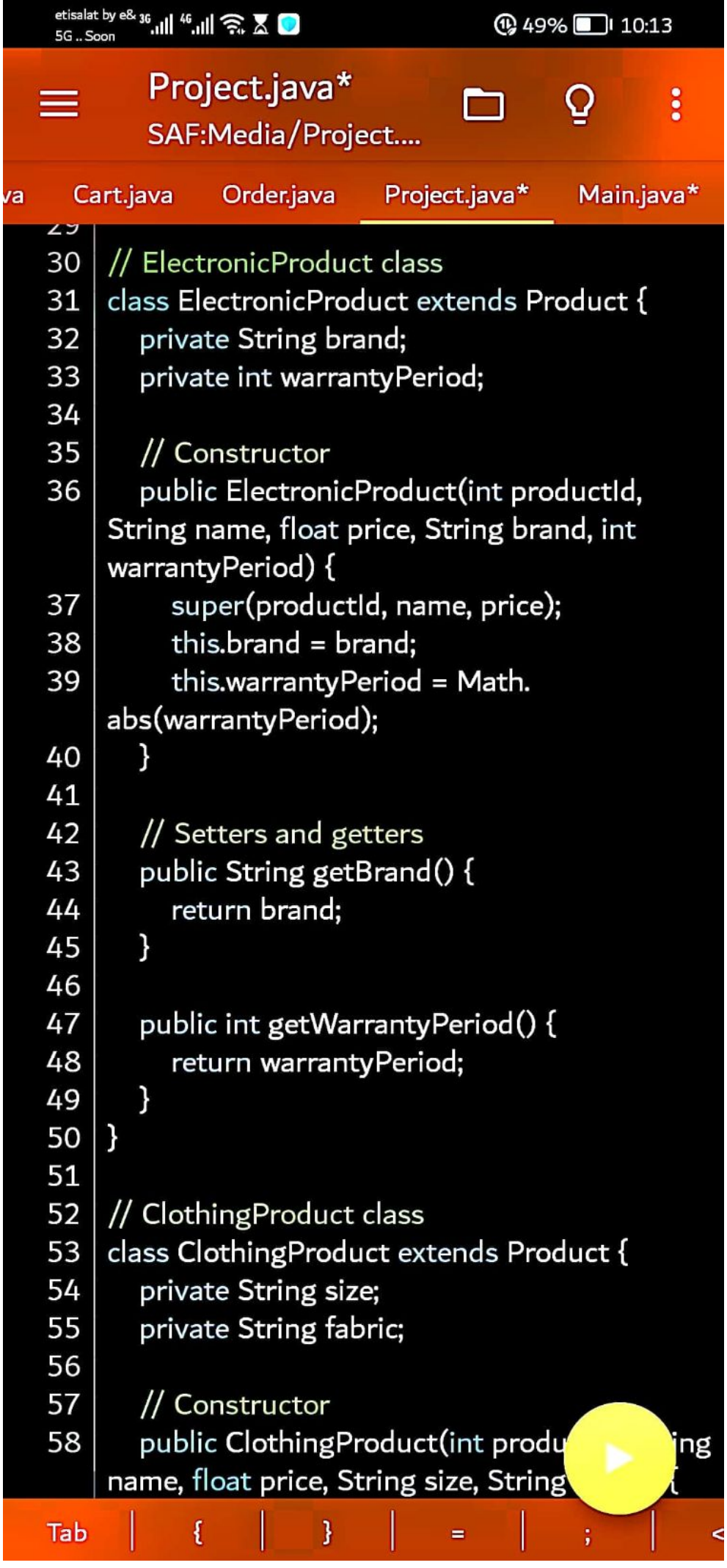
Project.java*
SAF:Media/Project....

