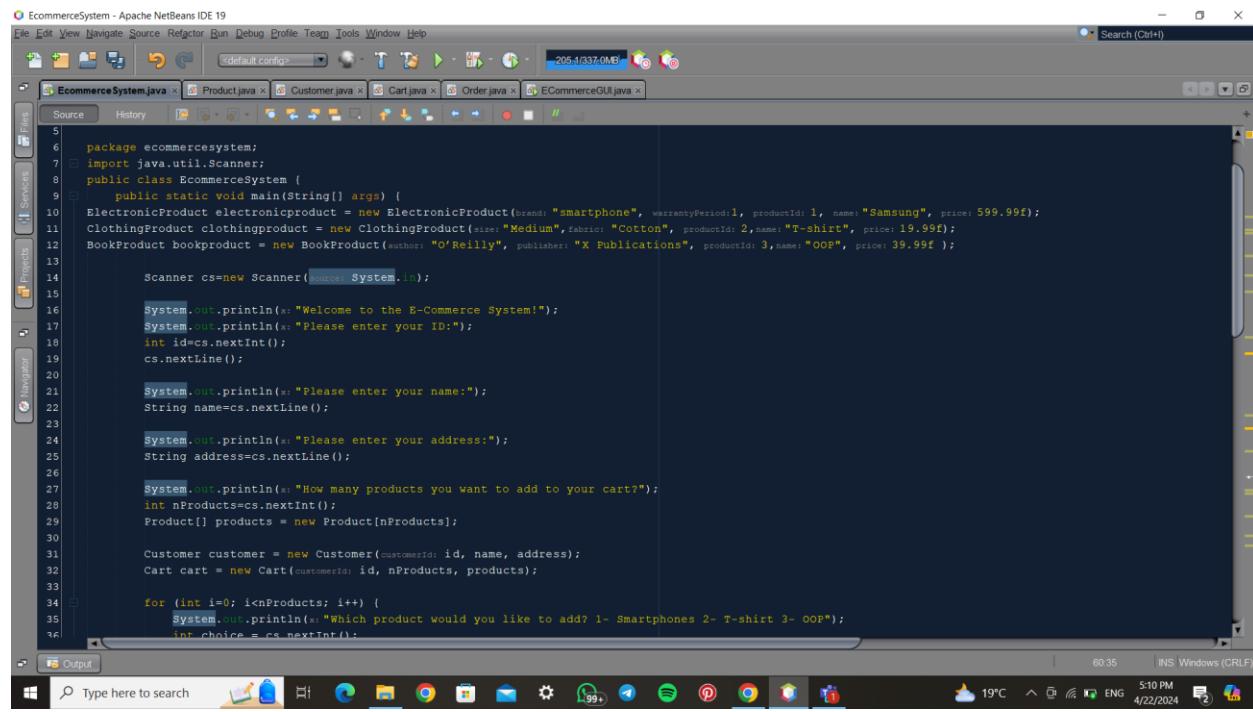


Name: Heba Ashraf AbdelMohsen Sayed Ahmed Elsharaby

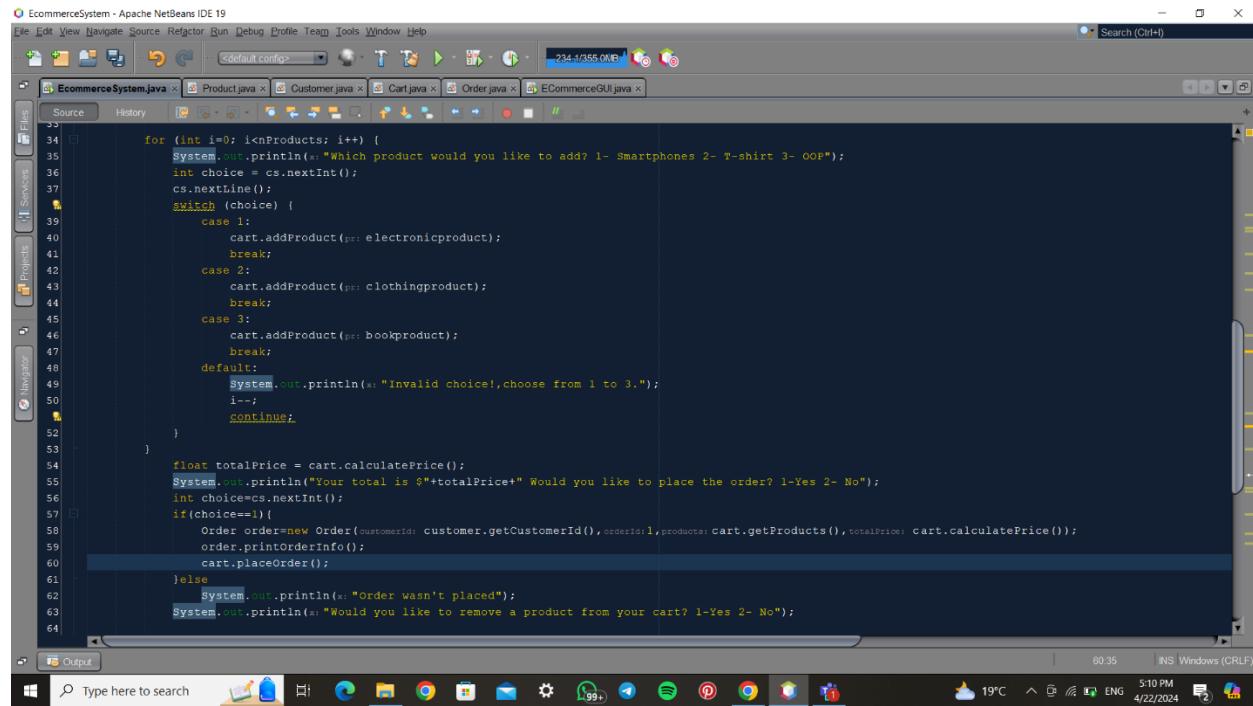
Id: 23012127

Project code:



EcommerceSystem - Apache NetBeans IDE 19

```
5 package ecommerceystem;
6 import java.util.Scanner;
7 public class EcommerceSystem {
8     public static void main(String[] args) {
9         ElectronicProduct electronicproduct = new ElectronicProduct(brand: "smartphone", warrantyPeriod:1, productId: 1, name: "Samsung", price: 599.99f);
10        ClothingProduct clothingproduct = new ClothingProduct(size: "Medium", fabric: "Cotton", productId: 2, name: "T-shirt", price: 19.99f);
11        BookProduct bookproduct = new BookProduct(author: "O'Reilly", publisher: "X Publications", productId: 3, name: "OOP", price: 39.99f );
12
13        Scanner cs=new Scanner(source: System.in);
14
15        System.out.println("Welcome to the E-Commerce System!");
16        System.out.println("Please enter your ID:");
17        int id=cs.nextInt();
18        cs.nextLine();
19
20        System.out.println("Please enter your name:");
21        String name=cs.nextLine();
22
23        System.out.println("Please enter your address:");
24        String address=cs.nextLine();
25
26        System.out.println("How many products you want to add to your cart?");
27        int nProducts=cs.nextInt();
28        Product[] products = new Product[nProducts];
29
30        Customer customer = new Customer(customerId: id, name, address);
31        Cart cart = new Cart(customerId: id, nProducts, products);
32
33        for (int i=0; i<nProducts; i++) {
34            System.out.println("Which product would you like to add? 1- Smartphones 2- T-shirt 3- OOP");
35            int choice = cs.nextInt();
36
37            switch (choice) {
38                case 1:
39                    cart.addProduct(pr: electronicproduct);
40                    break;
41                case 2:
42                    cart.addProduct(pr: clothingproduct);
43                    break;
44                case 3:
45                    cart.addProduct(pr: bookproduct);
46                    break;
47                default:
48                    System.out.println("Invalid choice!,choose from 1 to 3.");
