

Name : حبيبة محمد فتحي ابراهيم

ID: 23011065

The project about ECommerce System

### Class Product :

```
1
2 // class Product
3 public class Product { 16 usages 3 inheritors
4     protected int productId; 3 usages
5     protected String name; 4 usages
6     protected float price; 5 usages
7
8
9     public int getProductId() { no usages
10         return productId;
11     }
12
13     public void setProductId(int productId) { no usages
14         this.productId = productId;
15     }
16
17     public String getName() { 1 usage
18         return name;
19     }
20
21     public void setName(String name) { no usages
22         this.name = name;
23     }
24
```

```
25     public float getPrice() { 2 usages
26         return price;
27     }
28
29     public void setPrice(float price) { no usages
30         this.price = price;
31     }
32
33     public Product(){} 3 usages
34
35
36         public Product( String name, float price) { no usages
37             // this.productId = productId;
38             this.name = name;
39             this.price = price;
40         }
41
42     public Product(int productId, String name, float price) { 3 usages
43         this.productId = productId;
44         this.name = name;
45         this.price = price;
46     }
47 }
48 |
```

## Class Electronic product :

```
1
2 // class ElectronicProduct
3 public class ElectronicProduct extends Product { no usages
4     private String brand; 3 usages
5     private int warrantyPeriod; 3 usages
6
7
8     public ElectronicProduct(){ no usages
9
10
11     public ElectronicProduct( int productId,String name,float price,String brand, int warrantyPeriod){ no usages
12         super(productId, name, price);
13         this.brand = brand;
14         this.warrantyPeriod = warrantyPeriod;
15     }
16
17
18 >     public String getBrand(){ return brand; }
21
22 >     public void setBrand(String brand) { this.brand = brand; }
25
26 >     public int getWarrantyPeriod(){ return warrantyPeriod; }
29
30 >     public void setWarrantyPeriod(int warrantyPeriod) { this.warrantyPeriod = warrantyPeriod; }
33 }
34 |
```

## Class Clothing Product :

```
1  public class ClothingProduct extends Product { 2 usages
2      private String size; 3 usages
3      private String fabric; 3 usages
4
5
6      public ClothingProduct(){ no usages
7
8      public ClothingProduct(int productId, String size, float price, String fabric, String name) { 1 usage
9          super(productId, name, price);
10         this.size = size;
11         this.fabric = fabric;
12     }
13
14
15
16     public String getSize() { no usages
17         return size;
18     }
19
20     public void setSize(String size) { no usages
21         this.size = size;
22     }
23
24     public String getFabric() { no usages
25         return fabric;
26     }
27
28     public void setFabric(String fabric) { no usages
29         this.fabric = fabric;
30     }
31
32 }
33 |
```

## Class Book Product :

```
1  public class BookProduct extends Product{ no usages
2
3      private String author; 3 usages
4      private String publisher; 3 usages
5
6      public BookProduct(){ no usages
7
8      public BookProduct(int productId,String name , float price, String publisher, String author) { no usages
9          super(productId, name, price);
10         this.author = author;
11         this.publisher = publisher;
12     }
13
14
15
16     public String getAuthor() { no usages
17         return author;
18     }
19
20     public void setAuthor(String author) { no usages
21         this.author = author;
22     }
23
24     public String getPublisher() { no usages
25         return publisher;
26     }
27
28     public void setPublisher(String publisher) { no usages
29         this.publisher = publisher;
30     }
31
32 }
33
```

## Class Customer :

```
1 public class Customer { no usages
2     private int customerId; 3 usages
3     private String name; 3 usages
4     private String address; 3 usages
5
6     public Customer(){} no usages
7
8     public Customer(int customerId, String name, String address) { no usages
9         this.customerId = customerId;
10        this.name = name;
11        this.address = address;
12    }
13
14    public int getCustomerId() { no usages
15        return customerId;
16    }
17
18    public void setCustomerId(int customerId) { no usages
19        this.customerId = customerId;
20    }
21
22    public String getName() { no usages
23        return name;
24    }
25
26    public void setName(String name) { no usages
27        this.name = name;
28    }
29
30    public String getAddress() { no usages
31        return address;
32    }
33    public void setAddress(String address) { no usages
34        this.address = address;
35    } }
```

