      } catch (NumberFormatException e) {
76          JOptionPane.showMessageDialog(this, "Please enter a valid number.", "Error", JOptionPane.ERROR_MESSAGE);
77          return 0;
78      }
79  }
80  private void addOrders(int numOrders) {
81      String customerId = customerIdField.getText();
82      String customerName = customerNameField.getText();
83      String customerAddress = customerAddressField.getText();
84      if (customerId.isEmpty() || customerName.isEmpty() || customerAddress.isEmpty()) {
85          JOptionPane.showMessageDialog(this, "Please enter all customer details.", "Error", JOptionPane.ERROR_MESSAGE);
86      }
87  }
```

GUI

Start Page × Product.java × ElectronicProduct.java × ClothingProduct.java × BookProduct.java × Customer.java × Cart.java × Order.java × EcommerceSystem.java × EcommerceSystemGUI.java ×

Source History

```
80     if (customerId.isEmpty() || customerName.isEmpty() || customerAddress.isEmpty()) {
81         JOptionPane.showMessageDialog(this, "Please enter all customer details.", "Error", JOptionPane.ERROR_MESSAGE);
82         return;
83     }
84     Customer customer = new Customer(Integer.parseInt(customerId), customerName, customerAddress);
85     for (int i = 0; i < numOrders; i++) {
86         String[] products = {"Smartphone", "T-shirt", "OOP Book"};
87         String selectedProduct = (String) JOptionPane.showInputDialog(this, "Choose a product:", "Product Selection", JOptionPane.QUESTION_MESSAGE, null, products, products[0]);
88         if (selectedProduct != null) {
89             Product product;
90             switch (selectedProduct) {
91                 case "Smartphone":
92                     product = new ElectronicProduct(1, "Smartphone", 599.9f, "Samsung", 1);
93                     break;
94                 case "T-shirt":
95                     product = new ClothingProduct(2, "T-shirt", 19.99f, "Medium", "Cotton");
96                     break;
97                 case "OOP Book":
98                     product = new BookProduct(3, "OOP", 39.99f, "O'Reilly", "X Publications");
99                     break;
100                default:
101                    product = null;
102            }
103            if (product != null) {
