

Programming Project

Nour El-Din Amgad Abdel Fattah Ahmed Mohamed Alwa
ID: 23011596

April 2024

```
● ● ●
1 package EcommerceClasses;
2
3 public class Product {
4     private int ProductId;
5     private String name;
6     private float price;
7
8     Product(int ProductId, String name, float price){
9         this.ProductId = Math.abs(ProductId);
10        this.name = name;
11        this.price = Math.abs(price);
12    }
13
14    Product(){
15        ProductId = 0;
16        name = null;
17        price = 0;
18    }
19
20    public void setProductId(int ProductId){
21        this.ProductId = ProductId;
22    }
23
24    public int getProductId(){
25        return ProductId;
26    }
27
28    public void setName(String name){
29        this.name = name;
30    }
31
32    public String getName(){
33        return name;
34    }
35
36    public void setPrice(float price){
37        this.price = price;
38    }
39
40    public float getPrice(){
41        return price;
42    }
43
44 }
45
```

```
1 package EcommerceClasses;
2 public class ElectronicProduct extends Product{
3     private String brand;
4     private int warrantyPeriod;
5
6
7     public ElectronicProduct(int productId, String name, float price, String brand, int warrantyPeriod){
8         super(productId, name, price);
9         this.brand = brand;
10        this.warrantyPeriod = Math.abs(warrantyPeriod);
11    }
12
13    ElectronicProduct(){
14        brand = null;
15        warrantyPeriod = 0;
16    }
17
18    public void setBrand(String brand){
19        this.brand = brand;
20    }
21
22    public String getBrand(){
23        return brand;
24    }
25
26    public void setWarrantyPeriod(int warrantyPeriod){
27        this.warrantyPeriod = warrantyPeriod;
28    }
29
30    public int getWarrantyPeriod(){
31        return warrantyPeriod;
32    }
33
34 }
35
```

2: ElectronicProduct

```
● ● ●
1 package EcommerceClasses;
2 public class ClothingProduct extends Product{
3     private String size;
4     private String fabric;
5
6     public ClothingProduct(int productId, String name, float price, String size, String fabric){
7         super(productId, name, price);
8         this.size = size;
9         this.fabric = fabric;
10    }
11
12    public ClothingProduct(){
13        size = null;
14        fabric = null;
15    }
16
17    public void setSize(String size){
18        this.size = size;
19    }
20
21    public String getSize(){
22        return size;
23    }
24
25    public void setFabric(String fabric){
26        this.fabric = fabric;
27    }
28
29    public String getFabric(){
30        return fabric;
31    }
32 }
33
```

3: ClothingProduct

```
● ● ●
1 package EcommerceClasses;
2 public class BookProduct extends Product{
3     private String author;
4     private String publisher;
5
6     public BookProduct(int productId, String name, float price, String author, String publisher){
7         super(productId, name, price);
8         this.author = author;
9         this.publisher = publisher;
10    }
11
12    public BookProduct(){
13        author = null;
14        publisher = null;
15    }
16
17    public void setAuthor(String author){
18        this.author = author;
19    }
20
21    public String getAuthor(){
22        return author;
23    }
24
25    public void setPublisher(String publisher){
26        this.publisher = publisher;
27    }
28
29    public String getPublisher(){
30        return publisher;
31    }
32
33 }
34
```

4: BookProduct

```
● ● ●
1 package EcommerceClasses;
2 public class Customer {
3     private int customerId;
4     private String name;
5     private String address;
6
7     public Customer(int customerId, String name, String address){
8         this.customerId = Math.abs(customerId);
9         this.name = name;
10        this.address = address;
11    }
12
13    public Customer(){
14        customerId = 0;
15        name = null;
16        address = null;
17    }
18
19    public void setCustomerId(int customerId){
20        this.customerId = customerId;
21    }
22
23    public int getCustomerId(){
24        return customerId;
25    }
26
27    public void setName(String name){
28        this.name = name;
29    }
30
31    public String getName(){
32        return name;
33    }
34
35    public void setAddress(String address){
36        this.address = address;
37    }
38
39    public String getAddress(){
40        return address;
41    }
42
43
44 }
45
```

