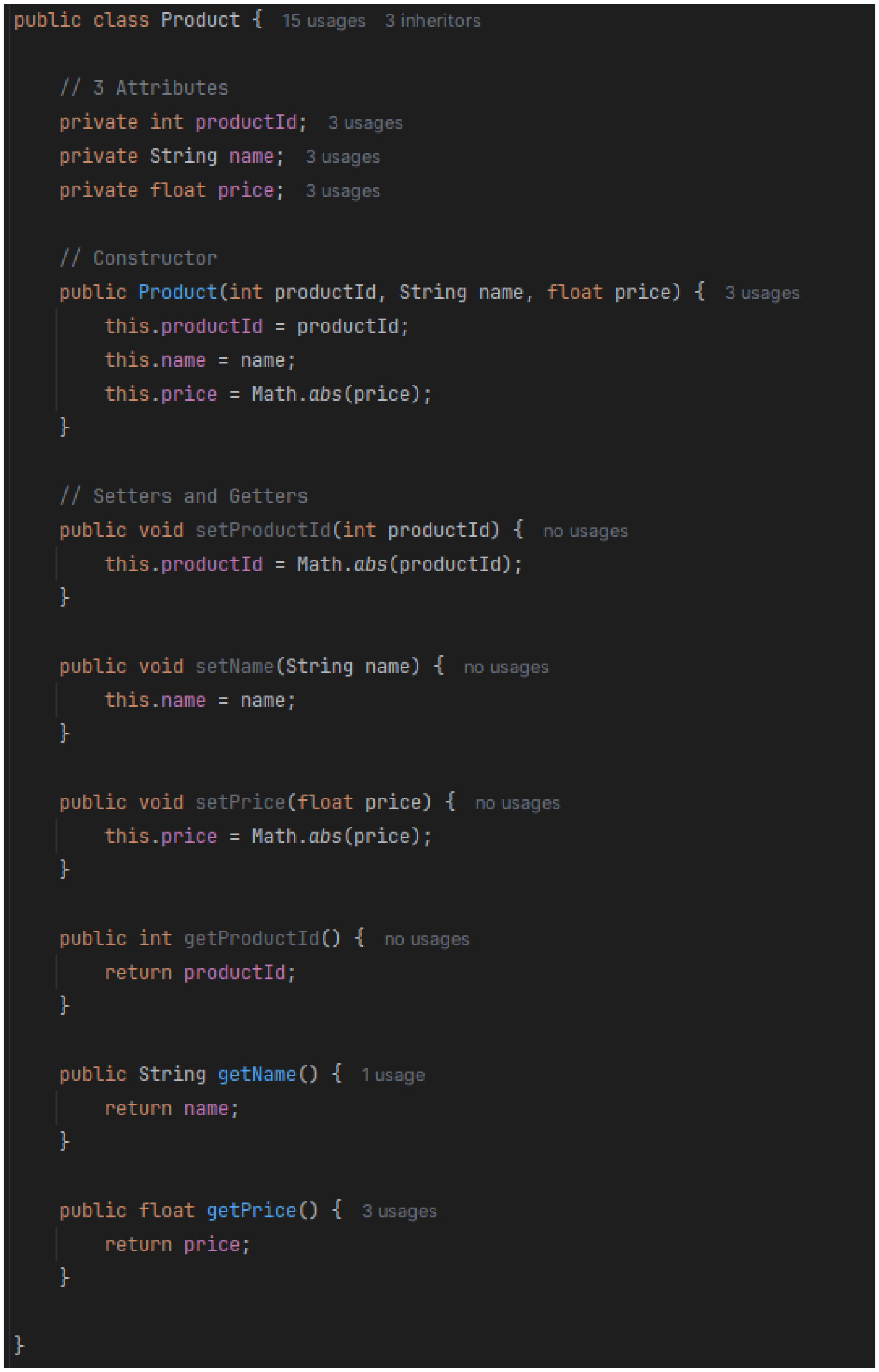
# Programing Project #:

*Name:Yaqeen alradi mousa obaid* *ID:22011498*

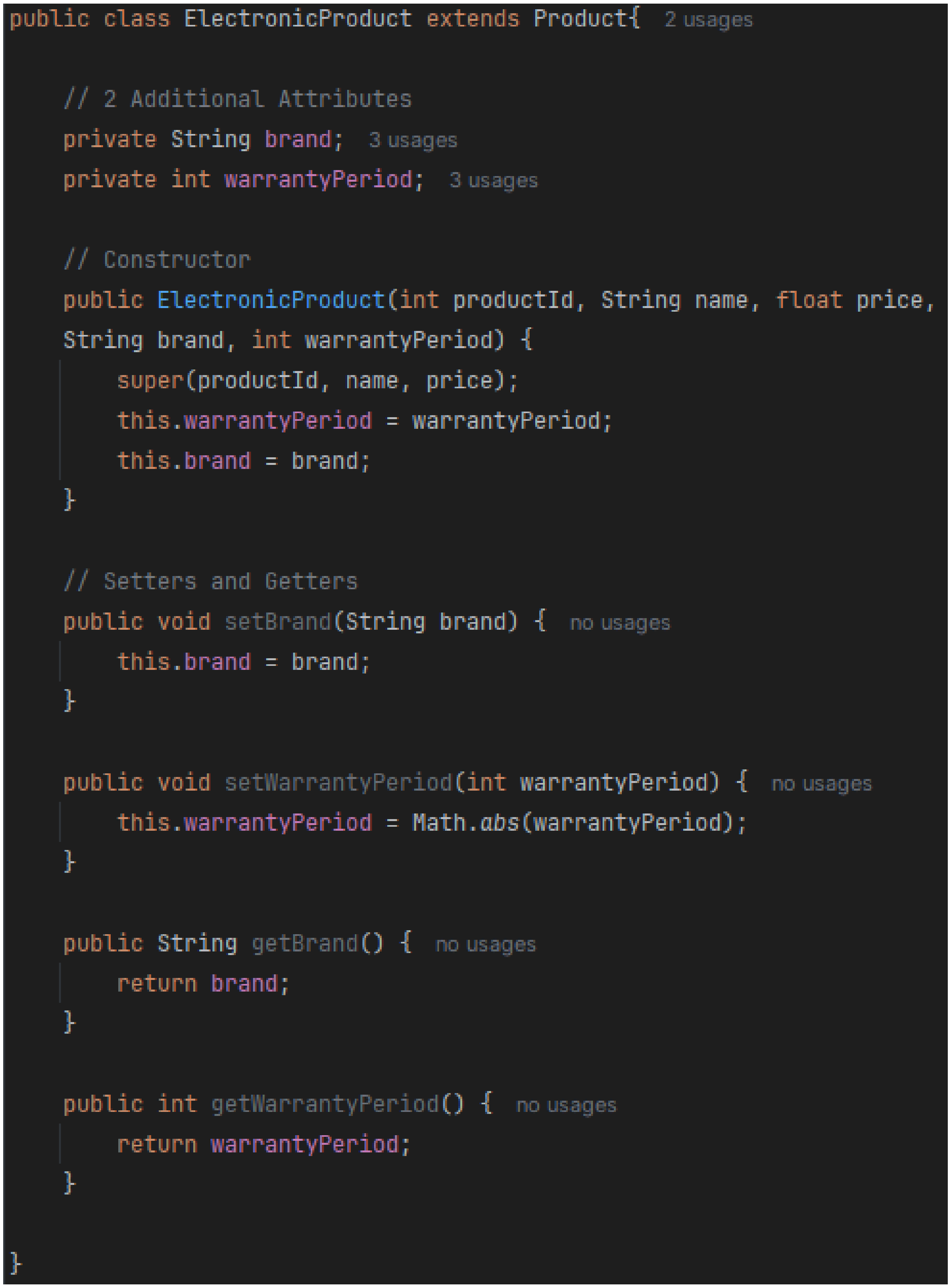
---

***Product Class:***



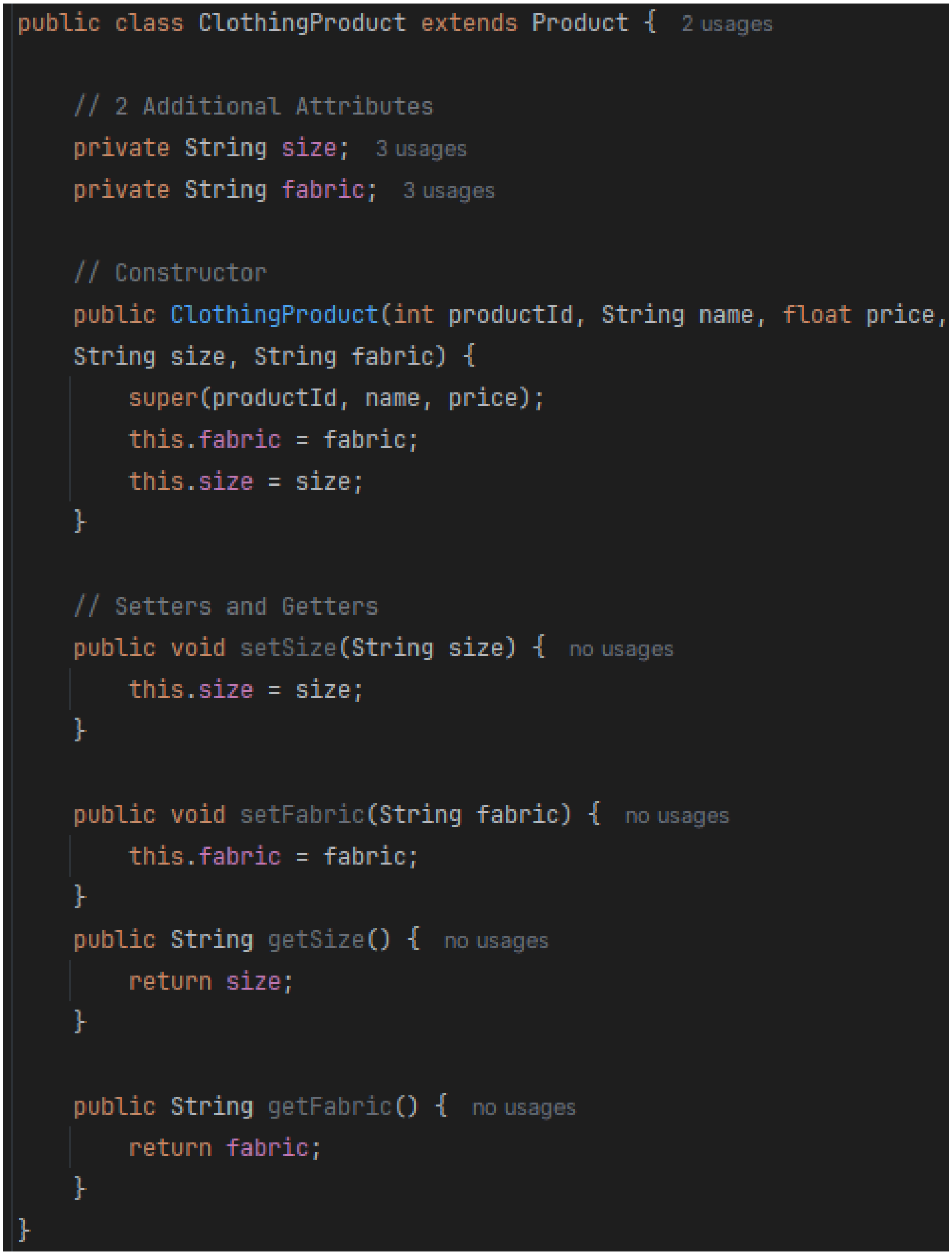
```
public class Product {  15 usages  3 inheritors

    // 3 Attributes
    private int productId;  3 usages
    private String name;  3 usages
    private float price;  3 usages

    // Constructor
    public Product(int productId, String name, float price) {  3 usages
        this.productId = productId;
        this.name = name;
        this.price = Math.abs(price);
    }

    // Setters and Getters
    public void setProductId(int productId) {  no usages
        this.productId = Math.abs(productId);
    }

    public void setName(String name) {  no usages
        this.name = name;
    }

    public void setPrice(float price) {  no usages
        this.price = Math.abs(price);
    }

    public int getProductId() {  no usages
        return productId;
    }

    public String getName() {  1 usage
        return name;
    }

    public float getPrice() {  3 usages
        return price;
    }

}
```