## Class Cart :

```
1  public class Cart { 2 usages
2
3      private int customerId; 4 usages
4      private int nProducts; 5 usages
5      private Product [] Products; 7 usages
6
7      public Cart(){ } no usages
8
9      public Cart(int customerId, int nProducts) { no usages
10         this.customerId = Math.abs(customerId);
11         this.nProducts = Math.abs(nProducts);
12         this.Products = new Product[Math.abs(nProducts)];
13     }
14
15     > public int getCustomerId() { return customerId; }
16
17
18     > public void setCustomerId(int customerId) { this.customerId = customerId; }
19
20
21
22     > public int getnProducts() { return nProducts; }
23
24
25
26     > public void setnProducts(int nProducts) { this.nProducts = nProducts; }
27
28
29
30     > public Product[] getProducts() { return Products; }
31
32
33
34     > public void setProducts(Product[] Products) { this.Products = Products; }
35
36
37
38
39
40     public void addProduct(Product product, int index){ no usages
41         if(index>=0 && index < nProducts){
42             Products[index]=product;
43         }
44     }
45 }
```

```
46
47     public void removeProduct(int index){ no usages
48         if(index>=0 && index < nProducts){
49             Products[index]= null;
50         }
51     }
52 }
53 public float calculatePrice(){ 1 usage
54     float totalPrice=0;
55     for(Product product : Products){
56         if(product != null){
57             totalPrice+=product.price;
58         }
59     }
60     return totalPrice;
61 }
62 > public Order placeOrder() { return new Order(customerId,Products,calculatePrice()); }
63
64
65
66     public void addProduct(String label1) { no usages
67
68     }
69 }
70 |
```



## Class Order:

```
1 public class Order { 2 usages
2
3     private int customerId; 4 usages
4     private int orderedId; 4 usages
5     private Product[] Products; 5 usages
6     private float totalPrice; 4 usages
7
8
9     public Order(int customerId1, Product[] Products1, float calculatePrice){} 1 usage
10
11     public int getCustomerId() { no usages
12         return customerId;
13     }
14
15     public void setCustomerId(int customerId) { no usages
16         this.customerId = customerId;
17     }
18
19     public int getOrderedId() { no usages
20         return orderedId;
21     }
22
23     public void setOrderedId(int orderedId) { no usages
24         this.orderedId = orderedId;
25     }
26
27     public Product[] getProducts() { no usages
28         return Products;
29     }
30
31     public void setProducts(Product[] Products) { no usages
32         this.Products = Products;
33     }
```

```

35     public float getTotalPrice() { no usages
36         return totalPrice;
37     }
38
39     public void setTotalPrice(float totalPrice) { no usages
40         this.totalPrice = totalPrice;
41     }
42     public Order(int customerId, int orderedId, Product[] Products) { no usages
43         this.customerId = Math.abs(customerId);
44         this.orderedId = Math.abs(orderedId);
45         this.Products = Products;
46         this.totalPrice = Math.abs(totalPrice);
47     }
48     private double calculateTotalPrice(){ 1 usage
49         float totalPrice=0;
50         for(Product product : Products){
51             if(product != null){
52                 totalPrice+=product.getPrice();
53             }
54         }
55         return totalPrice;
56     }
57     public void printOrderInfo(){ no usages
58         System.out.println("Here 's Your order 's summary :");
59         System.out.println("Order ID: "+ orderedId);
60         System.out.println("Customer ID: "+customerId);
61         System.out.println("Products: ");
62         for(Product product : Products){
63             if(product!=null){
64                 System.out.println(product.getName() +" - $" + product.getPrice() );
65             }
66         }
67         System.out.printf("Total Price: $ %.5f",calculateTotalPrice());
68     } }

