

- EcommerceSystem

```
5 package ecommercesystem;
6
7 /**
8  *
9  * @author Ziad
10 */
11
12
13
14 public class EcommerceSystem {
15
16     /**
17      * @param args the command line arguments
18      */
19     public static void main(String[] args) {
20
21
22         ElectronicProduct p1=new ElectronicProduct();
23         p1.setName( name:"smartphone");
24         p1.setBrand( brand:"samsung");
25         p1.setPrice( price:599.9f);
26         p1.setProductid( productid:1);
27         p1.setWarrantyPeriod( WarrantyPeriod:1);
28
29         ClothingProduct p2=new ClothingProduct();
30         p2.setName( name:"T-shirt");
31         p2.setFabric( fabric:"cotton");
32         p2.setPrice( price:19.99f);
33         p2.setProductid( productid:2);
34         p2.setSize( size:"medium");
35
36         BookProduct p3=new BookProduct();
37         p3.setName( name:"OOP");
38         p3.setAuthor( author:"O'Reilly");
39         p3.setProductid( productid:3);
40         p3.setPrice( price:39.99f);
41         p3.setPublisher( publisher:"X publication");
42
43
44         Customer c=new Customer();
45
46         c.setCart();
47         c.setO();
48
49
50         java.awt.EventQueue.invokeLater(new Runnable() {
51             public void run() {
52                 new Ecommerce__System(p1, p2, p3, c, cart:c.getCart(), o:c.getO()).setVisible(b:true);
53             }
54         });
55
56
57
58
59     }
60 }
61
```

- **Product class**

```
12 public class Product {
13     int productid;
14     String name;
15     float price;
16
17     public int getProductid() {
18         return productid;
19     }
20
21     public void setProductid(int productid) {
22         this.productid = Math.abs(a:productid);
23     }
24
25     public String getName() {
26         return name;
27     }
28
29     public void setName(String name) {
30         this.name = name;
31     }
32
33     public float getPrice() {
34         return price;
35     }
36
37     public void setPrice(float price) {
38         this.price = Math.abs(a:price);
39     }
40
41
42
43 }
```

- Electronic product class

```
5 package ecommerceSystem;
6 /**
7  *
8  * @author Ziad
9  */
10 public class ElectronicProduct extends Product {
11
12     String brand;
13     int WarrantyPeriod;
14
15     public String getBrand() {
16         return brand;
17     }
18
19     public void setBrand(String brand) {
20         this.brand = brand;
21     }
22
23     public int getWarrantyPeriod() {
24         return WarrantyPeriod;
25     }
26
27     public void setWarrantyPeriod(int WarrantyPeriod) {
28         this.WarrantyPeriod = Math.abs(a:WarrantyPeriod);
29     }
30
31     public int getProductid() {
32         return productid;
33     }
34
35     public void setProductid(int productid) {
36         this.productid = Math.abs(a:productid);
37     }
38     public String getName() {
39         return name;
40     }
41
42     public void setName(String name) {
43         this.name = name;
44     }
45
46     public float getPrice() {
47         return price;
48     }
49
50     public void setPrice(float price) {
51         this.price = Math.abs(a:price);
52     }
53
54
55
56
57
58 }
59
```

- Clothing product class

```
5 package ecommerceSystem;
6
7 /**
8  *
9  * @author Ziad
10 */
11 public class ClothingProduct extends Product {
12
13     String size, fabric;
14
15     public String getSize() {
16         return size;
17     }
18
19     public void setSize(String size) {
20         this.size = size;
21     }
22
23     public String getFabric() {
24         return fabric;
25     }
26
27     public void setFabric(String fabric) {
28         this.fabric = fabric;
29     }
30
31
32
33
34 }
35
```

- **Book product class**

```
5 package ecommerceSystem;
6
7 /**
8  *
9  * @author Ziad
10 */
11 public class BookProduct extends Product {
12
13     String author,publisher;
14
15     public String getAuthor() {
16         return author;
17     }
18
19     public void setAuthor(String author) {
20         this.author = author;
21     }
22
23     public String getPublisher() {
24         return publisher;
25     }
26
27     public void setPublisher(String publisher) {
28         this.publisher = publisher;
29     }
30
31
32
33
34
35
36 }
```

