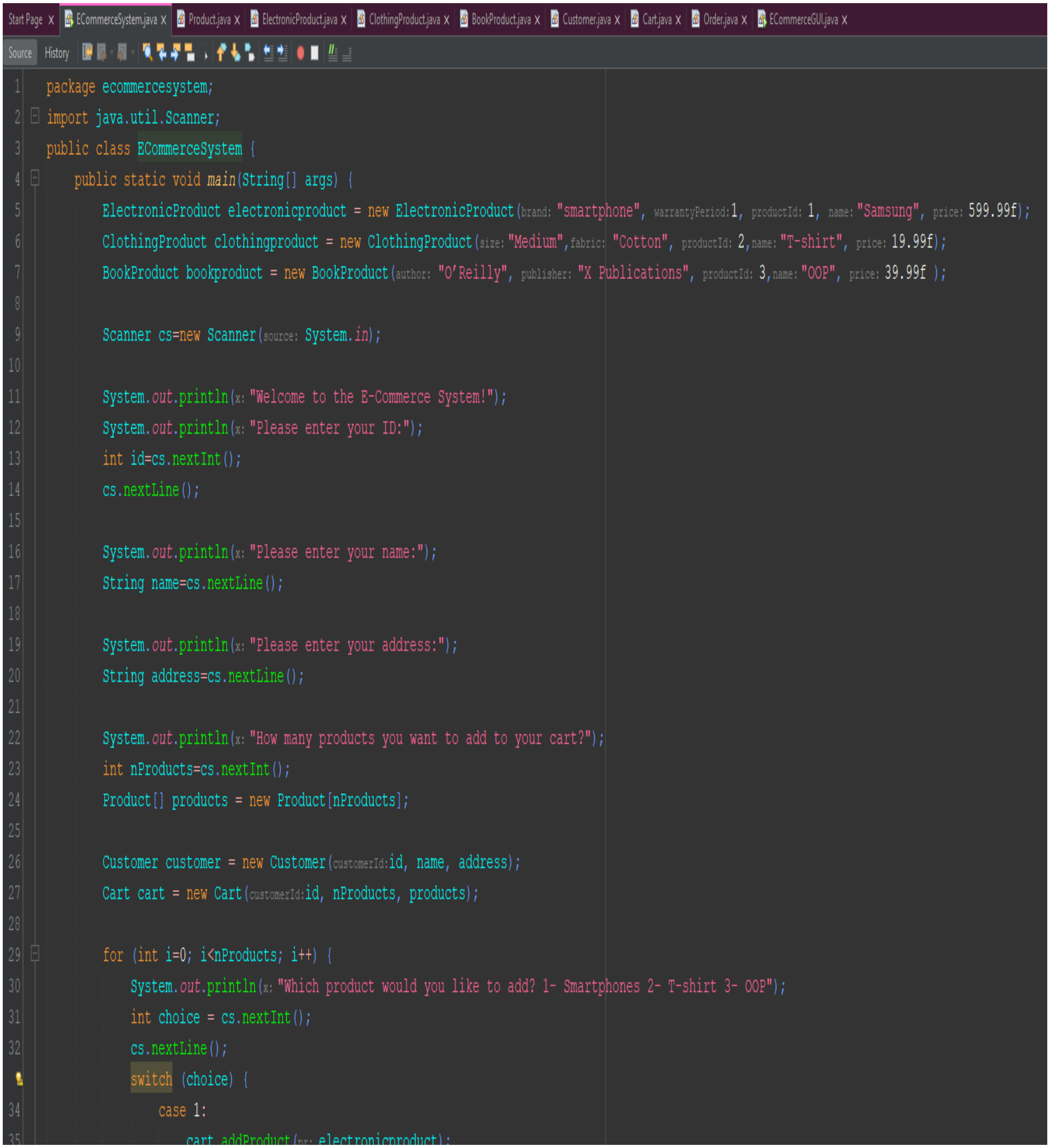
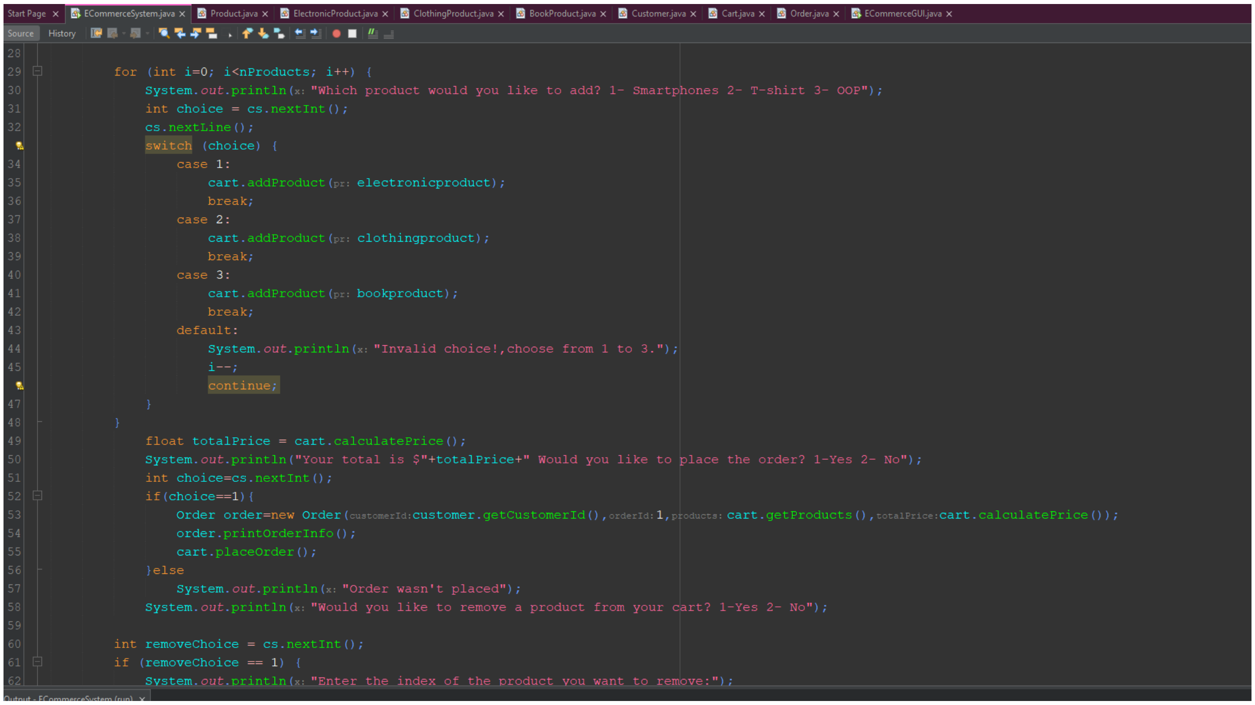
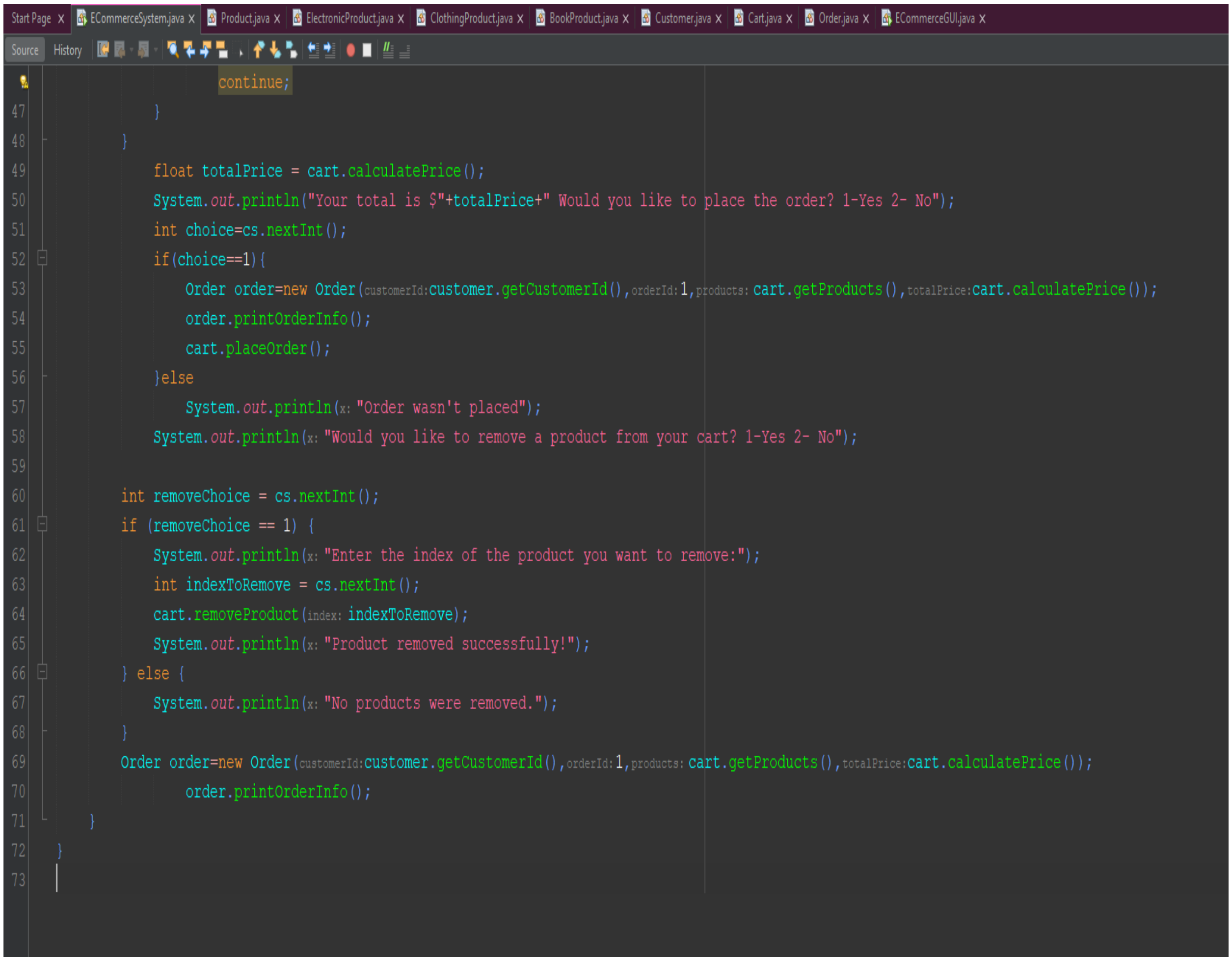


```java
package ecommercesystem;
import java.util.Scanner;
public class ECommerceSystem {
    public static void main(String[] args) {
        ElectronicProduct electronicproduct = new ElectronicProduct(brand: "smartphone", warrantyPeriod:1, productId: 1, name: "Samsung", price: 599.99f);
        ClothingProduct clothingproduct = new ClothingProduct(size: "Medium",fabric: "Cotton", productId: 2,name: "T-shirt", price: 19.99f);
        BookProduct bookproduct = new BookProduct(author: "O'Reilly", publisher: "X Publications", productId: 3,name: "OOP", price: 39.99f );

        Scanner cs=new Scanner(source: System.in);

        System.out.println(x: "Welcome to the E-Commerce System!");
        System.out.println(x: "Please enter your ID:");
        int id=cs.nextInt();
        cs.nextLine();

        System.out.println(x: "Please enter your name:");
        String name=cs.nextLine();

        System.out.println(x: "Please enter your address:");
        String address=cs.nextLine();

        System.out.println(x: "How many products you want to add to your cart?");
        int nProducts=cs.nextInt();
        Product[] products = new Product[nProducts];

        Customer customer = new Customer(customerId:id, name, address);
        Cart cart = new Cart(customerId:id, nProducts, products);

        for (int i=0; i<nProducts; i++) {
            System.out.println(x: "Which product would you like to add? 1- Smartphones 2- T-shirt 3- OOP");
            int choice = cs.nextInt();
            cs.nextLine();
            switch (choice) {
                case 1:
                    cart.addProduct(pr: electronicproduct);
```

