

EcommerceSystem.java	test.java	Customer.java	Product.java X
----------------------	-----------	---------------	----------------

Product.java > ...

```
1  public class Product
2  {
3      protected int productId ;
4      protected String name ;
5      protected float price ;
6
7      public Product(int productId , String name , float price)
8      {
9          setProductId(productId);
10         setName(name);
11         setPrice(price);
12     }
13
14     public int getProductId() {
15         return productId;
16     }
17
18     private void setProductId(int productId){
19         this.productId = Math.abs(productId);
20     }
21
22     public String getName() {
23         return name;
24     }
25
26     private void setName(String name) {
27         this.name = name;
28     }
29
30     public float getPrice() {
31         return price;
32     }
33
34     private void setPrice(float price){
35         this.price = Math.abs(price);
36     }
37 }
38
```

EcommerceSystem.java

test.java

Customer.java

ElectronicProduct.java X

Cart.java

Order.java

ElectronicProduct.java > ElectronicProduct > getBrand()

```
1 public class ElectronicProduct extends Product {
2
3     private String brand ;
4     private int warrentyPeriod ;
5
6
7     public ElectronicProduct(int productId , String name , float price , String brand , int warrentyPeriod)
8     {
9         super(productId, name, price);
10        setBrand(brand);
11        setWarrentyPeriod(warrentyPeriod);
12    }
13
14
15    public String getBrand() {
16        return brand;
17    }
18
19 > private void setBrand(String brand) { ...
22
23    public int getWarrentyPeriod() {
24        return warrentyPeriod;
25    }
26
27    private void setWarrentyPeriod(int warrentyPeriod) {
28        this.warrentyPeriod = Math.abs(warrentyPeriod);
29    }
30
31 }
32
```

```
1  public class BookProduct extends Product
2  {
3      private String author , publisher ;
4
5  > public BookProduct(int productId , String name , float price , String author , String publisher) ...
11
12     public String getAuthor() {
13         return author;
14     }
15
16     public void setAuthor(String author) {
17         this.author = author;
18     }
19
20     public String getPublisher() {
21         return publisher;
22     }
23
24     public void setPublisher(String publisher) {
25         this.publisher = publisher;
26     }
27 }💡
28
```

ClothingProduct.java } wasn't finished with the GUI yet in the discussion

```
1 public class ClothingProduct extends Product {
2
3
4     private String size , fabric ;
5
6     public ClothingProduct(int productId , String name , float price , String size , String fabric)
7     {
8         super(productId, name, price);
9         setFabric(fabric);
10        setSize(size);
11    }
12
13
14    public String getSize() {
15        return size;
16    }
17
18    private void setSize(String size) {
19        this.size = size;
20    }
21
22    public String getFabric() {
23        return fabric;
24    }
25
26    private void setFabric(String fabric) {
27        this.fabric = fabric;
28    }
29 }
30
```



```
1 public class Customer {
2
3     private int customerId ;
4     private String name , address ;
5
6     public Customer(int customerId , String name , String address)
7     {
8         setName(name);
9         setCustomerId(customerId);
10        setAddress(address);
11    }
12
13    public int getCustomerId() {
14        return customerId;
15    }
16
17    private void setCustomerId(int customerId) {
18        this.customerId = Math.abs(customerId);
19    }
20
21    public String getName() {
22        return name;
23    }
24
25    private void setName(String name) {
26        this.name = name;
27    }
28
29    public String getAddress() {
30        return address;
31    }
32
33    private void setAddress(String address) {
34        this.address = address;
35    }
36
37 }💡
```

Go Run ...