- Cart class

```
1  public class Cart {
12
13     int nProducts;
14     Product products[] ;
15     int productnum=0;
16
17
18
19
20
21     public int getnProducts() {
22         return nProducts;
23     }
24
25     public void setnProducts(int nProducts) {
26         this.nProducts = Math.abs(a:nProducts);
27         products=new Product[nProducts];
28     }
29
30     public void addProduct(Product p){
31
32         products[productnum]=p;
33
34         productnum++;
35     }
36
37     public void removeProduct(String productName) {
38         boolean found = false;
39         int indexToRemove = -1;
40
41         // Find the index of the product with the given name
42         for (int i = 0; i < productnum; i++) {
43             if (products[i].getName().equals(anObject:productName)) {
44                 found =true;
45                 indexToRemove = i;
46                 break;
47             }
48
49         if (found) {
50             for (int j = indexToRemove; j < productnum - 1; j++) {
51                 products[j] = products[j + 1];
52             }
53             // Set the last element to null to avoid memory leaks
54             products[productnum - 1] = null;
55             productnum--;
56             System.out.println("Product " + productName + " removed successfully.");
57         } else {
58             System.out.println("Product " + productName + " not found.");
59         }
60     }
61 }
62
63 }
```

- Customer class

```
11 public class Customer {
12
13     int customerid;
14     String name,address;
15     Cart cart ;
16     Order o;
17
18     public Order getO() {
19         return o;
20     }
21
22     public void setO() {
23         this.o = new Order();
24         o.orderid=1;
25     }
26
27
28
29
30
31     public int getCustomerid() {
32         return customerid;
33     }
34
35     public void setCustomerid(int customerid) {
36         this.customerid = Math.abs( a:customerid);
37     }
38
39     public String getName() {
40         return name;
41     }
42
43     public void setName(String name) {
44         this.name = name;
45     }
46
47     public String getAddress() {
48         return address;
49     }
50
51     public void setAddress(String address) {
52         this.address = address;
53     }
54
55     public Cart getCart() {
56         return cart;
57     }
58
59     public void setCart() {
60         this.cart =new Cart();
61     }
62
63
64
65 }
66
```

- Order class

```
7  import javax.swing.JTextArea;
8
9  /**
10   *
11   * @author Ziad
12   */
13  public class Order extends Cart {
14
15      int orderid;
16      float totalprice;
17
18      public int getOrderid() {
19          return orderid;
20      }
21
22      public void setOrderid(int orderid) {
23          this.orderid = orderid;
24      }
25
26
27      public Product[] getProducts() {
28          return products;
29      }
30
31      public void setProducts(Product[] products) {
32          this.products = products;
33      }
34
35      public int getProductnum() {
36          return productnum;
37      }
38
39      public void setProductnum(int productnum) {
40          this.productnum = productnum;
41      }
42
43      public float calctotalprice(Product arr[],int nProducts){
44
45          for(int i=0; i<nProducts; i++){
46              totalprice+=arr[i].getPrice();
47          }
48          return totalprice;
49      }
50
51
52
53      public float getTotalprice() {
54          return totalprice;
55      }
56
57      public void printOrderInfo(int customerid,int nProducts,Product arr[],JTextArea textArea ){
58          StringBuilder sb = new StringBuilder();
59          sb.append( str:"Here 's your order summary \n");
60          sb.append("Order Id: "+orderid+"\n Customer Id: "+customerid+"\n");
61          sb.append( str:"products: \n");
62          calctotalprice(arr,nProducts);
63
64          for(int i=0; i<nProducts; i++){
65              if(arr[i]!=null)
66                  sb.append( str:arr[i].getName()).append( str:" - ").append( f:arr[i].getPrice()).append( str:"\n");
67          }
68          sb.append("The total price is :"+getTotalprice());
69          textArea.setText( t:sb.toString());
70      }
71  }
```