49                    i--;
50                    continue;
51            }
52        }
53        float totalPrice = cart.calculatePrice();
54        System.out.println("Your total is $" +totalPrice + " Would you like to place the order? 1-Yes 2- No");
55        int choice=cs.nextInt();
56        if(choice==1){
57            Order order=new Order(customerId: customer.getCustomerId(), orderId:1,products: cart.getProducts(), totalPrice: cart.calculatePrice());
58            order.printOrderInfo();
59            cart.placeOrder();
60        }
61        else
62            System.out.println("Order wasn't placed");
63        System.out.println("Would you like to remove a product from your cart? 1-Yes 2- No");
64    }
65}
```



```
34        for (int i=0; i<nProducts; i++) {
35            System.out.println("Which product would you like to add? 1- Smartphones 2- T-shirt 3- OOP");
36            int choice = cs.nextInt();
37            cs.nextLine();
38            switch (choice) {
39                case 1:
40                    cart.addProduct(pr: electronicproduct);
41                    break;
42                case 2:
43                    cart.addProduct(pr: clothingproduct);
44                    break;
45                case 3:
46                    cart.addProduct(pr: bookproduct);
47                    break;
48                default:
49                    System.out.println("Invalid choice!,choose from 1 to 3.");
50                    i--;
51                    continue;
52            }
53        }
54        float totalPrice = cart.calculatePrice();
55        System.out.println("Your total is $" +totalPrice + " Would you like to place the order? 1-Yes 2- No");
56        int choice=cs.nextInt();
57        if(choice==1){
58            Order order=new Order(customerId: customer.getCustomerId(), orderId:1,products: cart.getProducts(), totalPrice: cart.calculatePrice());
59            order.printOrderInfo();
60            cart.placeOrder();
61        }
62        else
63            System.out.println("Order wasn't placed");
64        System.out.println("Would you like to remove a product from your cart? 1-Yes 2- No");
65    }
66}
```