***Electronic Product Class:***



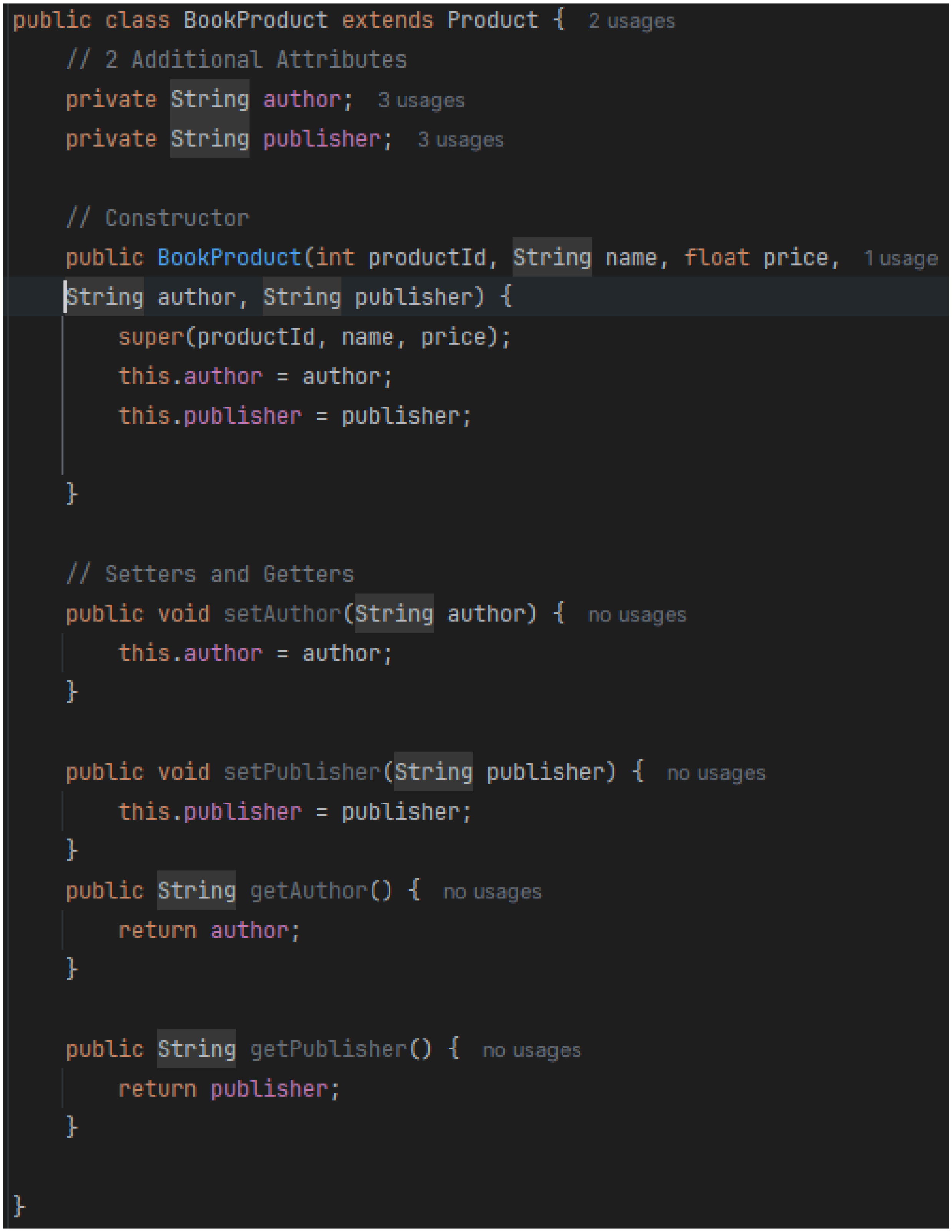
```
public class ElectronicProduct extends Product{  2 usages

    // 2 Additional Attributes
    private String brand;  3 usages
    private int warrantyPeriod;  3 usages

    // Constructor
    public ElectronicProduct(int productId, String name, float price,
    String brand, int warrantyPeriod) {
        super(productId, name, price);
        this.warrantyPeriod = warrantyPeriod;
        this.brand = brand;
    }

    // Setters and Getters
    public void setBrand(String brand) {  no usages
        this.brand = brand;
    }

    public void setWarrantyPeriod(int warrantyPeriod) {  no usages
        this.warrantyPeriod = Math.abs(warrantyPeriod);
    }

    public String getBrand() {  no usages
        return brand;
    }

    public int getWarrantyPeriod() {  no usages
        return warrantyPeriod;
    }

}
```

***Clothing Product Class:***



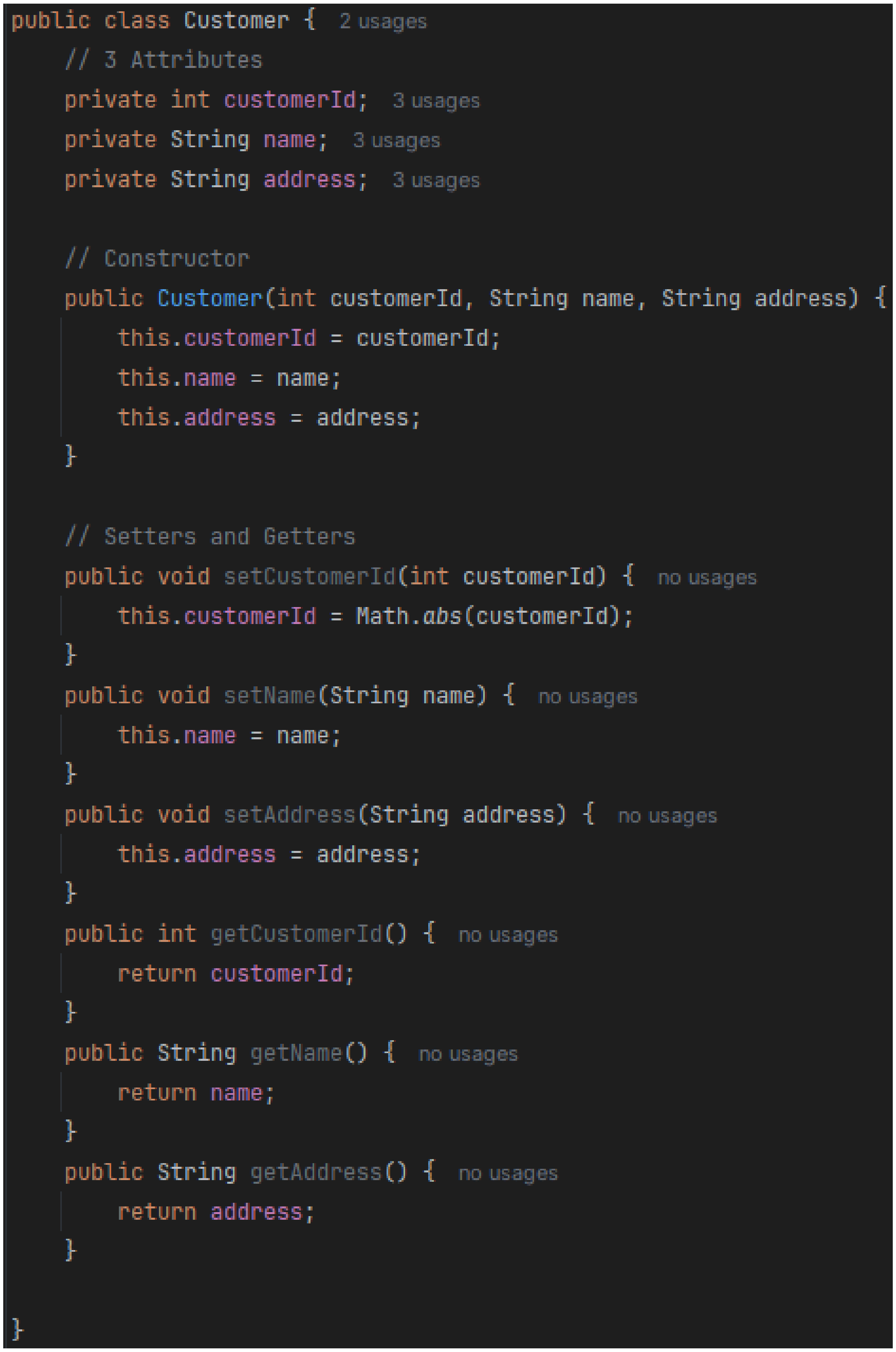
```java
public class ClothingProduct extends Product {  2 usages

    // 2 Additional Attributes
    private String size;  3 usages
    private String fabric;  3 usages

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

    // Setters and Getters
    public void setSize(String size) {  no usages
        this.size = size;
    }

    public void setFabric(String fabric) {  no usages
        this.fabric = fabric;
    }
    public String getSize() {  no usages
        return size;
    }

    public String getFabric() {  no usages
        return fabric;
    }
}
```

***Book Product Class:***



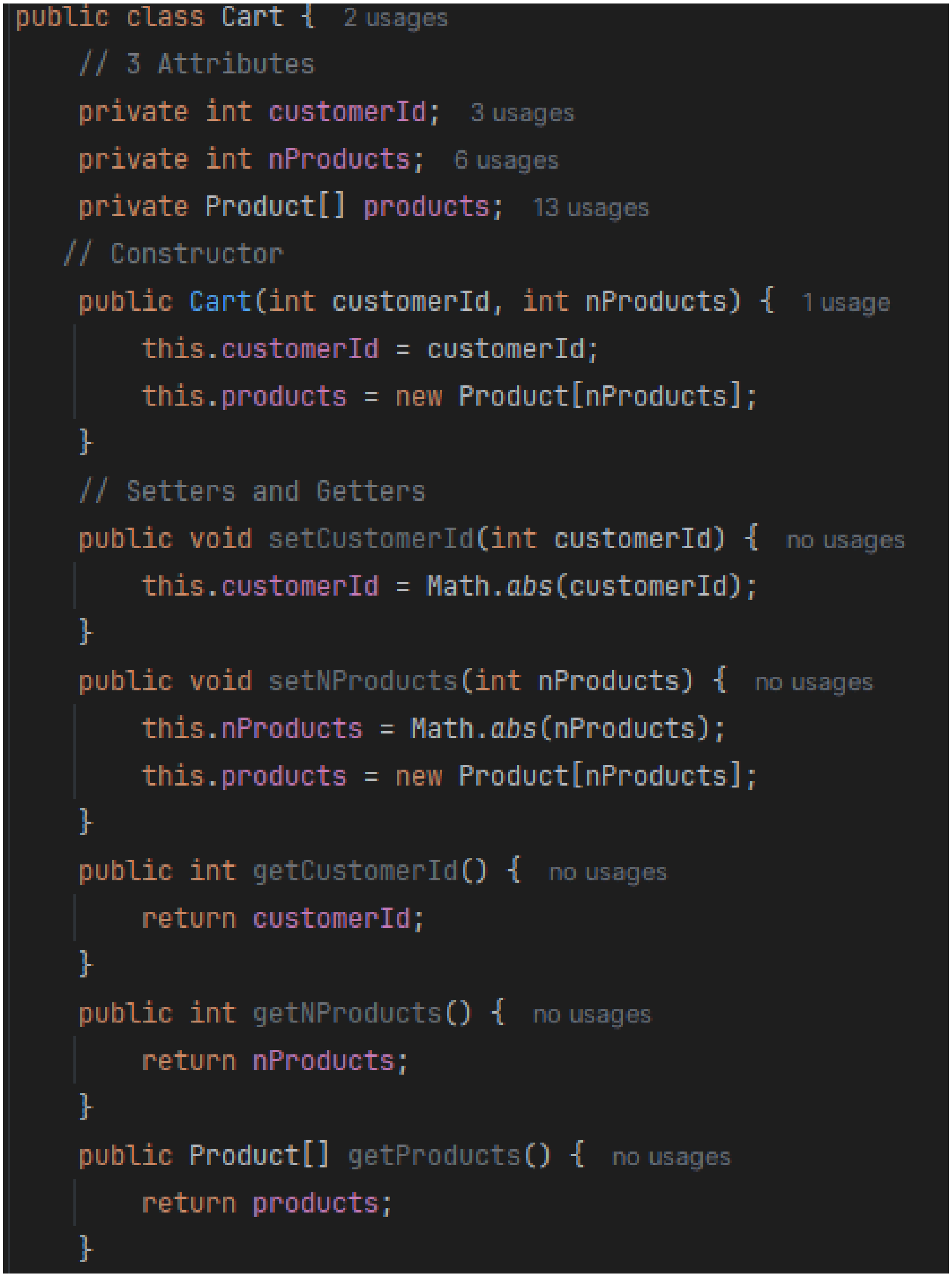
```
public class BookProduct extends Product {  2 usages
    // 2 Additional Attributes
    private String author;  3 usages
    private String publisher;  3 usages

    // Constructor
    public BookProduct(int productId, String name, float price,  1 usage
    String author, String publisher) {
        super(productId, name, price);
        this.author = author;
        this.publisher = publisher;

    }

    // Setters and Getters
    public void setAuthor(String author) {  no usages
        this.author = author;
    }

    public void setPublisher(String publisher) {  no usages
        this.publisher = publisher;
    }
    public String getAuthor() {  no usages
        return author;
    }

    public String getPublisher() {  no usages
        return publisher;
    }

}
```