- GUI class

```

12 import java.util.TimerTask;
13 import javax.swing.*;
14
15 public class Ecommerce__System extends JFrame {
16
17     // Declare GUI components
18     private JButton add_pro_btn;
19     private JComboBox<String> productComboBox;
20     private JButton place_ord_btn;
21     private JLabel jLabel1;
22     private JLabel jLabel2;
23     private JLabel jLabel3;
24     private JLabel jLabel4;
25     private JLabel jLabel5;
26     private JPanel jPanel1;
27     private JScrollPane jScrollPane1;
28     private JScrollPane jScrollPane2;
29     private JScrollPane jScrollPane3;
30     private JScrollPane jScrollPane4;
31     private javax.swing.JScrollPane jScrollPane5;
32     private JTextArea address;
33     private JTextArea id;
34     private JTextArea n_Products;
35     private JTextArea name;
36     private int addCounter = 0;
37     private javax.swing.JTextArea jTextArea1;
38
39     // Other variable declarations...
40     private ElectronicProduct p1;
41     private ClothingProduct p2;
42     private BookProduct p3;
43     private Customer c;
44     private Cart cart;
45     private Order o;
46
47     public Ecommerce__System(ElectronicProduct p1, ClothingProduct p2, BookProduct p3, Customer c, Cart cart, Order o) {
48         this.p1 = p1;
49         this.p2 = p2;
50         this.p3 = p3;
51         this.c = c;
52         this.cart = cart;
53         this.o = o;
54
55         initComponents(p1,p2,p3,c,card,o);
56
57         addListeners(p1,p2,p3,c,card,o);
58         setInitialValues();
59     }
60
61     private void initComponents(ElectronicProduct p1, ClothingProduct p2, BookProduct p3, Customer c, Cart cart, Order o) {
62
63         jPanel1 = new javax.swing.JPanel();
64         jLabel1 = new javax.swing.JLabel();
65         jLabel2 = new javax.swing.JLabel();
66         jLabel3 = new javax.swing.JLabel();
67         jLabel5 = new javax.swing.JLabel();
68         jLabel4 = new javax.swing.JLabel();
69         productComboBox = new javax.swing.JComboBox<>();
70         add_pro_btn = new javax.swing.JButton();
71         place_ord_btn = new javax.swing.JButton();
72         jScrollPane1 = new javax.swing.JScrollPane();
73         name = new javax.swing.JTextArea();
74         jScrollPane2 = new javax.swing.JScrollPane();
75         id = new javax.swing.JTextArea();
76         jScrollPane3 = new javax.swing.JScrollPane();

