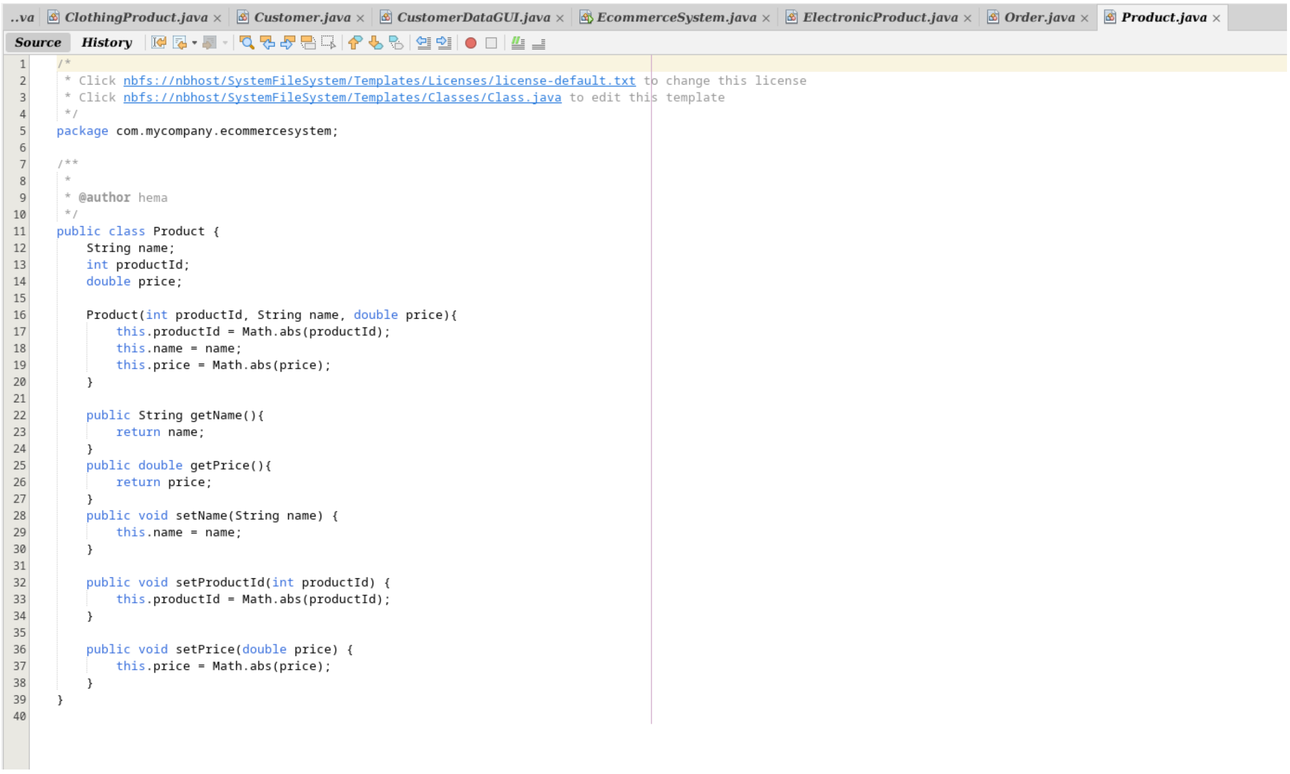
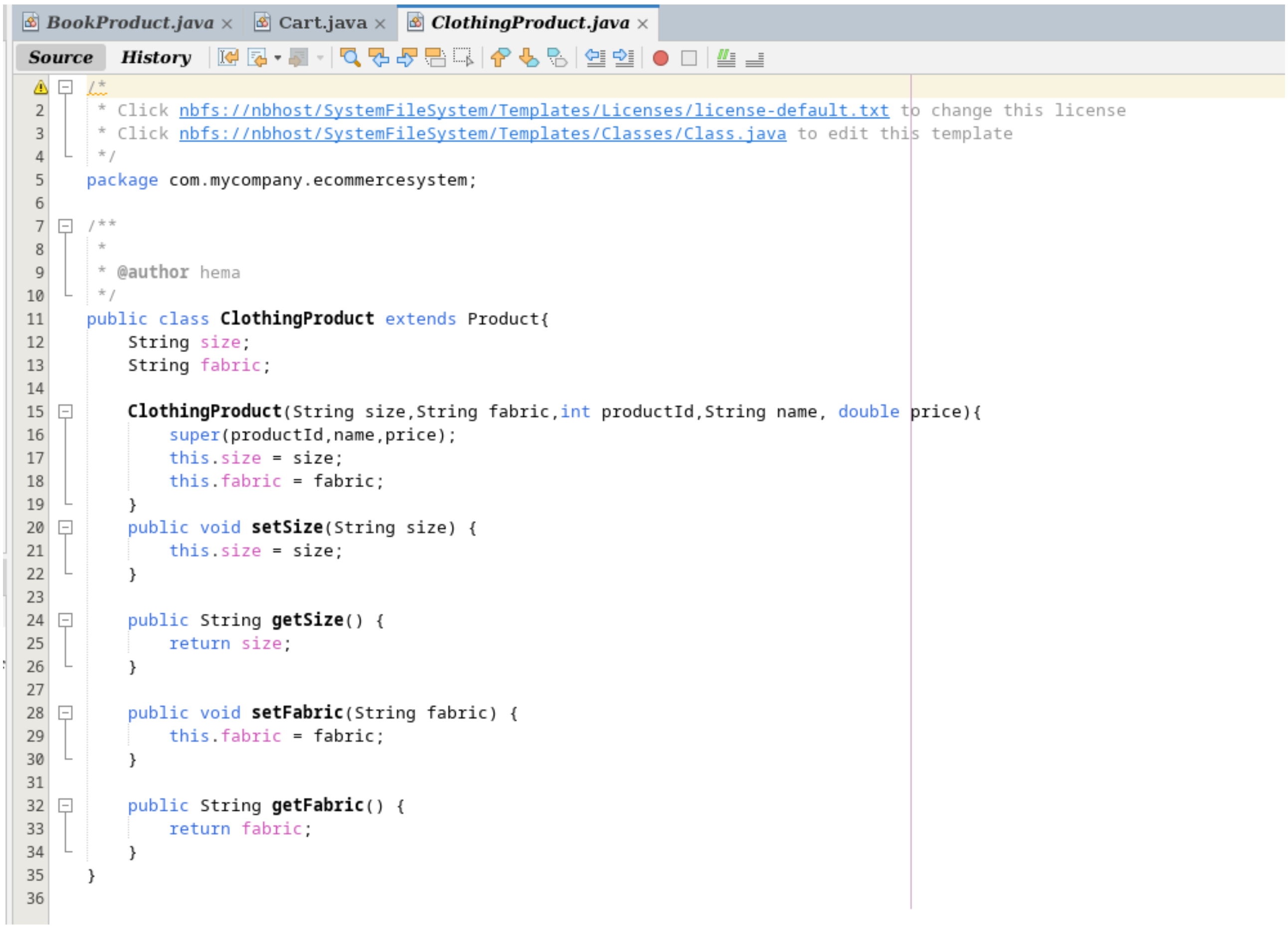


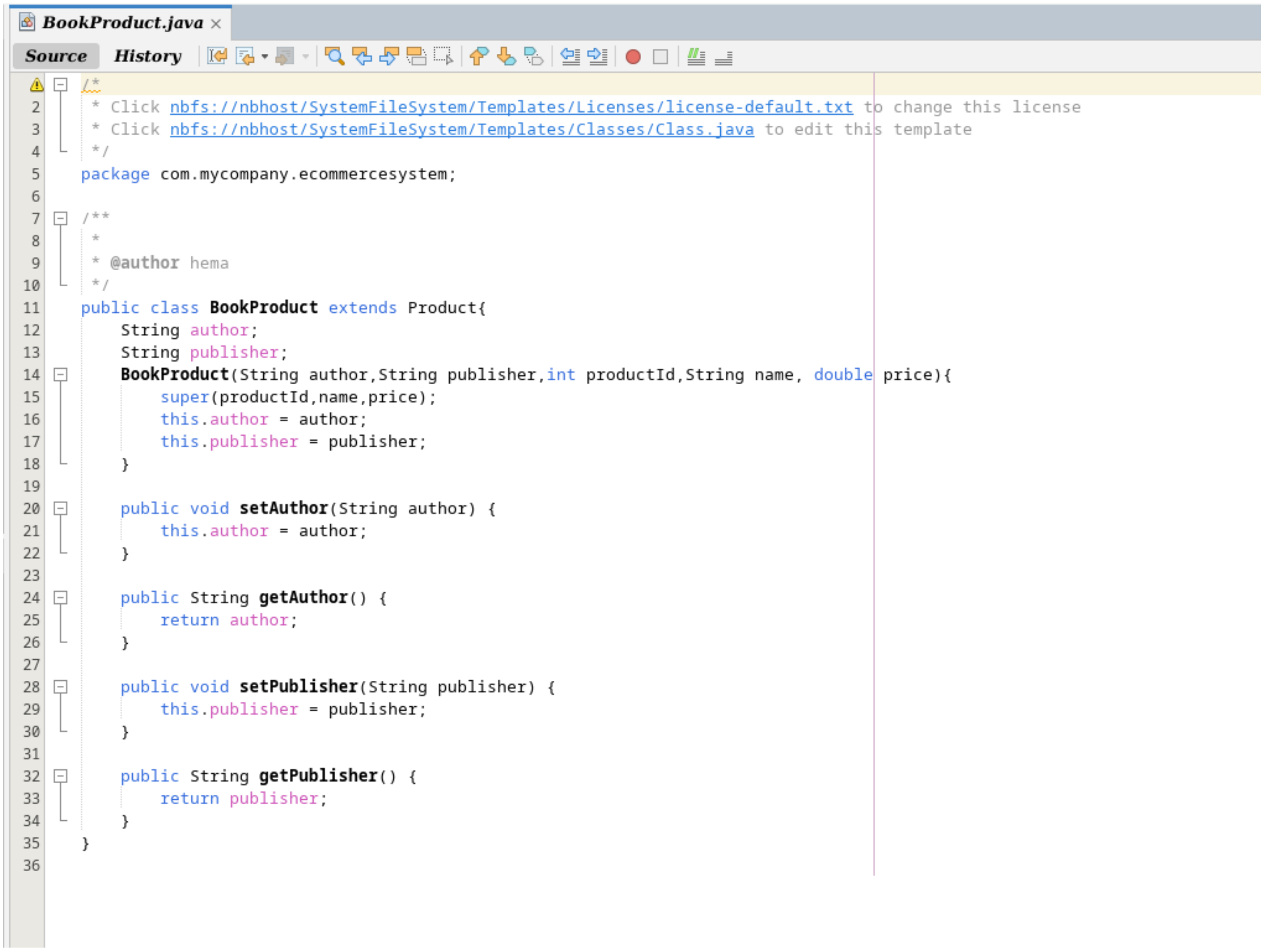
```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package com.mycompany.ecommercesystem;

/**
 *
 * @author hema
 */
public class Product {
    String name;
    int productId;
    double price;

    Product(int productId, String name, double price){
        this.productId = Math.abs(productId);
        this.name = name;
        this.price = Math.abs(price);
    }

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

    public void setProductId(int productId) {
        this.productId = Math.abs(productId);
    }

    public void setPrice(double price) {
        this.price = Math.abs(price);
    }
}
```