***Customer Class:***

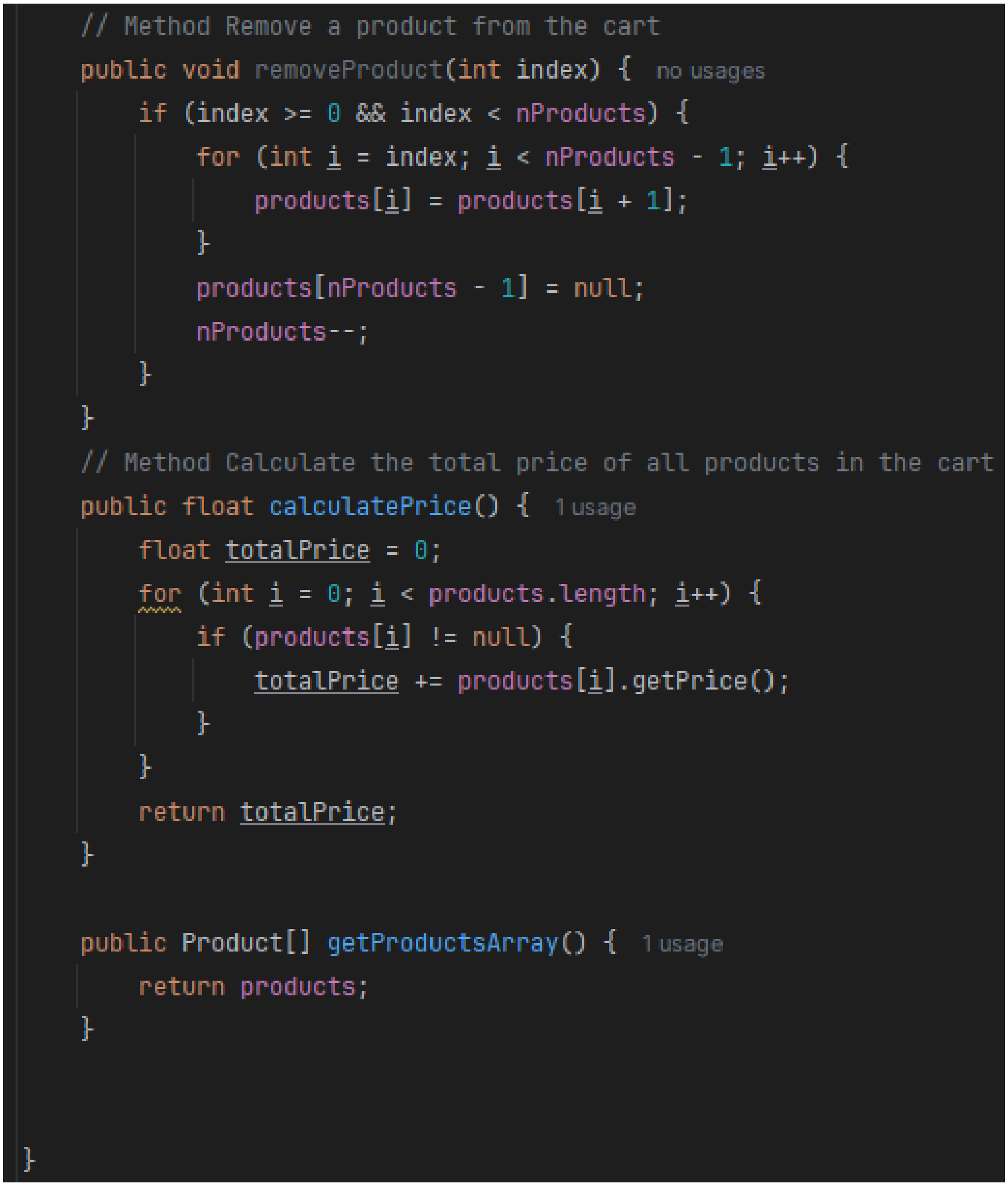


```
public class Customer {  2 usages
    // 3 Attributes
    private int customerId;  3 usages
    private String name;  3 usages
    private String address;  3 usages

    // Constructor
    public Customer(int customerId, String name, String address) {
        this.customerId = customerId;
        this.name = name;
        this.address = address;
    }

    // Setters and Getters
    public void setCustomerId(int customerId) {  no usages
        this.customerId = Math.abs(customerId);
    }
    public void setName(String name) {  no usages
        this.name = name;
    }
    public void setAddress(String address) {  no usages
        this.address = address;
    }
    public int getCustomerId() {  no usages
        return customerId;
    }
    public String getName() {  no usages
        return name;
    }
    public String getAddress() {  no usages
        return address;
    }

}
```

***Cart Class:***

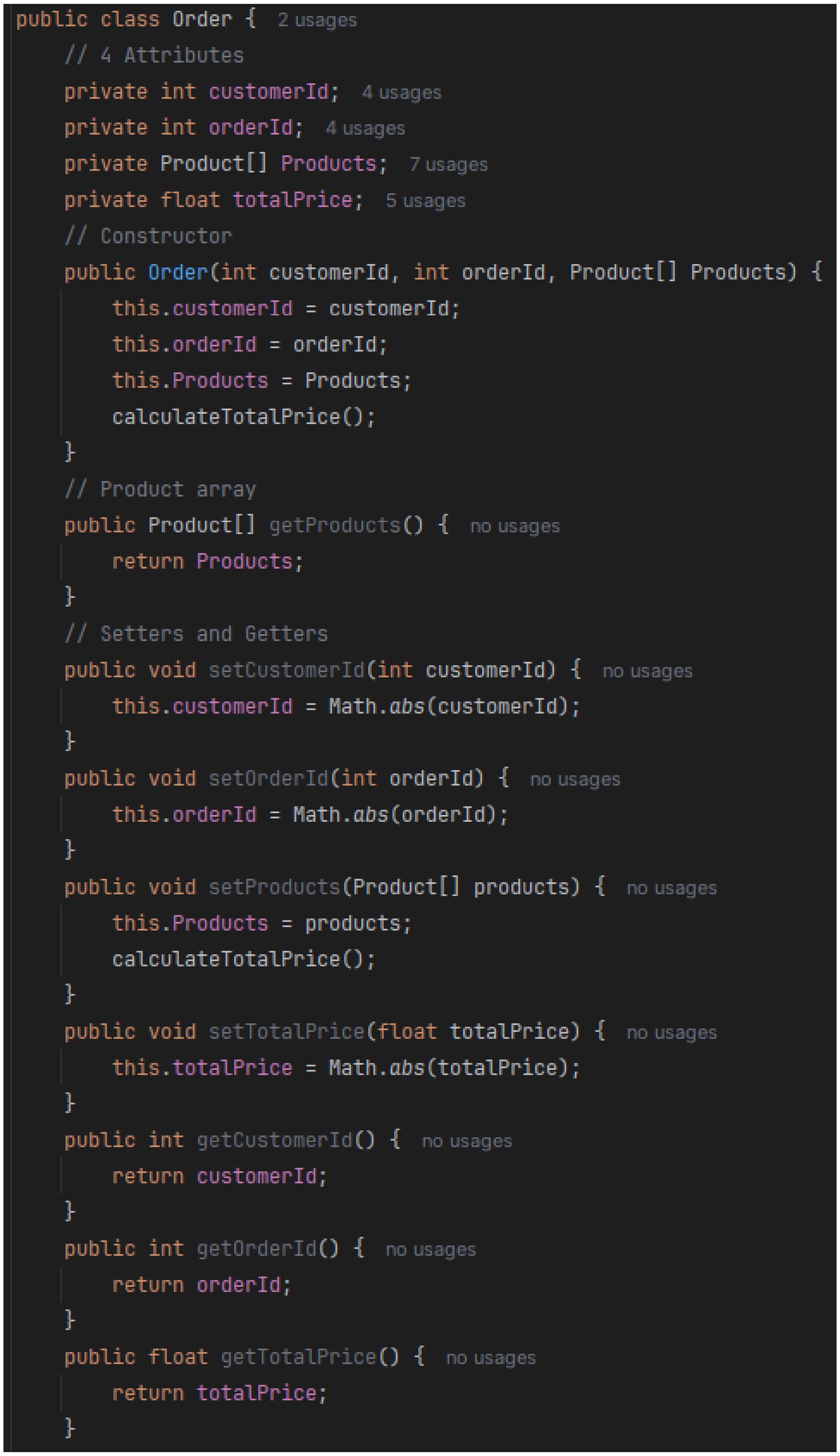


```java
public class Cart {  2 usages
    // 3 Attributes
    private int customerId;  3 usages
    private int nProducts;  6 usages
    private Product[] products;  13 usages
   // Constructor
    public Cart(int customerId, int nProducts) {  1 usage
        this.customerId = customerId;
        this.products = new Product[nProducts];
    }
    // Setters and Getters
    public void setCustomerId(int customerId) {  no usages
        this.customerId = Math.abs(customerId);
    }
    public void setNProducts(int nProducts) {  no usages
        this.nProducts = Math.abs(nProducts);
        this.products = new Product[nProducts];
    }
    public int getCustomerId() {  no usages
        return customerId;
    }
    public int getNProducts() {  no usages
        return nProducts;
    }
    public Product[] getProducts() {  no usages
        return products;
    }
```

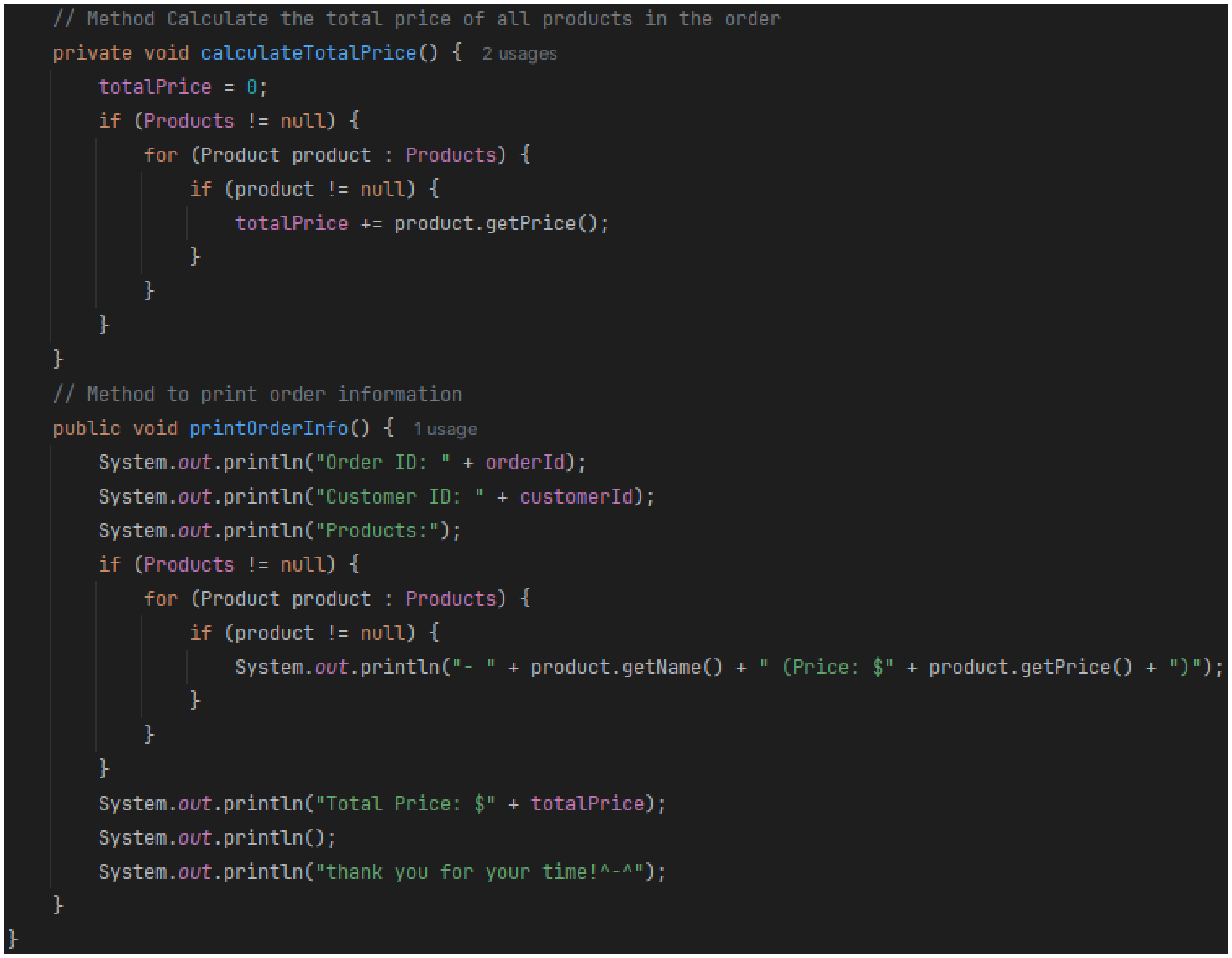


```java
    // Method Remove a product from the cart
    public void removeProduct(int index) {  no usages
        if (index >= 0 && index < nProducts) {
            for (int i = index; i < nProducts - 1; i++) {
                products[i] = products[i + 1];
            }
            products[nProducts - 1] = null;
            nProducts--;
        }
    }
    // Method Calculate the total price of all products in the cart
    public float calculatePrice() {  1 usage
        float totalPrice = 0;
        for (int i = 0; i < products.length; i++) {
            if (products[i] != null) {
                totalPrice += products[i].getPrice();
            }
        }
        return totalPrice;
    }

    public Product[] getProductsArray() {  1 usage
        return products;
    }


}
```

