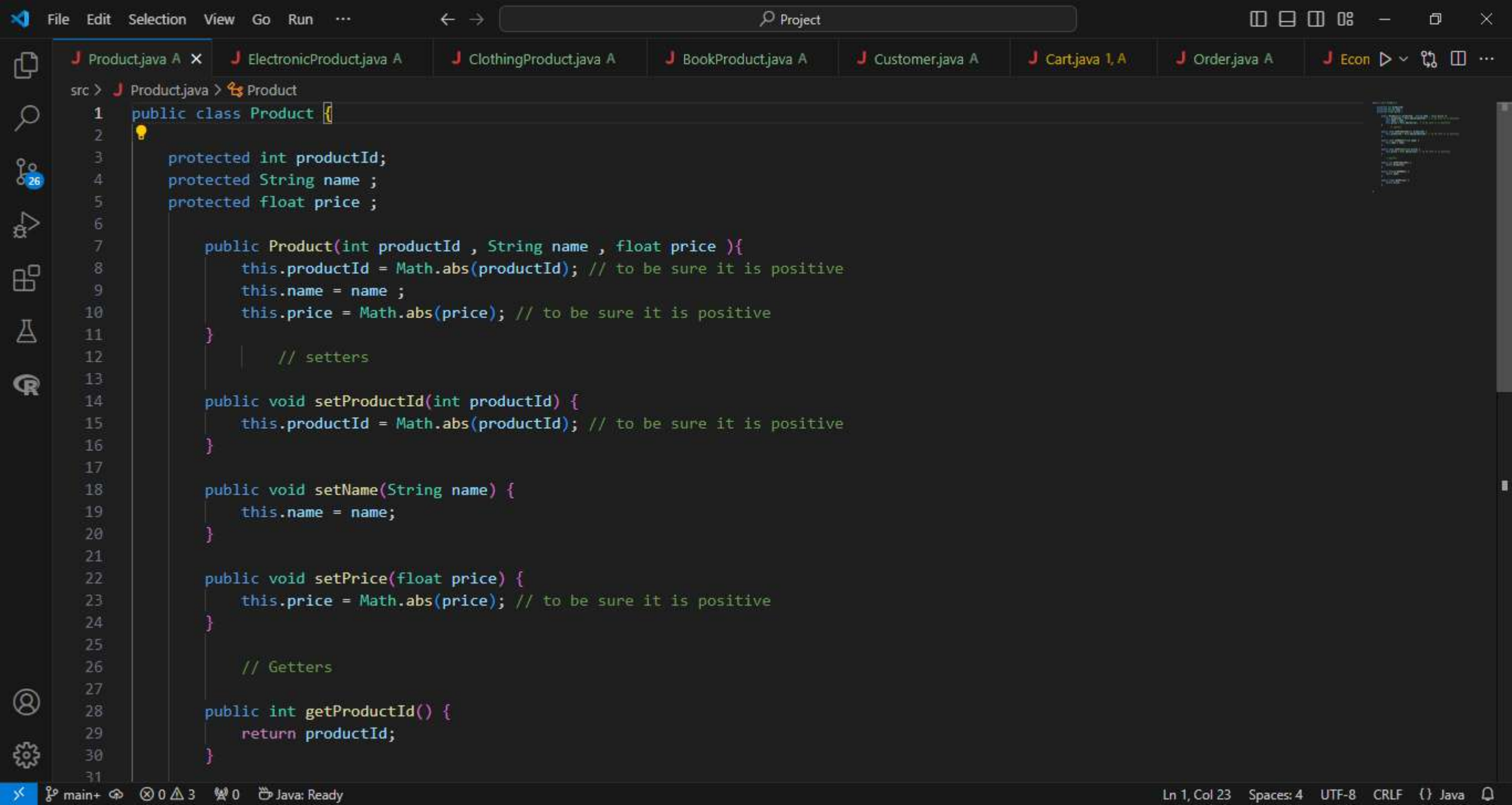
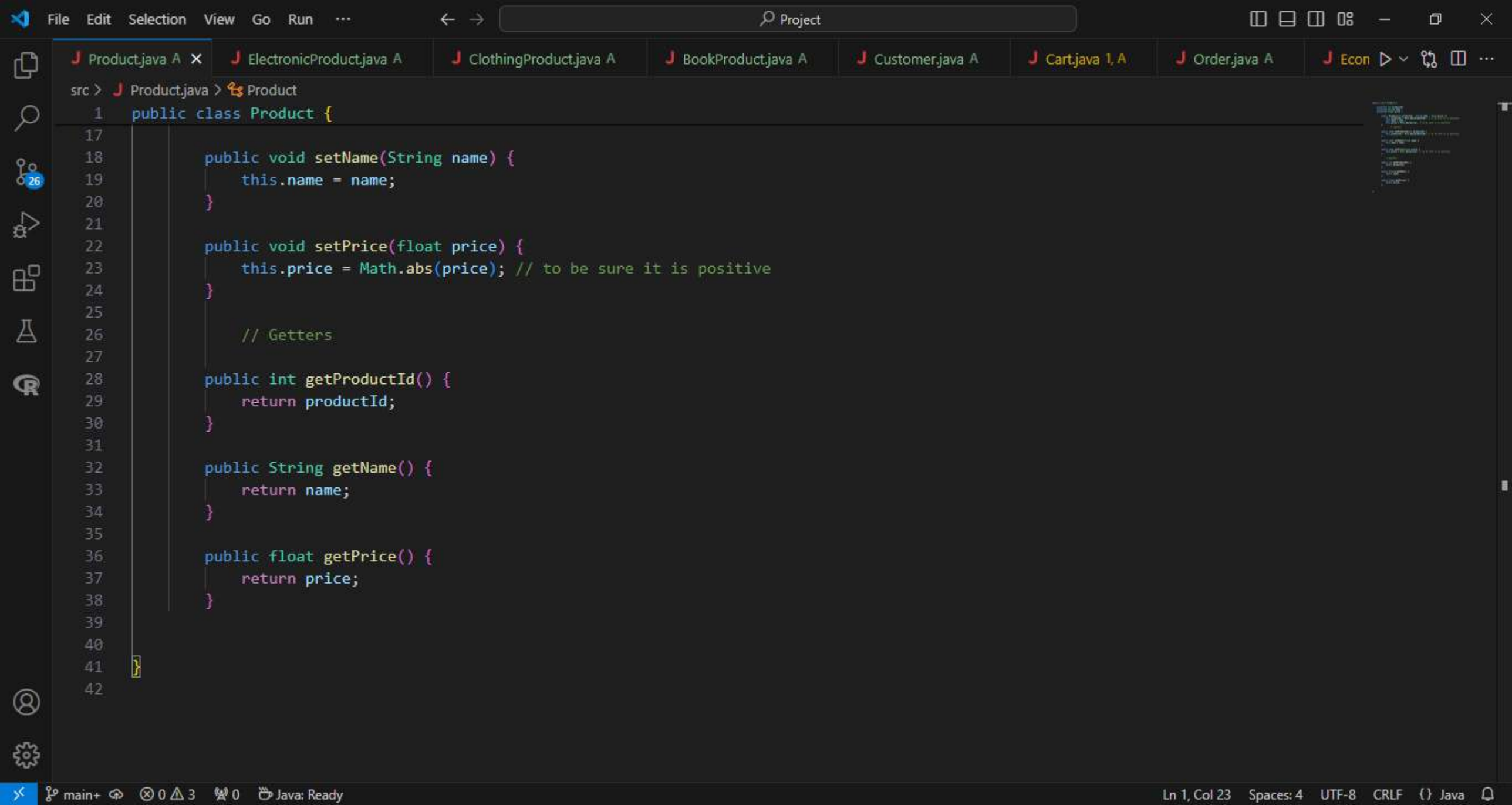
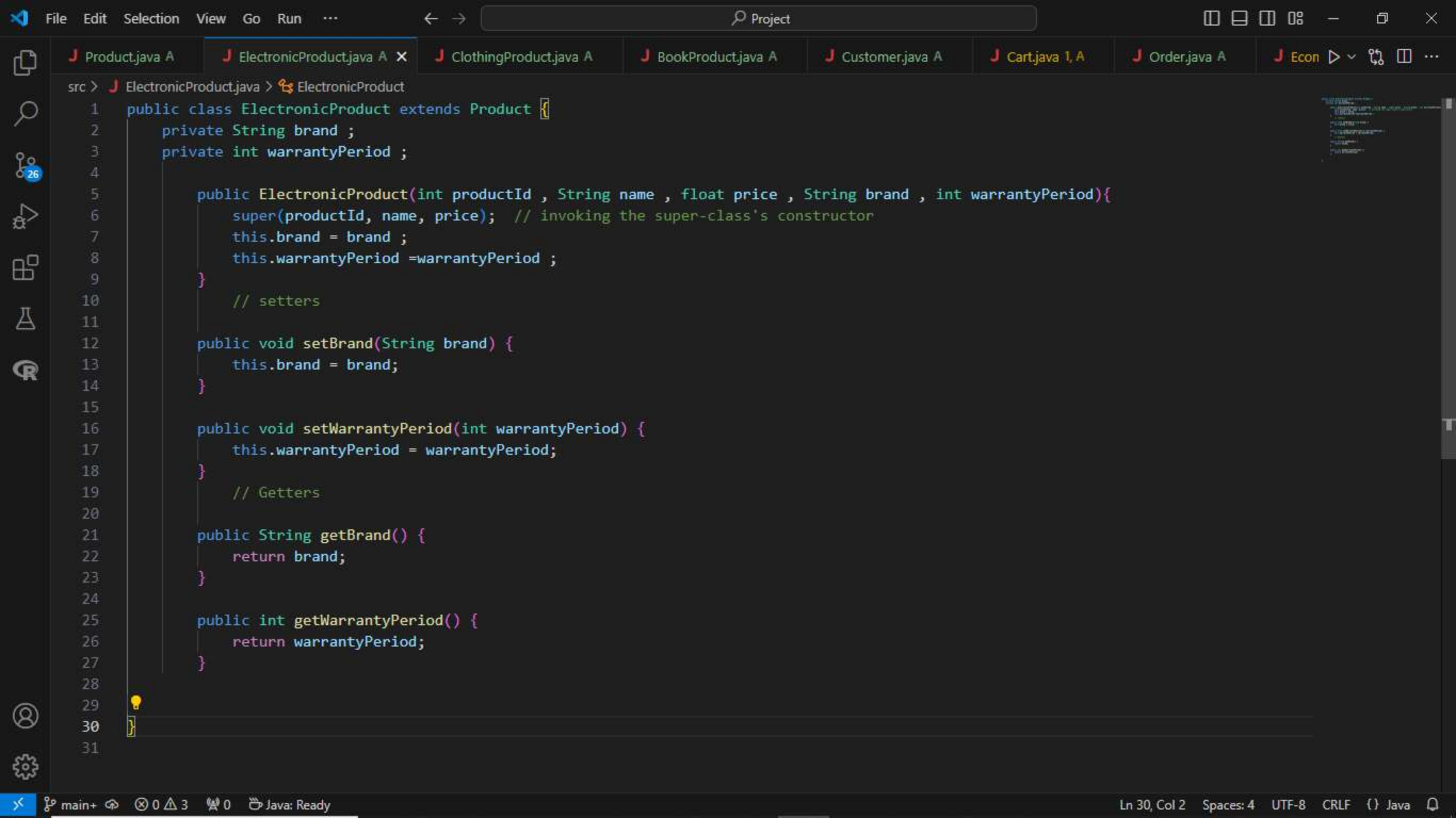
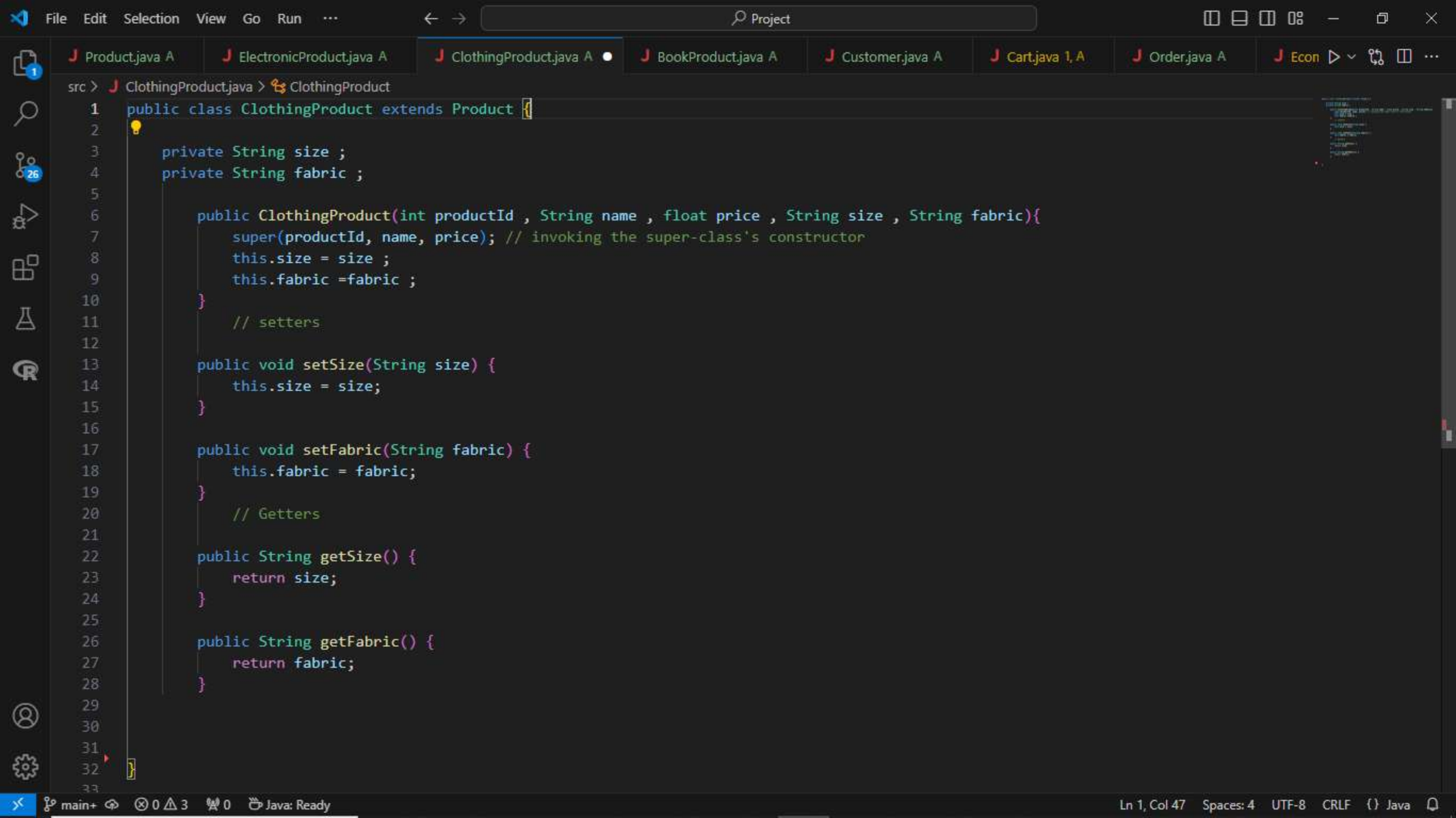
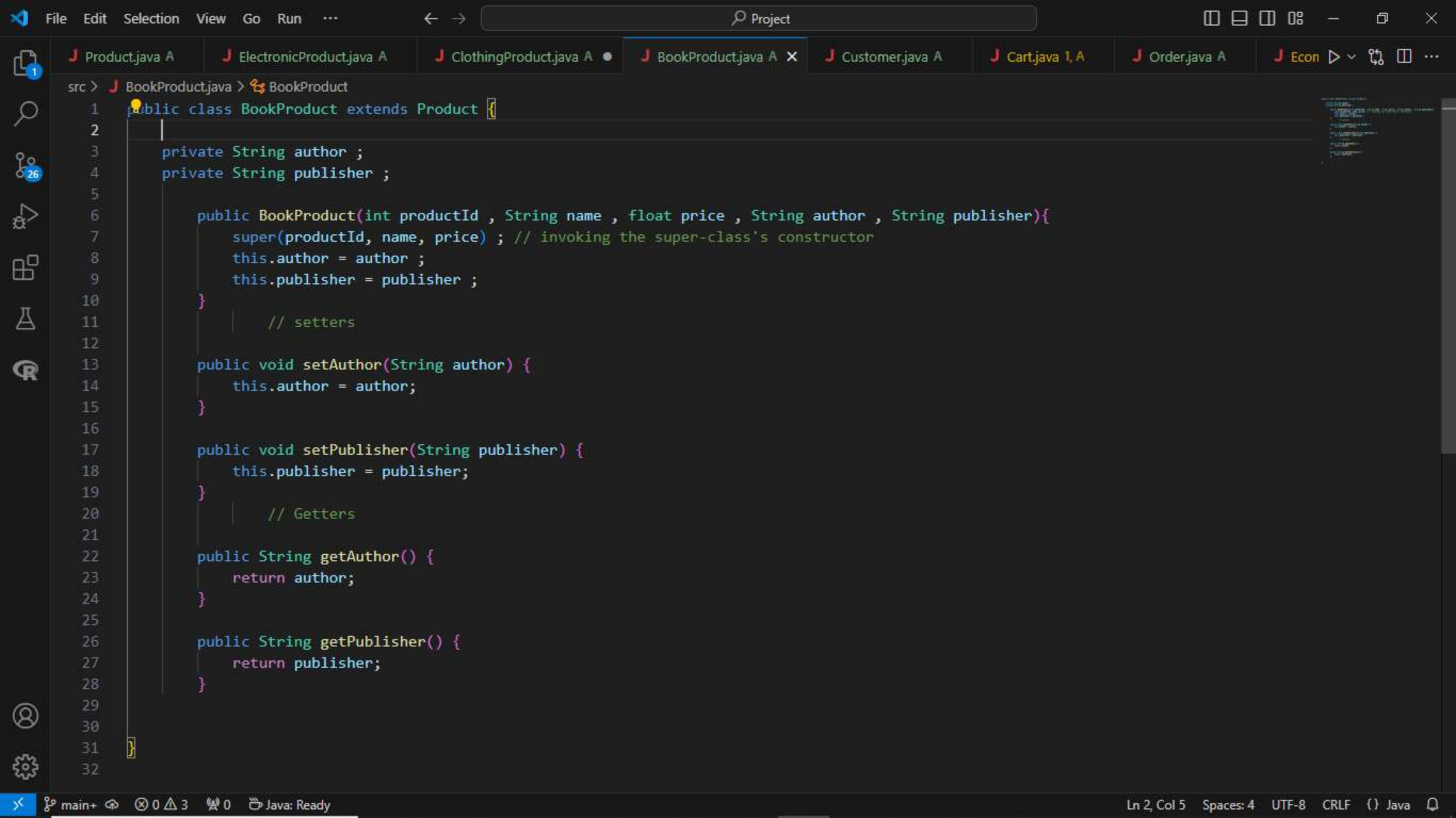