EcommerceSystem - Apache NetBeans IDE 19

```
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("Order wasn't placed");
System.out.println("Would you like to remove a product from your cart? 1-Yes 2- No");

int removeChoice = cs.nextInt();
if(removeChoice == 1) {
    System.out.println("Enter the index of the product you want to remove:");
    int indexToRemove = cs.nextInt();
    cart.removeProduct(index: indexToRemove);
    System.out.println("Product removed successfully!");
} else {
    System.out.println("No products were removed.");
}
}
```

EcommerceSystem - Apache NetBeans IDE 19

```
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(productId);
        this.name = name;
        this.price = Math.abs(price);
    }

    public int getProductId() {
        return productId;
    }

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

    public String getName() {
        return name;
    }

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

    public float getPrice() {
        return price;
    }

    public void setPrice(float price) {
```

EcommerceSystem - Apache NetBeans IDE 19

```
Source History EcommerceSystem.java Product.java Customer.java Cart.java Order.java ECommerceGUI.java
```

```
38 }
39 }
40 class ElectronicProduct extends Product{
41     private String brand;
42     private int warrantyPeriod;
43
44     public ElectronicProduct(String brand, int warrantyPeriod, int productId, String name, float price) {
45         super(productId, name, price);
46         this.brand = brand;
47         this.warrantyPeriod = Math.abs(warrantyPeriod);
48     }
49
50     public String getBrand() {
51         return brand;
52     }
53     public void setBrand(String brand) {
54         this.brand = brand;
55     }
56     public int getWarrantyPeriod() {
57         return warrantyPeriod;
58     }
59     public void setWarrantyPeriod(int warrantyPeriod) {
60         this.warrantyPeriod = Math.abs(warrantyPeriod);
61     }
62 }
63 class ClothingProduct extends Product{
64     private String size;
65     private String fabric;
66
67     public ClothingProduct(String size, String fabric, int productId, String name, float price) {
68         super(productId, name, price);
69         this.size = size;
70     }
71
72     public String getSize() {
73         return size;
74     }
75     public void setSize(String size) {
76         this.size = size;
77     }
78
79     public String getFabric() {
80         return fabric;
81     }
82
83     public void setFabric(String fabric) {
84         this.fabric = fabric;
85     }
86 }
87 class BookProduct extends Product{
88     private String author;
89     private String publisher;
90
91     public BookProduct(String author, String publisher, int productId, String name, float price) {
92         super(productId, name, price);
93         this.author = author;
94         this.publisher = publisher;
95     }
96
97     public String getAuthor() {
```