Product class



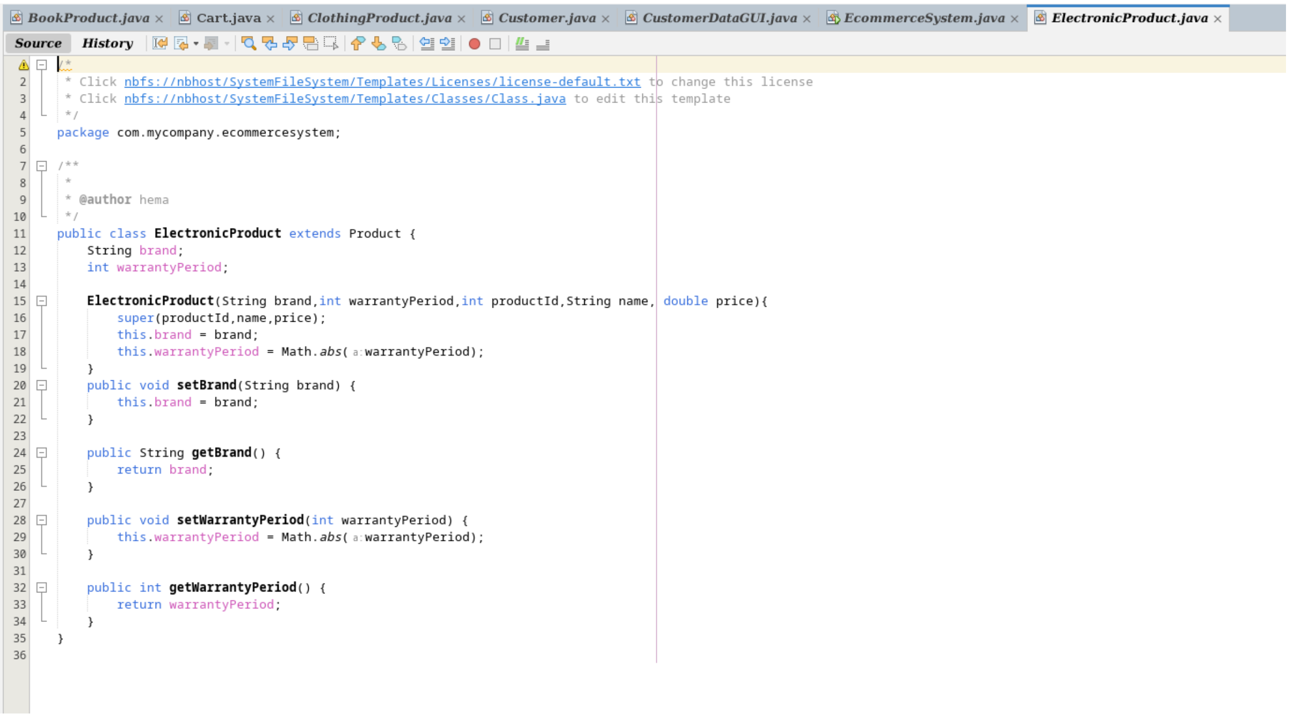
```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package com.mycompany.ecommercesystem;

/**
 *
 * @author hema
 */
public class ClothingProduct extends Product{
    String size;
    String fabric;

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

    public String getSize() {
        return size;
    }

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

    public String getFabric() {
        return fabric;
    }
}
```

Clothing Product class



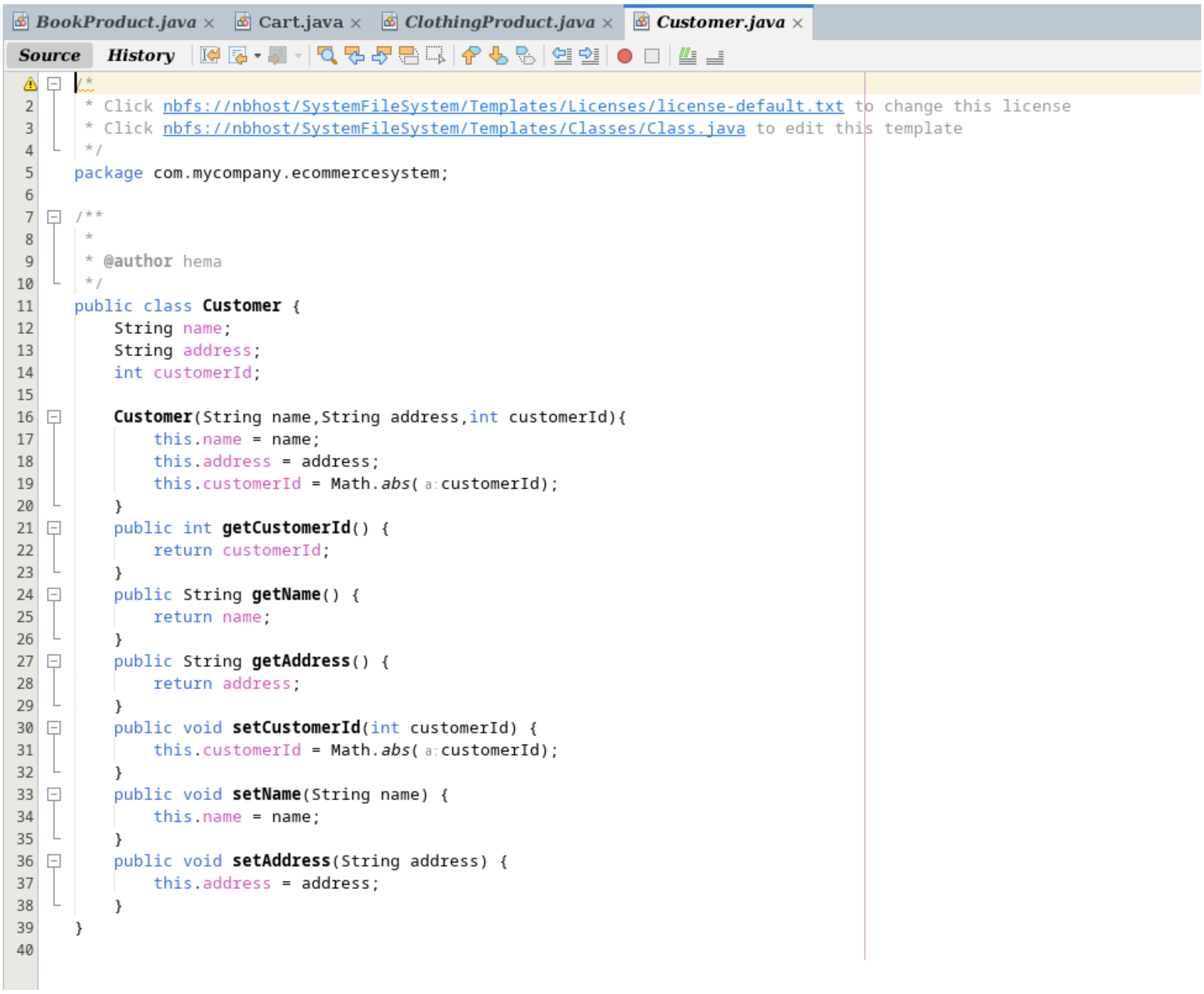
```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package com.mycompany.ecommercesystem;

/**
 *
 * @author hema
 */
public class BookProduct extends Product{
    String author;
    String publisher;
    BookProduct(String author,String publisher,int productId,String name, double price){
        super(productId,name,price);
        this.author = author;
        this.publisher = publisher;
    }

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

    public String getAuthor() {
        return author;
    }

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

    public String getPublisher() {
        return publisher;
    }
}
```