***Order Class:***

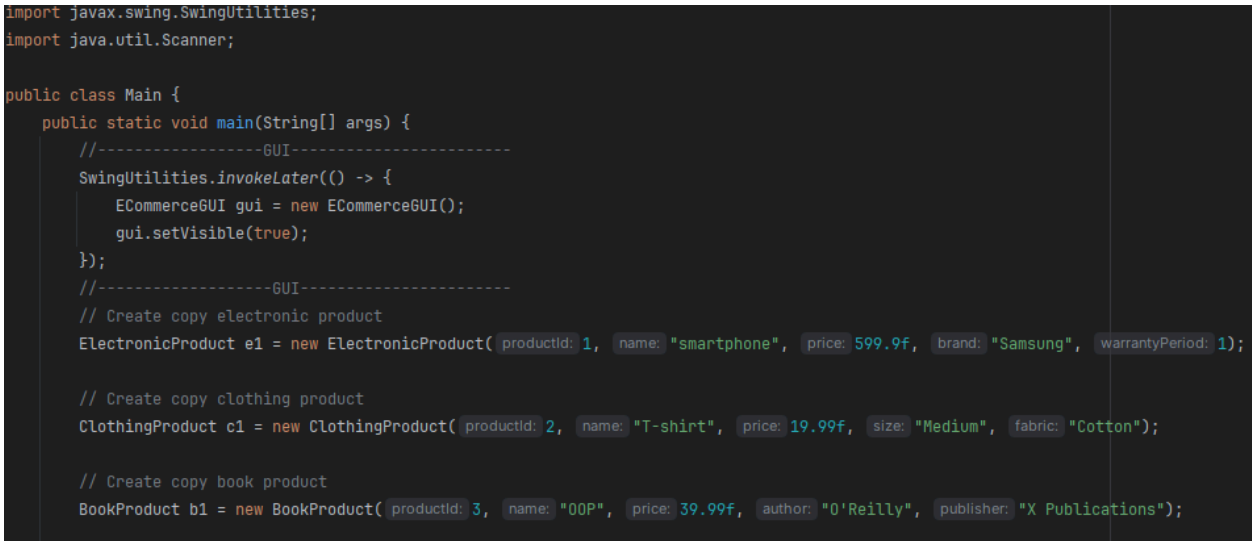


```java
public class Order {  2 usages
    // 4 Attributes
    private int customerId;  4 usages
    private int orderId;  4 usages
    private Product[] Products;  7 usages
    private float totalPrice;  5 usages
    // Constructor
    public Order(int customerId, int orderId, Product[] Products) {
        this.customerId = customerId;
        this.orderId = orderId;
        this.Products = Products;
        calculateTotalPrice();
    }
    // Product array
    public Product[] getProducts() {  no usages
        return Products;
    }
    // Setters and Getters
    public void setCustomerId(int customerId) {  no usages
        this.customerId = Math.abs(customerId);
    }
    public void setOrderId(int orderId) {  no usages
        this.orderId = Math.abs(orderId);
    }
    public void setProducts(Product[] products) {  no usages
        this.Products = products;
        calculateTotalPrice();
    }
    public void setTotalPrice(float totalPrice) {  no usages
        this.totalPrice = Math.abs(totalPrice);
    }
    public int getCustomerId() {  no usages
        return customerId;
    }
    public int getOrderId() {  no usages
        return orderId;
    }
    public float getTotalPrice() {  no usages
        return totalPrice;
    }
```

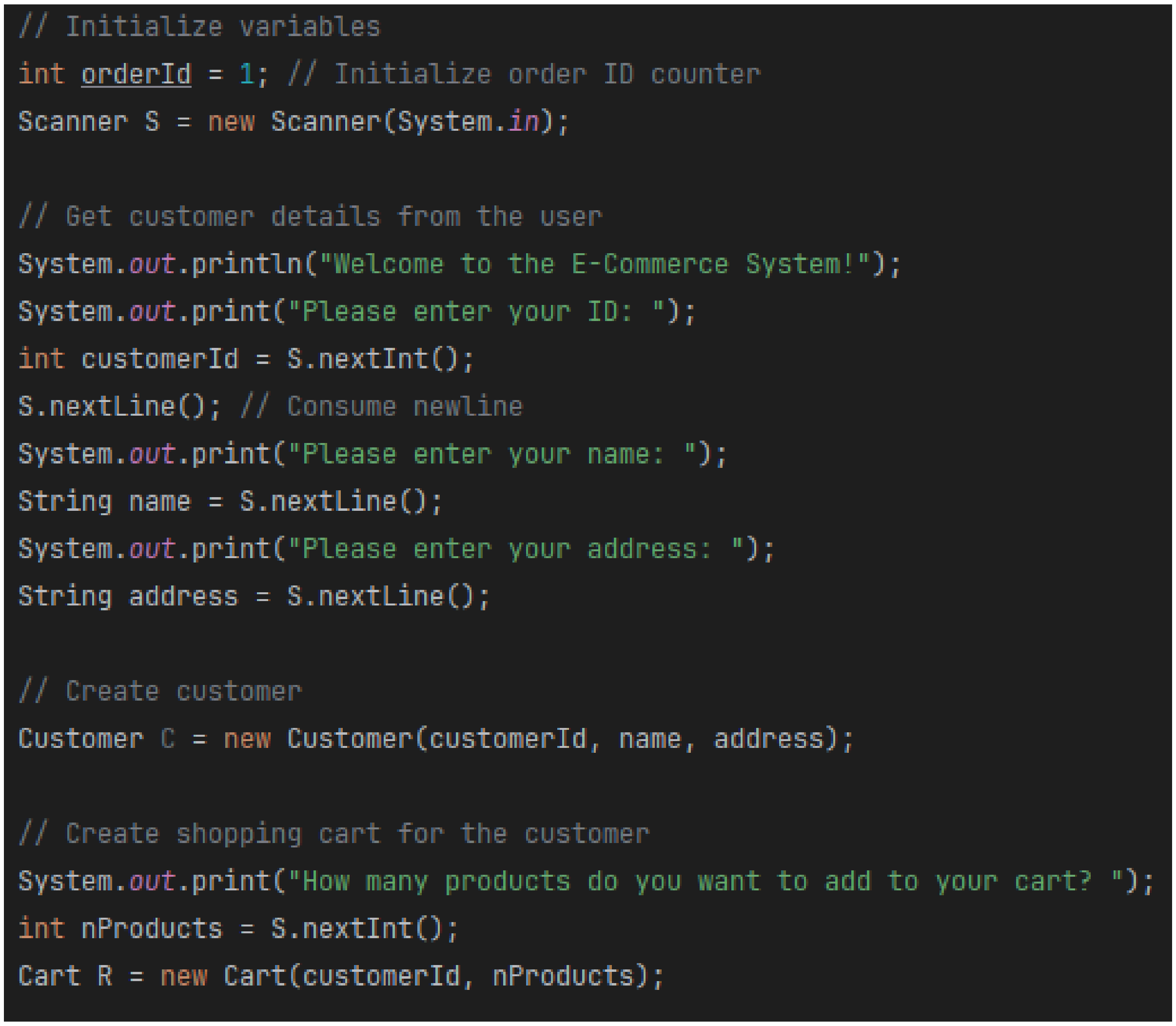


```
    // Method Calculate the total price of all products in the order
    private void calculateTotalPrice() {  2 usages
        totalPrice = 0;
        if (Products != null) {
            for (Product product : Products) {
                if (product != null) {
                    totalPrice += product.getPrice();
                }
            }
        }
    }
    // Method to print order information
    public void printOrderInfo() {  1 usage
        System.out.println("Order ID: " + orderId);
        System.out.println("Customer ID: " + customerId);
        System.out.println("Products:");
        if (Products != null) {
            for (Product product : Products) {
                if (product != null) {
                    System.out.println("- " + product.getName() + " (Price: $" + product.getPrice() + ")");
                }
            }
        }
        System.out.println("Total Price: $" + totalPrice);
        System.out.println();
        System.out.println("thank you for your time!^-^");
    }
}
```

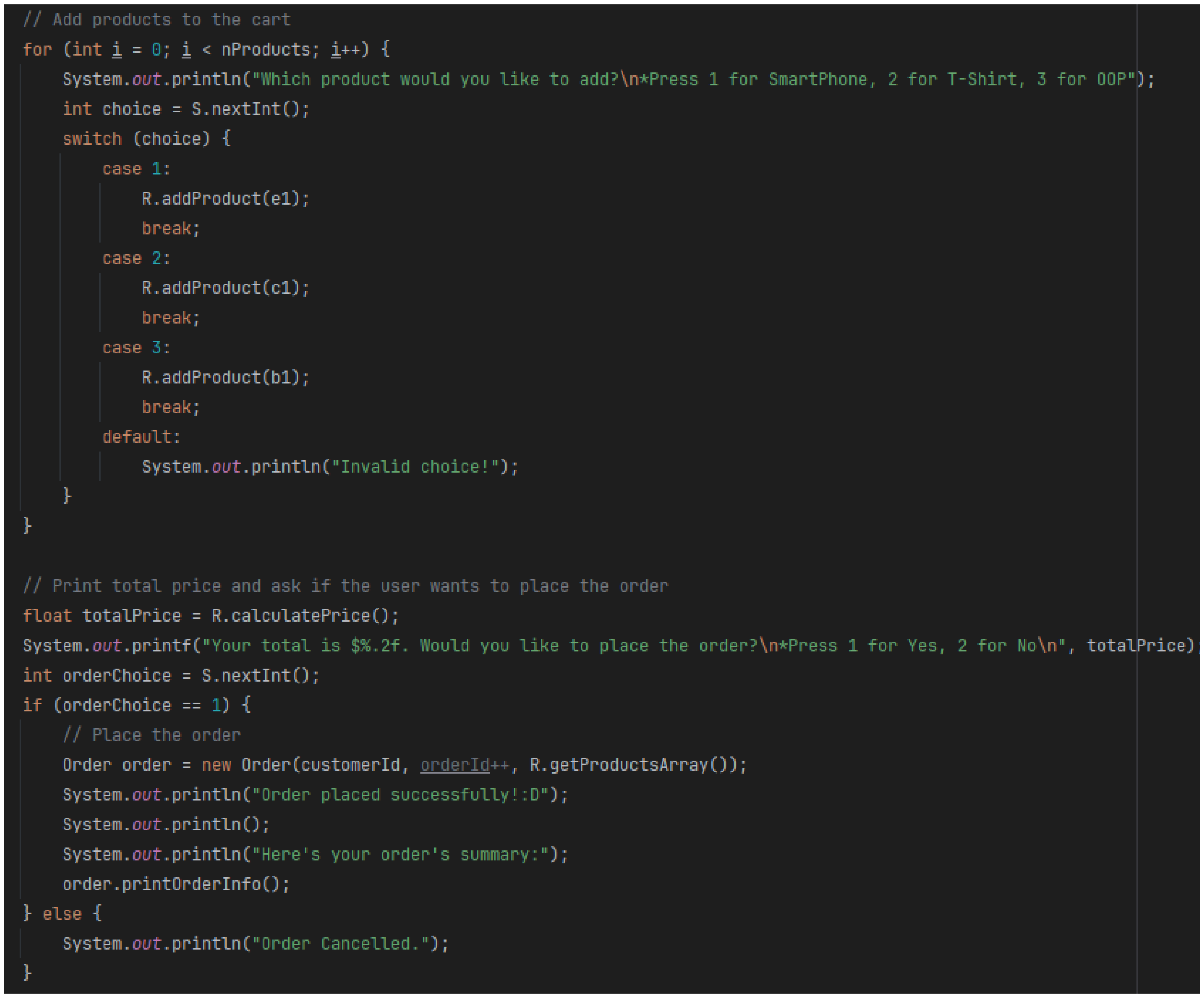
***Main Class:***



```java
import javax.swing.SwingUtilities;
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        //-----------------GUI-----------------------
        SwingUtilities.invokeLater(() -> {
            ECommerceGUI gui = new ECommerceGUI();
            gui.setVisible(true);
        });
        //------------------GUI----------------------
        // Create copy electronic product
        ElectronicProduct e1 = new ElectronicProduct( productId: 1,  name: "smartphone",  price: 599.9f,  brand: "Samsung",  warrantyPeriod: 1);

        // Create copy clothing product
        ClothingProduct c1 = new ClothingProduct( productId: 2,  name: "T-shirt",  price: 19.99f,  size: "Medium",  fabric: "Cotton");

        // Create copy book product
        BookProduct b1 = new BookProduct( productId: 3,  name: "OOP",  price: 39.99f,  author: "O'Reilly",  publisher: "X Publications");
```