🔍 OOP Project 1

EcommerceSystem.java

test.java

Cart.java

Order.java

Cart.java > Cart

```
1 public class Cart
2 {
3     private int customerId , nProducts , nLoggedProducts = 0 ;
4     Product[] products ;
5
6     public Cart(int customerId , int nProducts)
7     {
8         setCustomerId(customerId);
9         setnProducts(nProducts);
10        products = new Product[nProducts];
11    }
12
13    public int getnLoggedProducts() {
14        return nLoggedProducts;
15    }
16
17    public int getCustomerId() {
18        return customerId;
19    }
20
21    private void setCustomerId(int customerId) {
22        this.customerId = Math.abs(customerId) ;
23    }
24
25    public int getnProducts() {
26        return nProducts;
27    }
28
29    private void setnProducts(int nProducts) {
30        this.nProducts = Math.abs(nProducts);
31    }
```

EcommerceSystem.java test.java Cart.java X Order.java

Cart.java > Cart > calculatePrice()

```
1 public class Cart
32
33     public void addProduct(Product p)
34     {
35         if(nLoggedProducts < nProducts)
36         {
37             products[nLoggedProducts] = p ;
38             nLoggedProducts++ ;
39         }
40         else
41             System.out.println(x:"you exceeded the maximum number of products");
42     }
43
44     public void removeProduct(Product p)
45     {
46         for(int i = 0 ; i < nLoggedProducts ; i++)
47         {
48             if(products[i].getProductId() == p.getProductId())
49             {
50                 for(int j = i ; j < nLoggedProducts-1 ; j++)    //? an array that shift the rest of the products in the array after one is removed
51                     products[j] = products[j+1] ;              //? (this's so if the array is full and you remove a product and add another this new
52                     //? product object will be added to the last array element removing the one already there)
53             }
54             products[nLoggedProducts-1] = null ;
55             nLoggedProducts-- ;
56             return ;
57         }
58     }
59
60     System.out.println(x:"not found");
61 }
```

EcommerceSystem.java

test.java

Cart.java X

Order.java

Cart.java > ...

```
1  public class Cart
62
63      public float calculatePrice()
64      {
65          float totalPrice = 0 ;
66          for(int i = 0 ; i < products.length ; i++)
67              totalPrice += products[i].getPrice() ;
68
69          return totalPrice ;
70      }
71
72      public Order placOrder()
73      {
74          Order order1 = new Order(this.getCustomerId() , Order.getOrderIds() , this.calculatePrice() );
75          order1.setProducts(this.products);
76          return order1 ;
77      }
78  }
79
```


EcommerceSystem.java	test.java	Cart.java	Order.java X
----------------------	-----------	-----------	--------------

Order.java > Order

```
1  import javax.swing.JOptionPane;
2
3  public class Order
4  {
5      private int customerId , orderId ;
6      private float totalPrice ;
7      private static int orderIds = 1 ;      //? used as a counter for the order ids of the store
8      private Product[] products ;
9
10     public Order(int customerId , int orderId , float totalPrice )
11     {
12         setCustomerId(customerId);
13         setOrderId(orderIds);
14         setTotalPrice(totalPrice);
15         orderIds++ ;
16     }
17
18     public Order()
19     {}
20
21     public int getCustomerId() {
22         return customerId;
23     }
24
25     private void setCustomerId(int customerId) {
26         this.customerId = Math.abs(customerId) ;
27     }
28
29     public int getOrderId() {
30         return orderId;
31     }
32
33     private void setOrderId(int orderId) {
34         this.orderId = orderId;
35     }
36
37     public float getTotalPrice() {
38         return totalPrice;
39     }
40 }
```

Ln 40, Col 1 Spaces: 4 UTF-8

Order.java > Order

```
3 public class Order
39 {
40
41     private void setTotalPrice(float totalPrice) {
42         this.totalPrice = totalPrice;
43     }
44
45     public Product[] getProducts() {
46         return products;
47     }
48
49     public void setProducts(Product[] products) {
50         this.products = products;
51     }
52
53     public static int getOrderIds() {
54         return orderIds;
55     }
56
57     public void printOrderInfo()
58     {
59         StringBuilder message = new StringBuilder();
60
61         message.append(str:"~~~~~\nHere's your order's summary: \nOrder Id: ").append(getOrderId())
62             .append(str:"\nCustomer-Id: ").append(getCustomerId()).append(str:"\n~~~~~\n");
63
64         for(int i = 0; i < products.length; i++)
65             message.append(products[i].getName()).append(str:" - $").append(products[i].getPrice()).append(str:"\n");
66
67         message.append(str:"~~~~~\nTotal price: $").append(totalPrice)
68             .append(str:"\n~~~~~\nlooking forward seeing you soon ^_^\n");
69
70         JOptionPane.showMessageDialog(parentComponent:null, message.toString());
71     }
72
73 }
74
```

