

```
        }

    public void removeProduct(int index) {
        if (index >= 0 && index < nProducts) {
            products[index] = null;
        } else {
            System.out.println("Invalid index.");
        }
    }

    public float calculatePrice() {
        float totalPrice = 0;
        for (Product product : products) {
            if (product != null) {
                totalPrice += product.getPrice();
            }
        }
        return totalPrice;
    }

    public void placeOrder() {
        System.out.println("Order placed successfully.");
    }
}
```



The screenshot shows a Java IDE interface with the following details:

- Project Structure:** BankingAccount, data structures, EcommerceSystem, Source Packages, ecommerceSystem (selected), BookProduct.java, Cart.java, ClothingProduct.java, Customer.java, EcommerceSystem.java, ElectronicProduct.java, Order.java, Product.java.
- Code Editor:** The main editor window displays Java code for the `EcommerceSystem` class. The code handles product addition to a cart, printing invalid product type errors, and prompting the user to place an order. It then calculates the total price, places the order, prints order information, or thanks the user for visiting.
- Navigator:** The Navigator panel on the left shows the members of the `EcommerceSystem` class, including `EcommerceSystem()`, `createCustomer(Scanner)`, and `main(String[] args)`.
- Bottom Bar:** The taskbar shows various pinned icons and system status indicators like battery level, signal strength, and system volume.

```
case 3:  
    cart.addProduct(product:bookProduct, index: i);  
    break;  
default:  
    System.out.println(x: "Invalid product type.");  
    break;  
}  
  
tem.out.print(x: "Do you want to place an order for the products in the cart? (yes/no): ");  
ing response = scanner.nextLine();  
(response.equalsIgnoreCase(anotherstring: "yes")) {  
    float totalPrice = cart.calculatePrice();  
    System.out.println("Total Price: $" + totalPrice);  
    cart.placeOrder();  
    // Print order info  
    System.out.println(x: "\nOrder Information:");  
    Order order = new Order(customerId: customer.getCustomerId(), orderId:1, products:cart.getProducts(), (float) cart.calculatePrice());  
    order.printOrderInfo();  
} else {  
    System.out.println(x: "Thank you for visiting!");  
}  
  
static Customer createCustomer(Scanner scanner) {  
    tem.out.println(x: "Enter customer details:");  
    tem.out.println(x: "Name: ");  
    String name = scanner.nextLine();  
    tem.out.println(x: "Address: ");  
    String address = scanner.nextLine();  
    tem.out.println(x: "Phone Number: ");  
    String phone = scanner.nextLine();  
    tem.out.println(x: "Email ID: ");  
    String email = scanner.nextLine();  
    tem.out.println(x: "Password: ");  
    String password = scanner.nextLine();  
    tem.out.println(x: "Confirm Password: ");  
    String confirmPassword = scanner.nextLine();  
    if (password.equals(confirmPassword)) {  
        Customer customer = new Customer(name, address, phone, email, password);  
        return customer;  
    } else {  
        System.out.println(x: "Passwords do not match");  
    }  
}
```

```
Page X | cart.java X | Prog_project.java X | order.java X | product.java X | book_p.java X |  
History |                   
package prog_project;  
import javax.swing.JOptionPane;  
  
public class Prog_project {  
  
    public static void main(String[] args) {  
        // Create product instances  
        electpro eli = new electpro();  
        clothing_p drip = new clothing_p();  
        book_p bookworm = new book_p();  
  
        // Set product details  
        eli.setBrand(brand: "Samsung");  
        eli.setId(id: 1);  
        eli.setPrice(price: 599.9f);  
        eli.setName(name: "Smartphone");  
        eli.setWarranty_period(warranty_period: 1);  
  
        drip.setFabric(fabric: "Cotton");  
        drip.setId(id: 2);  
        drip.setName(name: "T-shirt");  
        drip.setPrice(price: 19.99f);  
        drip.setSize(size: "Medium");  
  
        bookworm.setAuthor(author: "O Reilly");  
        bookworm.setId(id: 3);  
        bookworm.setPrice(price: 39.99f);  
        bookworm.setPublisher(publisher: "X Publications");  
        bookworm.setName(name: "OOP");  
  
        // Create customer instance  
        customer cuh = new customer();  
  
        // Set customer details  
        String name = JOptionPane.showInputDialog(parentComponent: null, message:  
cuh.setName(name);
```