84.30 INS Windows (CRLF)

EcommerceSystem - Apache NetBeans IDE 19

```
Source History EcommerceSystem.java Product.java Customer.java Cart.java Order.java ECommerceGUI.java
```

```
66     public ClothingProduct(String size, String fabric, int productId, String name, float price) {
67         super(productId, name, price);
68         this.size = size;
69         this.fabric = fabric;
70     }
71
72     public String getSize() {
73         return size;
74     }
75     public void setSize(String size) {
76         this.size = size;
77     }
78
79     public String getFabric() {
80         return fabric;
81     }
82
83     public void setFabric(String fabric) {
84         this.fabric = fabric;
85     }
86 }
87 class BookProduct extends Product{
88     private String author;
89     private String publisher;
90
91     public BookProduct(String author, String publisher, int productId, String name, float price) {
92         super(productId, name, price);
93         this.author = author;
94         this.publisher = publisher;
95     }
96
97     public String getAuthor() {
```

84.30 INS Windows (CRLF)

EcommerceSystem - Apache NetBeans IDE 19

```
public BookProduct(String name, 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;
}
```

Output

EcommerceSystem - Apache NetBeans IDE 19

```
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(customerId);
        this.name = name;
        this.address = address;
    }

    public int getCustomerId() {
        return customerId;
    }

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

    public String getName() {
        return name;
    }

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

    public String getAddress() {
        return address;
    }
```

Output

EcommerceSystem - Apache NetBeans IDE 19

```
private int customerId;
private String name;
private String address;

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

public int getCustomerId() {
    return customerId;
}

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

Output

EcommerceSystem - Apache NetBeans IDE 19

```
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(:customerId);
        this.nProducts = Math.abs(:nProducts);
        this.products = products;
    }

    public int getCustomerId() {
        return customerId;
    }

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

    public int getnProducts() {
        return nProducts;
    }

    public void setnProducts(int n) {
        if (n>0){
            nProducts=n;
        }
    }
}
```

Output

EcommerceSystem - Apache NetBeans IDE 19

```
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
default config 251.6/355.0MB Search (Ctrl+F)
EcommerceSystem.java x Product.java x Customer.java x Cart.java x Order.java x ECommerceGUI.java x
Source History
38     }
39     else {
40         nProducts= Math.abs(s- n);
41     }
42 }
43
44 public Product[] getProducts() {
45     return products;
46 }
47
48 public void setProducts(Product[] products) {
49     this.products = products;
50 }
51
52 public void addProduct (Product pr){
53     for(int i = 0 ; i<products.length ;i++){
54         if (products[i]==null){
55             products[i]=pr;
56             break;
57         }
58     }
59 }
60 public void removeProduct(int index) {if (index >= 0 && index < nProducts) {
61     for (int i = index; i < nProducts - 1; i++) {
62         products[i] = products[i + 1];
63     }
64     products[nProducts - 1] = null;
65     nProducts--;
66 }
67 } else {
68     System.out.println("Invalid index. Product not removed.");
69 }
70 }
Output
Type here to search 37.25 INS Windows (CRLF)
19°C 5:11 PM 4/22/2024
```

EcommerceSystem - Apache NetBeans IDE 19