Go Run Terminal Help

← →

OOP Project 1

EcommerceSystem.java ×

test.java

Cart.java

Order.java

EcommerceSystem.java > EcommerceSystem > main(String[])

```
1 import javax.swing.JOptionPane;
2
3 public class EcommerceSystem
4 {
5     Run | Debug
6     public static void main(String[] args) {
7         Product p1 = new ElectronicProduct(productId:1, name:"smartphone", price:599.99f , brand:"Samsung", warrentyPeriod:1);
8         Product p2 = new ElectronicProduct(productId:2, name:"Iron",price:299.99f, brand:"Toshiba", warrentyPeriod:2);
9         Product p3 = new ElectronicProduct(productId:3, name:"vacuum cleaner",price:599.99f, brand:"Fresh", warrentyPeriod:3);
10        Product p4 = new ElectronicProduct(productId:4, name:"fridge",price:1999.99f, brand:"Samsung", warrentyPeriod:5);
11        Product p5 = new ClothingProduct(productId:5, name:"T-shirt", price:19.99f, size:"small", fabric:"cotton");
12        Product p6 = new ClothingProduct(productId:6, name:"pants", price:59.99f, size:"medium", fabric:"jeans");
13        Product p7 = new ClothingProduct(productId:7, name:"chemise", price:29.99f, size:"large", fabric:"cotton");
14        Product p8 = new ClothingProduct(productId:8, name:"jacket", price:79.99f, size:"x-large", fabric:"leather");
15        Product p9 = new BookProduct(productId:9, name:"OOP", price:39.99f , author:"O'Reilly", publisher:"X Publications");
16        Product p10 = new BookProduct(productId:10, name:"Harry Potter and the chambre of secrets",price:49.99f , author:"J. K. Rolling", publisher:
17        Product p11 = new BookProduct(productId:11, name:"Fire and Blood", price:45.0f , author:"George R. R. Martin", publisher:"bla bla bla2");
18        Product p12 = new BookProduct(productId:12, name:"Rich dad poor dad", price:29.99f , author:"Robert T. Kiyosaki and Sharon Lechter", publish
19
20        int customerId = Integer.parseInt(JOptionPane.showInputDialog(message:"Welcome to DigitMall ^_^\nenter you id")) ;
21        String name = JOptionPane.showInputDialog(message:"enter you name");
22        String address = JOptionPane.showInputDialog(message:"enter you address");
23        Customer c1 = new Customer(customerId, name, address);
24
25        int nProducts = Integer.parseInt(JOptionPane.showInputDialog(message:"how many products would you like to add to your cart? "));
26        Cart cart1 = new Cart(c1.getCustomerId(), nProducts);
27        for(int i = 0 ; i < nProducts ; )
28        {
29            int option = Integer.parseInt(JOptionPane.showInputDialog("which product would you like to add ? 1- smartphone 2- Iron "+
30            " 3- vacuum cleaner 4- fridge 5- t-shirt 6- pants 7- chemise "+
31            " 8- jacket 9- OOP 10- harry potter & the chambre of secrets "+
32            "11- Fire & Blood 12- rich dad poor dad"));
33            boolean validOption = true ;
```



```
3 public class EcommerceSystem
5     public static void main(String[] args)

34         switch (option)
35         {
36             case 1:
37                 cart1.addProduct(p1);
38                 break;
39             case 2:
40                 cart1.addProduct(p2);
41                 break;
42             case 3:
43                 cart1.addProduct(p3);
44                 break;
45             case 4:
46                 cart1.addProduct(p4);
47                 break;
48             case 5:
49                 cart1.addProduct(p5);
50                 break;
51             case 6:
52                 cart1.addProduct(p6);
53                 break;
54             case 7:
55                 cart1.addProduct(p7);
56                 break;
57             case 8:
58                 cart1.addProduct(p8);
59                 break;
60             case 9:
61                 cart1.addProduct(p9);
62                 break;
63             case 10:
64                 cart1.addProduct(p10);
65                 break;
66             case 11:
67                 cart1.addProduct(p11);
68                 break;
69             case 12:
70                 cart1.addProduct(p12);
```