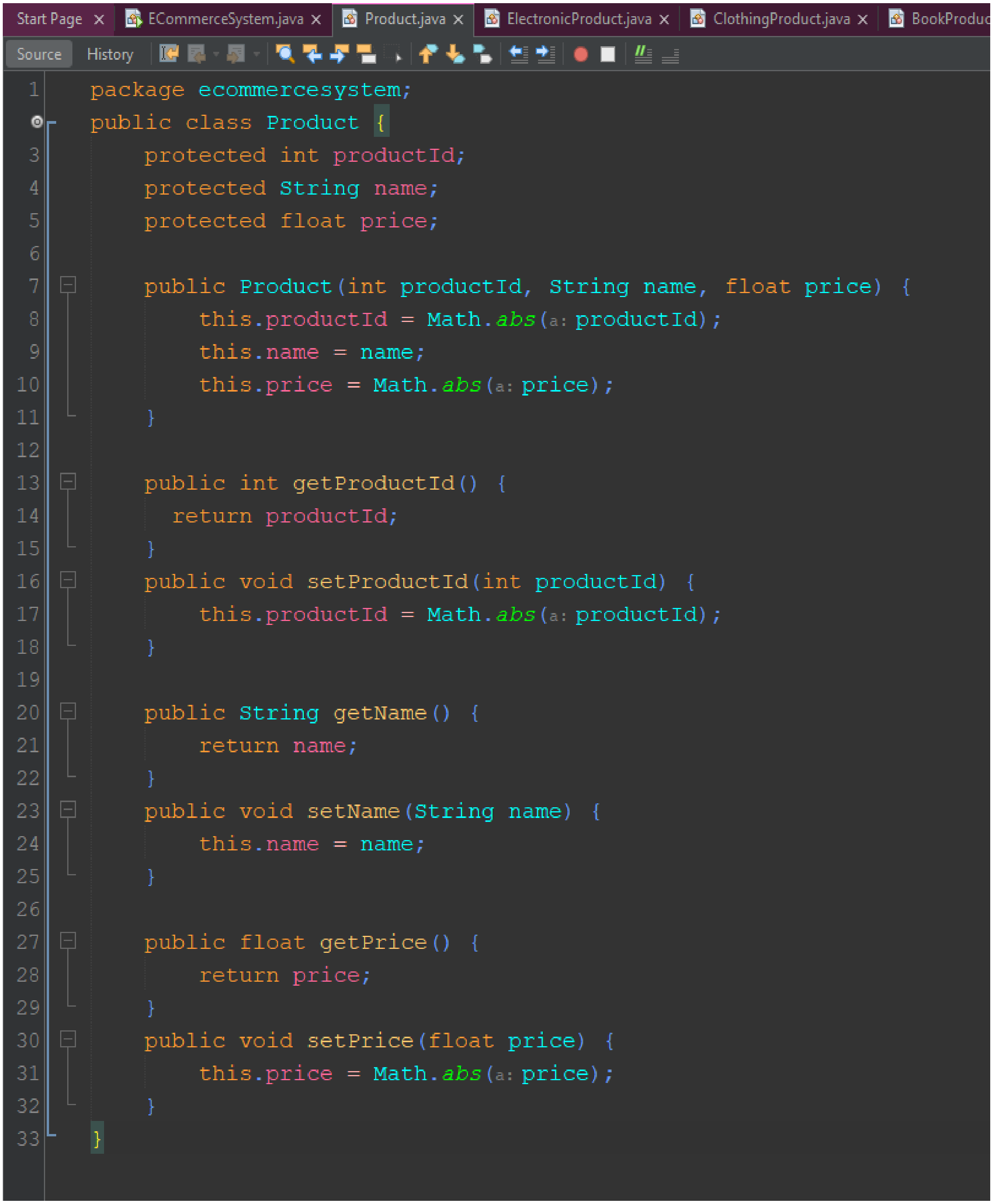


```java

        for (int i=0; i<nProducts; i++) {
            System.out.println(x: "Which product would you like to add? 1- Smartphones 2- T-shirt 3- OOP");
            int choice = cs.nextInt();
            cs.nextLine();
            switch (choice) {
                case 1:
                    cart.addProduct(pr: electronicproduct);
                    break;
                case 2:
                    cart.addProduct(pr: clothingproduct);
                    break;
                case 3:
                    cart.addProduct(pr: bookproduct);
                    break;
                default:
                    System.out.println(x: "Invalid choice!,choose from 1 to 3.");
                    i--;
                    continue;
            }
        }
            float totalPrice = cart.calculatePrice();
            System.out.println("Your total is $"+totalPrice+" Would you like to place the order? 1-Yes 2- No");
            int choice=cs.nextInt();
            if(choice==1){
                Order order=new Order(customerId:customer.getCustomerId(),orderId:1,products: cart.getProducts(),totalPrice:cart.calculatePrice());
                order.printOrderInfo();
                cart.placeOrder();
            }else
                System.out.println(x: "Order wasn't placed");
            System.out.println(x: "Would you like to remove a product from your cart? 1-Yes 2- No");

        int removeChoice = cs.nextInt();
        if (removeChoice == 1) {
            System.out.println(x: "Enter the index of the product you want to remove:");
```

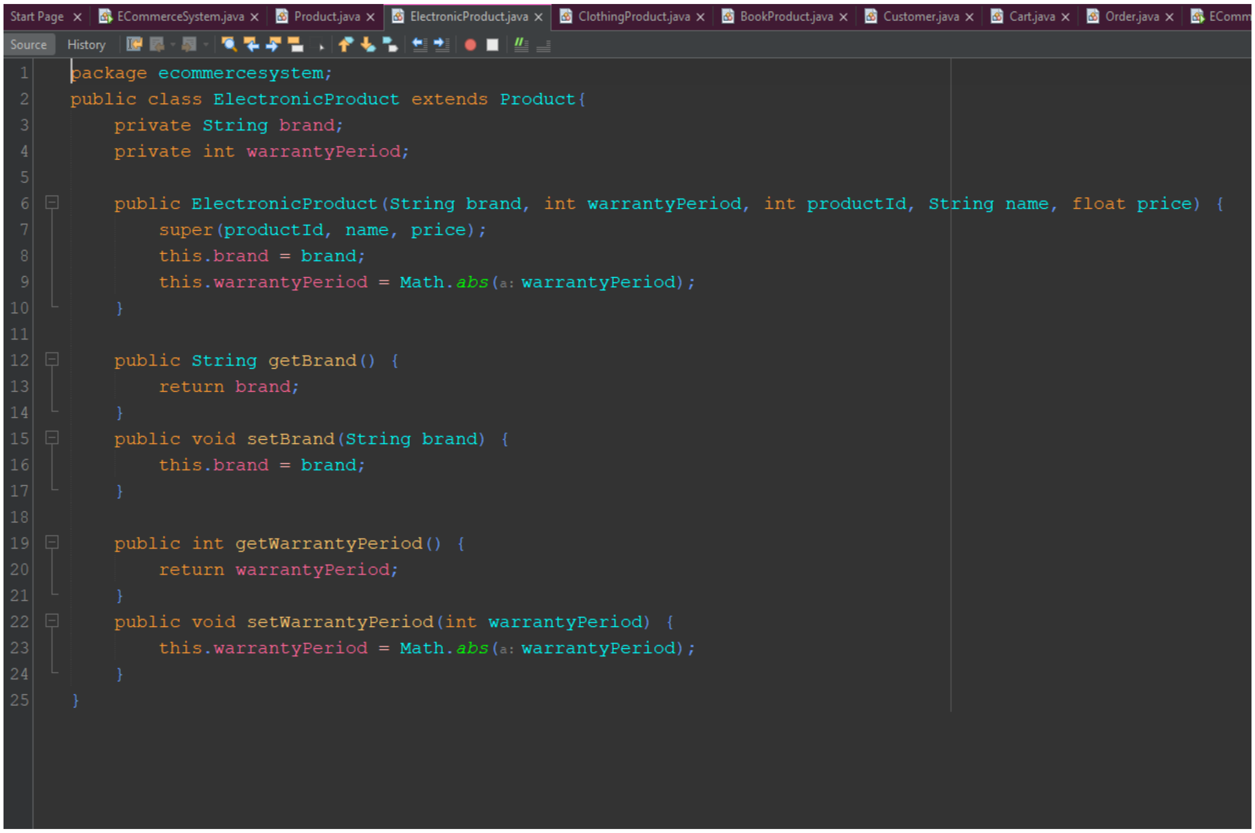


```java
                    continue;
            }
        }
            float totalPrice = cart.calculatePrice();
            System.out.println("Your total is $"+totalPrice+" Would you like to place the order? 1-Yes 2- No");
            int choice=cs.nextInt();
            if(choice==1){
                Order order=new Order(customerId:customer.getCustomerId(),orderId:1,products: cart.getProducts(),totalPrice:cart.calculatePrice());
                order.printOrderInfo();
                cart.placeOrder();
            }else
                System.out.println(x: "Order wasn't placed");
            System.out.println(x: "Would you like to remove a product from your cart? 1-Yes 2- No");

        int removeChoice = cs.nextInt();
        if (removeChoice == 1) {
            System.out.println(x: "Enter the index of the product you want to remove:");
```