```

## Classes about GUI

### Class login(Ecommerce System )

```
1  import javax.swing.*;
2  import java.awt.Color;
3  import java.awt.Font;
4  import java.awt.event.ActionEvent;
5  import java.awt.event.ActionListener;
6
7
8  public class EcommerceSystemGUI extends JFrame implements ActionListener { 4 usages
9
10     //panel
11
12     JPanel n = new JPanel(); 13 usages
13     JTextField t1=new JTextField(); 2 usages
14     JTextField t2=new JTextField(); 2 usages
15     JTextField t3=new JTextField(); 2 usages
16     JTextField t4=new JTextField(); 2 usages
17     JButton b1=new JButton( text: "Create"); 6 usages
18     JLabel l1=new JLabel( text: "Welcome to the E-Commerce System !"); 3 usages
19     JLabel l2=new JLabel( text: "ID:"); 3 usages
20     JLabel l3=new JLabel( text: "Name:"); 3 usages
21     JLabel l4=new JLabel( text: "Address:"); 3 usages
22     JLabel l5=new JLabel( text: "How many products you want to add to your cart ?"); 3 usages
23     public EcommerceSystemGUI() 2 usages
24     {
25
26         this.setTitle("EcommerceSystem");
27         this.setSize( width: 500, height: 600);
28         this.setVisible(true);
29         this.setResizable(false);
30         this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
31         this.setLocation( x: 400, y: 100);
32         n.setBackground(Color.pink);
33         this.add(n);
34         n.setLayout(null);
35         l1.setBounds( x: 80, y: 10, width: 400, height: 100);
```

```

36     l1.setFont(new Font( name: "Welcome to the E-Commerce System !",Font.PLAIN, size: 20));
37     n.add(l1);
38
39     l2.setBounds( x: 100, y: 100, width: 200, height: 30);
40     l2.setFont(new Font( name: "ID:",Font.PLAIN, size: 20));
41     n.add(l2);
42     t1.setBounds( x: 200, y: 100, width: 200, height: 30);
43     n.add(t1);
44
45     l3.setBounds( x: 100, y: 150, width: 200, height: 30);
46     l3.setFont(new Font( name: "Name:",Font.PLAIN, size: 20));
47     n.add(l3);
48     t2.setBounds( x: 200, y: 150, width: 200, height: 30);
49     n.add(t2);
50
51     l4.setBounds( x: 100, y: 200, width: 200, height: 30);
52     l4.setFont(new Font( name: "Address:",Font.PLAIN, size: 20));
53     n.add(l4);
54     t3.setBounds( x: 200, y: 200, width: 200, height: 30);
55     n.add(t3);
56
57     l5.setBounds( x: 15, y: 300, width: 450, height: 30);
58     l5.setFont(new Font( name: "How many products you want to add to your cart ?",Font.PLAIN, size: 20));
59     n.add(l5);
60     t4.setBounds( x: 130, y: 350, width: 200, height: 30);
61     n.add(t4);
62
63     b1.setBounds( x: 160, y: 400, width: 150, height: 50);
64     b1.setBackground(Color.lightGray);
65     b1.setFont(new Font( name: "Create",Font.PLAIN, size: 30));
66     //b1.setBorder();
67     n.add(b1);
68     b1.addActionListener( k: this);

```

```

69     }
70
71
72     @Override
73     public void actionPerformed(ActionEvent e) {
74         if(e.getSource()==b1){
75             SwingUtilities.invokeLater(()->new ProductList());
76             //ProductList p=new ProductList();
77
78             //p.setNp(Integer.parseInt(t4.getText()));
79             //Cart cart=new Cart(Integer.parseInt(t1.getText()),Integer.parseInt(t4.getText()));
80
81         }
82     }
83 }
84
85
86