Book Product class



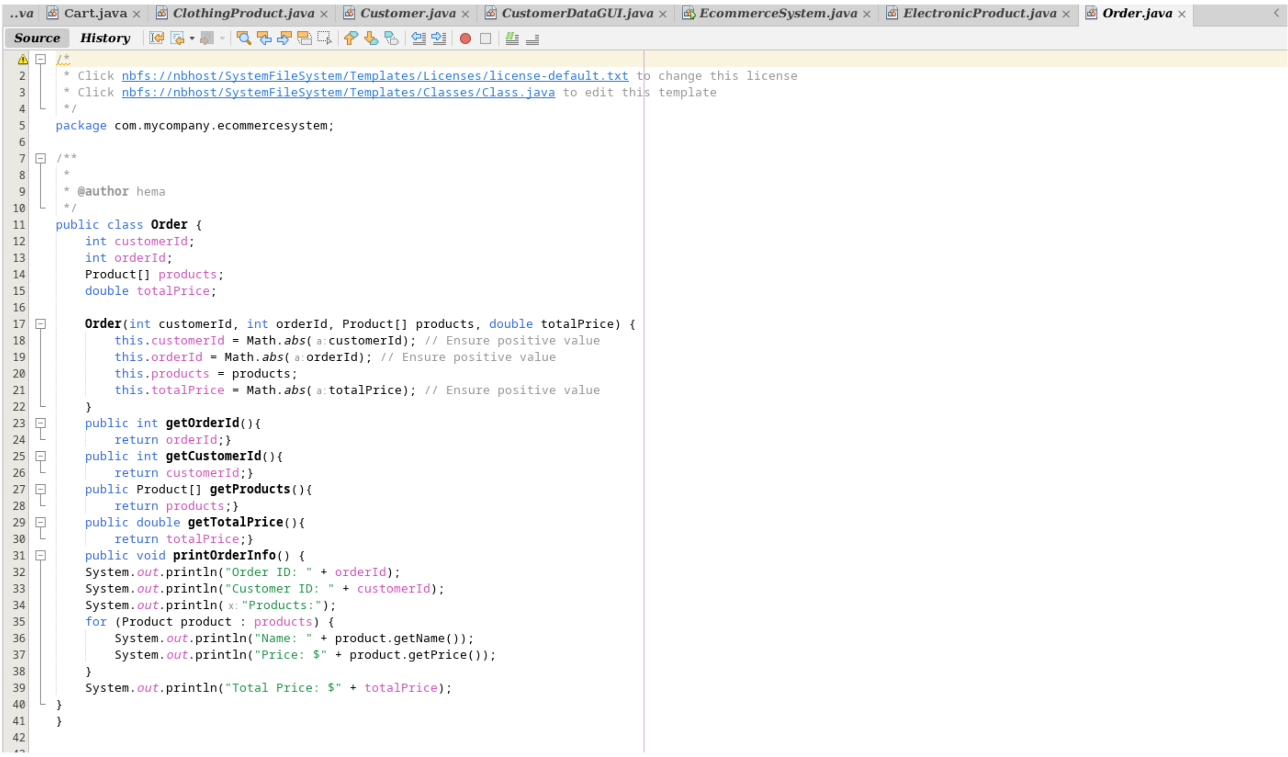
```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package com.mycompany.ecommercesystem;

/**
 *
 * @author hema
 */
public class ElectronicProduct extends Product {
    String brand;
    int warrantyPeriod;

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

    public String getBrand() {
        return brand;
    }

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

    public int getWarrantyPeriod() {
        return warrantyPeriod;
    }
}
```

Electronic Product class



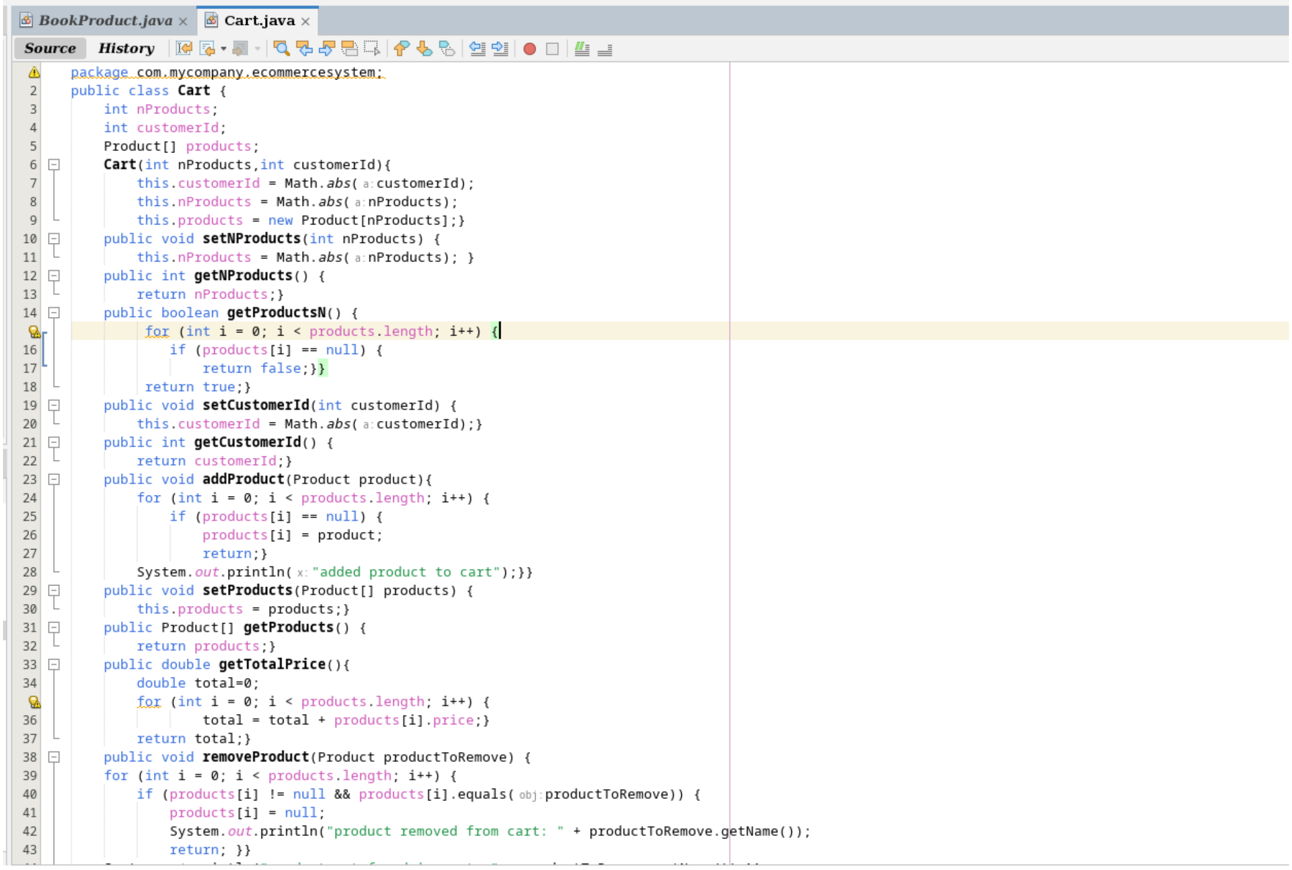
```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package com.mycompany.ecommercesystem;

/**
 *
 * @author hema
 */
public class Customer {
    String name;
    String address;
    int customerId;

    Customer(String name,String address,int customerId){
        this.name = name;
        this.address = address;
        this.customerId = Math.abs( a:customerId);
    }
    public int getCustomerId() {
        return customerId;
    }
    public String getName() {
        return name;
    }
    public String getAddress() {
        return address;
    }
    public void setCustomerId(int customerId) {
        this.customerId = Math.abs( a:customerId);
    }
    public void setName(String name) {
        this.name = name;
    }
    public void setAddress(String address) {
        this.address = address;
    }
}
```

Customer Class

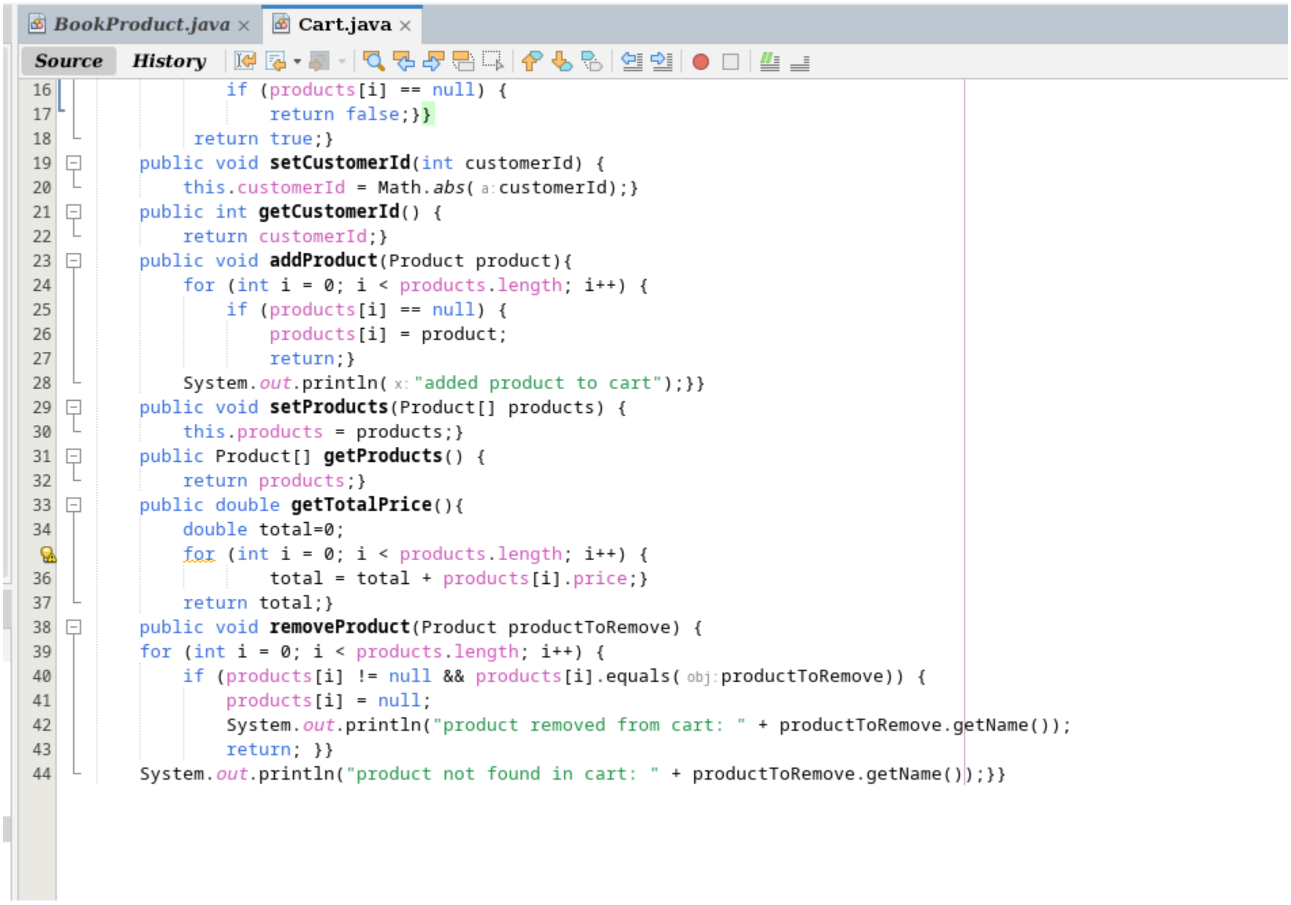


```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package com.mycompany.ecommercesystem;

/**
 *
 * @author hema
 */
public class Order {
    int customerId;
    int orderId;
    Product[] products;
    double totalPrice;

    Order(int customerId, int orderId, Product[] products, double totalPrice) {
        this.customerId = Math.abs( a:customerId); // Ensure positive value
        this.orderId = Math.abs( a:orderId); // Ensure positive value
        this.products = products;
        this.totalPrice = Math.abs( a:totalPrice); // Ensure positive value
    }
    public int getOrderId(){
        return orderId;}
    public int getCustomerId(){
        return customerId;}
    public Product[] getProducts(){
        return products;}
    public double getTotalPrice(){
        return totalPrice;}
    public void printOrderInfo() {
    System.out.println("Order ID: " + orderId);
    System.out.println("Customer ID: " + customerId);
    System.out.println( x:"Products:");
    for (Product product : products) {
        System.out.println("Name: " + product.getName());
        System.out.println("Price: $" + product.getPrice());
    }
    System.out.println("Total Price: $" + totalPrice);
 }
 }
```