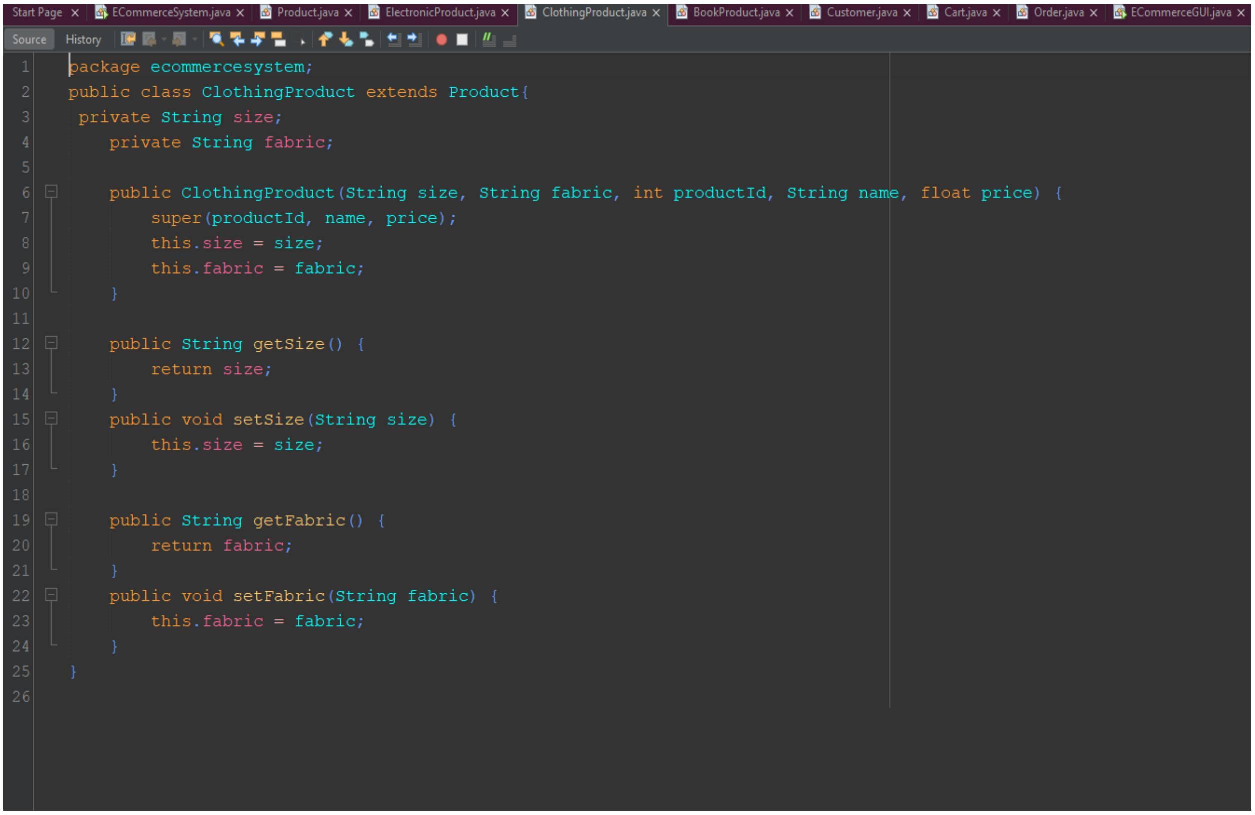
```java
            int indexToRemove = cs.nextInt();
            cart.removeProduct(index: indexToRemove);
            System.out.println(x: "Product removed successfully!");
        } else {
            System.out.println(x: "No products were removed.");
        }
        Order order=new Order(customerId:customer.getCustomerId(),orderId:1,products: cart.getProducts(),totalPrice:cart.calculatePrice());
                order.printOrderInfo();
    }
}
```



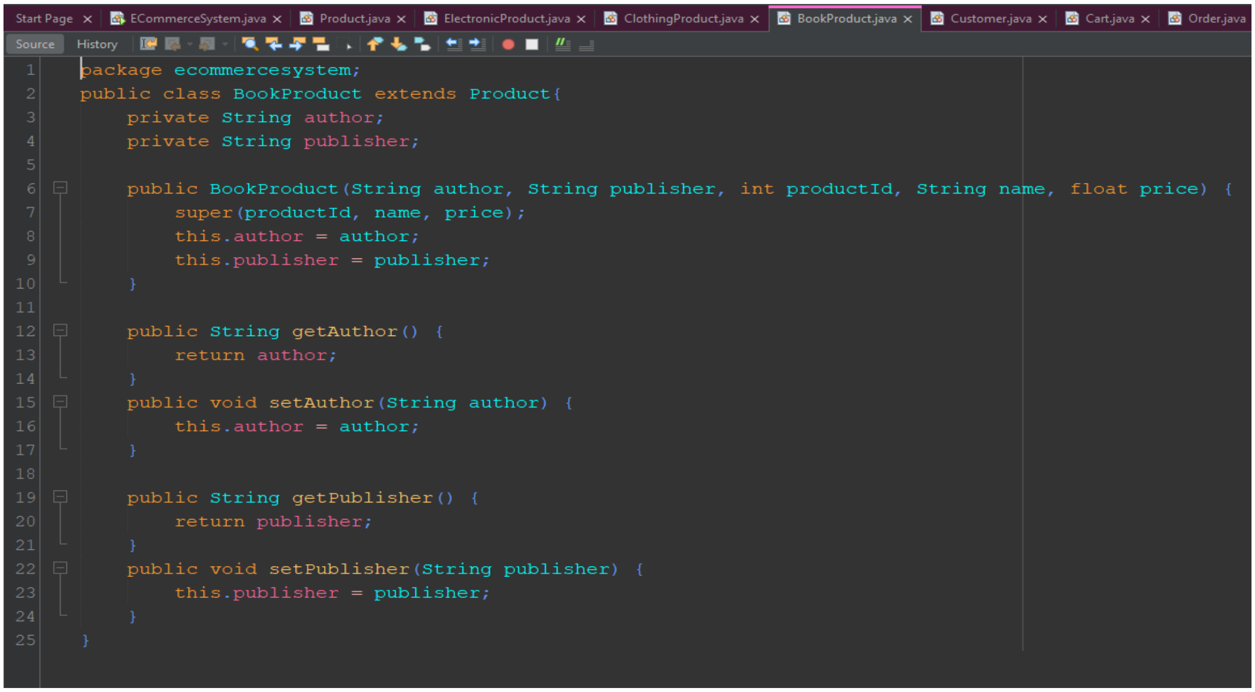
```java
package ecommercesystem;
public class Product {
    protected int productId;
    protected String name;
    protected float price;

    public Product(int productId, String name, float price) {
        this.productId = Math.abs(a: productId);
        this.name = name;
        this.price = Math.abs(a: price);
    }

    public int getProductId() {
      return productId;
    }
    public void setProductId(int productId) {
        this.productId = Math.abs(a: productId);
    }

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

    public float getPrice() {
        return price;
    }
    public void setPrice(float price) {
        this.price = Math.abs(a: price);
    }
}
```



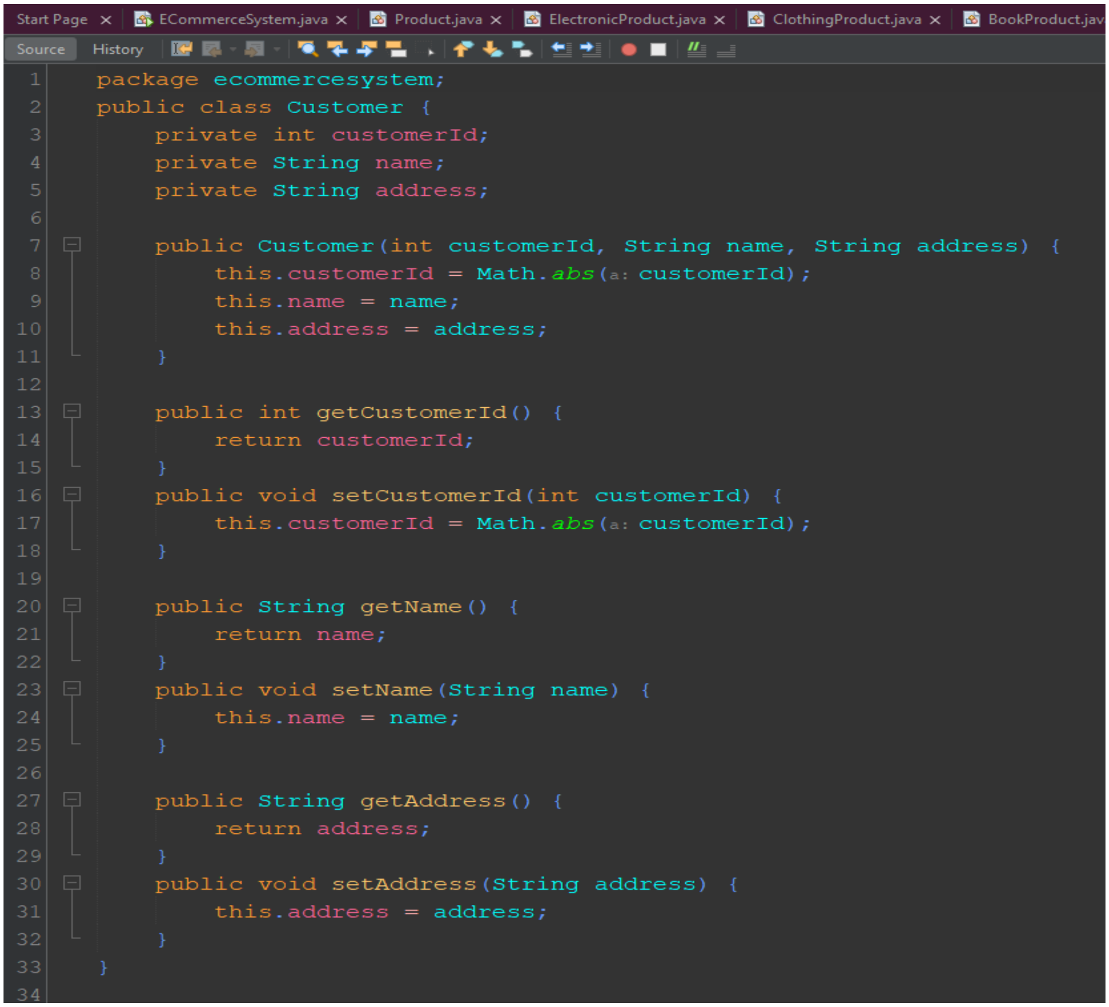
```java
package ecommercesystem;
public class ElectronicProduct extends Product{
    private String brand;
    private int warrantyPeriod;

    public ElectronicProduct(String brand, int warrantyPeriod, int productId, String name, float price) {
        super(productId, name, price);
        this.brand = brand;
        this.warrantyPeriod = Math.abs(a: warrantyPeriod);
    }

    public String getBrand() {
        return brand;
    }
    public void setBrand(String brand) {
        this.brand = brand;
    }

    public int getWarrantyPeriod() {
        return warrantyPeriod;
    }
    public void setWarrantyPeriod(int warrantyPeriod) {
        this.warrantyPeriod = Math.abs(a: warrantyPeriod);
    }
}
```