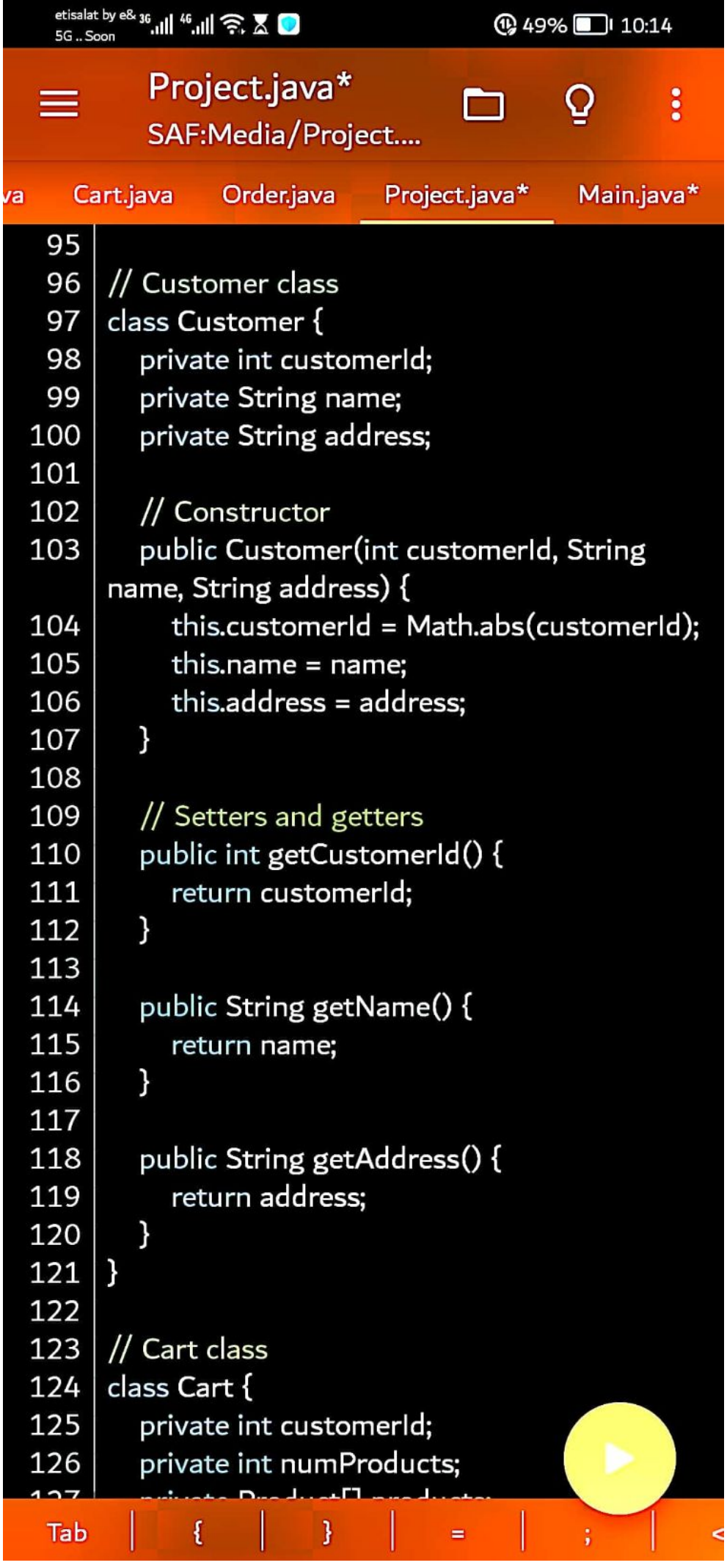
va | Cart.java | Order.java | Project.java* | Main.java*

```
3  // Product class
4  class Product {
5      protected int productId;
6      protected String name;
7      protected float price;
8
9      // Constructor
10     public Product(int productId, String name,
   float price) {
11         this.productId = Math.abs(productId);
12         this.name = name;
13         this.price = Math.abs(price);
14     }
15
16     // Setters and getters
17     public int getProductId() {
18         return productId;
19     }
20
21     public String getName() {
22         return name;
23     }
24
25     public float getPrice(){
26         return price;
27     }
28 }
29
30 // ElectronicProduct class
31 class ElectronicProduct extends Product {
32     private String brand;
33     private int warrantyPeriod;
34
```

Tab | { | } | = | ; | <


etisalat by e& 5G .. Soon
49% 10:13
Project.java*
SAF:Media/Project....