5: Customer

```
1 package EcommerceClasses;
2
3 public class Cart {
4     private int customerId;
5     private int nProducts;
6     private Product[] products;
7     private int i = 0;
8
9     public Cart(int customerId, int nProducts){
10         this.customerId = customerId;
11         this.nProducts = nProducts;
12         this.products = new Product[nProducts];
13     }
14
15
16     public Cart(){
17         this.customerId = 1;
18         this.nProducts = 1;
19         this.products = new Product[1];
20     }
21
22
23     public void setnProducts(int nProducts){
24         this.nProducts = nProducts;
25     }
26
27
28     public int getnProducts(){
29         return nProducts;
30     }
31
32
33     public void setCustomerId(int customerId){
34         this.customerId = customerId;
35     }
36
37     public int getCustomerId(){
38         return customerId;
39     }
40
41     public void addProduct(Product p){
42         products[i++] = p;
43     }
44
45     public float getTotalPrice(){
46         return this.calculatePrice();
47     }
48
49     public float calculatePrice() {
50         float totalPrice = 0;
51         for (int i = 0; i < nProducts; i++) {
52             totalPrice += products[i].getPrice();
53         }
54         return totalPrice;
55     }
56
57     public void removeProduct(){
58         products[i++] = null;
59     }
60
61     public Order placeOrder() {    7
62         return new Order(this.customerId, this.products, this.calculatePrice());
63     }
64 }
65
```

```
● ● ●
1 package EcommerceClasses;
2 public class Order {
3     private int customerId;
4     private int orderId;
5     private Product[] products;
6     private float totalPrice;
7     private String str = "";
8
9     public Order(int customerId, Product[] products, float totalPrice) {
10         this.customerId = Math.abs(customerId);
11         this.orderId = 1;
12         this.products = products;
13         this.totalPrice = Math.abs(totalPrice);
14     }
15
16     public int getOrderId(){
17         return this.orderId;
18     }
19
20     public String getStr(){
21         for (Product product : products) {
22             if (product != null) {
23                 String tmp = product.getName() + " - $" + product.getPrice() + '\n';
24                 str = str +tmp;
25             }
26         }
27
28         return str;
29     }
30
31     public int getCustomerId(){
32         return this.customerId;
33     }
34
35     public float getTotalPrice(){
36         return this.totalPrice;
37     }
38
39     public void printOrderInfo() {
40         System.out.println("Order ID: " + orderId);
41         System.out.println("Customer ID: " + customerId);
42         System.out.println("Product:");
43         for (Product product : products) {
44             if (product != null) {
45                 String tmp = product.getName() + " - $" + product.getPrice() + '\n';
46                 System.out.printf("%s - %.3f\n", product.getName(), product.getPrice());
47                 str = str +tmp;
48             }
49         }
50         System.out.printf("Total Price: %.3f\n", totalPrice);
51     }
52 }
53
```