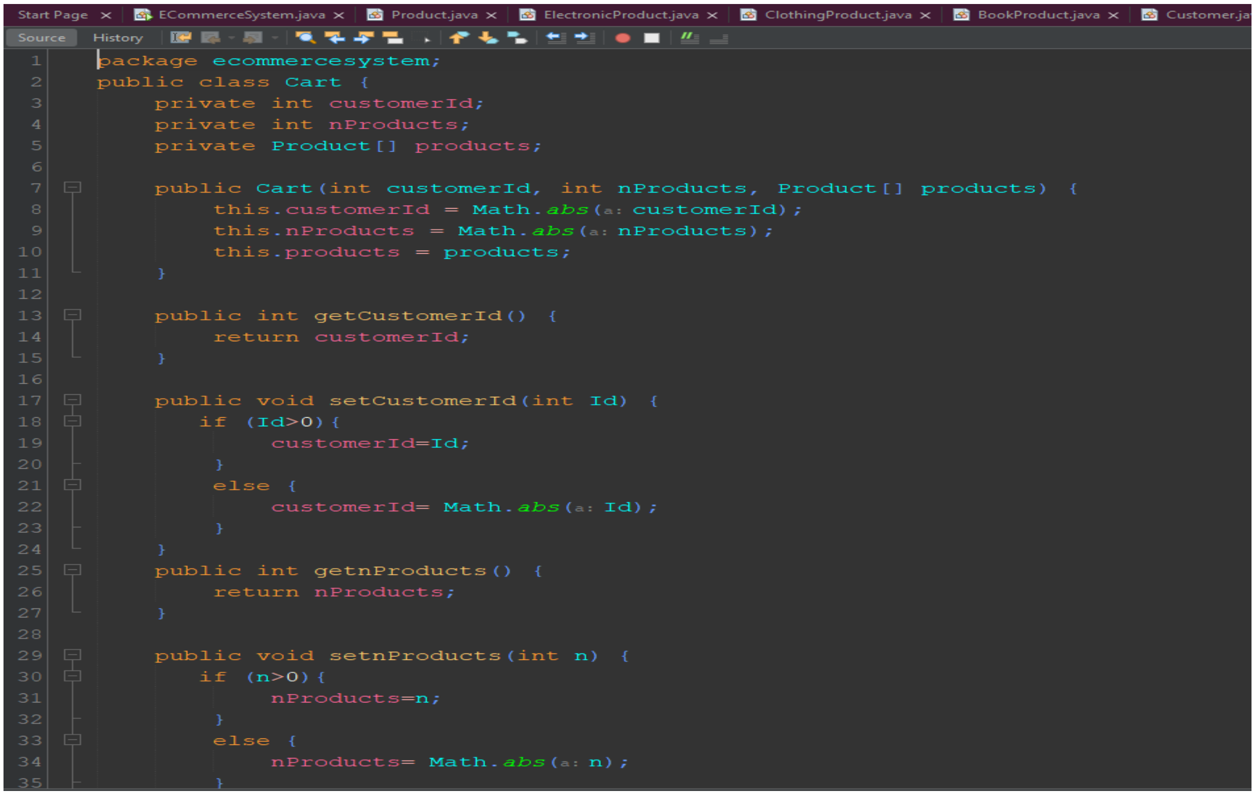
```java
package ecommercesystem;
public class ClothingProduct extends Product{
 private String size;
    private String fabric;

    public ClothingProduct(String size, String fabric, int productId, String name, float price) {
        super(productId, name, price);
        this.size = size;
        this.fabric = fabric;
    }

    public String getSize() {
        return size;
    }
    public void setSize(String size) {
        this.size = size;
    }

    public String getFabric() {
        return fabric;
    }
    public void setFabric(String fabric) {
        this.fabric = fabric;
    }
}
```



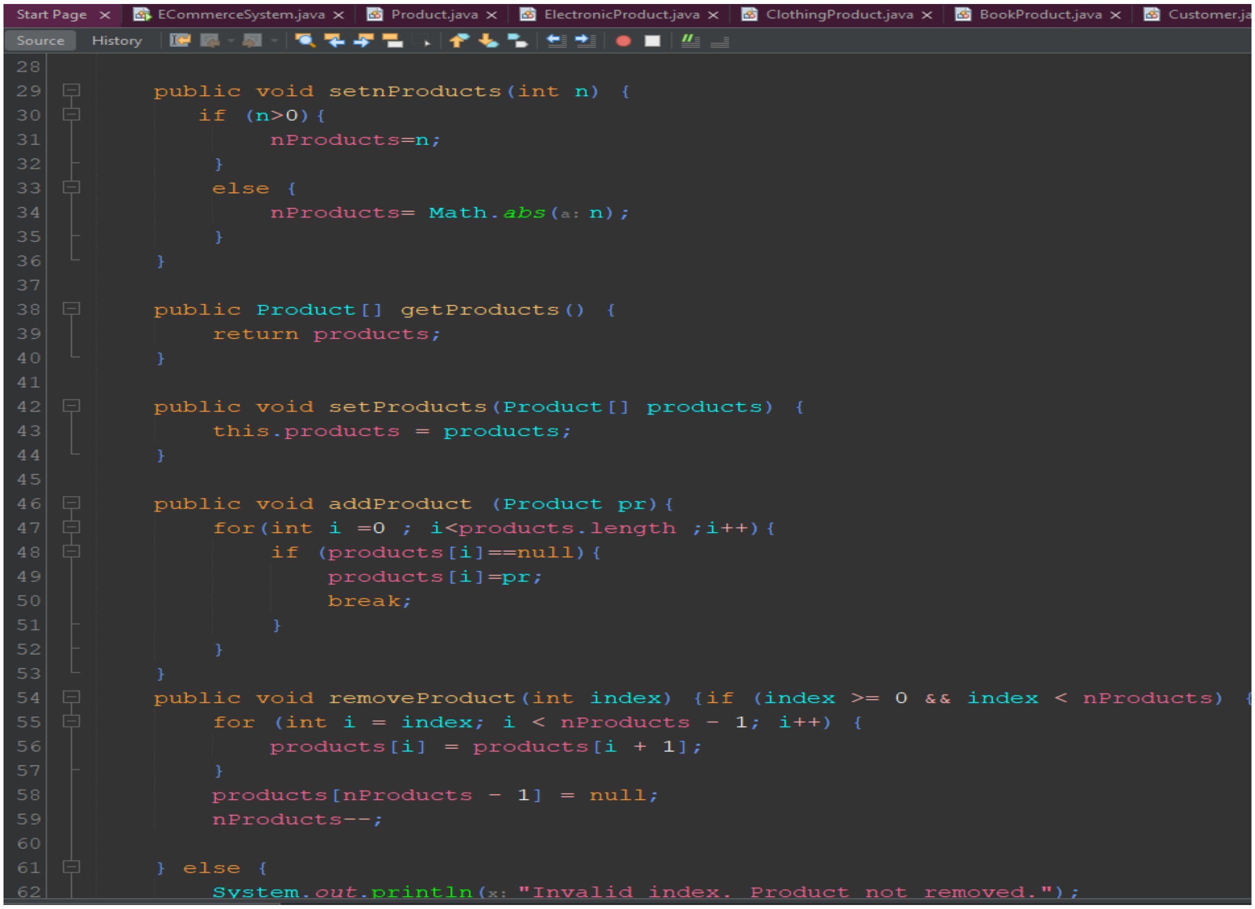
```java
package ecommercesystem;
public class BookProduct extends Product{
    private String author;
    private String publisher;

    public BookProduct(String author, String publisher, int productId, String name, float price) {
        super(productId, name, price);
        this.author = author;
        this.publisher = publisher;
    }

    public String getAuthor() {
        return author;
    }
    public void setAuthor(String author) {
        this.author = author;
    }

    public String getPublisher() {
        return publisher;
    }
    public void setPublisher(String publisher) {
        this.publisher = publisher;
    }
}
```