Order Class



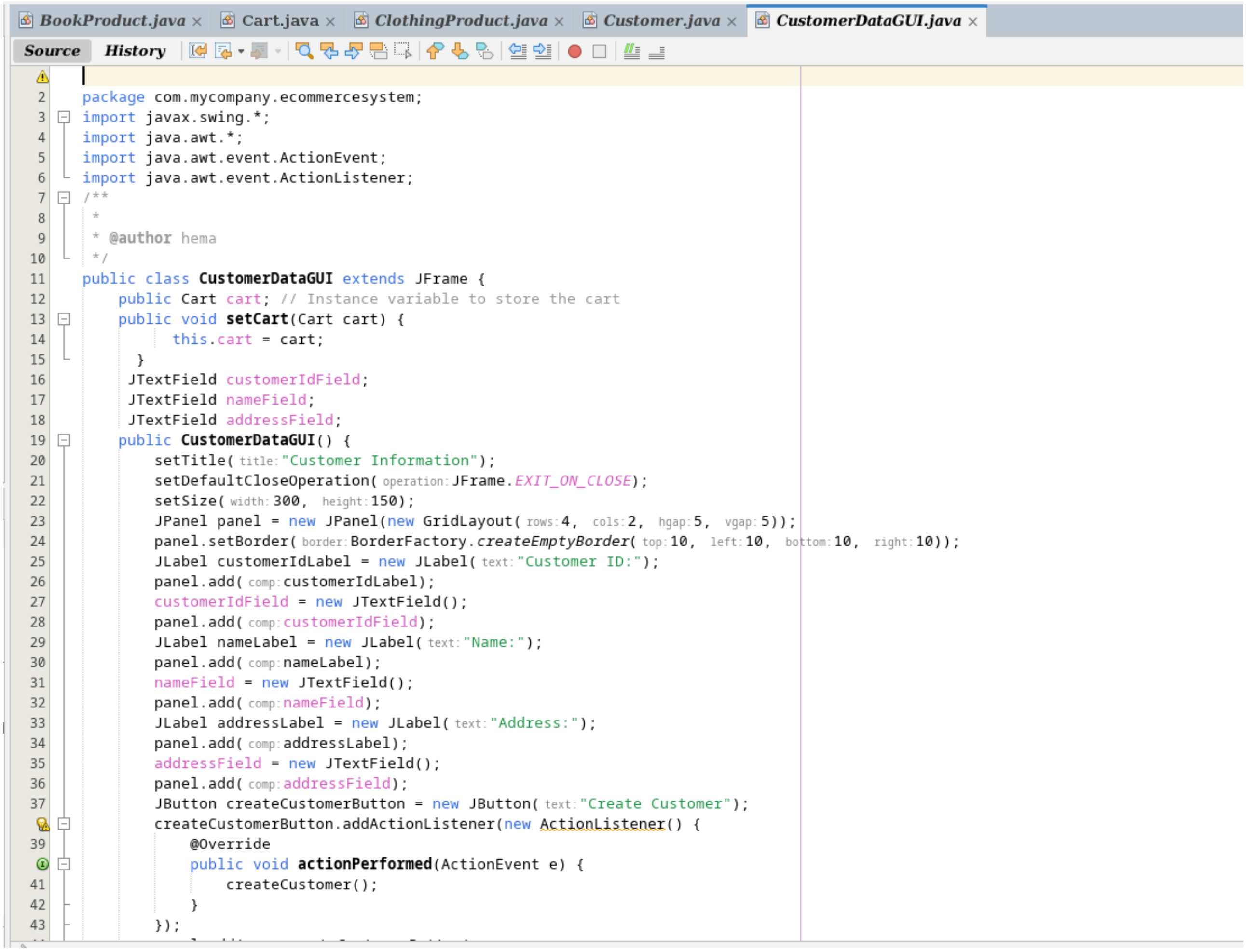
```
package com.mycompany.ecommercesystem;
public class Cart {
    int nProducts;
    int customerId;
    Product[] products;
    Cart(int nProducts,int customerId){
        this.customerId = Math.abs( a: customerId);
        this.nProducts = Math.abs( a: nProducts);
        this.products = new Product[nProducts];}
    public void setNProducts(int nProducts) {
        this.nProducts = Math.abs( a: nProducts); }
    public int getNProducts() {
        return nProducts;}
    public boolean getProductsN() {
         for (int i = 0; i < products.length; i++) {
            if (products[i] == null) {
                return false;}}
         return true;}
    public void setCustomerId(int customerId) {
        this.customerId = Math.abs( a: customerId);}
    public int getCustomerId() {
        return customerId;}
    public void addProduct(Product product){
        for (int i = 0; i < products.length; i++) {
            if (products[i] == null) {
                products[i] = product;
                return;}
        System.out.println( x: "added product to cart");}}
    public void setProducts(Product[] products) {
        this.products = products;}
    public Product[] getProducts() {
        return products;}
    public double getTotalPrice(){
        double total=0;
        for (int i = 0; i < products.length; i++) {
                total = total + products[i].price;}
        return total;}
    public void removeProduct(Product productToRemove) {
    for (int i = 0; i < products.length; i++) {
        if (products[i] != null && products[i].equals( obj: productToRemove)) {
            products[i] = null;
            System.out.println("product removed from cart: " + productToRemove.getName());
            return; }}
```



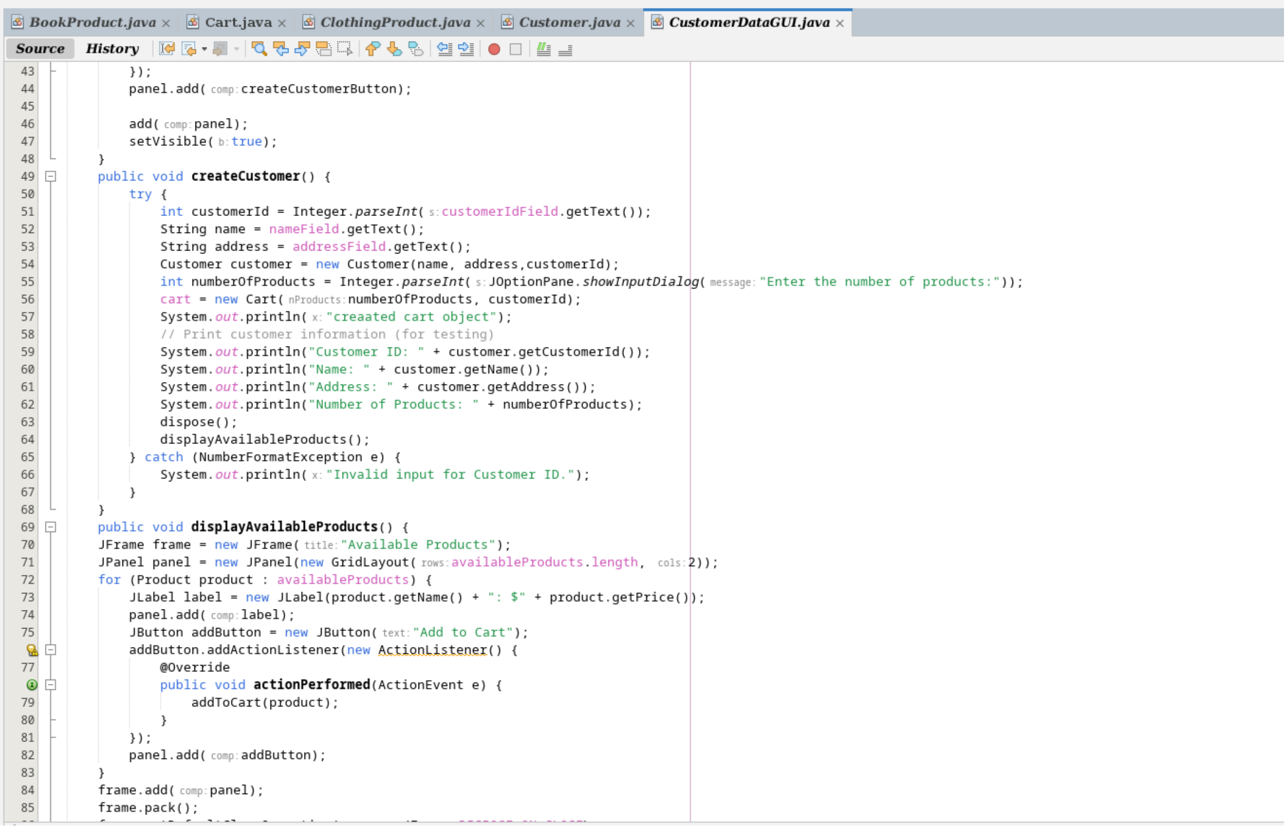
```
            if (products[i] == null) {
                return false;}}
         return true;}
    public void setCustomerId(int customerId) {
        this.customerId = Math.abs( a: customerId);}
    public int getCustomerId() {
        return customerId;}
    public void addProduct(Product product){
        for (int i = 0; i < products.length; i++) {
            if (products[i] == null) {
                products[i] = product;
                return;}
        System.out.println( x: "added product to cart");}}
    public void setProducts(Product[] products) {
        this.products = products;}
    public Product[] getProducts() {
        return products;}
    public double getTotalPrice(){
        double total=0;
        for (int i = 0; i < products.length; i++) {
                total = total + products[i].price;}
        return total;}
    public void removeProduct(Product productToRemove) {
    for (int i = 0; i < products.length; i++) {
        if (products[i] != null && products[i].equals( obj: productToRemove)) {
            products[i] = null;
            System.out.println("product removed from cart: " + productToRemove.getName());
            return; }}
```

```
    System.out.println("product not found in cart: " + productToRemove.getName());}}
```