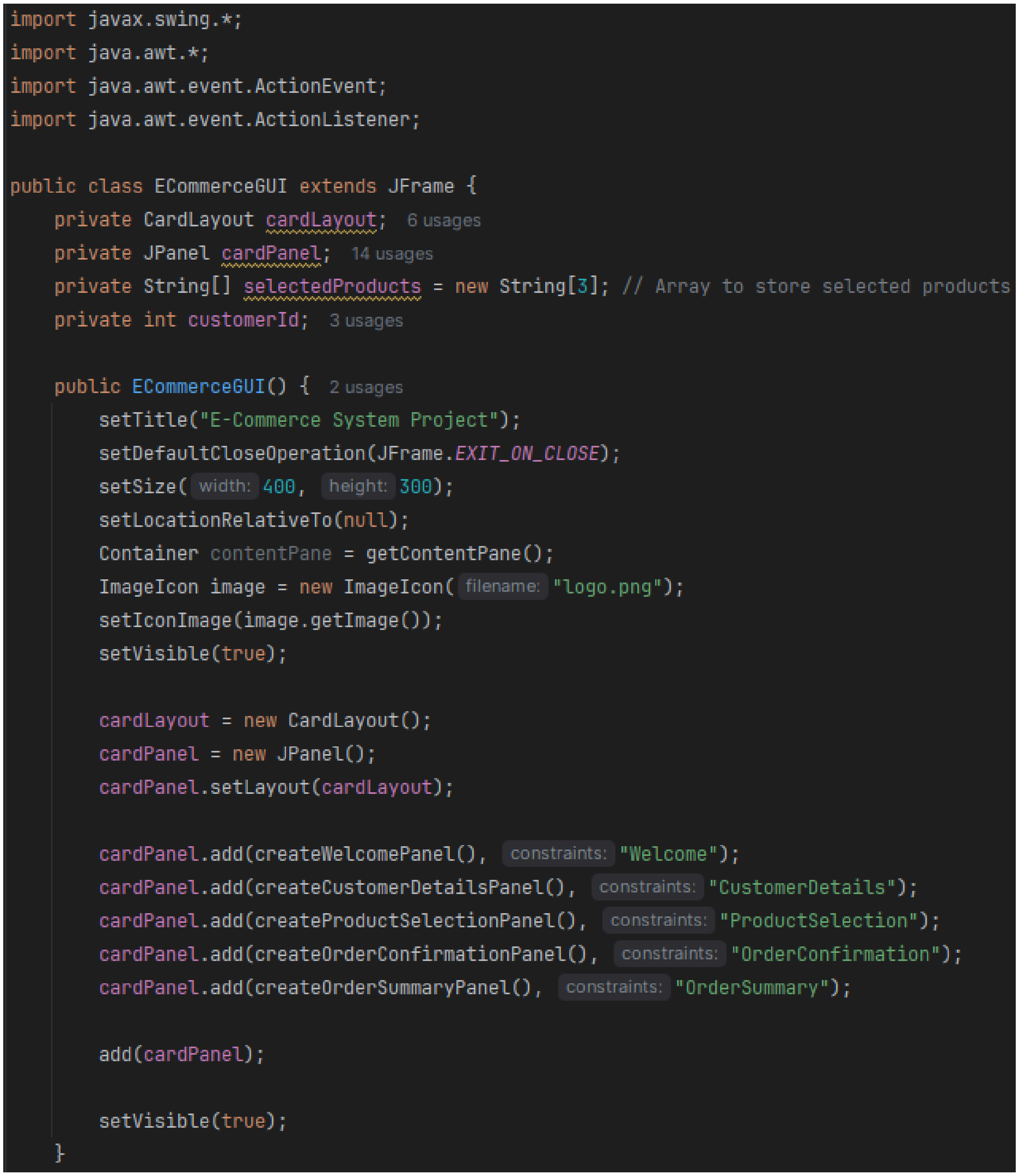


```java
// Initialize variables
int orderId = 1; // Initialize order ID counter
Scanner S = new Scanner(System.in);

// Get customer details from the user
System.out.println("Welcome to the E-Commerce System!");
System.out.print("Please enter your ID: ");
int customerId = S.nextInt();
S.nextLine(); // Consume newline
System.out.print("Please enter your name: ");
String name = S.nextLine();
System.out.print("Please enter your address: ");
String address = S.nextLine();

// Create customer
Customer C = new Customer(customerId, name, address);

// Create shopping cart for the customer
System.out.print("How many products do you want to add to your cart? ");
int nProducts = S.nextInt();
Cart R = new Cart(customerId, nProducts);
```

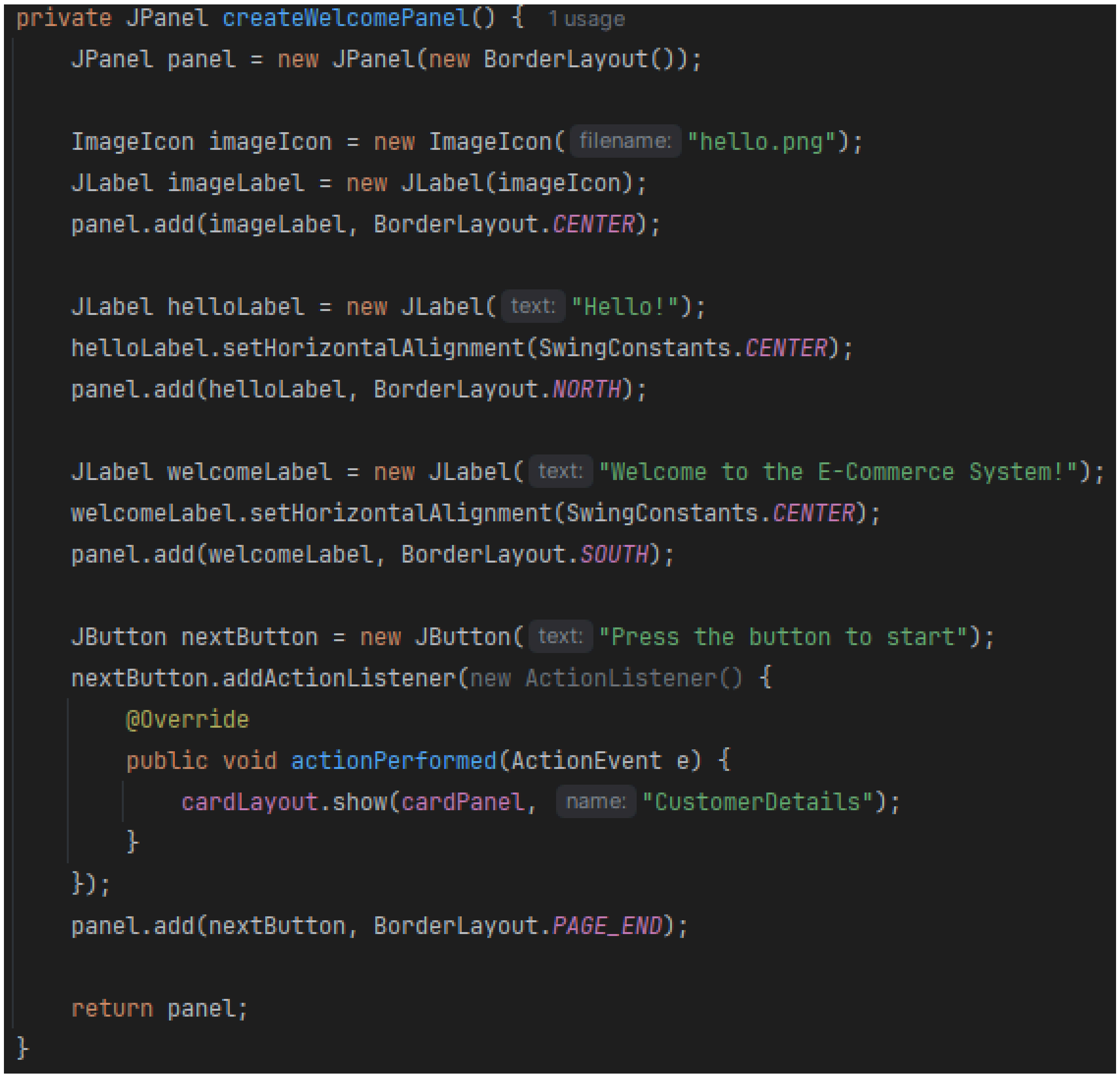


```java
// Add products to the cart
for (int i = 0; i < nProducts; i++) {
    System.out.println("Which product would you like to add?\n*Press 1 for SmartPhone, 2 for T-Shirt, 3 for OOP");
    int choice = S.nextInt();
    switch (choice) {
        case 1:
            R.addProduct(e1);
            break;
        case 2:
            R.addProduct(c1);
            break;
        case 3:
            R.addProduct(b1);
            break;
        default:
            System.out.println("Invalid choice!");
    }
}

// Print total price and ask if the user wants to place the order
float totalPrice = R.calculatePrice();
System.out.printf("Your total is $%.2f. Would you like to place the order?\n*Press 1 for Yes, 2 for No\n", totalPrice)
int orderChoice = S.nextInt();
if (orderChoice == 1) {
    // Place the order
    Order order = new Order(customerId, orderId++, R.getProductsArray());
    System.out.println("Order placed successfully!:D");
    System.out.println();
    System.out.println("Here's your order's summary:");
    order.printOrderInfo();
} else {
    System.out.println("Order Cancelled.");
}
```