```

## Class Product List:

```
1  import java.awt.*;
2  import java.awt.event.ActionEvent;
3  import java.awt.event.ActionListener;
4  import javax.swing.*;
5
6
7  public class ProductList extends JFrame implements ActionListener { 2 usages
8
9
10     private int ind=0; no usages
11     protected Cart cart; 1 usage
12     ProductList [] array; no usages
13     // Order order=new Order(cart.getCustomerId(), 1, cart.getProducts());
14     EcommerceSystemGUI e1=new EcommerceSystemGUI(); no usages
15     JPanel panel1 = new JPanel(); no usages
16     JPanel panel2 = new JPanel(); no usages
17
18     JLabel label1 = new JLabel( text: "SmartPhone"); 3 usages
19     JLabel label2 = new JLabel( text: "$599.99"); 3 usages
20
21     JButton b1 = new JButton( text: " Add "); 5 usages
22     JButton b2 = new JButton( text: " Remove "); 5 usages
23
24     JLabel label7=new JLabel(); 9 usages
25
26     JLabel label3 = new JLabel( text: "T-Shirt"); 3 usages
27     JLabel label4 = new JLabel( text: "$19.99"); 3 usages
28
29     JButton b3 = new JButton( text: " Add "); 5 usages
30     JButton b4 = new JButton( text: " Remove "); 5 usages
31
32     JLabel label8=new JLabel(); 9 usages
33
34     JLabel label5 = new JLabel( text: "00P"); 3 usages
35     JLabel label6 = new JLabel( text: "$39.99"); 3 usages
```

```

36
37 JButton b5 = new JButton( text: " Add "); 5 usages
38 JButton b6 = new JButton( text: " Remove "); 5 usages
39
40 JLabel label9=new JLabel(); 9 usages
41 JLabel label10=new JLabel(); 1 usage
42
43 JButton button=new JButton( text: "Place Order"); 6 usages
44
45 > public void setCart(Cart c) { this.cart=c; }
48 public ProductList() { 1 usage
49
50     this.setTitle("Products");
51     this.setSize( width: 400, height: 900);
52     this.setVisible(true);
53     this.setResizable(false);
54     this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
55     this.setLocation( x: 400, y: 50);
56
57     label7.setBackground(Color.PINK);
58     label7.setOpaque(true);
59     label7.setBounds( x: 10, y: 80, width: 150, height: 60);
60     label7.setLayout(null);
61
62     label8.setBackground(Color.PINK);
63     label8.setOpaque(true);
64     label8.setBounds( x: 480, y: 80, width: 150, height: 60);
65     label8.setLayout(null);
66
67     label9.setBackground(Color.PINK);
68     label9.setOpaque(true);
69     label9.setBounds( x: 1120, y: 80, width: 150, height: 60);
70     label9.setLayout(null);

```



```
71
72     label1.setBounds(x: 200, y: 20, width: 100, height: 80);
73     label1.setFont(new Font(name: "SmartPhone", Font.BOLD, size: 15));
74     label7.add(label1);
75
76     label2.setBounds(x: 40, y: 20, width: 100, height: 80);
77     label2.setFont(new Font(name: "$599.99", Font.BOLD, size: 15));
78     label7.add(label2);
79
80     b1.setBounds(x: 200, y: 100, width: 100, height: 50);
81     b1.setBackground(Color.LIGHT_GRAY);
82     b1.setFont(new Font(name: "Add", Font.BOLD, size: 15));
83     b1.setForeground(Color.WHITE);
84     label7.add(b1);
85
86     b2.setBounds(x: 30, y: 100, width: 130, height: 50);
87     b2.setBackground(Color.DARK_GRAY);
88     b2.setFont(new Font(name: "Remove", Font.BOLD, size: 15));
89     b2.setForeground(Color.WHITE);
90     label7.add(b2);
91     button.setBounds(x: 100, y: 80, width: 30, height: 50);
92     button.setFont(new Font(name: "Place Order", Font.BOLD, size: 15));
93     label10.add(button);
94
95
96     label3.setBounds(x: 220, y: 20, width: 100, height: 80);
97     label3.setFont(new Font(name: "T-Shirt", Font.BOLD, size: 15));
98     label8.add(label3);
99
100     label4.setBounds(x: 60, y: 20, width: 100, height: 80);
101     label4.setFont(new Font(name: "$19.99", Font.BOLD, size: 15));
102     label8.add(label4);
103
```

```

48 public ProductList() { //usage
104     b3.setBounds(x: 200, y: 100, width: 100, height: 50);
105     b3.setBackground(Color.LIGHT_GRAY);
106     b3.setFont(new Font(name: "Add", Font.BOLD, size: 15));
107     b3.setForeground(Color.WHITE);
108     label8.add(b3);
109
110     b4.setBounds(x: 30, y: 100, width: 130, height: 50);
111     b4.setBackground(Color.DARK_GRAY);
112     b4.setFont(new Font(name: "Remove", Font.BOLD, size: 15));
113     b4.setForeground(Color.WHITE);
114     label8.add(b4);
115
116
117
118
119     label5.setBounds(x: 220, y: 20, width: 100, height: 80);
120     label5.setFont(new Font(name: "OOP", Font.BOLD, size: 15));
121     label9.add(label5);
122
123     label6.setBounds(x: 60, y: 20, width: 100, height: 80);
124     label6.setFont(new Font(name: "$39.99", Font.BOLD, size: 15));
125     label9.add(label6);
126
127     b5.setBounds(x: 200, y: 100, width: 100, height: 50);
128     b5.setBackground(Color.LIGHT_GRAY);
129     b5.setFont(new Font(name: "Add", Font.BOLD, size: 15));
130     b5.setForeground(Color.WHITE);
131     label9.add(b5);
132
133     b6.setBounds(x: 30, y: 100, width: 130, height: 50);
134     b6.setBackground(Color.DARK_GRAY);
135     b6.setFont(new Font(name: "Remove", Font.BOLD, size: 15));
136     b6.setForeground(Color.WHITE);