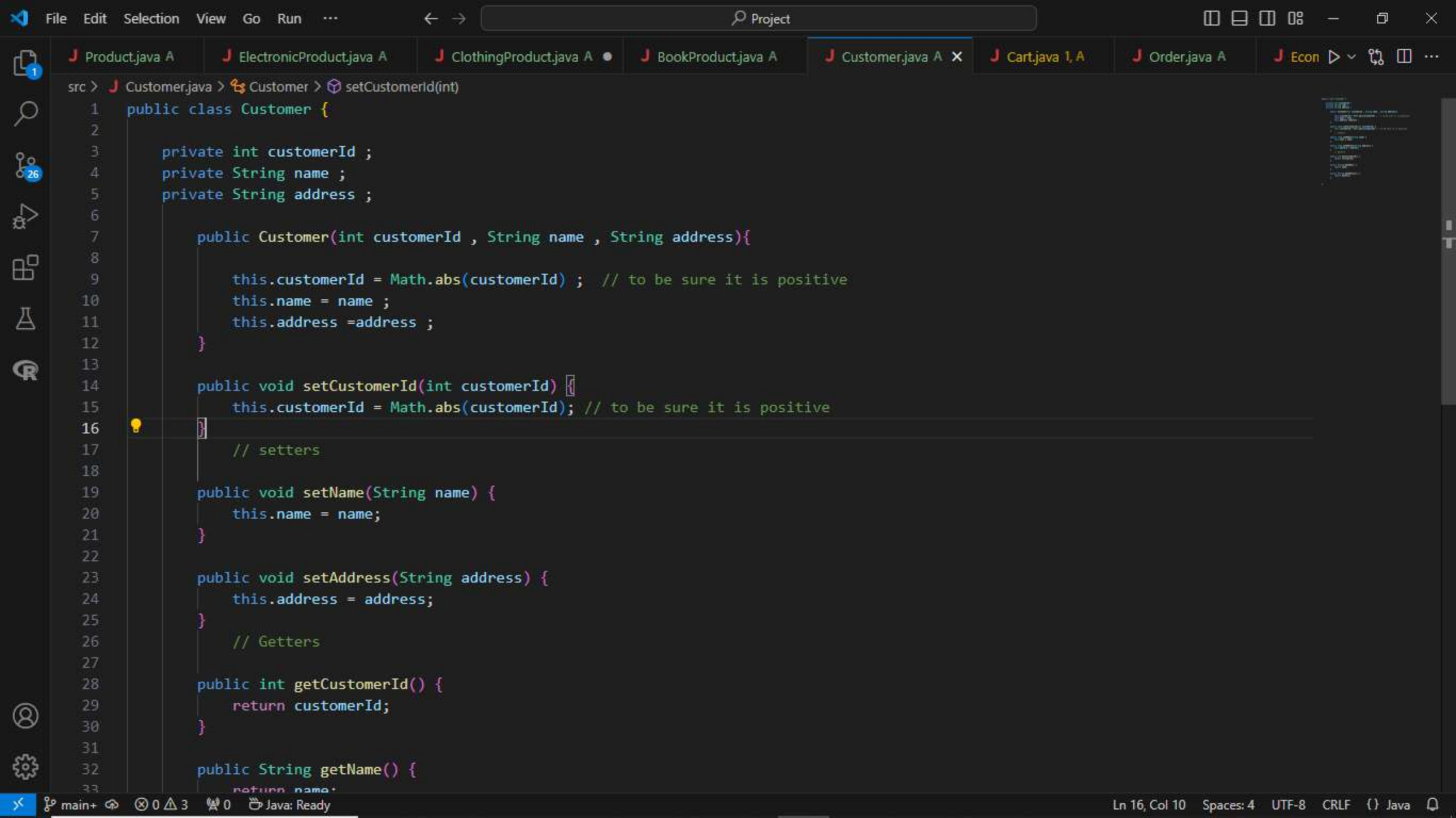

File Edit Selection View Go Run ···
Project
Product.java A
ElectronicProduct.java A
ClothingProduct.java A
BookProduct.java A
Customer.java A
Cart.java 1, A