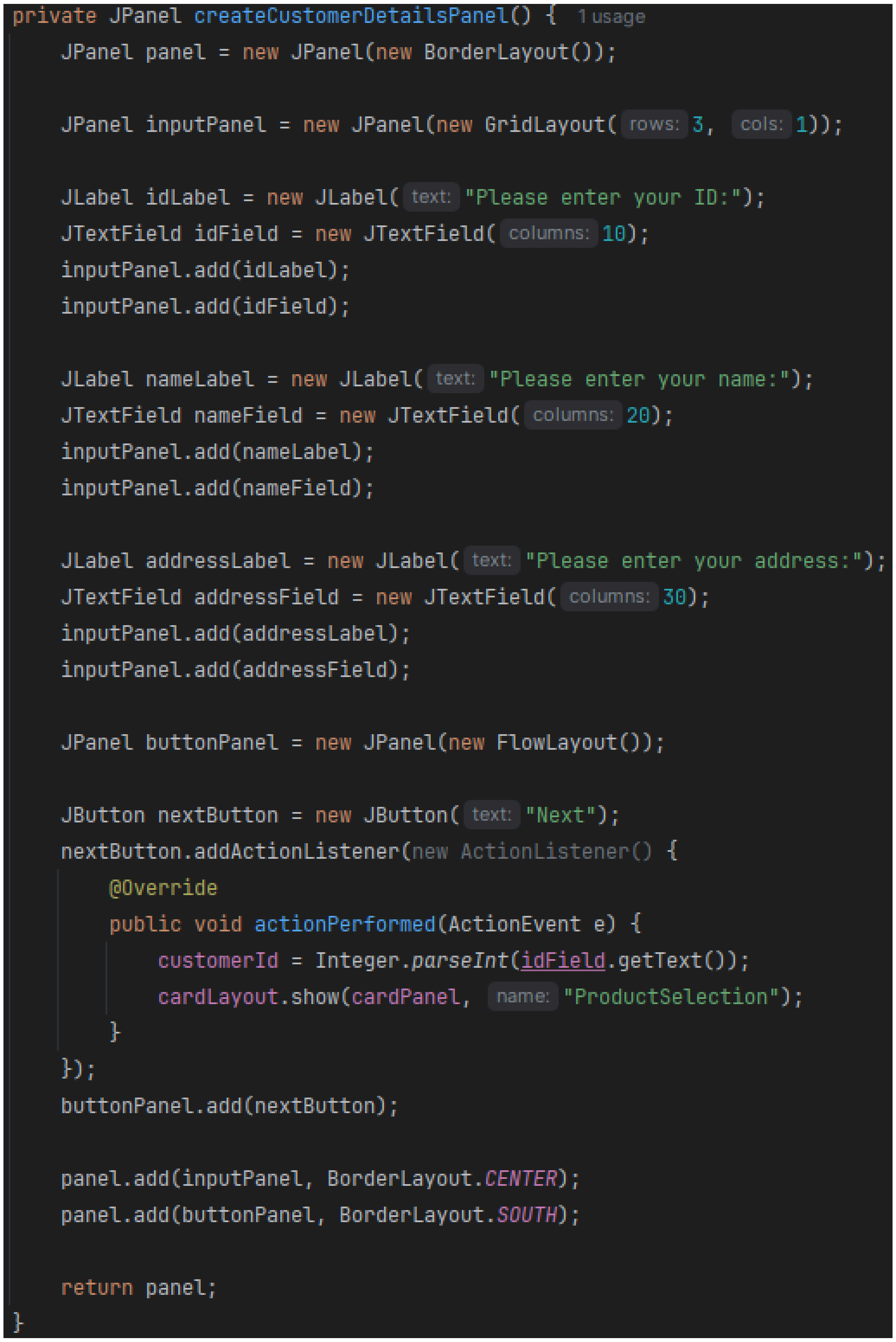
***GUI Class:***



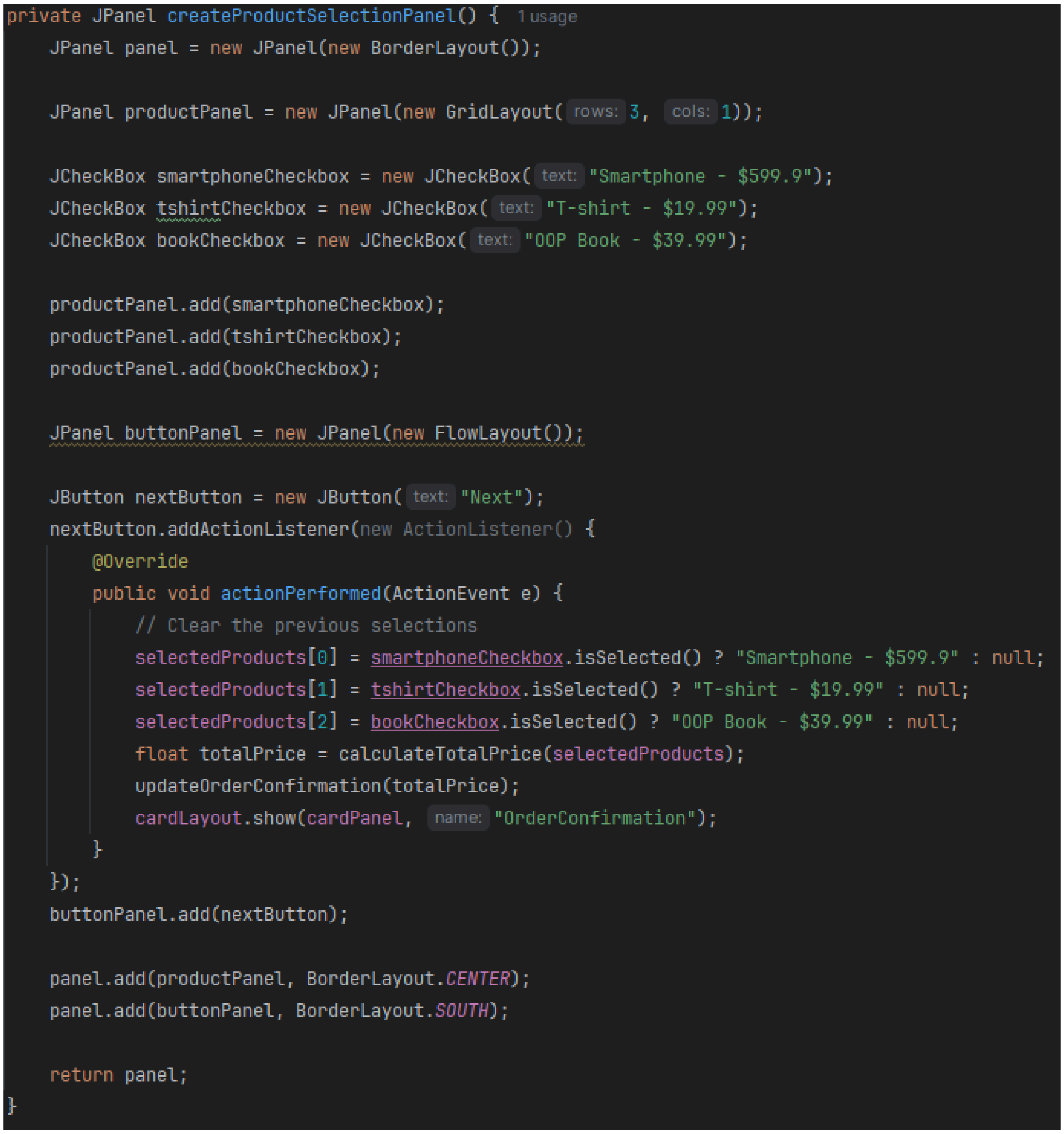
```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class ECommerceGUI extends JFrame {
    private CardLayout cardLayout;  6 usages
    private JPanel cardPanel;  14 usages
    private String[] selectedProducts = new String[3]; // Array to store selected products
    private int customerId;  3 usages

    public ECommerceGUI() {  2 usages
        setTitle("E-Commerce System Project");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setSize( width: 400,  height: 300);
        setLocationRelativeTo(null);
        Container contentPane = getContentPane();
        ImageIcon image = new ImageIcon( filename: "logo.png");
        setIconImage(image.getImage());
        setVisible(true);

        cardLayout = new CardLayout();
        cardPanel = new JPanel();
        cardPanel.setLayout(cardLayout);

        cardPanel.add(createWelcomePanel(),  constraints: "Welcome");
        cardPanel.add(createCustomerDetailsPanel(),  constraints: "CustomerDetails");
        cardPanel.add(createProductSelectionPanel(),  constraints: "ProductSelection");
        cardPanel.add(createOrderConfirmationPanel(),  constraints: "OrderConfirmation");
        cardPanel.add(createOrderSummaryPanel(),  constraints: "OrderSummary");

        add(cardPanel);

        setVisible(true);
    }
```



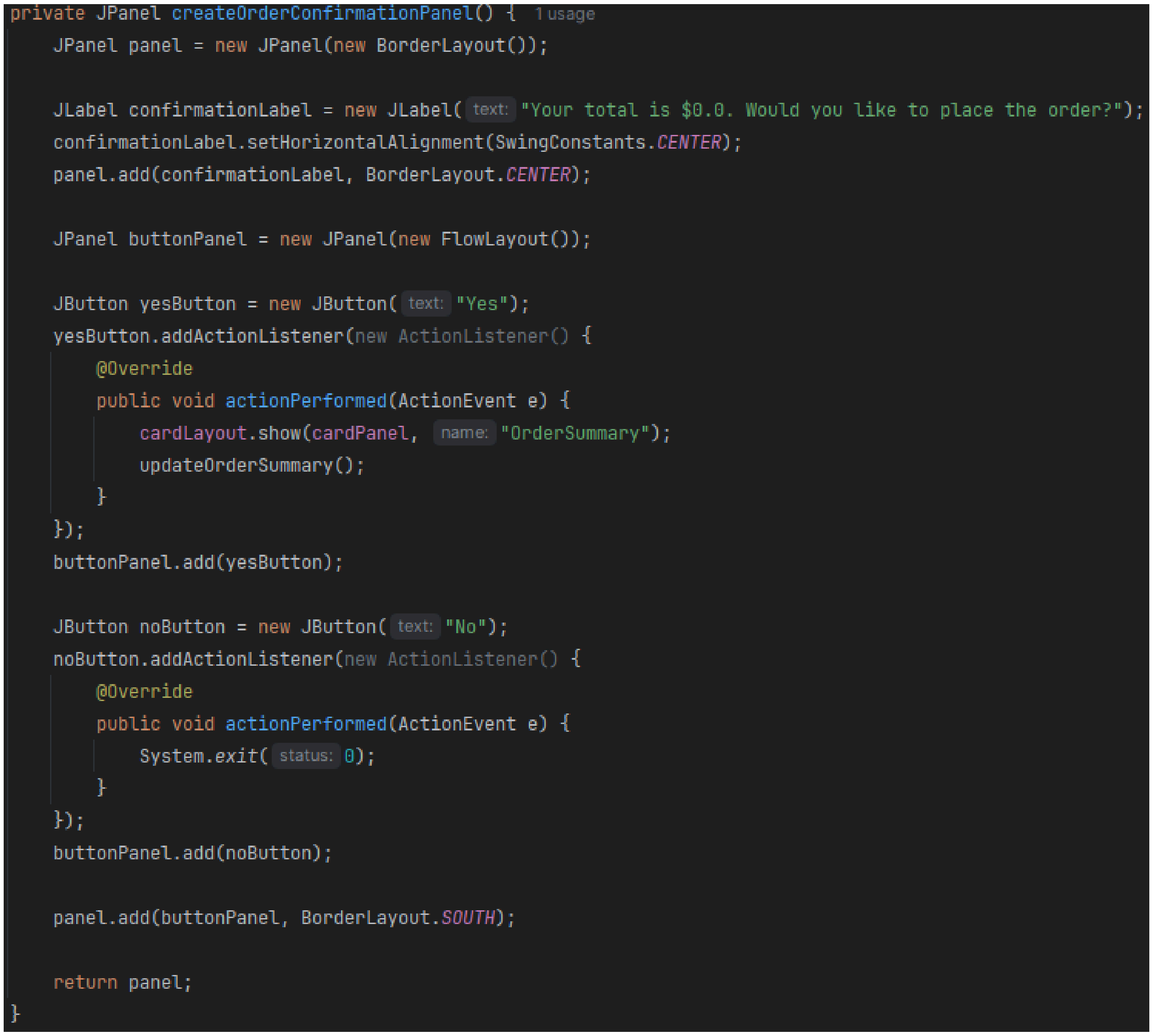
```java
private JPanel createWelcomePanel() {  1 usage
    JPanel panel = new JPanel(new BorderLayout());

    ImageIcon imageIcon = new ImageIcon( filename: "hello.png");
    JLabel imageLabel = new JLabel(imageIcon);
    panel.add(imageLabel, BorderLayout.CENTER);

    JLabel helloLabel = new JLabel( text: "Hello!");
    helloLabel.setHorizontalAlignment(SwingConstants.CENTER);
    panel.add(helloLabel, BorderLayout.NORTH);

    JLabel welcomeLabel = new JLabel( text: "Welcome to the E-Commerce System!");
    welcomeLabel.setHorizontalAlignment(SwingConstants.CENTER);
    panel.add(welcomeLabel, BorderLayout.SOUTH);

    JButton nextButton = new JButton( text: "Press the button to start");
    nextButton.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            cardLayout.show(cardPanel,  name: "CustomerDetails");
        }
    });
    panel.add(nextButton, BorderLayout.PAGE_END);

    return panel;
}
```