```



```
137     label9.add(b6);
138
139
140     this.setLayout(new GridLayout( rows: 4, cols: 1, hgap: 10, vgap: 1));
141     this.add(label7);this.add(label8);this.add(label9);this.add(button);
142     //this.getLayout().addLayoutComponent("Button", button);
143
144     button.addActionListener( e->this);
145
146 }
147
148 @Override
149   public void actionPerformed(ActionEvent e) {
150
151     if(e.getSource()==button){
152         SwingUtilities.invokeLater(()->new OrderPlace());
153     }
154 }
155
156
157 }
158
```

## Class Order Place:

```
1  import javax.swing.*;
2  import java.awt.*;
3  import java.awt.event.ActionEvent;
4
5  public class OrderPlace extends JFrame { 1 usage
6
7      JPanel panel1=new JPanel(); 9 usages
8      JLabel label1=new JLabel( text: "Here 's your order 's summary"); 3 usages
9      JLabel label2=new JLabel( text: "Order ID : "); 3 usages
10     JLabel label3=new JLabel( text: "Customer ID : "); 3 usages
11     JLabel label4=new JLabel( text: "Products : "); 3 usages
12     JLabel label5=new JLabel( text: "Total Price : $ "); 3 usages
13     JButton button=new JButton( text: "OK"); 2 usages
14
15     public OrderPlace(){ 1 usage
16
17         this.setTitle(" Your Place Order");
18         this.setSize( width: 500, height: 600);
19         this.setVisible(true);
20         this.setResizable(false);
21         this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
22         this.setLocation( x: 400, y: 100);
23         panel1.setBackground(Color.WHITE);
24         this.add(panel1);
25         panel1.setLayout(null);
26
27
28         label1.setBounds( x: 100, y: 0, width: 500, height: 100);
29         label1.setFont(new Font( name: "Here 's your order 's summary",Font.BOLD, size: 20));
30         panel1.add(label1);
31
32         label2.setBounds( x: 80, y: 40, width: 100, height: 100);
33         label2.setFont(new Font( name: "Order ID : ",Font.ITALIC, size: 20));
34         panel1.add(label2);
35     }
```

```

35
36     label3.setBounds( x: 80, y: 80, width: 150, height: 100);
37     label3.setFont(new Font( name: "Customer ID : ",Font.ITALIC, size: 20));
38     panel1.add(label3);
39
40     label4.setBounds( x: 80, y: 120, width: 150, height: 100);
41     label4.setFont(new Font( name: "Products : ",Font.ITALIC, size: 20));
42     panel1.add(label4);
43
44
45     label5.setBounds( x: 80, y: 350, width: 150, height: 80);
46     label5.setFont(new Font( name: "Total Price : $ ",Font.ITALIC, size: 20));
47     panel1.add(label5);
48
49     button.setBounds( x: 200, y: 450, width: 80, height: 50);
50     panel1.add(button);
51 }
52 }
53