The screenshot shows a Java IDE interface with multiple tabs open. The active tab is 'cart.java'. The code in the editor is as follows:

```
34 // Set customer details
35 String name = JOptionPane.showInputDialog(parentComponent: null, message: "Enter name:");
36 cuh.setName(name);
37
38 int id = Integer.parseInt(JOptionPane.showInputDialog(parentComponent: null, message: "Enter id:"));
39 cuh.setId(id);
40
41 String address = JOptionPane.showInputDialog(parentComponent: null, message: "Enter address:");
42 cuh.setAddress(address);
43
44 int n_products = Integer.parseInt(JOptionPane.showInputDialog(parentComponent: null, message: "How many items would you like to order?"));
45 cart carti = new cart(size: n_products);
46
47 for (int i = 0; i < n_products; i++) {
48     Object[] options = {drip.getName(), bookworm.getName(), eli.getName()};
49     int op = JOptionPane.showOptionDialog(parentComponent: null, message: "Which product would you like to add?", title: "Select Product", optionType: JOptionPane.DEFAULT_OPTION, messageType: JOptionPane.QUESTION_MESSAGE);
50     if (op == 0)
51         carti.addproduct(product: drip);
52     else if (op == 1)
53         carti.addproduct(product: bookworm);
54     else
55         carti.addproduct(product: eli);
56 }
57
58 order ord = new order();
59 ord.printorderinfo(cuh, carti);
60
61 int choice = JOptionPane.showConfirmDialog(parentComponent: null, message: "Do you want to remove something?", title: "Remove Product", optionType: JOptionPane.YES_NO_OPTION);
62 if (choice == JOptionPane.YES_OPTION) {
63     String productName = JOptionPane.showInputDialog(parentComponent: null, message: "Enter product name:");
64     carti.removeProduct(productName);
65     StringBuilder updatedOrder = new StringBuilder(str: "Your updated order is:\n");
66     updatedOrder.append(str: "Customer Name: ").append(str: cuh.getName()).append(str: "\n");
67     updatedOrder.append(str: "Customer ID: ").append(i: cuh.getId()).append(str: "\n");
68     updatedOrder.append(str: "Customer Address: ").append(str: cuh.getAddress()).append(str: "\n\n");
69 }
```

File Edit View Navigate Source Refactor Run Debug Profile Team Window Help

Project Object Apache NetBeans IDE 19 Search (Ctrl+F)

Start Page cart.java Prog_project.java order.java product.java book_p.java clothing_p.java customer.java electpro.java

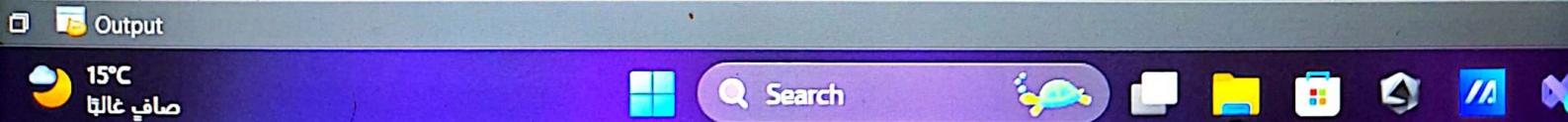
Source History

```
55 }  
56  
57 order ord = new order();  
ord.printorderinfo(cuh, carti);  
  
int choice = JOptionPane.showConfirmDialog(parentComponent: null, message: "Do you want to remove something?", title: "Remove Product", optionType: JOptionPane.YES_NO_OPTION);  
if (choice == JOptionPane.YES_OPTION) {  
    String productName = JOptionPane.showInputDialog(parentComponent: null, message: "Enter product name:");  
    carti.removeProduct(productName);  
    StringBuilder updatedOrder = new StringBuilder(str: "Your updated order is:\n");  
    updatedOrder.append(str: "Customer Name: ") .append(str: cuh.getName()).append(str: "\n");  
    updatedOrder.append(str: "Customer ID: ") .append(str: cuh.getId()).append(str: "\n");  
    updatedOrder.append(str: "Customer Address: ") .append(str: cuh.getAddress()).append(str: "\n\n");  
  
    updatedOrder.append(str: "Products in Cart:\n");  
    for (int i = 0; i < carti.getNProducts(); i++) {  
        updatedOrder.append(str: carti.products[i].getName() .append(str: " - $") .append(str: carti.products[i].getPrice()) .append(str: "\n"));  
    }  
    JOptionPane.showMessageDialog(parentComponent: null, message: updatedOrder.toString());  
} else {  
    JOptionPane.showMessageDialog(parentComponent: null, message: "Thank you for using our program <3");  
}  
}
```