```
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
default config 315.6/355.0MB Search (Ctrl+F)
EcommerceSystem.java x Product.java x Customer.java x Cart.java x Order.java x ECommerceGUI.java x
Source History
58 }
59 public void removeProduct(int index) {if (index >= 0 && index < nProducts) {
60     for (int i = index; i < nProducts - 1; i++) {
61         products[i] = products[i + 1];
62     }
63     products[nProducts - 1] = null;
64     nProducts--;
65 }
66 } else {
67     System.out.println("Invalid index. Product not removed.");
68 }
69 }
70 public float calculatePrice(){
71     float totalPrice=0;
72     for(int i =0; i<nProducts;i++){
73         totalPrice+= products[i].getPrice();
74     }
75     return totalPrice;
76 }
77
78 public void placeOrder(){
79     System.out.println("your order is placed successfully!");
80 }
81 }
82
83
84
85
Output
Type here to search 37.25 INS Windows (CRLF)
19°C 5:11 PM 4/22/2024
```

EcommerceSystem - Apache NetBeans IDE 19

```
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
default config 200.0350us Search (Ctrl+F)
EcommerceSystem.java x Product.java x Customer.java x Cart.java x Order.java x ECommerceGUI.java x
Source History + Services Projects Navigator Output
4  /*
5   * package ecommerceystem;
6   *
7   public class Order {
8       private int customerid;
9       private int orderId;
10      private Product[] products;
11      private float totalPrice;
12
13      public Order(int customerid, int orderId, Product[] products, float totalPrice) {
14          this.customerid = Math.abs(customerid);
15          this.orderId = Math.abs(orderId);
16          this.products = products;
17          this.totalPrice = Math.abs(totalPrice);
18      }
19
20      public void printOrderInfo() {
21          System.out.println("Here's your order's summary:");
22          System.out.println("Order's ID: " + orderId);
23          System.out.println("Customer's ID: " + customerid);
24          System.out.println("Products: ");
25          for (Product product : products) {
26              if (product != null) {
27                  System.out.println(product.getName() + " - $" + product.getPrice());
28              }
29          }
30          System.out.println("Total price: $" + totalPrice);
31      }
32  }
33
34 }
35
```

35.1 INS Windows (CRLF)

EcommerceSystem - Apache NetBeans IDE 19

```
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
default config 306.0365 MB Search (Ctrl+F)
EcommerceSystem.java x Product.java x Customer.java x Cart.java x Order.java x ECommerceGUI.java x
Source History + Services Projects Navigator Output
1  package ecommerceystem;
2  import javax.swing.*;
3  import java.awt.*;
4  import java.awt.event.ActionEvent;
5  import java.awt.event.ActionListener;
6
7  public class ECommerceGUI extends JFrame {
8      private JLabel customerIdLabel, nameLabel, addressLabel, totalPriceLabel;
9      private JTextField customerIdField, nameField, addressField;
10     private JComboBox<String> productsComboBox;
11     private JButton addProductButton, calculatePriceButton, printOrderButton;
12     private String[] productOptions = new String[3]; // Assuming a maximum of 3 products can be selected
13     private double totalPrice = 0;
14
15     public ECommerceGUI() {
16         setTitle(title: "E-Commerce System");
17         setSize(width: 400, height: 300);
18         setDefaultCloseOperation(operation: JFrame.EXIT_ON_CLOSE);
19         setLayout(new GridLayout(rows: 7, cols: 2));
20
21         customerIdLabel = new JLabel(text: "Customer ID:");
22         add(comp: customerIdLabel);
23         customerIdField = new JTextField();
24         add(comp: customerIdField);
25
26         nameLabel = new JLabel(text: "Name:");
27         add(comp: nameLabel);
28         nameField = new JTextField();
29         add(comp: nameField);
30
31         addressLabel = new JLabel(text: "Address:");
32         add(comp: addressLabel);
33
34         productsComboBox = new JComboBox<String>(options: productOptions);
35         add(comp: productsComboBox);
36
37         addProductButton = new JButton(text: "Add Product");
38         add(comp: addProductButton);
39
40         calculatePriceButton = new JButton(text: "Calculate Price");
41         add(comp: calculatePriceButton);
42
43         printOrderButton = new JButton(text: "Print Order");
44         add(comp: printOrderButton);
45
46         calculatePriceButton.addActionListener(actionListener: calculatePriceActionListener);
47         printOrderButton.addActionListener(actionListener: printOrderActionListener);
48
49         calculatePriceActionListener = new ActionListener() {
50             @Override
51             public void actionPerformed(ActionEvent event) {
52                 calculatePrice();
53             }
54         };
55
56         printOrderActionListener = new ActionListener() {
57             @Override
58             public void actionPerformed(ActionEvent event) {
59                 printOrder();
60             }
61         };
62
63     }
64
65     private void calculatePrice() {
66         totalPrice = 0;
67         for (int i = 0; i < productsComboBox.getItemCount(); i++) {
68             String selectedOption = productsComboBox.getSelectedItem();
69             if (selectedOption != null) {
70                 totalPrice += Double.parseDouble(selectedOption);
71             }
72         }
73         totalPriceLabel.setText("Total Price: $" + totalPrice);
74     }
75
76     private void printOrder() {
77         String customerId = customerIdField.getText();
78         String name = nameField.getText();
79         String address = addressField.getText();
80
81         System.out.println("Order Summary:");
82         System.out.println("Customer ID: " + customerId);
83         System.out.println("Name: " + name);
84         System.out.println("Address: " + address);
85
86         System.out.println("Selected Products:");
87         for (int i = 0; i < productsComboBox.getItemCount(); i++) {
88             String selectedOption = productsComboBox.getSelectedItem();
89             if (selectedOption != null) {
90                 System.out.println(selectedOption);
91             }
92         }
93
94         System.out.println("Total Price: $" + totalPrice);
95     }
96
97     private ActionListener calculatePriceActionListener;
98     private ActionListener printOrderActionListener;
99
100    public static void main(String[] args) {
101        SwingUtilities.invokeLater(() -> {
102            ECommerceGUI gui = new ECommerceGUI();
103            gui.setVisible(true);
104        });
105    }
106}
```

3.19 INS Windows (CRLF)

EcommerceSystem - Apache NetBeans IDE 19