```

## The main class:

```

1
2 import java.util.Scanner;
3
4 public class ECommerceSystem {
5     public static void main(String[] args) {
6         //TIP Press <shortcut actionId="ShowIntentionActions"/> with your caret at the highlighted text
7         // to see how IntelliJ IDEA suggests fixing it.
8
9         Scanner in=new Scanner(System.in);
10        System.out.println("Welcome to the E-Commerce System!");
11        System.out.println("Please enter your id : ");
12        int customerId=in.nextInt();
13        in.nextLine();
14        System.out.println("Please enter your name : ");
15        String name=in.nextLine();
16        System.out.println("Please enter your address : ");
17        String address=in.nextLine();
18
19        Customer c1=new Customer(customerId,name,address);
20
21        ElectronicProduct smartphone =new ElectronicProduct( productId: 1, name: "smartphone", price: 599.99f, brand: "Samsung", warrantyPeriod: 1);
22        ClothingProduct TShirt=new ClothingProduct( productId: 2, size: "Medium", price: 19.99f, fabric: "Cotton", name: "T-shirt");
23        BookProduct OOP= new BookProduct( productId: 3, name: "OOP", price: 39.99f, publisher: "O'Reilly", author: "X publication");
24
25
26        System.out.println("How many products do you want to add to your cart?");
27        int nProducts=in.nextInt();
28        Cart cart =new Cart(customerId,nProducts);
29

```

```

29
30     for(int i=0;i<Math.abs(nProducts);i++){
31         System.out.println("Which product would you like to add? 1- smartphone 2- T-shirt 3- OOP");
32         int productchoice=in.nextInt();
33         switch(productchoice){
34             case 1:
35                 cart.addProduct(smartphone, i);
36                 break;
37             case 2:
38
39                 cart.addProduct(TShirt, i);
40                 break;
41
42             case 3:
43                 cart.addProduct(OOP, i);
44                 break;
45             default:
46                 System.out.println("Invalid choice. Please try again.");
47                 i--;
48         }
49     }
50     System.out.println("Your total is $" + cart.calculatePrice() + " Would you like to place the order? 1-Yes 2- No");
51     int c=in.nextInt();
52     if(c==1){
53         Order order=new Order(customerId, orderedid: 1, cart.getProducts());
54         order.printOrderInfo();
55     }else{
56         System.out.println("Order not placed.");
57     }
58 }