104                float orderPrice = product.getPrice();
105                totalOrdersPrice += orderPrice;
106                outputTextArea.append("Order " + (i + 1) + " Summary:\n");
107                outputTextArea.append("Customer ID: " + customer.getCustomerId() + "\n");
108                outputTextArea.append("Customer Name: " + customer.getName() + "\n");
109                outputTextArea.append("Customer Address: " + customer.getAddress() + "\n");
110                outputTextArea.append("Product: " + selectedProduct + "\n");
111                outputTextArea.append("Total price: $" + orderPrice + "\n\n");
112            }
113        }
114        outputTextArea.append("Total Price of All Orders: $" + totalOrdersPrice + "\n");
115    }
116    public static void main(String[] args) {
117        SwingUtilities.invokeLater(new Runnable() {
118            public void run() {
119                new EcommerceSystemGUI();
120            }
121        });
122    }
123 }
```

GUI

Customer ID:

GUI

22010078

Customer Name:

Hazem Ahmed Helmy Abd El Haseeb

Customer Address:

Montaza-Alexandria

Number of Orders

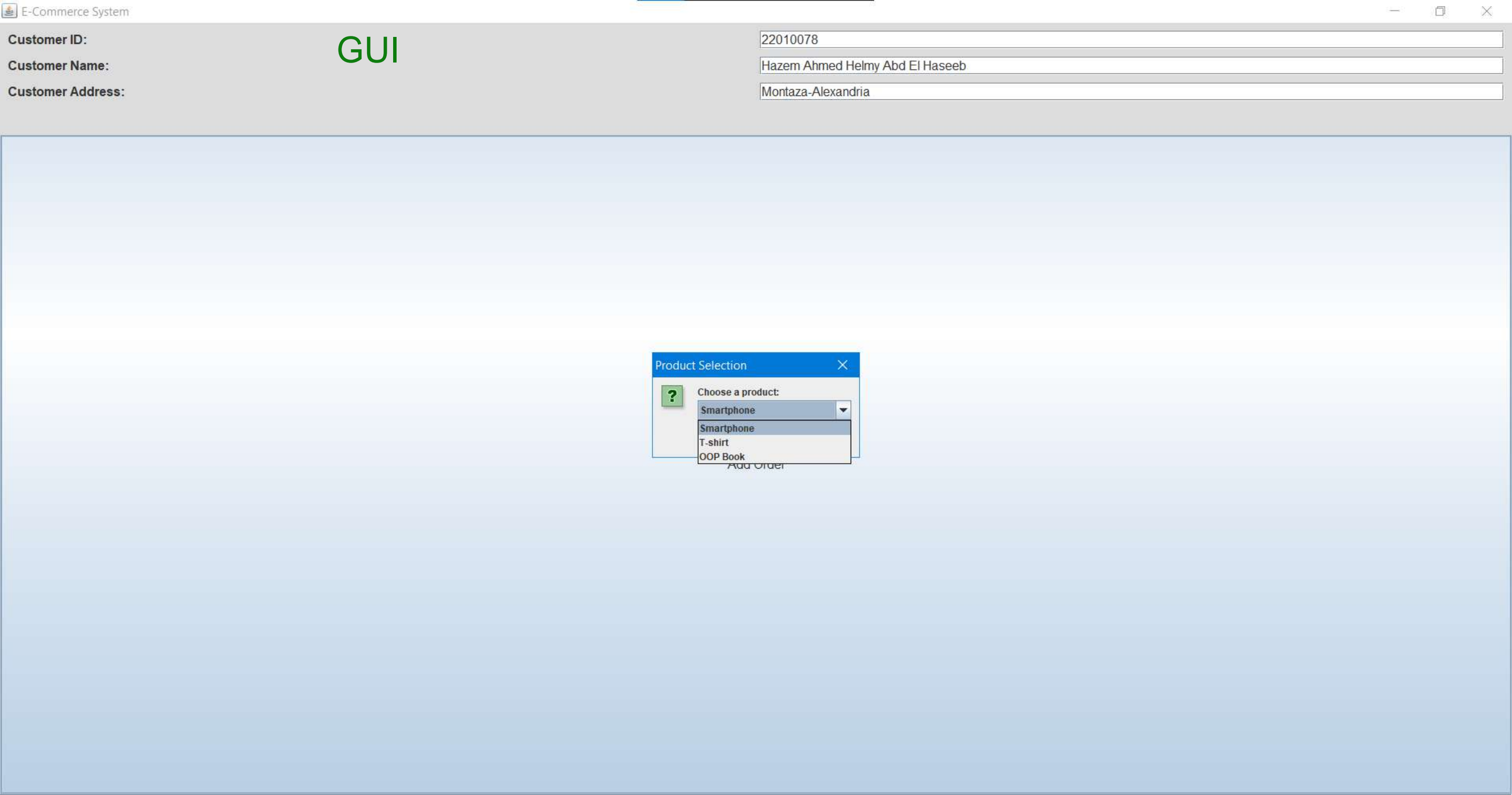
?

Enter the number of orders:

4

OK Cancel

Add Order



Customer ID:

22010078

Customer Name:

Hazem Ahmed Helmy Abd El Haseeb

Customer Address:

Montaza-Alexandria

GUI

Product Selection

?

Choose a product:

Smartphone

Smartphone

T-shirt

OOP Book

Add Order

Customer ID:

22010078

Customer Name:

Hazem Ahmed Helmy Abd El Haseeb

Customer Address:

Montaza-Alexandria

Add Order

Order 1 Summary:
Customer ID: 22010078
Customer Name: Hazem Ahmed Helmy Abd El Haseeb
Customer Address: Montaza-Alexandria
Product: Smartphone
Total price: \$599.9

Order 2 Summary:
Customer ID: 22010078
Customer Name: Hazem Ahmed Helmy Abd El Haseeb
Customer Address: Montaza-Alexandria
Product: T-shirt
Total price: \$19.99

Order 3 Summary:
Customer ID: 22010078
Customer Name: Hazem Ahmed Helmy Abd El Haseeb
Customer Address: Montaza-Alexandria
Product: T-shirt
Total price: \$19.99

Order 4 Summary:
Customer ID: 22010078
Customer Name: Hazem Ahmed Helmy Abd El Haseeb
Customer Address: Montaza-Alexandria
Product: OOP Book
Total price: \$39.99

Total Price of All Orders: \$679.87