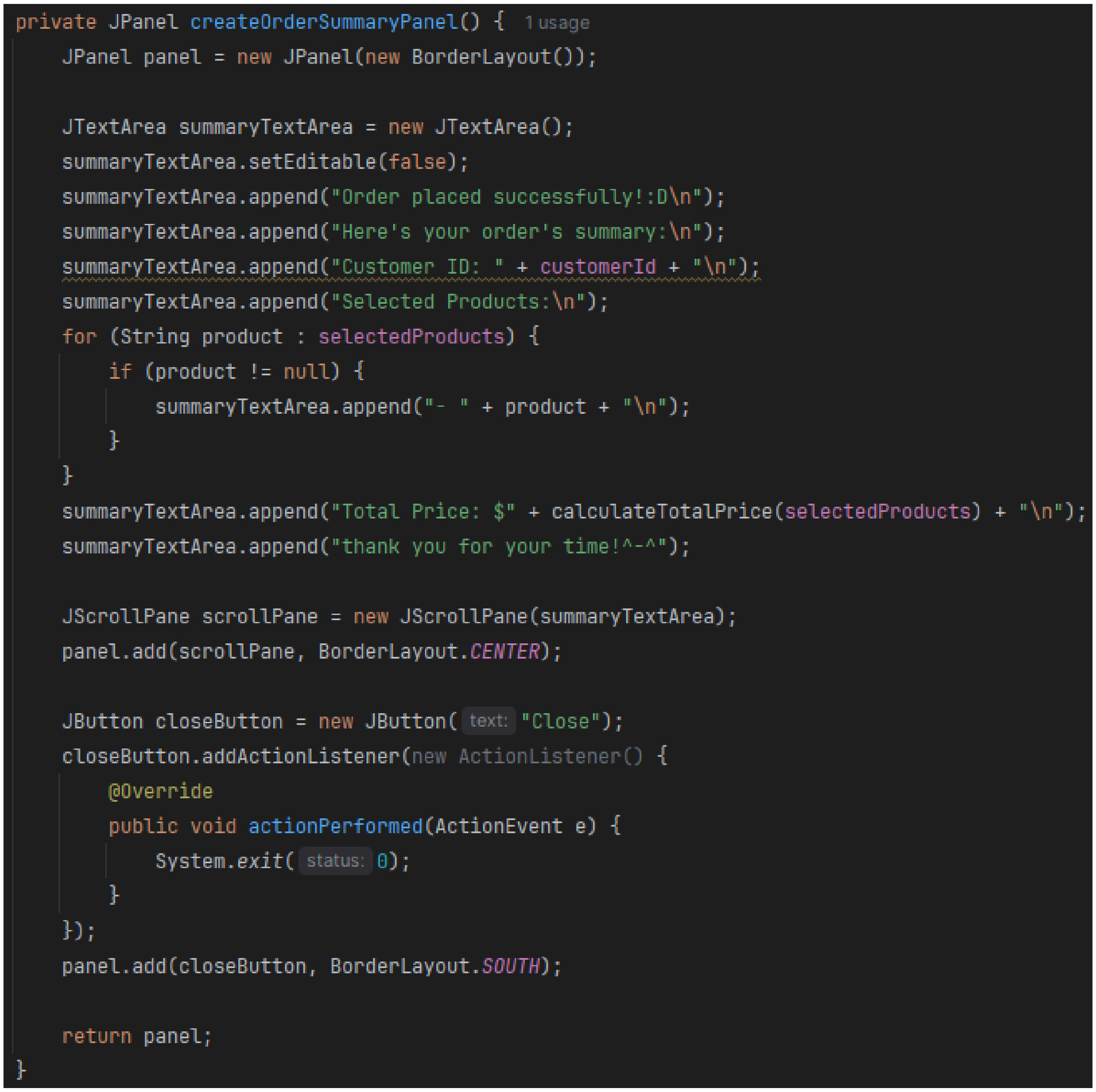
```java
private JPanel createCustomerDetailsPanel() {  1 usage
    JPanel panel = new JPanel(new BorderLayout());

    JPanel inputPanel = new JPanel(new GridLayout( rows: 3,  cols: 1));

    JLabel idLabel = new JLabel( text: "Please enter your ID:");
    JTextField idField = new JTextField( columns: 10);
    inputPanel.add(idLabel);
    inputPanel.add(idField);

    JLabel nameLabel = new JLabel( text: "Please enter your name:");
    JTextField nameField = new JTextField( columns: 20);
    inputPanel.add(nameLabel);
    inputPanel.add(nameField);

    JLabel addressLabel = new JLabel( text: "Please enter your address:");
    JTextField addressField = new JTextField( columns: 30);
    inputPanel.add(addressLabel);
    inputPanel.add(addressField);

    JPanel buttonPanel = new JPanel(new FlowLayout());

    JButton nextButton = new JButton( text: "Next");
    nextButton.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            customerId = Integer.parseInt(idField.getText());
            cardLayout.show(cardPanel,  name: "ProductSelection");
        }
    });
    buttonPanel.add(nextButton);

    panel.add(inputPanel, BorderLayout.CENTER);
    panel.add(buttonPanel, BorderLayout.SOUTH);

    return panel;
}
```



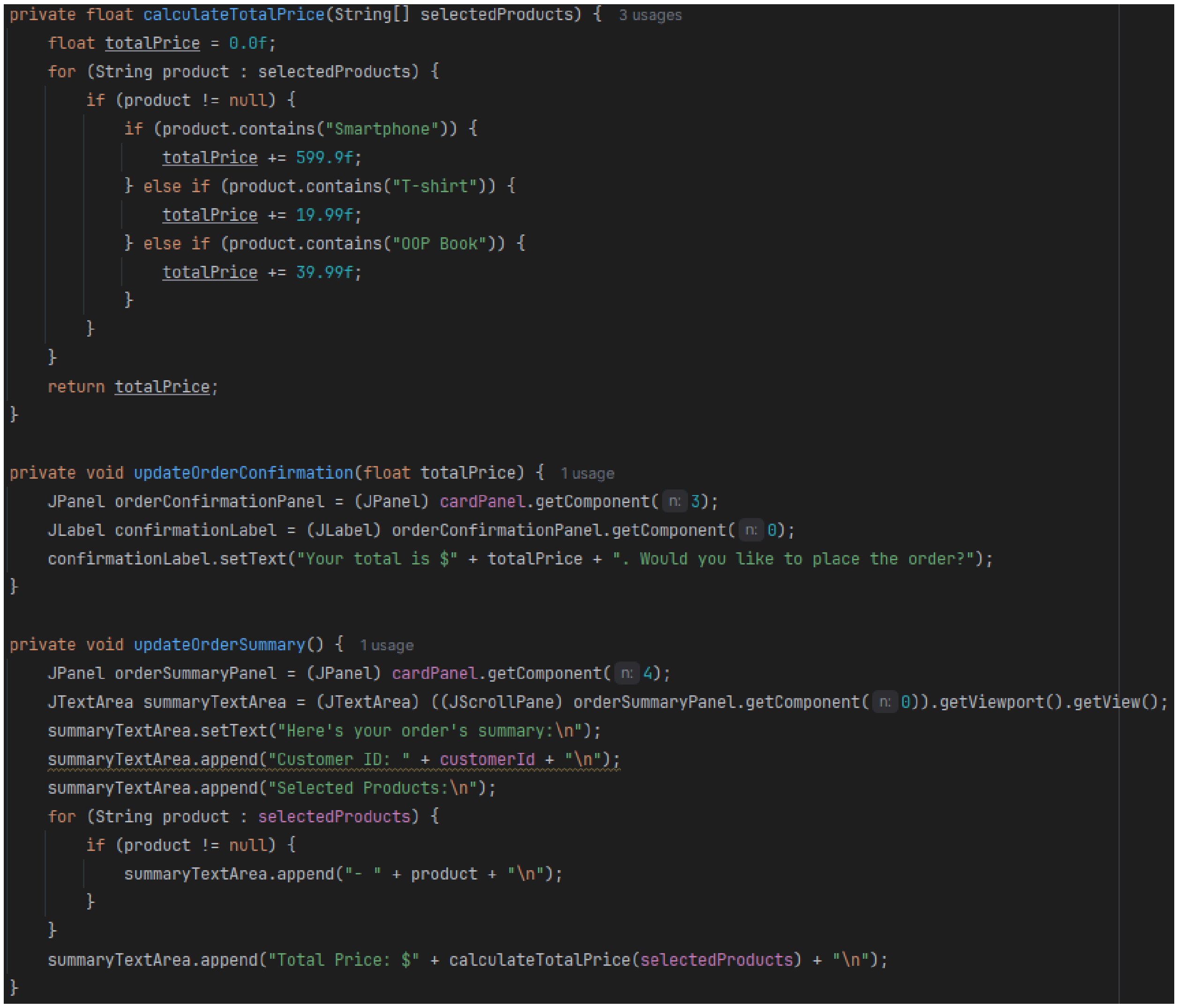
```java
private JPanel createProductSelectionPanel() {  1 usage
    JPanel panel = new JPanel(new BorderLayout());

    JPanel productPanel = new JPanel(new GridLayout( rows: 3,  cols: 1));

    JCheckBox smartphoneCheckbox = new JCheckBox( text: "Smartphone - $599.9");
    JCheckBox tshirtCheckbox = new JCheckBox( text: "T-shirt - $19.99");
    JCheckBox bookCheckbox = new JCheckBox( text: "OOP Book - $39.99");

    productPanel.add(smartphoneCheckbox);
    productPanel.add(tshirtCheckbox);
    productPanel.add(bookCheckbox);

    JPanel buttonPanel = new JPanel(new FlowLayout());

    JButton nextButton = new JButton( text: "Next");
    nextButton.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            // Clear the previous selections
            selectedProducts[0] = smartphoneCheckbox.isSelected() ? "Smartphone - $599.9" : null;
            selectedProducts[1] = tshirtCheckbox.isSelected() ? "T-shirt - $19.99" : null;
            selectedProducts[2] = bookCheckbox.isSelected() ? "OOP Book - $39.99" : null;
            float totalPrice = calculateTotalPrice(selectedProducts);
            updateOrderConfirmation(totalPrice);
            cardLayout.show(cardPanel,  name: "OrderConfirmation");
        }
    });
    buttonPanel.add(nextButton);

    panel.add(productPanel, BorderLayout.CENTER);
    panel.add(buttonPanel, BorderLayout.SOUTH);

    return panel;
}
```