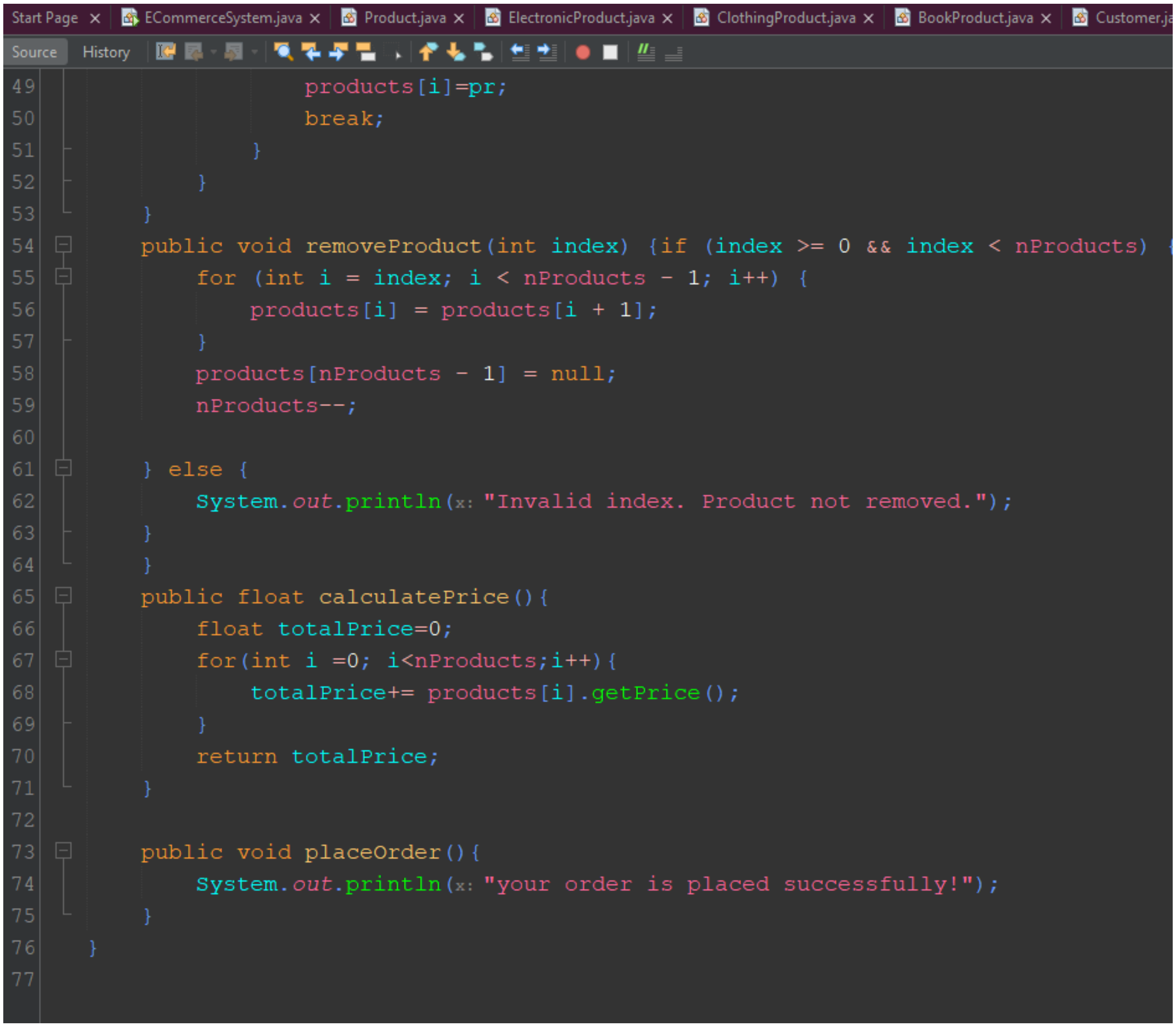
```java
package ecommercesystem;
public class Customer {
    private int customerId;
    private String name;
    private String address;

    public Customer(int customerId, String name, String address) {
        this.customerId = Math.abs(a: customerId);
        this.name = name;
        this.address = address;
    }

    public int getCustomerId() {
        return customerId;
    }
    public void setCustomerId(int customerId) {
        this.customerId = Math.abs(a: customerId);
    }

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

    public String getAddress() {
        return address;
    }
    public void setAddress(String address) {
        this.address = address;
    }
}

```



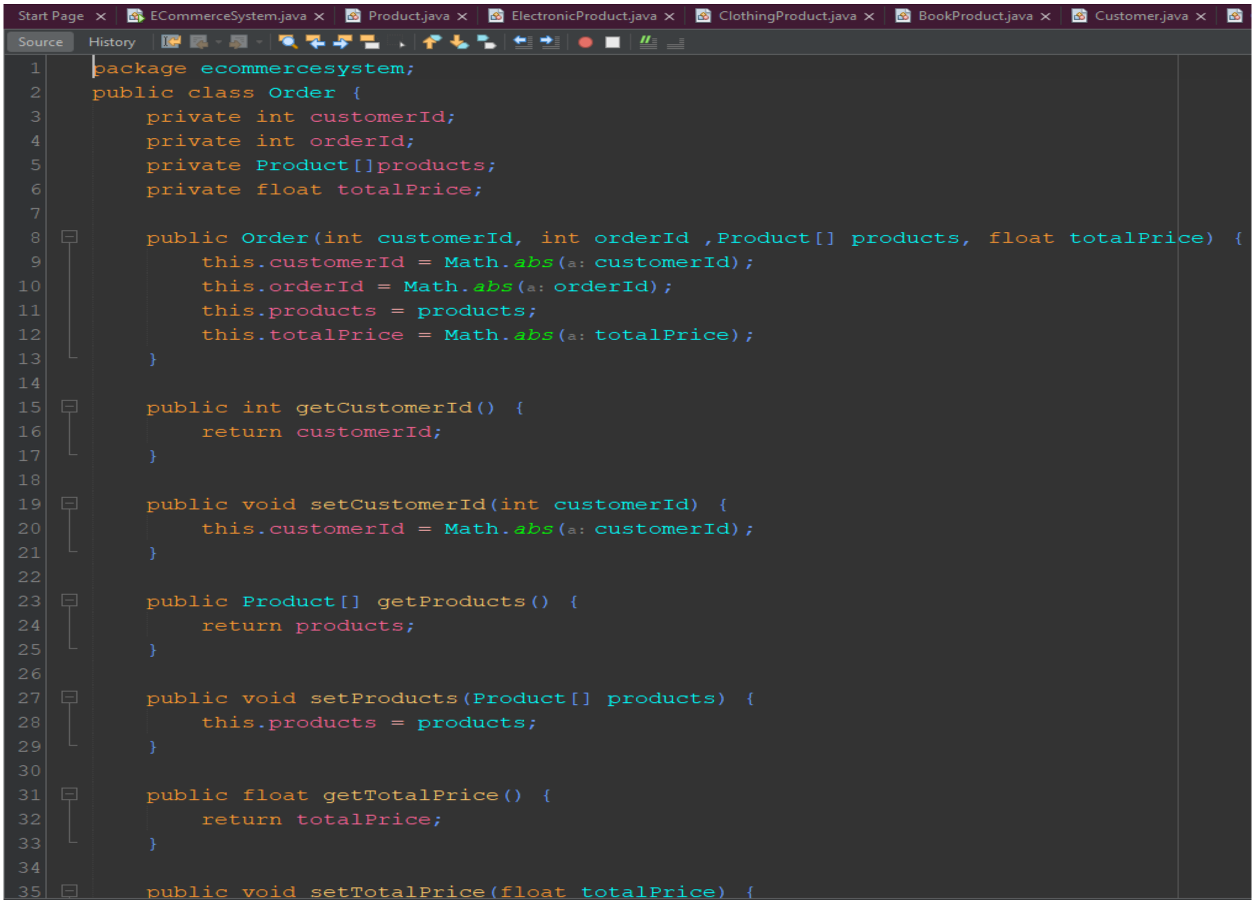
```java
package ecommercesystem;
public class Cart {
    private int customerId;
    private int nProducts;
    private Product[] products;

    public Cart(int customerId, int nProducts, Product[] products) {
        this.customerId = Math.abs(a: customerId);
        this.nProducts = Math.abs(a: nProducts);
        this.products = products;
    }

    public int getCustomerId() {
        return customerId;
    }

    public void setCustomerId(int Id) {
       if (Id>0){
            customerId=Id;
        }
        else {
            customerId= Math.abs(a: Id);
        }
    }
    public int getnProducts() {
        return nProducts;
    }

    public void setnProducts(int n) {
       if (n>0){
            nProducts=n;
        }
        else {
            nProducts= Math.abs(a: n);
        }
```



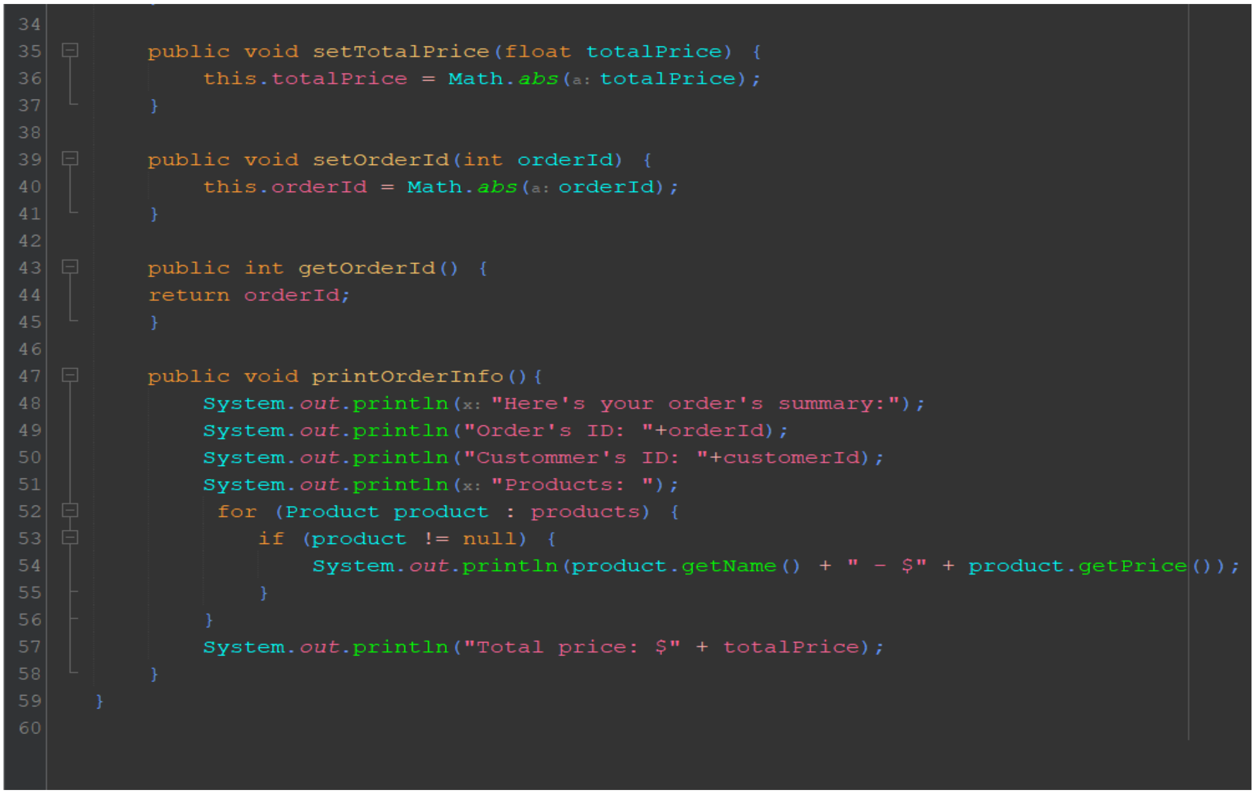
```java

    public void setnProducts(int n) {
       if (n>0){
            nProducts=n;
        }
        else {
            nProducts= Math.abs(a: n);
        }
    }

    public Product[] getProducts() {
        return products;
    }

    public void setProducts(Product[] products) {
        this.products = products;
    }

    public void addProduct (Product pr){
        for(int i =0 ; i<products.length ;i++){
            if (products[i]==null){
                products[i]=pr;
                break;
            }
        }
    }
    public void removeProduct(int index) {if (index >= 0 && index < nProducts) {
        for (int i = index; i < nProducts - 1; i++) {
            products[i] = products[i + 1];
        }
        products[nProducts - 1] = null;
        nProducts--;

    } else {
        System.out.println(x: "Invalid index. Product not removed.");
```