va
Cart.java
Order.java
Project.java*
Main.java*
// ElectronicProduct class
class ElectronicProduct extends Product {
private String brand;
private int warrantyPeriod;
// Constructor
public ElectronicProduct(int productId, String name, float price, String brand, int warrantyPeriod) {
super(productId, name, price);
this.brand = brand;
this.warrantyPeriod = Math.abs(warrantyPeriod);
}
// Setters and getters
public String getBrand() {
return brand;
}
public int getWarrantyPeriod() {
return warrantyPeriod;
}
}
// ClothingProduct class
class ClothingProduct extends Product {
private String size;
private String fabric;
// Constructor
public ClothingProduct(int produ ing name, float price, String size, String
Tab
{
}
=
;

```
52  // ClothingProduct class
53  class ClothingProduct extends Product {
54      private String size;
55      private String fabric;
56
57      // Constructor
58      public ClothingProduct(int productId, String
    name, float price, String size, String fabric) {
59          super(productId, name, price);
60          this.size = size;
61          this.fabric = fabric;
62      }
63
64      // Setters and getters
65      public String getSize() {
66          return size;
67      }
68
69      public String getFabric() {
70          return fabric;
71      }
72  }
73
74  // BookProduct class
75  class BookProduct extends Product {
76      private String author;
77      private String publisher;
78
79      // Constructor
80      public BookProduct(int productId, String
    name, float price, String author, String
    publisher) {
81          super(productId, name, price);
82          this.author = author;
83          this.publisher = publisher;
84      }
85
86      // Setters and getters
87      public String getAuthor() {
88          return author;
89      }
90
91      public String getPublisher() {
92          return publisher;
93      }
94  }
```


etisalat by e&
5G .. Soon
49%
10:14
Project.java*
SAF:Media/Project....
va
Cart.java
Order.java
Project.java*
Main.java*
95
96 // Customer class
97 class Customer {
98 private int customerId;
99 private String name;
100 private String address;
101
102 // Constructor
103 public Customer(int customerId, String name, String address) {
104 this.customerId = Math.abs(customerId);
105 this.name = name;
106 this.address = address;
107 }
108
109 // Setters and getters
110 public int getCustomerId() {
111 return customerId;
112 }
113
114 public String getName() {
115 return name;
116 }
117
118 public String getAddress() {
119 return address;
120 }
121 }
122
123 // Cart class
124 class Cart {
125 private int customerId;
126 private int numProducts;
Tab
{
}
=
;

```
123 // Cart class
124 class Cart {
125     private int customerId;
126     private int numProducts;
127     private Product[] products;
128 
129     // Constructor
130     public Cart(int customerId, int
    numProducts) {
131         this.customerId = Math.abs(customerId);
132         this.numProducts = Math.