```

```

78     n_Products = new javax.swing.JTextArea();
79     jScrollPane4 = new javax.swing.JScrollPane();
80     adress = new javax.swing.JTextArea();
81     jScrollPane5 = new javax.swing.JScrollPane();
82     jTextArea1 = new javax.swing.JTextArea();
83
84     setDefaultCloseOperation( operation: javax.swing.WindowConstants.EXIT_ON_CLOSE);
85
86     jLabel1.setFont(new java.awt.Font( name: "Segoe UI", style:1, size:14)); // NOI18N
87     jLabel1.setText( text: "ID :");
88
89     jLabel2.setFont(new java.awt.Font( name: "Segoe UI", style:1, size:14)); // NOI18N
90     jLabel2.setText( text: "Name :");
91
92     jLabel3.setFont(new java.awt.Font( name: "Segoe UI", style:1, size:14)); // NOI18N
93     jLabel3.setText( text: "Adress :");
94
95     jLabel5.setText( text: " how many products would you like to add ?");
96
97     jLabel4.setText( text: "Select Product :");
98     productComboBox.addItem( item: p1.getName());
99     productComboBox.addItem( item: p2.getName());
100    productComboBox.addItem( item: p3.getName());
101
102
103    add_pro_btn.setText( text: "Add product");
104
105
106    place_ord_btn.setText( text: "place order");
107
108
109    name.setColumns( columns:20);
110    name.setRows( rows:1);
111
112    jScrollPane1.setViewportView( view: name);
113
114    id.setColumns( columns:20);
115    id.setRows( rows:1);
116    jScrollPane2.setViewportView( view: id);
117
118    n_Products.setColumns( columns:20);
119    n_Products.setRows( rows:1);
120    jScrollPane3.setViewportView( view: n_Products);
121
122    adress.setColumns( columns:20);
123    adress.setRows( rows:1);
124    jScrollPane4.setViewportView( view: adress);
125
126    javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout( host: jPanel1);
127    jPanel1.setLayout( mgr: jPanel1Layout);
128    jPanel1Layout.setHorizontalGroup(
129        group: jPanel1Layout.createParallelGroup( alignment: javax.swing.GroupLayout.Alignment.LEADING)
130        .addGroup( group: jPanel1Layout.createSequentialGroup()
131            .addComponent( component: jLabel5)
132            .addGroup( group: jPanel1Layout.createSequentialGroup()
133                .addComponent( component: jLabel1)
134                .addComponent( component: jLabel2)
135                .addComponent( component: jLabel3)
136                .addComponent( component: productComboBox)
137                .addComponent( component: jLabel4)
138                .addComponent( component: jScrollPane3)
139                .addComponent( component: jScrollPane4)
140                .addComponent( component: add_pro_btn)
141                .addComponent( component: place_ord_btn)
142            )
143        )
144        .addGroup( group: jPanel1Layout.createSequentialGroup()
145            .addComponent( component: jScrollPane1)
146            .addComponent( component: jScrollPane2)
147            .addComponent( component: jScrollPane5)
148        )
149    );
150    jPanel1Layout.setVerticalGroup(
151        group: jPanel1Layout.createParallelGroup( alignment: javax.swing.GroupLayout.Alignment.LEADING)
152        .addGroup( group: jPanel1Layout.createSequentialGroup()
153            .addComponent( component: jLabel5)
154            .addComponent( component: jScrollPane1)
155            .addComponent( component: jScrollPane2)
156            .addComponent( component: jScrollPane3)
157            .addComponent( component: jScrollPane4)
158            .addComponent( component: add_pro_btn)
159            .addComponent( component: place_ord_btn)
160        )
161    );
162    layout.setVerticalGroup(
163        group: jPanel1Layout.createParallelGroup( alignment: javax.swing.GroupLayout.Alignment.LEADING)
164        .addGroup( group: jPanel1Layout.createSequentialGroup()
165            .addComponent( component: jScrollPane1)
166            .addComponent( component: jScrollPane2)
167            .addComponent( component: jScrollPane3)
168            .addComponent( component: jScrollPane4)
169            .addComponent( component: add_pro_btn)
170            .addComponent( component: place_ord_btn)
171        )
172    );
173
174
175