Output

15°C 4/22/2024 5:42 AM ENG INS Windo

```
1 package prog_project;
2 import java.util.*;
3
4 public class cart extends customer {
5     private int customerId;
6     private int nProducts = 0;
7     private int size;
8     product[] products;
9
10    public cart(int size) {
11        this.size = size;
12        products = new product[size];
13    }
14
15    public int getCustomerId() {
16        return customerId;
17    }
18
19    public int getNProducts() {
20        return nProducts;
21    }
22
23    public void setcustomerId(int customerId) {
24        this.customerId = customerId;
25    }
26
27    public void addproduct(product product) {
28        if (nProducts < size) {
29            products[nProducts++] = product;
30        } else {
31            System.out.println("Cart is full");
32        }
33    }
34    public void removeProduct(String productName) {
35        for (int i = 0; i < nProducts; i++) {
36            if (products[i].getName().equals(anObject: productName)) {
37                // Shift the elements to the left to fill the gap
38            }
39        }
40    }
41}
```



```
public void removeProduct(String productName) {
    for (int i = 0; i < nProducts; i++) {
        if (products[i].getName().equals(anObject: productName)) {
            // Shift the elements to the left to fill the gap
            for (int j = i; j < nProducts - 1; j++) {
                products[j] = products[j + 1];
            }
            // Reduce the number of products
            nProducts--;
            // Exit the loop after the first occurrence is removed
            return;
        }
    }
    // If the product is not found
    System.out.println("Product " + productName + " not found in the cart");
}

public double calcprice() {
    double sum = 0;
    for (int i = 0; i < nProducts; i++) {
        sum += products[i].getPrice();
    }
    return sum;
}

public void placeorder(customer customer) {
    order order = new order();
    order.printorderinfo(cuh:customer, carti: this);
}

public void productslist(){
    for(int i=0;i<nProducts;i++){
        System.out.println((i+1)+"："+products[i].getName()+"->" +products[i].getPrice());
    }
}
```



Page X | cart.java X | Prog_project.java X | orderjava X | product.java X | book_p.java X | clothing_p.java X | customer.java X | electpro.java X

History |

```
package prog_project;
import javax.swing.JOptionPane;

public class order extends customer {
    int order_id = 1;

    public void printorderinfo(customer cuh, cart carti) {
        StringBuilder orderInfo = new StringBuilder();
        orderInfo.append(str: "Order ID: ").append(i: order_id).append(str: "\n");
        orderInfo.append(str: "Customer ID: ").append(i: cuh.getId()).append(str: "\n");
        orderInfo.append(str: "Total Price: ").append(d: carti.calcprice()).append(str: "\n");
        orderInfo.append(str: "Products in Cart:\n");
        for (int i = 0; i < carti.getNProducts(); i++) {
            orderInfo.append(str: carti.products[i].getName() + " - $").append(f: carti.products[i].getPrice()).append(str: "\n");
        }
        JOptionPane.showMessageDialog(parentComponent: null, message: orderInfo.toString(), title: "Order Summary", messageType: JOptionPane.INFORMATION_MESSAGE);
    }
}
```

Start Page X cart.java X Prog_project.java X order.java X product.java X book_p.java X

Source History

```
4  */
5  package prog_project;
6
7
8  public class product {
9      private int id;
10     private String name;
11     private float price;
12
13     public int getId() {
14         return id;
15     }
16
17     public String getName() {
18         return name;
19     }
20
21     public float getPrice() {
22         return price;
23     }
24
25     public void setId(int id) {
26         this.id = id;
27     }
28
29     public void setName(String name) {
30         this.name = name;
31     }
32
33     public void setPrice(float price) {
34         this.price = price;
35     }
36
37
38
39 }
40
```

Output

15°C
صافٰ غلباً



Search



```
* Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.j
* Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.j
*/
package prog_project;

/**
 *
 * @author ASUS
 */
public class book_p extends product{
    String author;
    String publisher;

    public String getAuthor() {
        return author;
    }

    public String getPublisher() {
        return publisher;
    }

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

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

```
1  /*
2  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to
3  * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this
4  */
5 package prog_project;
6
7 /**
8 *
9 * @author ASUS
10 */
11 public class clothing_p extends product{
12     String size;
13     String fabric;
14
15     public String getSize() {
16         return size;
17     }
18
19     public String getFabric() {
20         return fabric;
21     }
22
23     public void setSize(String size) {
24         this.size = size;
25     }
26
27     public void setFabric(String fabric) {
28         this.fabric = fabric;
29     }
30
31 }
32
```



Start Page | cart.java | Prog_project.java | order.java | product.java | book_p.java | ...

Source History

```
package prog_project;
import java.util.logging.Logger;

/**
 * 
 * @author ASUS
 */
public class customer {
    int id;
    String name;
    String address;

    public int getId() {
        return id;
    }

    public String getName() {
        return name;
    }

    public String getAddress() {
        return address;
    }

    public void setId(int id) {
        this.id = id;
    }

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

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

Output

15°C

Start Page X | cart.java X | Prog_project.java X | order.java X | product.java X | bc

Source History

```
1  /*
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.ja-
4   */
5 package prog_project;
6
7 public class electpro extends product{
8     String brand;
9     int warranty_period;
10
11
12     public String getBrand() {
13         return brand;
14     }
15
16     public int getWarranty_period() {
17         return warranty_period;
18     }
19
20     public void setBrand(String brand) {
21         this.brand = brand;
22     }
23
24     public void setWarranty_period(int warranty_period) {
25         this.warranty_period = warranty_period;
26     }
27
28
29
30 }
31
```

Output



15°C
صافٍ غالباً



Search