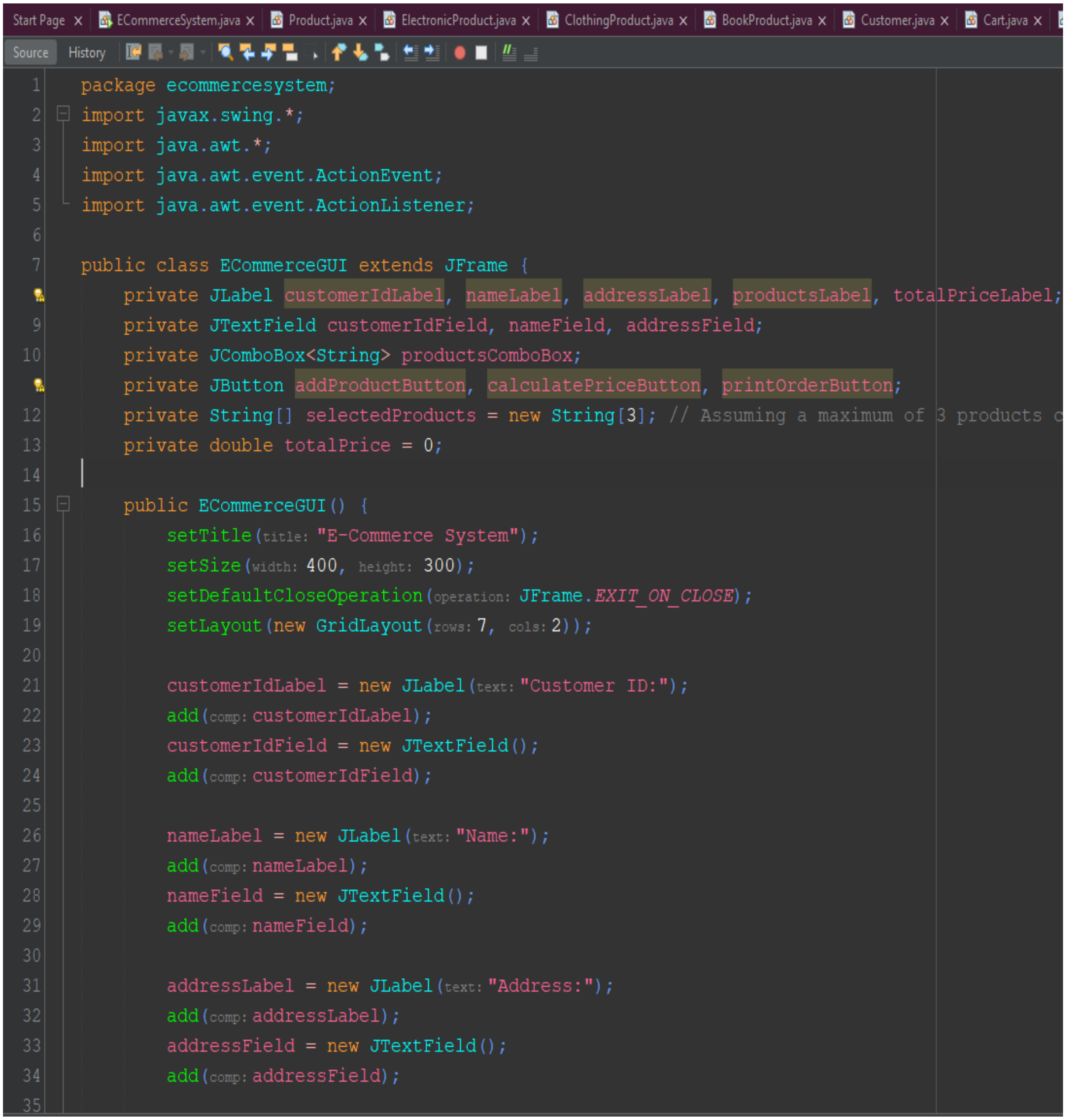
```
                products[i]=pr;
                break;
            }
        }
    }
    public void removeProduct(int index) {if (index >= 0 && index < nProducts) {
        for (int i = index; i < nProducts - 1; i++) {
            products[i] = products[i + 1];
        }
        products[nProducts - 1] = null;
        nProducts--;

    } else {
        System.out.println(x: "Invalid index. Product not removed.");
    }
    }
    public float calculatePrice(){
        float totalPrice=0;
        for(int i =0; i<nProducts;i++){
            totalPrice+= products[i].getPrice();
        }
        return totalPrice;
    }

    public void placeOrder(){
        System.out.println(x: "your order is placed successfully!");
    }
}
```



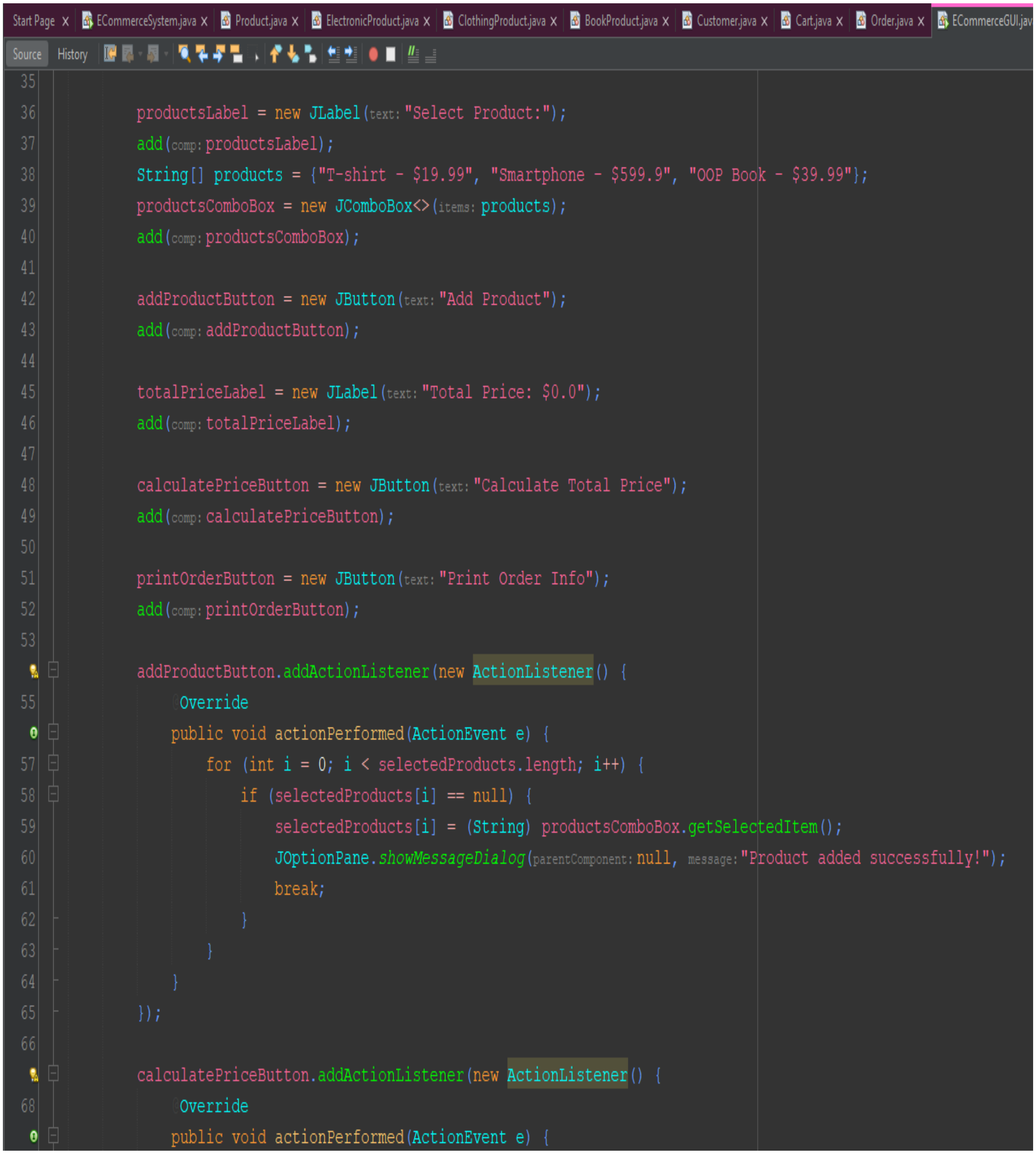
```java
package ecommercesystem;
public class Order {
    private int customerId;
    private int orderId;
    private Product[]products;
    private float totalPrice;

    public Order(int customerId, int orderId ,Product[] products, float totalPrice) {
        this.customerId = Math.abs(a: customerId);
        this.orderId = Math.abs(a: orderId);
        this.products = products;
        this.totalPrice = Math.abs(a: totalPrice);
    }

    public int getCustomerId() {
        return customerId;
    }

    public void setCustomerId(int customerId) {
        this.customerId = Math.abs(a: customerId);
    }

    public Product[] getProducts() {
        return products;
    }

    public void setProducts(Product[] products) {
        this.products = products;
    }

    public float getTotalPrice() {
        return totalPrice;
    }

    public void setTotalPrice(float totalPrice) {
```