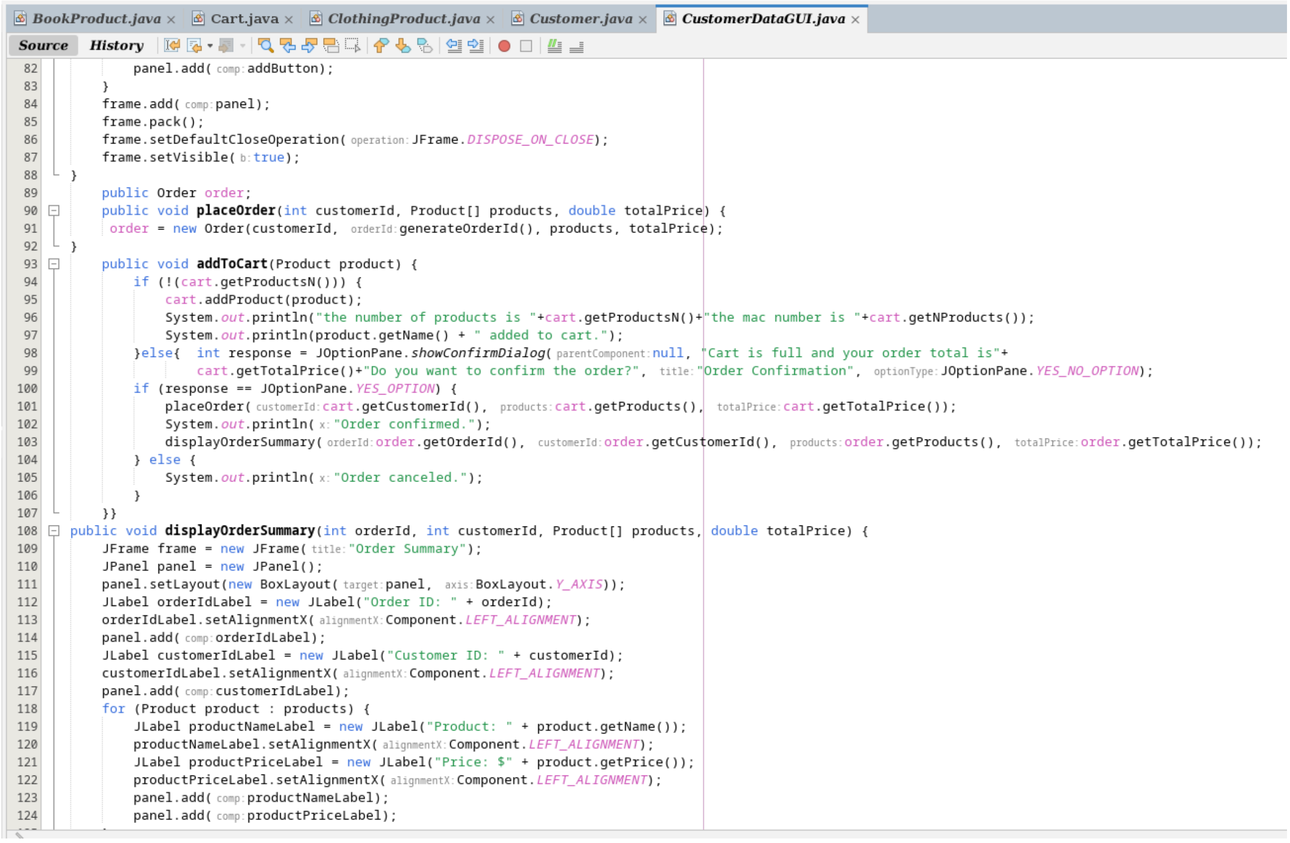
Cart class (Two images)



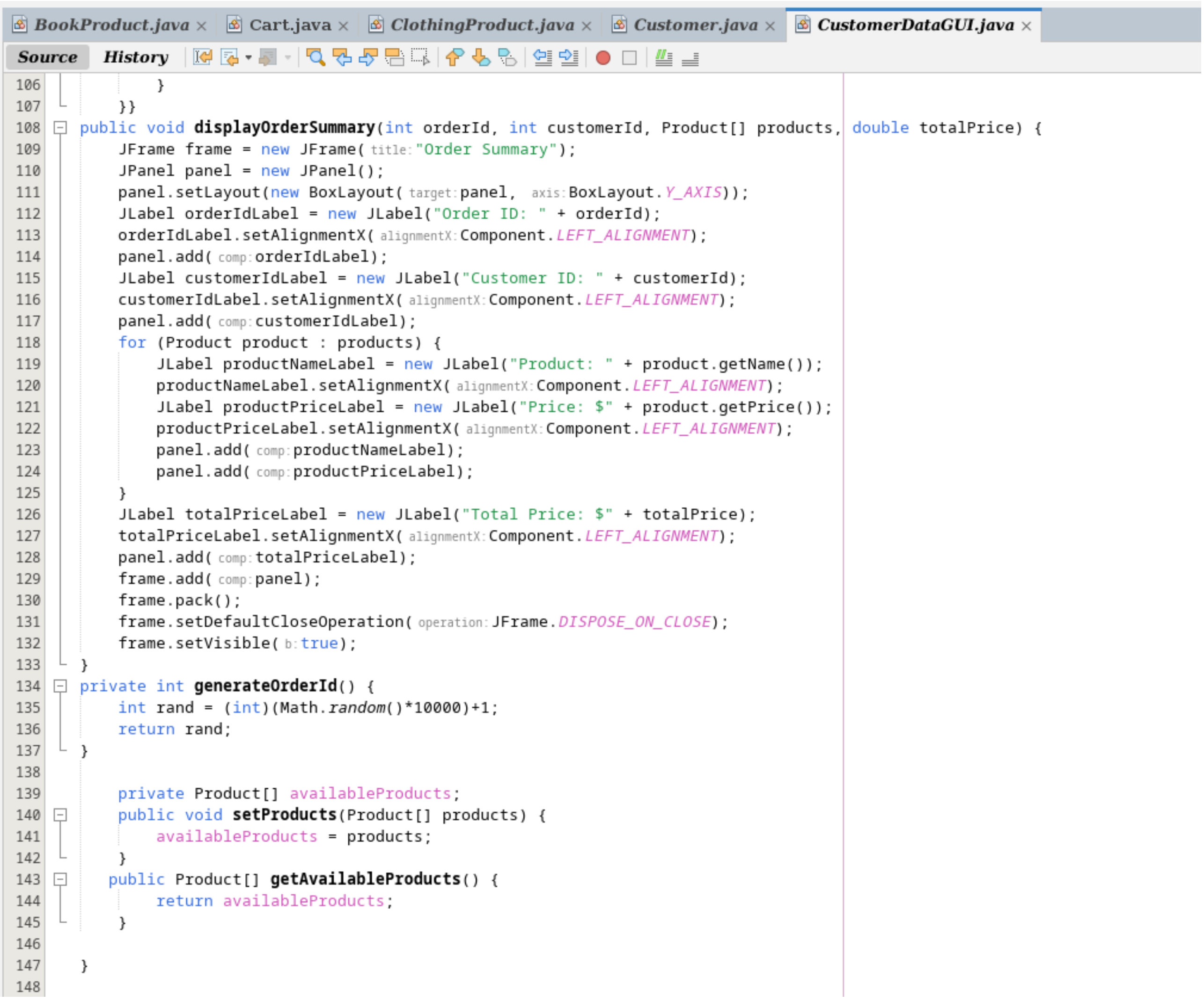
```java
package com.mycompany.ecommercesystem;
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
/**
 *
 * @author hema
 */
public class CustomerDataGUI extends JFrame {
    public Cart cart; // Instance variable to store the cart
    public void setCart(Cart cart) {
        this.cart = cart;
     }
     JTextField customerIdField;
     JTextField nameField;
     JTextField addressField;
    public CustomerDataGUI() {
        setTitle( title:"Customer Information");
        setDefaultCloseOperation( operation: JFrame.EXIT_ON_CLOSE);
        setSize( width: 300,  height: 150);
        JPanel panel = new JPanel(new GridLayout( rows: 4,  cols: 2,  hgap: 5,  vgap: 5));
        panel.setBorder( border: BorderFactory.createEmptyBorder( top: 10,  left: 10,  bottom: 10,  right: 10));
        JLabel customerIdLabel = new JLabel( text: "Customer ID:");
        panel.add( comp: customerIdLabel);
        customerIdField = new JTextField();
        panel.add( comp: customerIdField);
        JLabel nameLabel = new JLabel( text: "Name:");
        panel.add( comp: nameLabel);
        nameField = new JTextField();
        panel.add( comp: nameField);
        JLabel addressLabel = new JLabel( text: "Address:");
        panel.add( comp: addressLabel);
        addressField = new JTextField();
        panel.add( comp: addressField);
        JButton createCustomerButton = new JButton( text: "Create Customer");
        createCustomerButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                createCustomer();
            }
        });
```



```java
        });
        panel.add( comp: createCustomerButton);

        add( comp: panel);
        setVisible( b: true);
    }
    public void createCustomer() {
        try {
            int customerId = Integer.parseInt( s: customerIdField.getText());
            String name = nameField.getText();
            String address = addressField.getText();
            Customer customer = new Customer(name, address,customerId);
            int numberOfProducts = Integer.parseInt( s: JOptionPane.showInputDialog( message: "Enter the number of products:"));
            cart = new Cart( nProducts: numberOfProducts, customerId);
            System.out.println( x: "creaated cart object");
            // Print customer information (for testing)
            System.out.println("Customer ID: " + customer.getCustomerId());
            System.out.println("Name: " + customer.getName());
            System.out.println("Address: " + customer.getAddress());
            System.out.println("Number of Products: " + numberOfProducts);
            dispose();
            displayAvailableProducts();
        } catch (NumberFormatException e) {
            System.out.println( x: "Invalid input for Customer ID.");
        }
    }
    public void displayAvailableProducts() {
    JFrame frame = new JFrame( title: "Available Products");
    JPanel panel = new JPanel(new GridLayout( rows: availableProducts.length,  cols: 2));
    for (Product product : availableProducts) {
        JLabel label = new JLabel(product.getName() + ": $" + product.getPrice());
        panel.add( comp: label);
        JButton addButton = new JButton( text: "Add to Cart");
        addButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                addToCart(product);
            }
        });
        panel.add( comp: addButton);
    }
    frame.add( comp: panel);
    frame.pack();
```