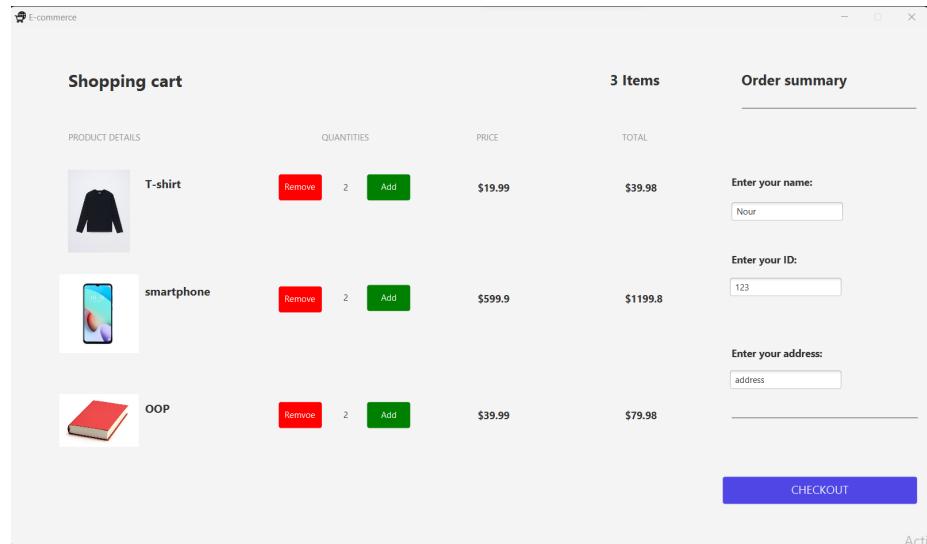
7: Order

```
1 import java.util.Scanner;
2 import EcommerceClasses.*;
3
4
5
6 public class ecommerce {
7
8     public static void main(String[] args) throws Exception {
9         Scanner input = new Scanner(System.in);
10
11     ElectronicProduct smartphone = new ElectronicProduct(1, "smartphone", 599.9f, "Samsung", 12);
12     ClothingProduct tshirt = new ClothingProduct(2, "t-shirt", 19.99f, "Medium", "Cotton");
13     BookProduct book = new BookProduct(3, "OOP", 39.99f, "O'Reilly", "X Publications");
14
15     String name, address;
16
17     System.out.print("Please enter your ID: \n");
18     int id = input.nextInt();
19
20     System.out.print("Please enter your name: \n");
21     name = input.next();
22
23     System.out.print("Please enter your address: \n");
24     address = input.nextLine();
25
26     Customer customer = new Customer(id, name, address);
27
28
29     System.out.println("How many products you want to add to your cart ? ");
30     int n = input.nextInt();
31
32
33     Cart cart = new Cart(customer.getCustomerId(), n);
34
35     while (n > 0) {
36         int chose = 0;
37         System.out.printf("Which product would you like to add? 1- %s 2- %s 3- %s\n", smartphone.getName(), tshirt.getName(), book.getName());
38         chose = input.nextInt();
39         switch (chose) {
40             case 1:
41                 cart.addProduct(smartphone);
42                 n--;
43                 break;
44
45             case 2:
46                 cart.addProduct(tshirt);
47                 n--;
48                 break;
49
50             case 3:
51                 cart.addProduct(book);
52                 n--;
53                 break;
54             default:
55                 System.out.println("Invalid chose");
56                 break;
57         }
58     }
59     System.out.println("Your total is "+cart.getTotalPrice() +". Would you like to place the order? 1- yes 2- No");
60     int place = input.nextInt();
61     if(place == 1){
62         System.out.println("Here's your order's summary:");
63         Order order = cart.placeOrder();
64         order.printOrderInfo();
65     }
66
67     input.close();
68 }
69 }
70 }
```

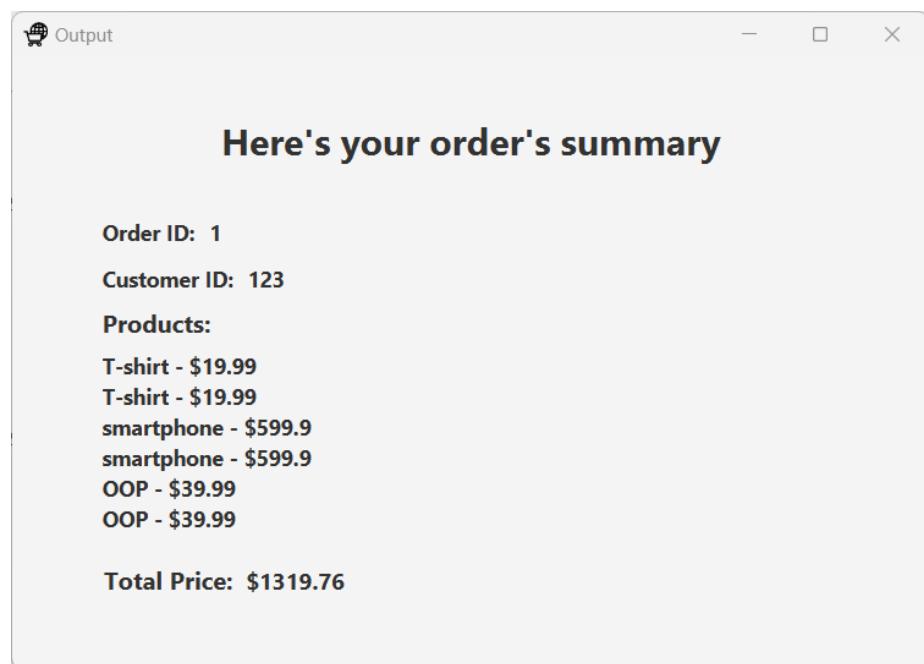
8: EcommerceSystem

```
Please enter your ID:  
23011596  
Please enter your name:  
Nour  
Please enter your address:  
address  
How many products you want to add to your cart ?  
4  
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  
Which product would you like to add? 1- smartphone 2- T-shirt 3- OOP  
1  
Your total is $679.87. Would you like to place the order? 1- yes 2- No  
1  
Here's your order's summary:  
Order ID: 1  
Customer ID: 23011596  
Product:  
T-shirt - $19.990  
OOP - $39.990  
T-shirt - $19.990  
smartphone - $599.900  
Total Price: $679.870
```