EcommerceSystem.java X

test.java

Cart.java

Order.java

EcommerceSystem.java > EcommerceSystem > main(String[])

```
3 public class EcommerceSystem
5     public static void main(String[] args)
71         break;
72     default:
73     {
74         JOptionPane.showMessageDialog(parentComponent:null, message:"invalid option");
75         validOption = false ;
76         break;
77     }
78 }
79 if(validOption)
80     i++;
81 }
82
83 Order order1 = new Order();
84 int option = Integer.parseInt(JOptionPane.showInputDialog("the total price is $" + cart1.calculatePrice() + "\nwould you like to place the o
85 switch (option).
86 {
87     case 1:
88     {
89         order1 = cart1.placOrder();
90         order1.printOrderInfo();
91         break;
92     }
93     case 2:
94     {
95         int option2 = Integer.parseInt(JOptionPane.showInputDialog(message:"would you like to remove any product from the cart and add anoth
96         switch (option2).
97         {
98             case 1:
99             {
100                 int option3 = Integer.parseInt(JOptionPane.showInputDialog("which product you want to remove ? 1- smartphone 2- Iron "+
101                 " 3- vacuum cleaner 4- fridge 5- t-shirt 6- pants "+
102                 " 7- chemise 8- jacket 9- OOP 10- harry potter & the chambre of
103                 "11- Fire & Blood 12- rich dad poor dad"));
104                 switch (option3)
105                 {
106                     case 1:
107                         cart1.removeProduct(p1);
```

Go Run Terminal Help ← → OOP Project 1

EcommerceSystem.java × test.java Cart.java Order.java

EcommerceSystem.java > EcommerceSystem > main(String[])

```
3 public class EcommerceSystem
5     public static void main(String[] args)
108         break;
109     case 2:
110         cart1.removeProduct(p2);
111         break;
112     case 3:
113         cart1.removeProduct(p3);
114         break;
115     case 4:
116         cart1.removeProduct(p4);
117         break;
118     case 5:
119         cart1.removeProduct(p5);
120         break;
121     case 6:
122         cart1.removeProduct(p6);
123         break;
124     case 7:
125         cart1.removeProduct(p7);
126         break;
127     case 8:
128         cart1.removeProduct(p8);
129         break;
130     case 9:
131         cart1.removeProduct(p9);
132         break;
133     case 10:
134         cart1.removeProduct(p10);
135         break;
136     case 11:
137         cart1.removeProduct(p11);
138         break;
139     case 12:
140         cart1.removeProduct(p12);
141         break;
142     default:
143         JOptionPane.showMessageDialog(parentComponent:null, message:"invalid option");
144         break;
```

Go Run Terminal Help

← →

🔍 OOP Project 1

EcommerceSystem.java × test.java Cart.java Order.java

EcommerceSystem.java > EcommerceSystem > main(String[])

```
3 public class EcommerceSystem
5     public static void main(String[] args)
144         break;
145     }
146     int option5 = Integer.parseInt(JOptionPane.showInputDialog(message:"which product would you like to add instead of the one y
147     switch (option5)
148     {
149         case 1:
150             cart1.addProduct(p1);
151             break;
152         case 2:
153             cart1.addProduct(p2);
154             break;
155         case 3:
156             cart1.addProduct(p3);
157             break;
158         case 4:
159             cart1.addProduct(p4);
160             break;
161         case 5:
162             cart1.addProduct(p5);
163             break;
164         case 6:
165             cart1.addProduct(p6);
166             break;
167         case 7:
168             cart1.addProduct(p7);
169             break;
170         case 8:
171             cart1.addProduct(p8);
172             break;
173         case 9:
174             cart1.addProduct(p9);
175             break;
176         case 10:
177             cart1.addProduct(p10);
178             break;
179         case 11:
180             cart1.addProduct(p11);
```