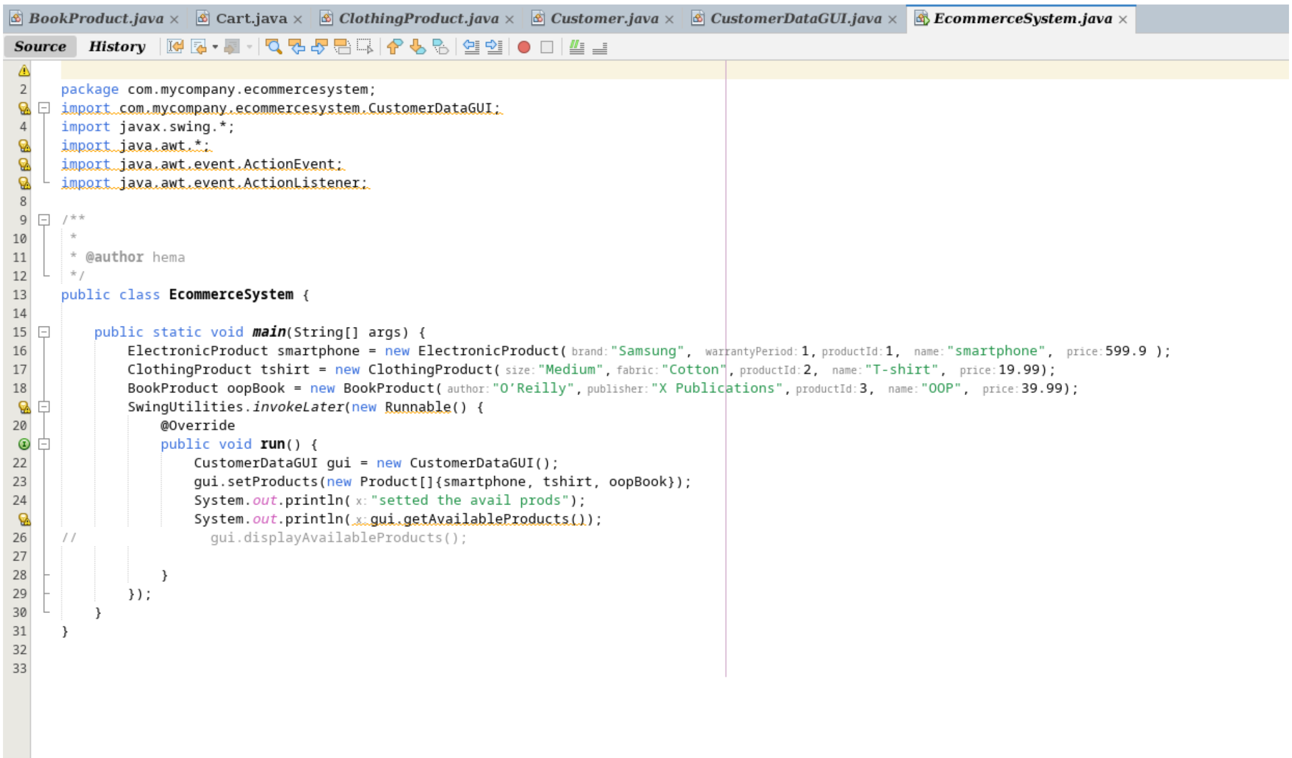


```
            panel.add( comp: addButton);
        }
        frame.add( comp: panel);
        frame.pack();
        frame.setDefaultCloseOperation( operation: JFrame.DISPOSE_ON_CLOSE);
        frame.setVisible( b: true);
}
        public Order order;
        public void placeOrder(int customerId, Product[] products, double totalPrice) {
         order = new Order(customerId,  orderId: generateOrderId(), products, totalPrice);
}
        public void addToCart(Product product) {
            if (!(cart.getProductsN())) {
                cart.addProduct(product);
                System.out.println("the number of products is "+cart.getProductsN()+"the mac number is "+cart.getNProducts());
                System.out.println(product.getName() + " added to cart.");
            }else{  int response = JOptionPane.showConfirmDialog( parentComponent: null, "Cart is full and your order total is"+
                    cart.getTotalPrice()+"Do you want to confirm the order?",  title: "Order Confirmation",  optionType: JOptionPane.YES_NO_OPTION);
            if (response == JOptionPane.YES_OPTION) {
                placeOrder( customerId: cart.getCustomerId(),  products: cart.getProducts(),  totalPrice: cart.getTotalPrice());
                System.out.println( x: "Order confirmed.");
                displayOrderSummary( orderId: order.getOrderId(),  customerId: order.getCustomerId(),  products: order.getProducts(),  totalPrice: order.getTotalPrice());
            } else {
                System.out.println( x: "Order canceled.");
            }
        }}
public void displayOrderSummary(int orderId, int customerId, Product[] products, double totalPrice) {
        JFrame frame = new JFrame( title: "Order Summary");
        JPanel panel = new JPanel();
        panel.setLayout(new BoxLayout( target: panel,  axis: BoxLayout.Y_AXIS));
        JLabel orderIdLabel = new JLabel("Order ID: " + orderId);
        orderIdLabel.setAlignmentX( alignmentX: Component.LEFT_ALIGNMENT);
        panel.add( comp: orderIdLabel);
        JLabel customerIdLabel = new JLabel("Customer ID: " + customerId);
        customerIdLabel.setAlignmentX( alignmentX: Component.LEFT_ALIGNMENT);
        panel.add( comp: customerIdLabel);
        for (Product product : products) {
            JLabel productNameLabel = new JLabel("Product: " + product.getName());
            productNameLabel.setAlignmentX( alignmentX: Component.LEFT_ALIGNMENT);
            JLabel productPriceLabel = new JLabel("Price: $" + product.getPrice());
            productPriceLabel.setAlignmentX( alignmentX: Component.LEFT_ALIGNMENT);
            panel.add( comp: productNameLabel);
            panel.add( comp: productPriceLabel);
```



```
            }
        }}
public void displayOrderSummary(int orderId, int customerId, Product[] products, double totalPrice) {
        JFrame frame = new JFrame( title: "Order Summary");
        JPanel panel = new JPanel();
        panel.setLayout(new BoxLayout( target: panel,  axis: BoxLayout.Y_AXIS));
        JLabel orderIdLabel = new JLabel("Order ID: " + orderId);
        orderIdLabel.setAlignmentX( alignmentX: Component.LEFT_ALIGNMENT);
        panel.add( comp: orderIdLabel);
        JLabel customerIdLabel = new JLabel("Customer ID: " + customerId);
        customerIdLabel.setAlignmentX( alignmentX: Component.LEFT_ALIGNMENT);
        panel.add( comp: customerIdLabel);
        for (Product product : products) {
            JLabel productNameLabel = new JLabel("Product: " + product.getName());
            productNameLabel.setAlignmentX( alignmentX: Component.LEFT_ALIGNMENT);
            JLabel productPriceLabel = new JLabel("Price: $" + product.getPrice());
            productPriceLabel.setAlignmentX( alignmentX: Component.LEFT_ALIGNMENT);
            panel.add( comp: productNameLabel);
            panel.add( comp: productPriceLabel);
        }
        JLabel totalPriceLabel = new JLabel("Total Price: $" + totalPrice);
        totalPriceLabel.setAlignmentX( alignmentX: Component.LEFT_ALIGNMENT);
        panel.add( comp: totalPriceLabel);
        frame.add( comp: panel);
        frame.pack();
        frame.setDefaultCloseOperation( operation: JFrame.DISPOSE_ON_CLOSE);
        frame.setVisible( b: true);
}
private int generateOrderId() {
        int rand = (int)(Math.random()*10000)+1;
        return rand;
}

        private Product[] availableProducts;
        public void setProducts(Product[] products) {
            availableProducts = products;
        }
       public Product[] getAvailableProducts() {
            return availableProducts;
        }

}
```

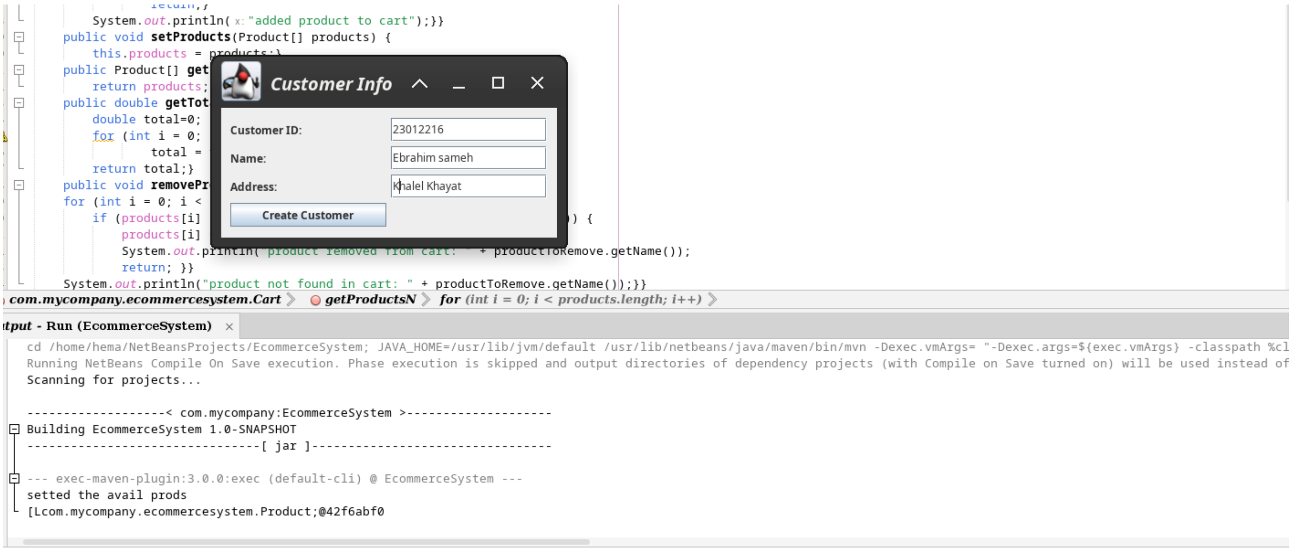
Gui class (4 images)



```java

package com.mycompany.ecommercesystem;
import com.mycompany.ecommercesystem.CustomerDataGUI;
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

/**
 *
 * @author hema
 */
public class EcommerceSystem {

    public static void main(String[] args) {
        ElectronicProduct smartphone = new ElectronicProduct( brand: "Samsung",  warrantyPeriod: 1, productId: 1,  name: "smartphone",  price: 599.9 );
        ClothingProduct tshirt = new ClothingProduct( size: "Medium", fabric: "Cotton", productId: 2,  name: "T-shirt",  price: 19.99);
        BookProduct oopBook = new BookProduct( author: "O'Reilly", publisher: "X Publications", productId: 3,  name: "OOP",  price: 39.99);
        SwingUtilities.invokeLater(new Runnable() {
            @Override
            public void run() {
                CustomerDataGUI gui = new CustomerDataGUI();
                gui.setProducts(new Product[]{smartphone, tshirt, oopBook});
                System.out.println( x: "setted the avail prods");
                System.out.println( x: gui.getAvailableProducts());
//                  gui.displayAvailableProducts();

            }
        });
    }
}