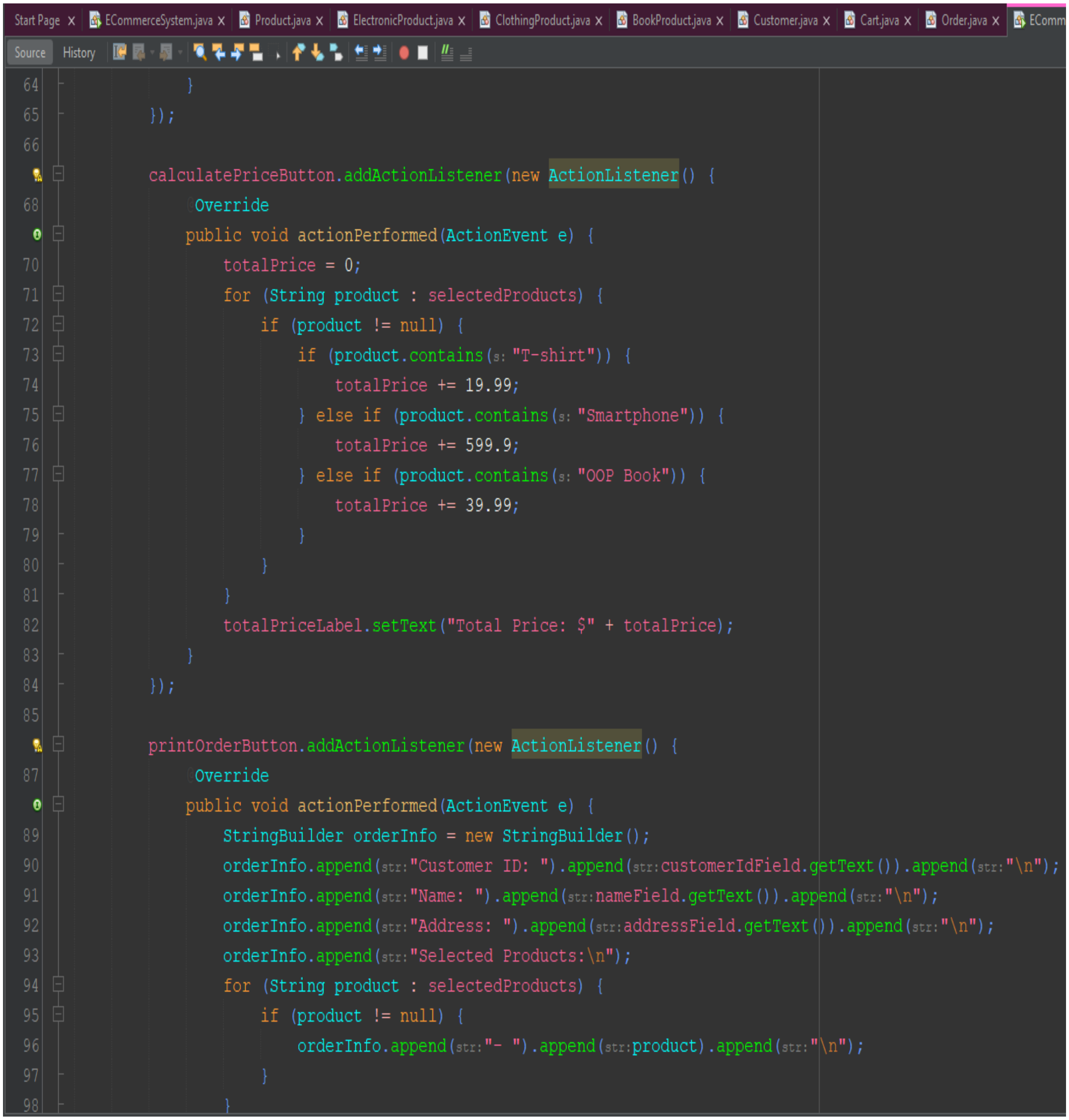
```java

    public void setTotalPrice(float totalPrice) {
        this.totalPrice = Math.abs(a: totalPrice);
    }

    public void setOrderId(int orderId) {
        this.orderId = Math.abs(a: orderId);
    }

    public int getOrderId() {
    return orderId;
    }

    public void printOrderInfo(){
        System.out.println(x: "Here's your order's summary:");
        System.out.println("Order's ID: "+orderId);
        System.out.println("Custommer's ID: "+customerId);
        System.out.println(x: "Products: ");
         for (Product product : products) {
            if (product != null) {
                System.out.println(product.getName() + " - $" + product.getPrice());
            }
        }
        System.out.println("Total price: $" + totalPrice);
    }
}

```



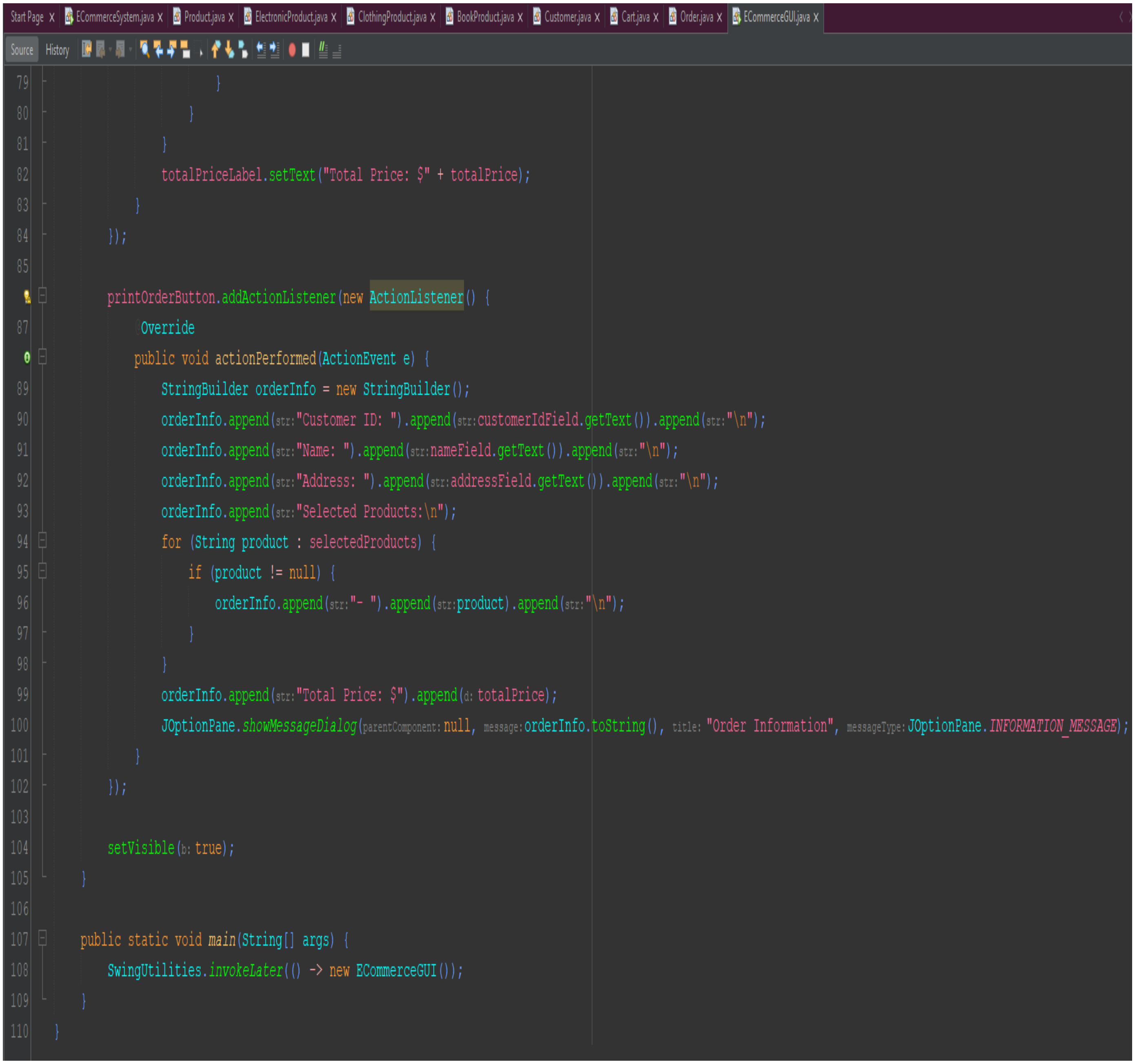
```java
package ecommercesystem;
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class ECommerceGUI extends JFrame {
    private JLabel customerIdLabel, nameLabel, addressLabel, productsLabel, totalPriceLabel;
    private JTextField customerIdField, nameField, addressField;
    private JComboBox<String> productsComboBox;
    private JButton addProductButton, calculatePriceButton, printOrderButton;
    private String[] selectedProducts = new String[3]; // Assuming a maximum of 3 products c
    private double totalPrice = 0;

    public ECommerceGUI() {
        setTitle(title: "E-Commerce System");
        setSize(width: 400, height: 300);
        setDefaultCloseOperation(operation: JFrame.EXIT_ON_CLOSE);
        setLayout(new GridLayout(rows: 7, cols: 2));

        customerIdLabel = new JLabel(text: "Customer ID:");
        add(comp: customerIdLabel);
        customerIdField = new JTextField();
        add(comp: customerIdField);

        nameLabel = new JLabel(text: "Name:");
        add(comp: nameLabel);
        nameField = new JTextField();
        add(comp: nameField);

        addressLabel = new JLabel(text: "Address:");
        add(comp: addressLabel);
        addressField = new JTextField();
        add(comp: addressField);

```



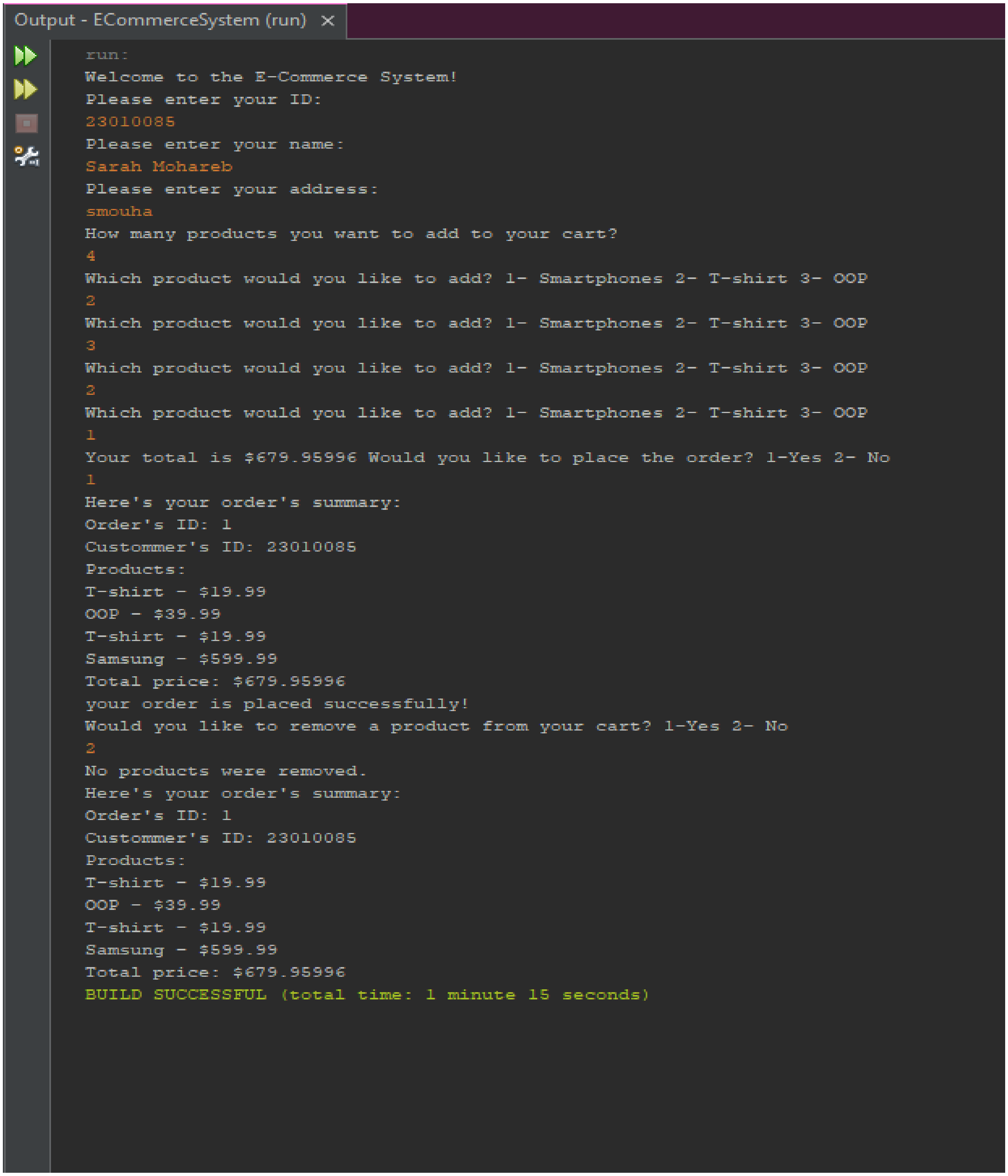
```

        productsLabel = new JLabel(text: "Select Product:");
        add(comp: productsLabel);
        String[] products = {"T-shirt - $19.99", "Smartphone - $599.9", "OOP Book - $39.99"};
        productsComboBox = new JComboBox<>(items: products);
        add(comp: productsComboBox);

        addProductButton = new JButton(text: "Add Product");
        add(comp: addProductButton);

        totalPriceLabel = new JLabel(text: "Total Price: $0.0");
        add(comp: totalPriceLabel);

        calculatePriceButton = new JButton(text: "Calculate Total Price");
        add(comp: calculatePriceButton);

        printOrderButton = new JButton(text: "Print Order Info");
        add(comp: printOrderButton);

        addProductButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                for (int i = 0; i < selectedProducts.length; i++) {
                    if (selectedProducts[i] == null) {
                        selectedProducts[i] = (String) productsComboBox.getSelectedItem();
                        JOptionPane.showMessageDialog(parentComponent: null, message: "Product added successfully!");
                        break;
                    }
                }
            }
        });

        calculatePriceButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
```