```

```

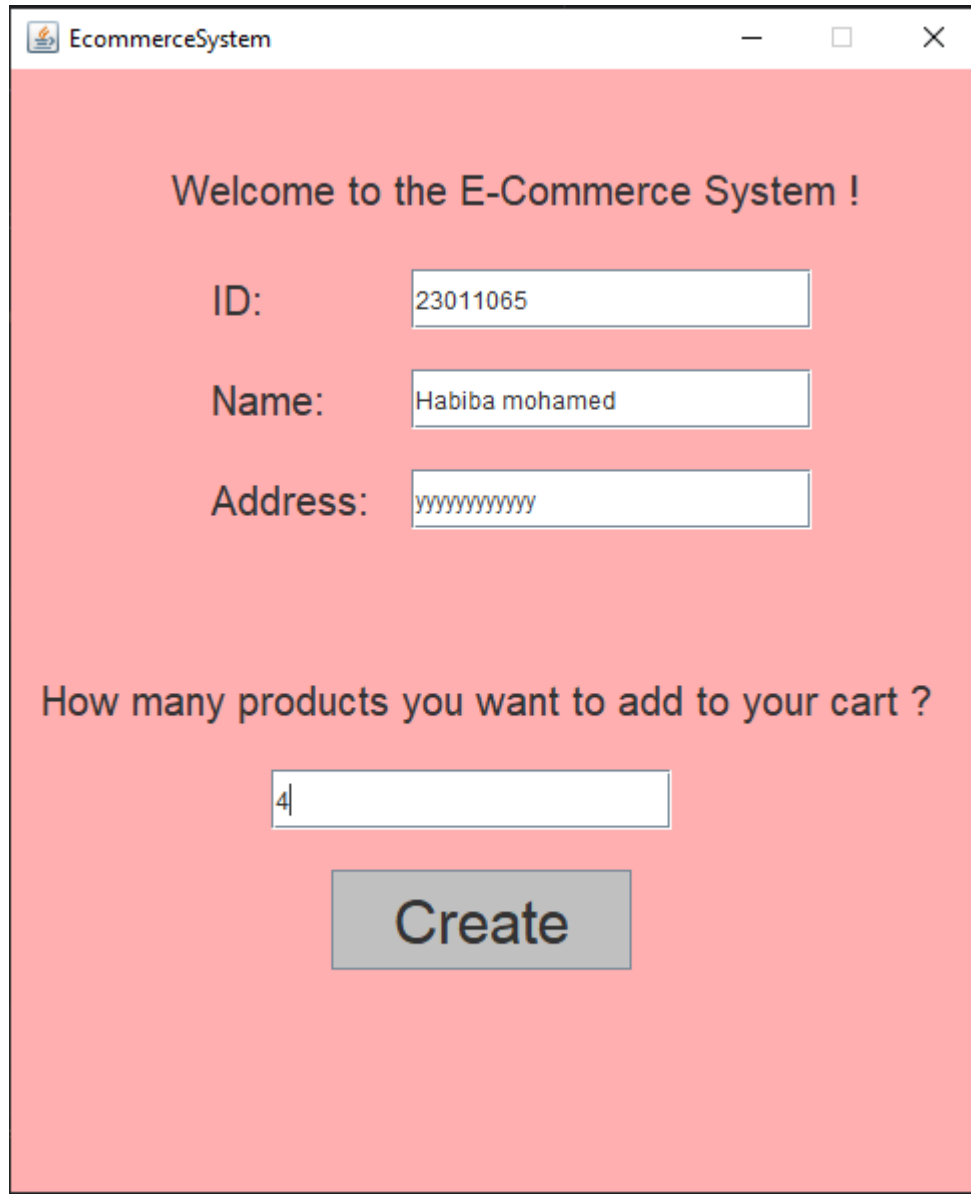
59
60 //Run GUI
61
62 //EcommerceSystemGUI E1=new EcommerceSystemGUI();
63 // ProductList p1=new ProductList();
64 // OrderPlace o1=new OrderPlace();
65
66
67
68 }
69 }

```

## The out put:

```
Run ECommerceSystem x
"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.1\lib\idea_rt.jar=54068:C:\Program Files\JetBrains\IntelliJ IDEA Co
Welcome to the E-Commerce System!
Please enter your id :
23011065
Please enter your name :
Habiba Mahamed
Please enter your address :
kkkkkkkkkkk
How many products do you want to add to your cart?
4
Which product would you like to add? 1- smartphone 2- T-shirt 3- OOP
1
Which product would you like to add? 1- smartphone 2- T-shirt 3- OOP
2
Which product would you like to add? 1- smartphone 2- T-shirt 3- OOP
3
Which product would you like to add? 1- smartphone 2- T-shirt 3- OOP
2
Your total is $679.95096 Would you like to place the order? 1-Yes 2- No
1
Here 's Your order 's summary :
Order ID: 1
Customer ID: 23011065
Products:
smartphone - $599.99
T-shirt - $19.99
OOP - $39.99
T-shirt - $19.99
Total Price: $ 679.95096
Process finished with exit code 0
|
```

## The out put of GUI:



The screenshot shows a window titled "EcommerceSystem" with a light red background. The window contains the following elements:

- A welcome message: "Welcome to the E-Commerce System !"
- Three input fields for user registration:
  - ID: 23011065
  - Name: Habiba mohamed
  - Address: yyyyyyyyyyy
- A question: "How many products you want to add to your cart ?"
- A single-digit input field containing the number "4".
- A "Create" button.



Products



\$599.99

SmartPhone

Remove

Add

\$19.99

T-Shirt

Remove

Add

\$39.99

OOP

Remove

Add

Place Order



Your Place Order



## Here 's your order 's summary

*Order ID :*

*Customer ID :*

*Products :*

*Total Price : \$*

OK