```

Ecommerce class (Main method)

The test cases showing the gui and the Console output:



```
        System.out.println( x: "added product to cart");}}
    public void setProducts(Product[] products) {
        this.products = products;}
    public Product[] get
        return products;
    public double getTot
        double total=0;
        for (int i = 0;
            total =
        return total;}
    public void removePr
    for (int i = 0; i <
        if (products[i]                                              )) {
            products[i]
            System.out.println("product removed from cart: " + productToRemove.getName());
            return; }}
    System.out.println("product not found in cart: " + productToRemove.getName());}}
```

com.mycompany.ecommercesystem.Cart 〉 getProductsN 〉 for (int i = 0; i < products.length; i++) 〉

Customer Info

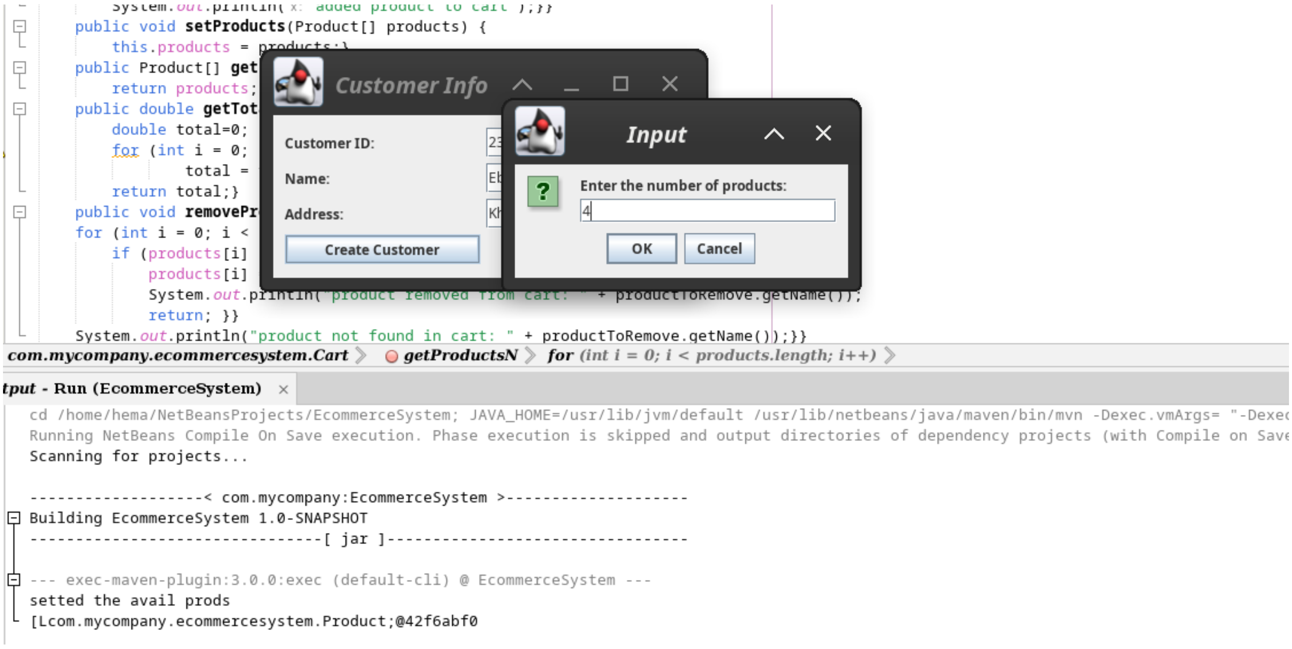
Customer ID: 23012216

Name: Ebrahim sameh

Address: Khalel Khayat

tput - Run (EcommerceSystem)

```
cd /home/hema/NetBeansProjects/EcommerceSystem; JAVA_HOME=/usr/lib/jvm/default /usr/lib/netbeans/java/maven/bin/mvn -Dexec.vmArgs= "-Dexec.args=${exec.vmArgs} -classpath %cl
Running NetBeans Compile On Save execution. Phase execution is skipped and output directories of dependency projects (with Compile on Save turned on) will be used instead of
Scanning for projects...

-------------------< com.mycompany:EcommerceSystem >--------------------
Building EcommerceSystem 1.0-SNAPSHOT
--------------------------------[ jar ]---------------------------------

--- exec-maven-plugin:3.0.0:exec (default-cli) @ EcommerceSystem ---
setted the avail prods
[Lcom.mycompany.ecommercesystem.Product;@42f6abf0
```



```
        System.out.println( x: "added product to cart");}}
    public void setProducts(Product[] products) {
        this.products = products;}
    public Product[] get
        return products;
    public double getTot
        double total=0;
        for (int i = 0;
            total =
        return total;}
    public void removePr
    for (int i = 0; i <
        if (products[i]
            products[i]
            System.out.println("product removed from cart: " + productToRemove.getName());
            return; }}
    System.out.println("product not found in cart: " + productToRemove.getName());}}
```

com.mycompany.ecommercesystem.Cart 〉 getProductsN 〉 for (int i = 0; i < products.length; i++) 〉

Customer Info

Customer ID: 23

Name: Eb

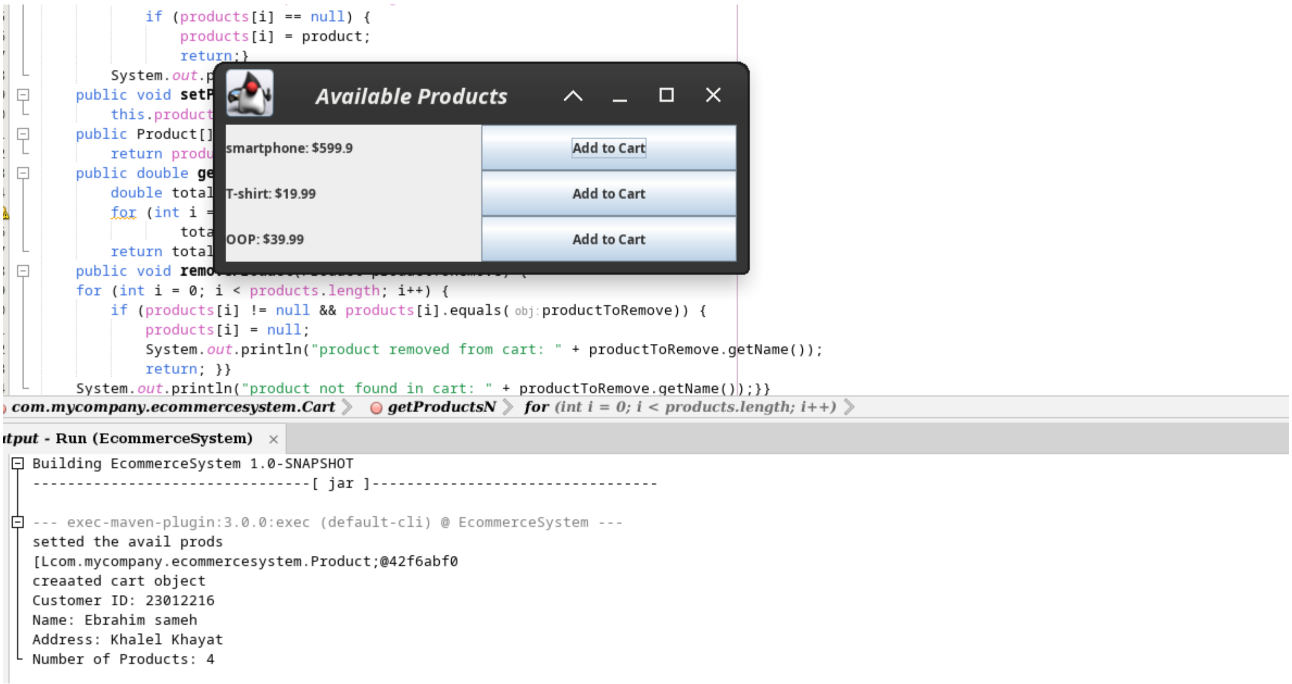
Address: Kh

Input

Enter the number of products: 4

tput - Run (EcommerceSystem)

```
cd /home/hema/NetBeansProjects/EcommerceSystem; JAVA_HOME=/usr/lib/jvm/default /usr/lib/netbeans/java/maven/bin/mvn -Dexec.vmArgs= "-Dexec
Running NetBeans Compile On Save execution. Phase execution is skipped and output directories of dependency projects (with Compile on Save
Scanning for projects...

-------------------< com.mycompany:EcommerceSystem >--------------------
Building EcommerceSystem 1.0-SNAPSHOT
--------------------------------[ jar ]---------------------------------

--- exec-maven-plugin:3.0.0:exec (default-cli) @ EcommerceSystem ---
setted the avail prods
[Lcom.mycompany.ecommercesystem.Product;@42f6abf0
```



```
            if (products[i] == null) {
                products[i] = product;
                return;}
        System.out.p
    public void setP
        this.product
    public Product[]
        return produ
    public double ge
        double total
        for (int i =
            tota
        return total
    public void remo
    for (int i = 0; i < products.length; i++) {
        if (products[i] != null && products[i].equals( obj: productToRemove)) {
            products[i] = null;
            System.out.println("product removed from cart: " + productToRemove.getName());
            return; }}
    System.out.println("product not found in cart: " + productToRemove.getName());}}
```

com.mycompany.ecommercesystem.Cart 〉 getProductsN 〉 for (int i = 0; i < products.length; i++) 〉