```
Source History ... default config ... 197.3/355.0KB Search (Ctrl+F)
EcommerceSystem.java x Product.java x Customer.java x Cart.java x Order.java x ECommerceGUI.java x

26     nameLabel = new JLabel(text: "Name:");
27     add(comp.nameLabel);
28     nameField = new JTextField();
29     add(comp.nameField);
30
31     addressLabel = new JLabel(text: "Address:");
32     add(comp.addressLabel);
33     addressField = new JTextField();
34     add(comp.addressField);
35
36     productsLabel = new JLabel(text: "Select Product:");
37     add(comp.productsLabel);
38     String[] products = {"T-shirt - $19.99", "Smartphone - $599.9", "OOP Book - $39.99"};
39     productsComboBox = new JComboBox(<items: products>);
40     add(comp.productsComboBox);
41
42     addProductButton = new JButton(text: "Add Product");
43     add(comp.addProductButton);
44
45     totalPriceLabel = new JLabel(text: "Total Price: $0.0");
46     add(comp.totalPriceLabel);
47
48     calculatePriceButton = new JButton(text: "Calculate Total Price");
49     add(comp.calculatePriceButton);
50
51     printOrderButton = new JButton(text: "Print Order Info");
52     add(comp.printOrderButton);
53
54     addProductButton.addActionListener(new ActionListener() {
55         @Override
56         public void actionPerformed(ActionEvent e) {
57             for (int i = 0; i < selectedProducts.length; i++) {
58                 if (selectedProducts[i] == null) {
59                     selectedProducts[i] = (String) productsComboBox.getSelectedItem();
60                     JOptionPane.showMessageDialog(parentComponent: null, message: "Product added successfully!");
61                     break;
62                 }
63             }
64         }
65     });
66
67     calculatePriceButton.addActionListener(new ActionListener() {
68         @Override
69         public void actionPerformed(ActionEvent e) {
70             totalPrice = 0;
71             for (String product : selectedProducts) {
72                 if (product != null) {
73                     if (product.contains(: "T-shirt")) {
74                         totalPrice += 19.99;
75                     } else if (product.contains(: "Smartphone")) {
76                         totalPrice += 599.9;
77                     } else if (product.contains(: "OOP Book")) {
78                         totalPrice += 39.99;
79                     }
80                 }
81             }
82             totalPriceLabel.setText("Total Price: $" + totalPrice);
83         }
84     });
85
86     printOrderButton.addActionListener(new ActionListener() {
87         @Override
88         public void actionPerformed(ActionEvent e) {
89             JOptionPane.showMessageDialog(parentComponent: null, message: "Order printed!");
90         }
91     });
92 }
```

EcommerceSystem - Apache NetBeans IDE 19