9: OutPut



10: GUI



11: result

```
 1 import java.io.IOException;
 2 import javafx.application.Application;
 3 import javafx.fxml.FXMLLoader;
 4 import javafx.scene.Parent;
 5 import javafx.scene.Scene;
 6 import javafx.scene.image.Image;
 7 import javafx.stage.Stage;
 8
 9 public class ecommerceGUI extends Application {
10     public static void main(String[] args) {
11         launch(args);
12     }
13
14     @Override
15     public void start(Stage stage) {
16         try {
17             Parent root = FXMLLoader.load(getClass().getResource("fx.fxml"));
18             Scene scene = new Scene(root);
19             scene.getStylesheets().add(getClass().getResource("imgs-style/style.css").toExternalForm());
20             stage.setScene(scene);
21             stage.setTitle("E-commerce");
22             stage.getIcons().add(new Image("imgs-style/57589.png"));
23             stage.setResizable(false);
24             stage.show();
25         } catch (IOException e) {
26             e.printStackTrace();
27         }
28     }
29 }
30 }
```

12: Main JavaFx class

```

1  import io.IoEException;
2  import eCommerceClasses.*;
3  import javafx.event.ActionEvent;
4  import javafx.fxml.FXMLLoader;
5  import javafx.fxml.FXMLLoader;
6  import javafx.scene.control.Button;
7  import javafx.scene.control.Label;
8  import javafx.scene.control.TextField;
9  import javafx.scene.image.Image;
10 import javafx.scene.Scene;
11 import javafx.scene.Stage;
12 import javafx.stage.Stage;
13
14 public class MainFx {
15     private Stage stage;
16     private Parent root;
17
18     @FXML
19     private Label count, count2,
20     count3, name1, name2, name3;
21     private Label price1, price2, price3, total1,
22     total2, total3;
23
24     @FXML
25     private Button dec, dec2, dec3, checkOut;
26
27     @FXML
28     private TextField name, id, address;
29
30     ElectronicProduct smartphone = new ElectronicProduct(1, "smartphone", 599.99, "Samsung", 12);
31     ClothingProduct tshirt = new ClothingProduct(2, "t-shirt", 19.99f, "Medium", "Cotton");
32     BookProduct book = new BookProduct(3, "OOP", 39.99f, "O'Reilly", "X Publications");
33
34
35     public void initialize() {
36         name1.setText(book.getName());
37         name2.setText(smartphone.getName());
38         name3.setText(tshirt.getName());
39
40         price1.setText("$" + Float.toString(tshirt.getPrice()));
41         price2.setText("$" + Float.toString(smartphone.getPrice()));
42         price3.setText("$" + Float.toString(book.getPrice()));
43
44         dec.setDisable(true);
45         dec2.setDisable(true);
46         dec3.setDisable(true);
47         checkOut.setDisable(true);
48
49     }
50
51     private void checkCheckoutButton() {
52
53         if (name.getText().isEmpty() && id.getText().isEmpty() && address.getText().isEmpty()) {
54             checkOut.setDisable(true);
55         } else {
56             checkOut.setDisable(false);
57         }
58
59     }
60
61     private void updateLabel(Label label, int delta) {
62         int num = Integer.parseInt(label.getText());
63         label.setText(Integer.toString(num + delta));
64
65     }
66     private void updateTotalPrice(Label price, Label totalPrice, Label count) {
67         float total = Float.parseFloat(price.getText().replace("$", "")) * Integer.parseInt(count.getText());
68         totalPrice.setText("$" + float.toString(total));
69
70     }
71
72     private void increment(Button button, Label count) {
73         if (Integer.parseInt(button.getText()) > 0)
74             button.setDisable(false);
75         else