Available Products

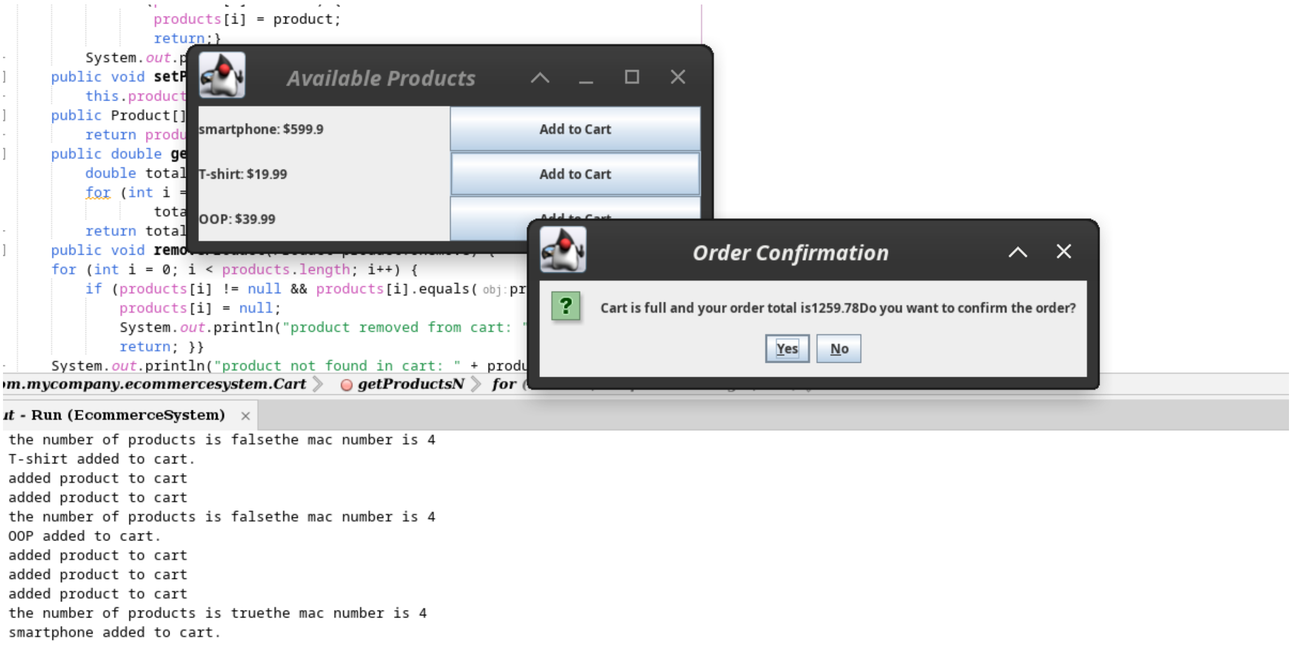
smartphone: $599.9

T-shirt: $19.99

OOP: $39.99

tput - Run (EcommerceSystem)

```
Building EcommerceSystem 1.0-SNAPSHOT
--------------------------------[ jar ]---------------------------------

--- exec-maven-plugin:3.0.0:exec (default-cli) @ EcommerceSystem ---
setted the avail prods
[Lcom.mycompany.ecommercesystem.Product;@42f6abf0
creaated cart object
Customer ID: 23012216
Name: Ebrahim sameh
Address: Khalel Khayat
Number of Products: 4
```



```
                products[i] = product;
                return;}
        System.out.p
    public void setP
        this.product
    public Product[]
        return produ
    public double ge
        double total
        for (int i =
                tota
        return total
    public void remo
    for (int i = 0; i < products.length; i++) {
        if (products[i] != null && products[i].equals( obj: pr
            products[i] = null;
            System.out.println("product removed from cart: "
            return; }}
    System.out.println("product not found in cart: " + produ
```

om.mycompany.ecommercesystem.Cart > getProductsN > for (
Available Products
Add to Cart
Add to Cart
Add to Cart
Order Confirmation
Yes
No
ut - Run (EcommerceSystem)

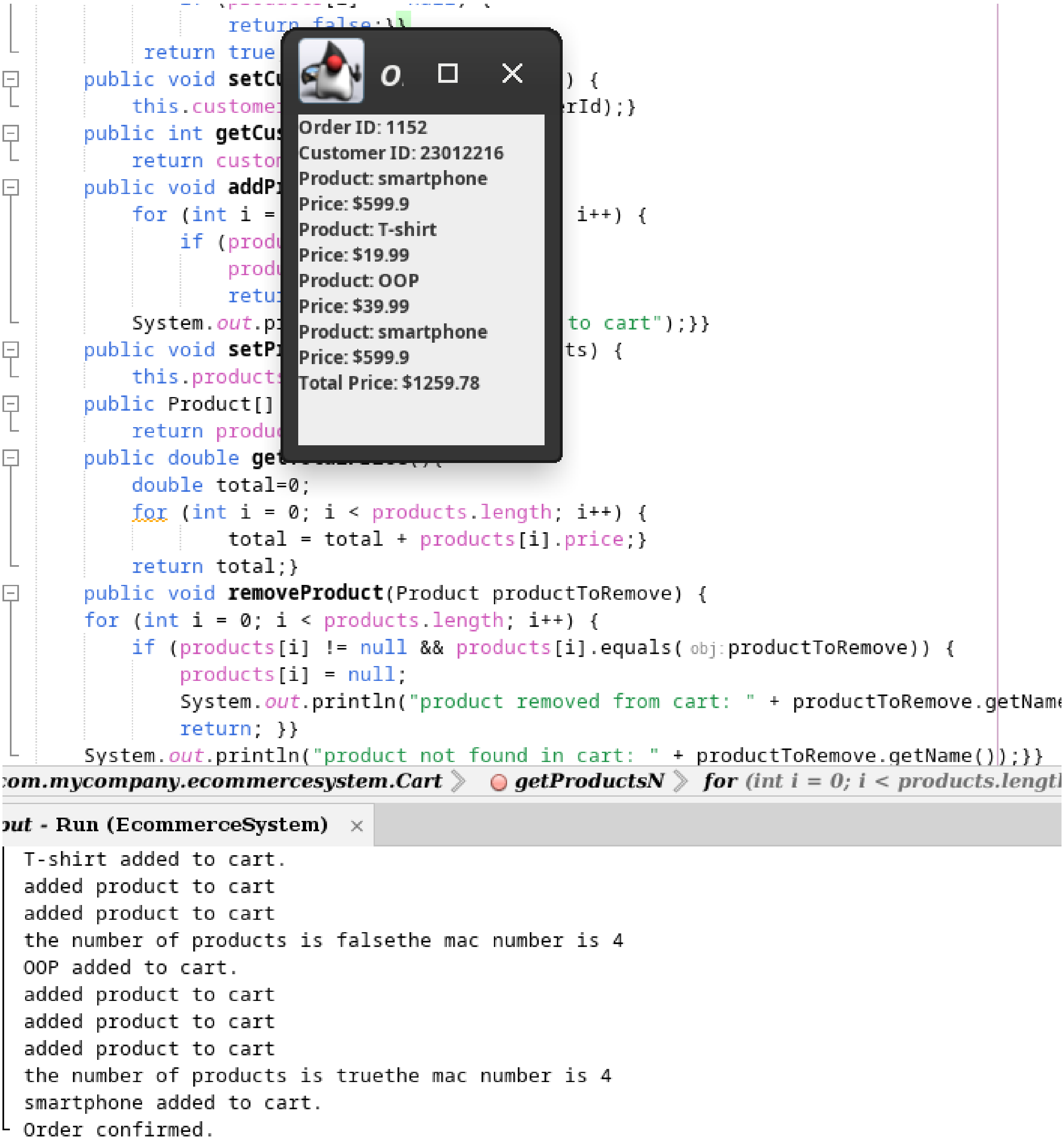
smartphone: $599.9

T-shirt: $19.99

OOP: $39.99

Cart is full and your order total is1259.78Do you want to confirm the order?

```
the number of products is falsethe mac number is 4
T-shirt added to cart.
added product to cart
added product to cart
the number of products is falsethe mac number is 4
OOP added to cart.
added product to cart
added product to cart
added product to cart
the number of products is truethe mac number is 4
smartphone added to cart.
```



```
            return false;}}
        return true
    public void setCu                    ) {
        this.custome                  rId);}
    public int getCus
        return custon
    public void addPr
        for (int i =                  i++) {
            if (prod
                prod
                retu
        System.out.p                   to cart");}}
    public void setPr                  ts) {
        this.products
    public Product[]
        return produ
    public double get
        double total=0;
        for (int i = 0; i < products.length; i++) {
                total = total + products[i].price;}
        return total;}
    public void removeProduct(Product productToRemove) {
    for (int i = 0; i < products.length; i++) {
        if (products[i] != null && products[i].equals( obj: productToRemove)) {
            products[i] = null;
            System.out.println("product removed from cart: " + productToRemove.getName
            return; }}
    System.out.println("product not found in cart: " + productToRemove.getName());}}
```

com.mycompany.ecommercesystem.Cart > getProductsN > for (int i = 0; i < products.lengt
O
out - Run (EcommerceSystem)

Order ID: 1152
Customer ID: 23012216
Product: smartphone
Price: $599.9
Product: T-shirt
Price: $19.99
Product: OOP
Price: $39.99
Product: smartphone
Price: $599.9
Total Price: $1259.78

```
T-shirt added to cart.
added product to cart
added product to cart
the number of products is falsethe mac number is 4
OOP added to cart.
added product to cart
added product to cart
added product to cart
the number of products is truethe mac number is 4
smartphone added to cart.
Order confirmed.
```