    abs(numProducts);
133         this.products = new
    Product[numProducts];
134     }
135 
136     // Setters and getters
137     public int getCustomerId() {
138         return customerId;
139     }
140 
141     public int getNumProducts() {
142         return numProducts;
143     }
144 
145     public Product[] getProducts() {
146         return products;
147     }
148 
149     // Other methods
150     public void addProduct(Product product, int
    index) {
151         products[index] = product;
152     }
153 
154     public void removeProduct(int index) {
155         products[index] = null;
156     }
157 
158     public float calculatePrice() {
159         float totalPrice = 0;
160         for (Product product : products) {
161             if (product != null) {
162                 totalPrice += product.getPrice();
163             }
164         }
165         return totalPrice;
166     }
167 
168     public void placeOrder() {
169         System.out.println("Your Order Success
    ");
170     }
171 }
172 
173 // Order class
174 class Order {
```

```
// Order class
class Order {
    private int customerId;
    private int orderId;
    private Product[] products;
    private float totalPrice;

    // Constructor
    public Order(int customerId, int orderId, Product[] products) {
        this.customerId = Math.abs(customerId);
        this.orderId = Math.abs(orderId);
        this.products = products;
        this.totalPrice = calculateTotalPrice();
    }

    // Other methods
    public void printOrderInfo() {
        System.out.println("Order ID: " + orderId);

        System.out.println("Customer ID: " + customerId);
        System.out.println("Products:");
        for (Product product : products){
          if (product != null){
            System.out.println("- " + product.getName() + ": $" + product.getPrice());
        }}
        System.out.println("Total Price: $" + totalPrice);
    }

    private float calculateTotalPrice() {
        float total = 0;
        for (Product product : products) {
            if (product != null) {
                total += product.getPrice();
            }
        }
        return total;
    }
}

// EcommerceSystem class
public class EcommerceSystem {
    public static void main(String[] args) {
```

```
// EcommerceSystem class
public class EcommerceSystem {
    public static void main(String[] args) {
        // Create products
        ElectronicProduct electronicProduct = new ElectronicProduct(1, "smartphone", 599.9f, "Samsung", 1);
        ClothingProduct clothingProduct = new ClothingProduct(2, "T-shirt", 19.99f, "Medium", "Cotton");
        BookProduct bookProduct = new BookProduct(3, "OOP", 39.99f, "O'Reilly", "X Publications");

        // Create customer
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter customer ID:");
        int customerId = Math.abs(scanner.nextInt());
        scanner.nextLine(); // Consume newline
        System.out.println("Enter customer name:");
        String customerName = scanner.nextLine();
        System.out.println("Enter customer address:");
        String customerAddress = scanner.nextLine();
        Customer customer = new Customer(customerId, customerName, customerAddress);

        // Create shopping cart
        System.out.println("How many products do you want to order?");
        int numProducts = Math.abs(scanner.nextInt());
        Cart cart = new Cart(customer.getCustomerId(), numProducts);

        // Add products to cart
        for (int i = 0; i < numProducts; i++) {
            System.out.println("Enter product type (1: Electronic, 2: Clothing, 3: Book):");
            int productType = Math.abs(scanner.nextInt());

            switch (productType) {
                case 1:
                    cart.addProduct(electronicProduct, i);
                    break;
                case 2:
                    cart.addProduct(clothingProduct, i);
                    break;
                case 3:
                    cart.addProduct(bookProduct, i);
                    break;
                default:
                    System.out.println("Invalid product type.");
                    break;
            }
        }


        // Ask user to place order
        System.out.println("Do you want to place an order? (yes/no)");
        String choice = scanner.next();
        Order order = new Order(customer.getCustomerId(), 1, cart.getProducts());
        if (choice.equals("yes")) {
            cart.placeOrder();
       // Assuming order placed, then print order info
            order.printOrderInfo();
        } else if(choice.equals("no")) {
            System.out.println("do you want to remove prudoct? (yes/no) ");
        String ans=scanner.next();
          if(ans.equals("yes")){
          System. out. println("what product you want to remove? ");
            int ind=scanner. nextInt();
            cart. removeProduct(ind-1);
            order. printOrderInfo();
            cart. placeOrder();
            }
            else{
            System.out.println("order canceled ");
}
        }
    }
}
```

```
Enter customer ID:
23011592
Enter customer name:
Nour Hatem Salah
Enter customer address:
Alex
How many products do you want to order?
4
Enter product type (1: Electronic, 2: Clothing, 3: Book):
1
Enter product type (1: Electronic, 2: Clothing, 3: Book):
2
Enter product type (1: Electronic, 2: Clothing, 3: Book):
2
Enter product type (1: Electronic, 2: Clothing, 3: Book):
3
Do you want to place an order? (yes/no)
yes
Your Order Success
Order ID: 1
Customer ID: 23011592
Products:
- smartphone: $599.9
- T-shirt: $19.99
- T-shirt: $19.99
- OOP: $39.99
Total Price: $679.87

[Program finished]
```