76             button.setDisable(true);
77
78     }
79
80     private void cartFilling(Label count, Label productName, Cart cart) {
81         for (int i = 0; i < Integer.parseInt(count.getText()); i++) {
82             if (productName.getText() == smartphone.getName()) {
83                 cart.addProduct(smartphone);
84             }
85             else if (productName.getText() == tshirt.getName()) {
86                 cart.addProduct(tshirt);
87             }
88             else
89                 cart.addProduct(book);
90         }
91
92     }
93
94     public void increment(ActionEvent e) {
95         updateLabel(count, 1);
96         updateTotalPrice(price1, total1, count);
97         btnDec1.setDisable(true);
98
99     }
100
101     public void decrement(ActionEvent e) {
102         updateLabel(count, -1);
103         btnDec1.setDisable(true);
104         updateTotalPrice(price1, total1, count);
105
106     }
107
108     public void increment2(ActionEvent e) {
109         updateLabel(count2, 1);
110         updateTotalPrice(price2, total2, count2);
111         btnDec2.setDisable(true);
112
113     }
114
115     public void decrement2(ActionEvent e) {
116         updateLabel(count2, -1);
117         btnDec2.setDisable(true);
118         updateTotalPrice(price2, total2, count2);
119
120     }
121
122     public void increment3(ActionEvent e) {
123         updateLabel(count3, 1);
124         updateTotalPrice(price3, total3, count3);
125         btnDec3.setDisable(true);
126
127     }
128
129     public void decrement3(ActionEvent e) {
130         updateLabel(count3, -1);
131         btnDec3.setDisable(true);
132         updateTotalPrice(price3, total3, count3);
133
134     }
135
136     public void numericId(ActionEvent e) {
137         checkCheckoutButton();
138
139     }
140
141     public void idField(ActionEvent e) {
142         checkCheckoutButton();
143
144     }
145
146     public void addressField(ActionEvent e) {
147         checkCheckoutButton();
148
149     }
150
151     public void checkout(ActionEvent e) throws IOException {
152
153         Customer customer = new Customer(Integer.parseInt(id.getText()), name.getText(), address.getText());
154         int size = Integer.parseInt(count.getText()) + Integer.parseInt(count2.getText()) + Integer.parseInt(count3.getText());
155
156         Cart cart = new Cart(customer.getCustomerId(), size);
157
158         cartFilling(count, name1, cart);
159         cartFilling(count2, name2, cart);
160         cartFilling(count3, name3, cart);
161
162         Order order = cart.placeOrder();
163
164         FXMLLoader loader = new FXMLLoader(getClass().getResource("outputFx.fxml"));
165         root = loader.load();
166
167         outputScene output = loader.getController();
168
169         output.displayOutputOrder();
170
171         stage = new Stage();
172         stage.setScene(new Scene(root));
173         stage.setTitle("Output");
174         stage.getIcons().add(new Image("imgs-style/57589.png"));
175
176         stage.show();
177
178     }
179
180 }
181
182

```

```
1 import EcommerceClasses.Order;
2 import javafx.fxml.FXML;
3 import javafx.scene.control.Label;
4
5 public class outputScene {
6     @FXML
7     private Label orderID, customerID, products, total;
8
9     public void displayOutput(Order order){
10         orderID.setText(Integer.toString(order.getOrderId()));
11         customerID.setText(Integer.toString(order.getCustomerId()));
12         products.setText(order.getStr());
13         total.setText("$"+Float.toString(order.getTotalPrice()));
14     }
15 }
16
```

14: output Scene Controller