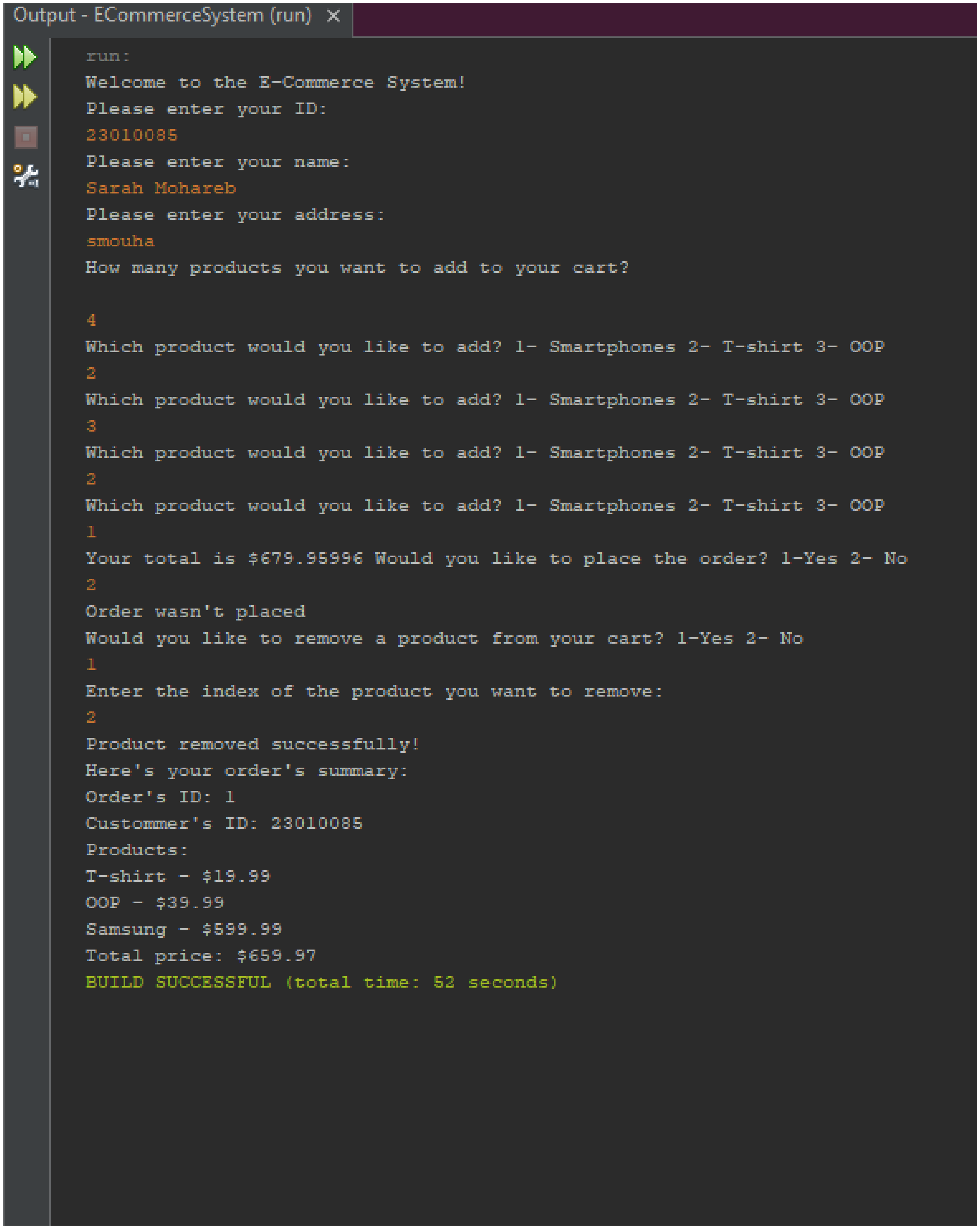
```java
            }
        });

        calculatePriceButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                totalPrice = 0;
                for (String product : selectedProducts) {
                    if (product != null) {
                        if (product.contains(s: "T-shirt")) {
                            totalPrice += 19.99;
                        } else if (product.contains(s: "Smartphone")) {
                            totalPrice += 599.9;
                        } else if (product.contains(s: "OOP Book")) {
                            totalPrice += 39.99;
                        }
                    }
                }
                totalPriceLabel.setText("Total Price: $" + totalPrice);
            }
        });

        printOrderButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                StringBuilder orderInfo = new StringBuilder();
                orderInfo.append(str:"Customer ID: ").append(str:customerIdField.getText()).append(str:"\n");
                orderInfo.append(str:"Name: ").append(str:nameField.getText()).append(str:"\n");
                orderInfo.append(str:"Address: ").append(str:addressField.getText()).append(str:"\n");
                orderInfo.append(str:"Selected Products:\n");
                for (String product : selectedProducts) {
                    if (product != null) {
                        orderInfo.append(str:"- ").append(str:product).append(str:"\n");
                    }
                }
```



```
                        }
                    }
                }
                totalPriceLabel.setText("Total Price: $" + totalPrice);
            }
        });

        printOrderButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                StringBuilder orderInfo = new StringBuilder();
                orderInfo.append(str:"Customer ID: ").append(str:customerIdField.getText()).append(str:"\n");
                orderInfo.append(str:"Name: ").append(str:nameField.getText()).append(str:"\n");
                orderInfo.append(str:"Address: ").append(str:addressField.getText()).append(str:"\n");
                orderInfo.append(str:"Selected Products:\n");
                for (String product : selectedProducts) {
                    if (product != null) {
                        orderInfo.append(str:"- ").append(str:product).append(str:"\n");
                    }
                }
                orderInfo.append(str:"Total Price: $").append(d: totalPrice);
                JOptionPane.showMessageDialog(parentComponent: null, message:orderInfo.toString(), title: "Order Information", messageType:JOptionPane.INFORMATION_MESSAGE);
            }
        });

        setVisible(b: true);
    }

    public static void main(String[] args) {
        SwingUtilities.invokeLater(() -> new ECommerceGUI());
    }
}
```



```
run:
Welcome to the E-Commerce System!
Please enter your ID:
23010085
Please enter your name:
Sarah Mohareb
Please enter your address:
smouha
How many products you want to add to your cart?
4
Which product would you like to add? 1- Smartphones 2- T-shirt 3- OOP
2
Which product would you like to add? 1- Smartphones 2- T-shirt 3- OOP
3
Which product would you like to add? 1- Smartphones 2- T-shirt 3- OOP
2
Which product would you like to add? 1- Smartphones 2- T-shirt 3- OOP
1
Your total is $679.95996 Would you like to place the order? 1-Yes 2- No
1
Here's your order's summary:
Order's ID: 1
Custommer's ID: 23010085
Products:
T-shirt - $19.99
OOP - $39.99
T-shirt - $19.99
Samsung - $599.99
Total price: $679.95996
your order is placed successfully!
Would you like to remove a product from your cart? 1-Yes 2- No
2
No products were removed.
Here's your order's summary:
Order's ID: 1
Custommer's ID: 23010085
Products:
T-shirt - $19.99
OOP - $39.99
T-shirt - $19.99
Samsung - $599.99
Total price: $679.95996
BUILD SUCCESSFUL (total time: 1 minute 15 seconds)
```



```
run:
Welcome to the E-Commerce System!
Please enter your ID:
23010085
Please enter your name:
Sarah Mohareb
Please enter your address:
smouha
How many products you want to add to your cart?

4
Which product would you like to add? 1- Smartphones 2- T-shirt 3- OOP
2
Which product would you like to add? 1- Smartphones 2- T-shirt 3- OOP
3
Which product would you like to add? 1- Smartphones 2- T-shirt 3- OOP
2
Which product would you like to add? 1- Smartphones 2- T-shirt 3- OOP
1
Your total is $679.95996 Would you like to place the order? 1-Yes 2- No
2
Order wasn't placed
Would you like to remove a product from your cart? 1-Yes 2- No
1
Enter the index of the product you want to remove:
2
Product removed successfully!
Here's your order's summary:
Order's ID: 1
Custommer's ID: 23010085
Products:
T-shirt - $19.99
OOP - $39.99
Samsung - $599.99
Total price: $659.97
BUILD SUCCESSFUL (total time: 52 seconds)
```