```
Source History ... default config ... 278.3/355.0KB Search (Ctrl+F)
EcommerceSystem.java x Product.java x Customer.java x Cart.java x Order.java x ECommerceGUI.java x

57     for (int i = 0; i < selectedProducts.length; i++) {
58         if (selectedProducts[i] == null) {
59             selectedProducts[i] = (String) productsComboBox.getSelectedItem();
60             JOptionPane.showMessageDialog(parentComponent: null, message: "Product added successfully!");
61             break;
62         }
63     }
64 }
65
66 calculatePriceButton.addActionListener(new ActionListener() {
67     @Override
68     public void actionPerformed(ActionEvent e) {
69         totalPrice = 0;
70         for (String product : selectedProducts) {
71             if (product != null) {
72                 if (product.contains(: "T-shirt")) {
73                     totalPrice += 19.99;
74                 } else if (product.contains(: "Smartphone")) {
75                     totalPrice += 599.9;
76                 } else if (product.contains(: "OOP Book")) {
77                     totalPrice += 39.99;
78                 }
79             }
80         }
81         totalPriceLabel.setText("Total Price: $" + totalPrice);
82     }
83 });
84
85 printOrderButton.addActionListener(new ActionListener() {
86     @Override
87     public void actionPerformed(ActionEvent e) {
88         JOptionPane.showMessageDialog(parentComponent: null, message: "Order printed!");
89     }
90 });

Output Type here to search 19°C 5:11 PM 4/22/2024
```

The screenshot shows the Apache NetBeans IDE 19 interface with the file `EcommerceSystem.java` open. The code implements a GUI for an ecommerce system, handling user input and displaying order information. The IDE includes a toolbar, a file tree on the left, and various tabs at the top.

```
87     @Override
88     public void actionPerformed(ActionEvent e) {
89         StringBuilder orderInfo = new StringBuilder();
90         orderInfo.append("Customer ID: ").append(customerIdField.getText()).append("\n");
91         orderInfo.append("Name: ").append(usernameField.getText()).append("\n");
92         orderInfo.append("Address: ").append(addressField.getText()).append("\n");
93         orderInfo.append("Selected Products:\n");
94         for (String product : selectedProducts) {
95             if (product != null) {
96                 orderInfo.append("- ").append(product).append("\n");
97             }
98         }
99         orderInfo.append("Total Price: $").append(totalPrice);
100        JOptionPane.showMessageDialog(parentComponent, message, title, messageType);
101    }
102}
103
104    setVisible(true);
105}
106
107    public static void main(String[] args) {
108        SwingUtilities.invokeLater(() -> new EcommerceGUI());
109    }
110}
```

Code output:

The screenshot shows the Apache NetBeans IDE 19 interface with the file `EcommerceSystem.java` open. A modal dialog titled "Order Information" is displayed, showing the details of an order. The dialog includes fields for Customer ID, Name, Address, and Selected Products, along with calculated values for Total Price and Subtotal.

Customer ID: 23012127
Name: heba
Address: miami
Selected Products:
- T-shirt - \$19.99
- Smartphone - \$599.9
- OOP Book - \$39.99
Total Price: \$659.88

Order Information

Customer ID: 23012127
Name: heba
Address: miami
Selected Products:
- T-shirt - \$19.99
- Smartphone - \$599.9
- OOP Book - \$39.99
Total Price: \$659.88