EcommerceSystem.java X

test.java

Cart.java

Order.java

EcommerceSystem.java > EcommerceSystem > main(String[])

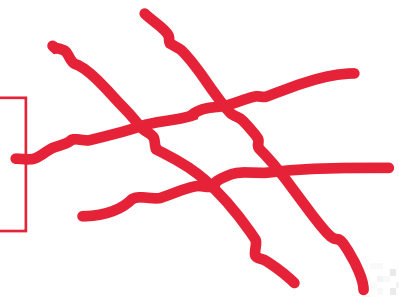
```
3 public class EcommerceSystem
5     public static void main(String[] args)
180         cart1.addProduct(p11);
181         break;
182     case 12:
183         cart1.addProduct(p12);
184         break;
185     default:
186         JOptionPane.showMessageDialog(parentComponent:null, message:"invalid option");
187         break;
188     }
189     int option4 = Integer.parseInt(JOptionPane.showInputDialog(message:"would you like to place the order ? 1- Yes 2- No "));
190     switch (option4)
191     {
192     case 1:
193     {
194         order1 = cart1.placOrder();
195         order1.printOrderInfo();
196         break;
197     }
198     case 2:
199         JOptionPane.showMessageDialog(parentComponent:null, message:"it was nice having you. GoodBye ^_^");
200         break;
201     default:
202         JOptionPane.showMessageDialog(parentComponent:null, message:"invalid option");
203         break;
204     }
205     break;
206 }
207 case 2:
208 {
209     JOptionPane.showMessageDialog(parentComponent:null, message:"it was nice having you. GoodBye ^_^");
210     break;
211 }
212 default:
213     JOptionPane.showMessageDialog(parentComponent:null, message:"invalld option");
214     break;
215 }
216 break;
```



```
215     }
216     break;
217 }
218 default:
219     JOptionPane.showMessageDialog(parentComponent:null, message:"invaild option");
220     break;
221 }
222
223 }#####
224 }
```



I wasn't finished with the GUI yet in the discussion



Input



Welcome to DigitMall ^_^

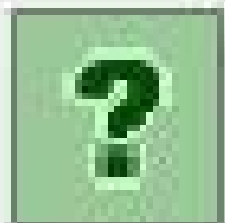
enter you id

23011586

OK

Cancel

Input



enter you name

Nada Mohamed

OK

Cancel

Input



enter you address

Alex

OK

Cancel

Input



how many products would you like to add to your cart?

OK

Cancel

Input



which product would you like to add ? 1- smartphone 2- Iron 3- vacuum cleaner 4- fridge 5- t-shirt 6- pants 7- chemise 8- jacket 9- OOP 10- harry potter & the chambre of secrets 11- Fire & Blood 12- rich dad poor dad

OK

Cancel

Input



which product would you like to add ? 1- smartphone 2- Iron 3- vacuum cleaner 4- fridge 5- t-shirt 6- pants 7- chemise 8- jacket 9- OOP 10- harry potter & the chambre of secrets 11- Fire & Blood 12- rich dad poor dad

4

OK

Cancel

Input



which product would you like to add ? 1- smartphone 2- Iron 3- vacuum cleaner 4- fridge 5- t-shirt 6- pants 7- chemise 8- jacket 9- OOP 10- harry potter & the chambre of secrets 11- Fire & Blood 12- rich dad poor dad

5

OK

Cancel

Input



the total price is \$2619.97

would you like to place the order ? 1- Yes 2- No

1

OK

Cancel



~~~~~  
**Here's your order's summary:**

**Order Id: 1**

**Customer-Id: 23011586**  
~~~~~

smartphone - \$599.99

fridge - \$1999.99

T-shirt - \$19.99
~~~~~

**Total price: \$2619.97**  
~~~~~

looking forward seeing you soon ^_^