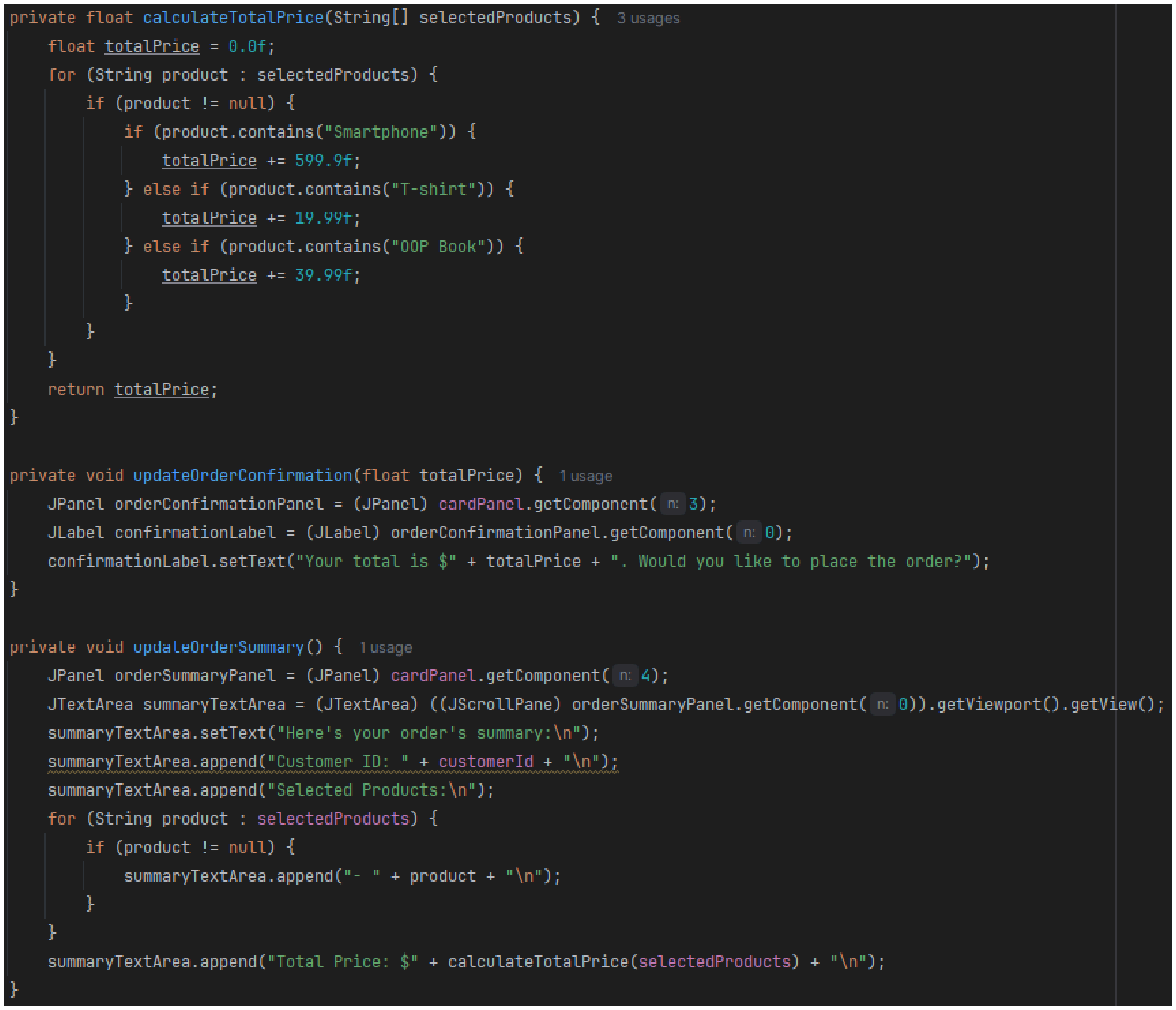
```java
private JPanel createOrderConfirmationPanel() {  1 usage
    JPanel panel = new JPanel(new BorderLayout());

    JLabel confirmationLabel = new JLabel( text: "Your total is $0.0. Would you like to place the order?");
    confirmationLabel.setHorizontalAlignment(SwingConstants.CENTER);
    panel.add(confirmationLabel, BorderLayout.CENTER);

    JPanel buttonPanel = new JPanel(new FlowLayout());

    JButton yesButton = new JButton( text: "Yes");
    yesButton.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            cardLayout.show(cardPanel,  name: "OrderSummary");
            updateOrderSummary();
        }
    });
    buttonPanel.add(yesButton);

    JButton noButton = new JButton( text: "No");
    noButton.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            System.exit( status: 0);
        }
    });
    buttonPanel.add(noButton);

    panel.add(buttonPanel, BorderLayout.SOUTH);

    return panel;
}
```



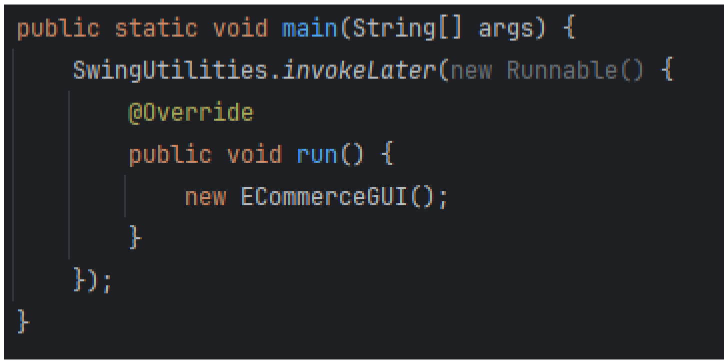
```java
private JPanel createOrderSummaryPanel() {  1 usage
    JPanel panel = new JPanel(new BorderLayout());

    JTextArea summaryTextArea = new JTextArea();
    summaryTextArea.setEditable(false);
    summaryTextArea.append("Order placed successfully!:D\n");
    summaryTextArea.append("Here's your order's summary:\n");
    summaryTextArea.append("Customer ID: " + customerId + "\n");
    summaryTextArea.append("Selected Products:\n");
    for (String product : selectedProducts) {
        if (product != null) {
            summaryTextArea.append("- " + product + "\n");
        }
    }
    summaryTextArea.append("Total Price: $" + calculateTotalPrice(selectedProducts) + "\n");
    summaryTextArea.append("thank you for your time!^-^");

    JScrollPane scrollPane = new JScrollPane(summaryTextArea);
    panel.add(scrollPane, BorderLayout.CENTER);

    JButton closeButton = new JButton( text: "Close");
    closeButton.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            System.exit( status: 0);
        }
    });
    panel.add(closeButton, BorderLayout.SOUTH);

    return panel;
}
```



```java
private float calculateTotalPrice(String[] selectedProducts) {  3 usages
    float totalPrice = 0.0f;
    for (String product : selectedProducts) {
        if (product != null) {
            if (product.contains("Smartphone")) {
                totalPrice += 599.9f;
            } else if (product.contains("T-shirt")) {
                totalPrice += 19.99f;
            } else if (product.contains("OOP Book")) {
                totalPrice += 39.99f;
            }
        }
    }
    return totalPrice;
}

private void updateOrderConfirmation(float totalPrice) {  1 usage
    JPanel orderConfirmationPanel = (JPanel) cardPanel.getComponent( n: 3);
    JLabel confirmationLabel = (JLabel) orderConfirmationPanel.getComponent( n: 0);
    confirmationLabel.setText("Your total is $" + totalPrice + ". Would you like to place the order?");
}

private void updateOrderSummary() {  1 usage
    JPanel orderSummaryPanel = (JPanel) cardPanel.getComponent( n: 4);
    JTextArea summaryTextArea = (JTextArea) ((JScrollPane) orderSummaryPanel.getComponent( n: 0)).getViewport().getView();
    summaryTextArea.setText("Here's your order's summary:\n");
    summaryTextArea.append("Customer ID: " + customerId + "\n");
    summaryTextArea.append("Selected Products:\n");
    for (String product : selectedProducts) {
        if (product != null) {
            summaryTextArea.append("- " + product + "\n");
        }
    }
    summaryTextArea.append("Total Price: $" + calculateTotalPrice(selectedProducts) + "\n");
}
```

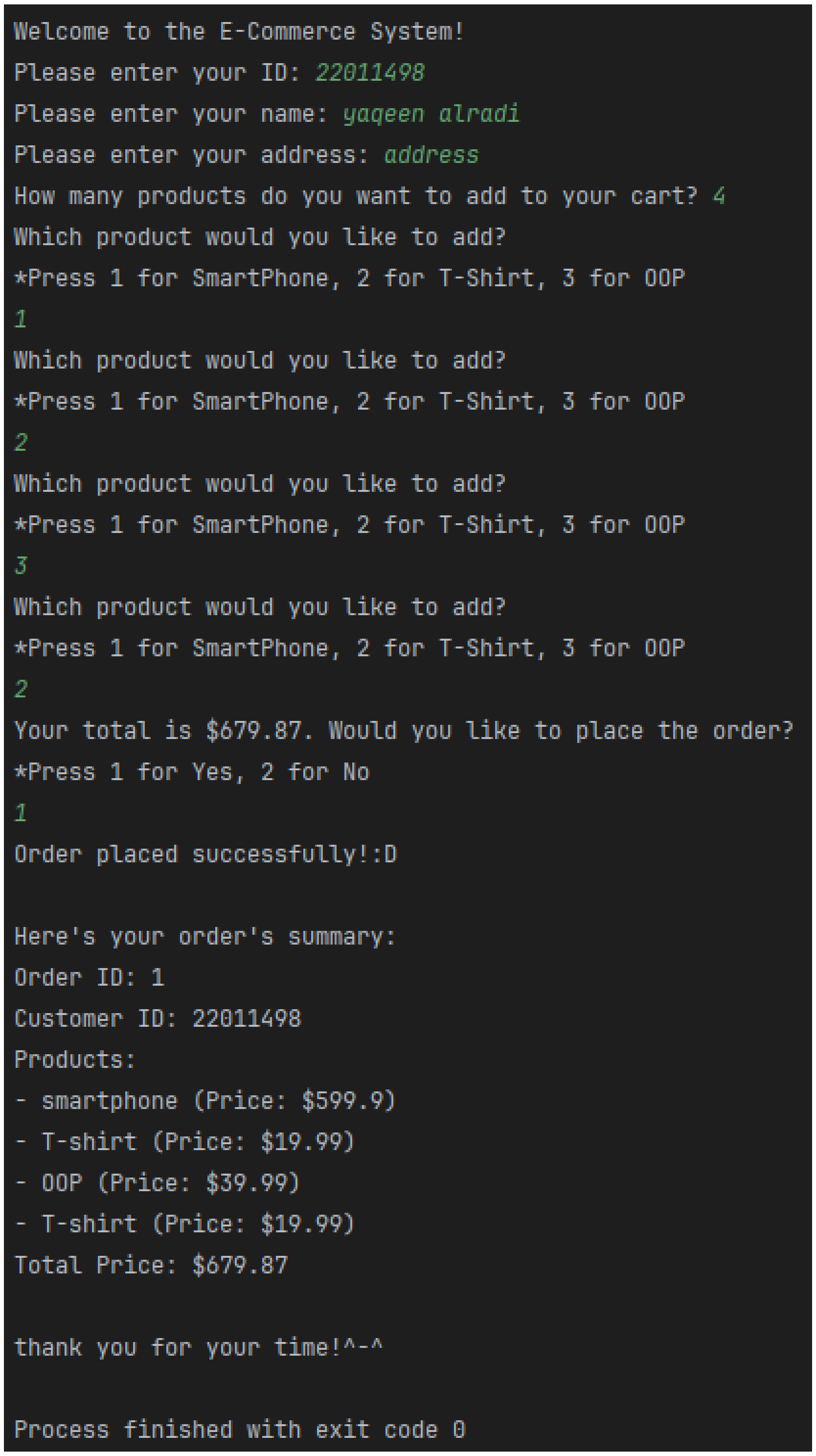


```java
private float calculateTotalPrice(String[] selectedProducts) {
    float totalPrice = 0.0f;
    for (String product : selectedProducts) {
        if (product != null) {
            if (product.contains("Smartphone")) {
                totalPrice += 599.9f;
            } else if (product.contains("T-shirt")) {
                totalPrice += 19.99f;
            } else if (product.contains("OOP Book")) {
                totalPrice += 39.99f;
            }
        }
    }
    return totalPrice;
}

private void updateOrderConfirmation(float totalPrice) {
    JPanel orderConfirmationPanel = (JPanel) cardPanel.getComponent(3);
    JLabel confirmationLabel = (JLabel) orderConfirmationPanel.getComponent(0);
    confirmationLabel.setText("Your total is $" + totalPrice + ". Would you like to place the order?");
}

private void updateOrderSummary() {
    JPanel orderSummaryPanel = (JPanel) cardPanel.getComponent(4);
    JTextArea summaryTextArea = (JTextArea) ((JScrollPane) orderSummaryPanel.getComponent(0)).getViewport().getView();
    summaryTextArea.setText("Here's your order's summary:\n");
    summaryTextArea.append("Customer ID: " + customerId + "\n");
    summaryTextArea.append("Selected Products:\n");
    for (String product : selectedProducts) {
        if (product != null) {
            summaryTextArea.append("- " + product + "\n");
        }
    }
    summaryTextArea.append("Total Price: $" + calculateTotalPrice(selectedProducts) + "\n");
}
```



```java
public static void main(String[] args) {
    SwingUtilities.invokeLater(new Runnable() {
        @Override
        public void run() {
            new ECommerceGUI();
        }
    });
}
```

***Terminal Output:***



```
Welcome to the E-Commerce System!
Please enter your ID: 22011498
Please enter your name: yaqeen alradi
Please enter your address: address
How many products do you want to add to your cart? 4
Which product would you like to add?
*Press 1 for SmartPhone, 2 for T-Shirt, 3 for OOP
1
Which product would you like to add?
*Press 1 for SmartPhone, 2 for T-Shirt, 3 for OOP
2
Which product would you like to add?
*Press 1 for SmartPhone, 2 for T-Shirt, 3 for OOP
3
Which product would you like to add?
*Press 1 for SmartPhone, 2 for T-Shirt, 3 for OOP
2
Your total is $679.87. Would you like to place the order?
*Press 1 for Yes, 2 for No
1
Order placed successfully!:D

Here's your order's summary:
Order ID: 1
Customer ID: 22011498
Products:
- smartphone (Price: $599.9)
- T-shirt (Price: $19.99)
- OOP (Price: $39.99)
- T-shirt (Price: $19.99)
Total Price: $679.87

thank you for your time!^-^

Process finished with exit code 0
```

***There's a GUI Output Too***