```

```

176         .addContainerGap ( pref: 9, max: Short.MAX_VALUE )
177     );
178
179     jTextArea1.setColumns ( columns: 20 );
180     jTextArea1.setRows ( rows: 5 );
181     jScrollPane5.setViewportViewView ( view: jTextArea1 );
182
183     javax.swing.GroupLayout layout = new javax.swing.GroupLayout ( host.getContentPane () );
184     getContentPane ().setLayout ( mgr: layout );
185     layout.setHorizontalGroup (
186         group: layout.createParallelGroup ( alignment: javax.swing.GroupLayout.Alignment.LEADING )
187         .addComponent ( component: jPanell, min: javax.swing.GroupLayout.DEFAULT_SIZE, pref: javax.swing.GroupLayout.DEFAULT_SIZE, max: Short.MAX_VALUE )
188         .addGroup ( group: layout.createSequentialGroup ()
189             .addContainerGap ()
190             .addComponent ( component: jScrollPane5, min: javax.swing.GroupLayout.PREFERRED_SIZE, pref: 287, max: javax.swing.GroupLayout.PREFERRED_SIZE )
191             .addContainerGap ( pref: javax.swing.GroupLayout.DEFAULT_SIZE, max: Short.MAX_VALUE )
192         );
193     layout.setVerticalGroup (
194         group: layout.createParallelGroup ( alignment: javax.swing.GroupLayout.Alignment.LEADING )
195         .addGroup ( group: layout.createSequentialGroup ()
196             .addComponent ( component: jPanell, min: javax.swing.GroupLayout.PREFERRED_SIZE, pref: javax.swing.GroupLayout.DEFAULT_SIZE, max: javax.swing.GroupLayout.PREFERRED_SIZE )
197             .addPreferredGap ( type: javax.swing.LayoutStyle.ComponentPlacement.RELATED )
198             .addComponent ( component: jScrollPane5, min: javax.swing.GroupLayout.PREFERRED_SIZE, pref: 92, max: javax.swing.GroupLayout.PREFERRED_SIZE )
199             .addContainerGap ( pref: javax.swing.GroupLayout.DEFAULT_SIZE, max: Short.MAX_VALUE )
200         );
201
202     pack ();
203
204
205
206
207


```

```

209 private void addListeners ( ElectronicProduct p1, ClothingProduct p2, BookProduct p3, Customer c, Cart cart, Order o ) {
210
211     add_pro_btn.addActionListener ( new java.awt.event.ActionListener () {
212         public void actionPerformed ( java.awt.event.ActionEvent evt ) {
213             if ( addCounter == 0 ) {
214                 c.cart.setnProducts ( nProducts: Integer.parseInt ( s: n_Products.getText () ) );
215                 c.setCustomerId ( customerId: Integer.parseInt ( s: id.getText () ) );
216                 if ( addCounter < c.getCart ().getnProducts () ) {
217                     int choice = productComboBox.getSelectedIndex () + 1;
218                     if ( choice == 1 ) {
219                         c.cart.addProduct ( p: p1 );
220                     } else if ( choice == 2 ) {
221                         c.cart.addProduct ( p: p2 );
222                     } else if ( choice == 3 ) {
223                         c.cart.addProduct ( p: p3 );
224                     }
225                     addCounter++;
226
227                     // If the counter reaches the limit, disable the button
228                     if ( addCounter == c.getCart ().getnProducts () ) {
229                         add_pro_btn.setEnabled ( b: false );
230                     }
231                 }
232             }
233         }
234     });
235
236     place_ord_btn.addActionListener ( new java.awt.event.ActionListener () {
237         public void actionPerformed ( java.awt.event.ActionEvent evt ) {
238             c.o.printOrderInfo ( customerId: c.getCustomerId (), nProducts: c.cart.getnProducts (), arr: c.cart.products, textArea: jTextArea1 );
239
240             java.util.Timer timer = new java.util.Timer ();
241             timer.schedule ( new TimerTask () {
242                 @Override
243                 public void run () {
244                     System.exit ( status: 0 );
245                 }
246             }, delay: 10000 ); // 10000 millisec
247             //
248             System.exit ( 20 );
249         }
250     });
251
252     private void setInitialValues () {
253         id.setText ( t: String.valueOf ( i: c.getCustomerId () ) );
254         name.setText ( t: c.getName () );
255         adress.setText ( t: c.getAdress () );
256         n_Products.setText ( t: "0" ); // Set default value for n_Products
257     }
258
259 }

```

- Output

— □ ×


ID :

Name :

Adress :

how many products would you like to add ?

Select Product :



Here 's your order summary

Order Id: 1


Customer Id: 22010102


products:

smartphone - 599.9

T-shirt - 19.99

OOP - 39.99



— □ ×


ID :

Name :

Adress :

how many products would you like to add ?

Select Product :



smartphone

T-shirt

OOP

Customer Id: 22010102

products:

smartphone - 599.9

T-shirt - 19.99

OOP - 39.99

The total price is :659.88