Order.java A
Econ
src > Product.java > Product
public class Product {
protected int productId;
protected String name ;
protected float price ;
public Product(int productId , String name , float price){
this.productId = Math.abs(productId); // to be sure it is positive
this.name = name ;
this.price = Math.abs(price); // to be sure it is positive
}
// setters
public void setProductId(int productId) {
this.productId = Math.abs(productId); // to be sure it is positive
}
public void setName(String name) {
this.name = name;
}
public void setPrice(float price) {
this.price = Math.abs(price); // to be sure it is positive
}
// Getters
public int getProductId() {
return productId;
}
main+
0 3
0
Java: Ready
Ln 1, Col 23
Spaces: 4
UTF-8
CRLF
Java

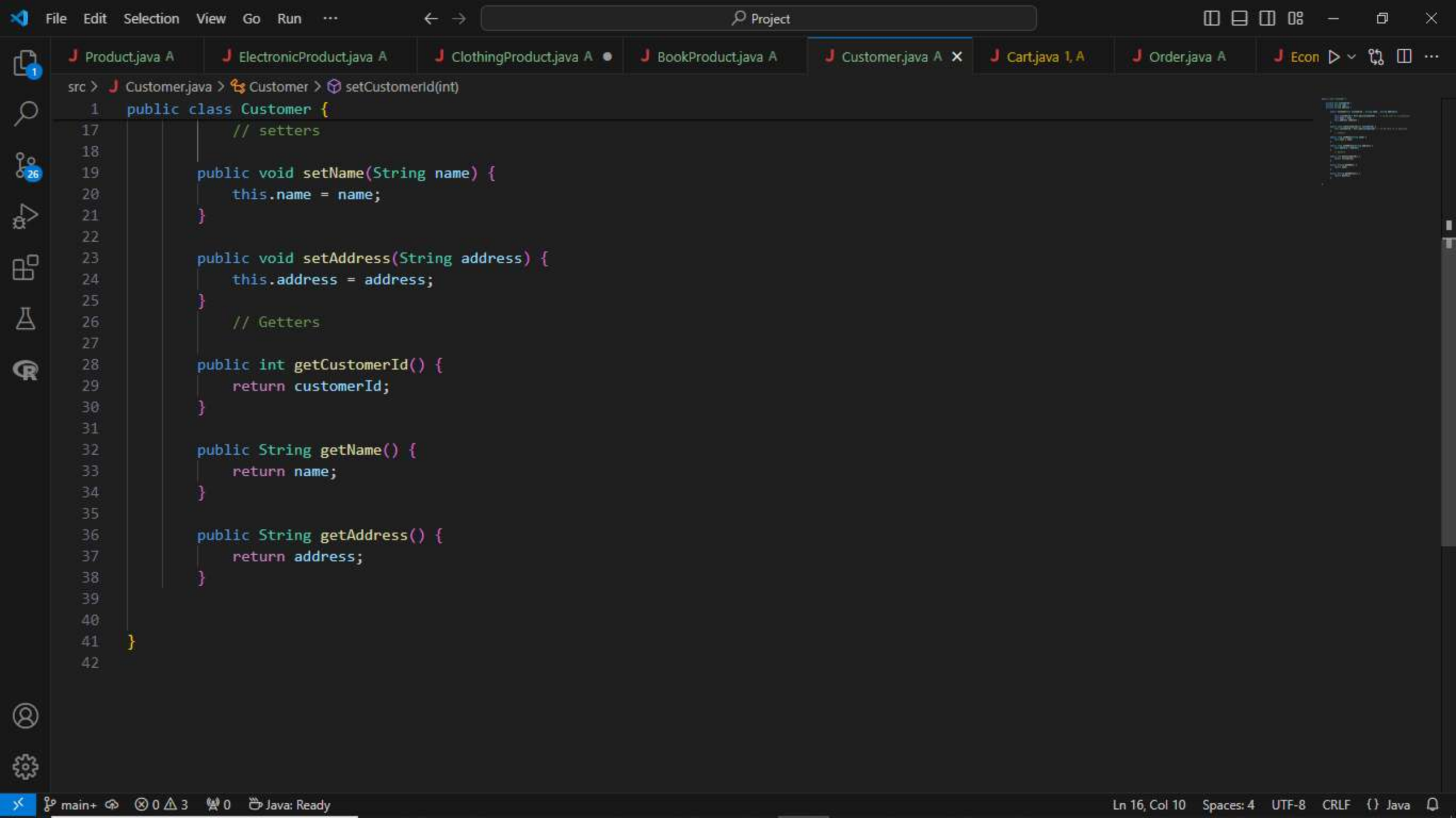

File Edit Selection View Go Run ···
Project
Product.java A
ElectronicProduct.java A
ClothingProduct.java A
BookProduct.java A
Customer.java A
Cart.java 1, A
Order.java A
Econ
src > Product.java > Product
public class Product {
public void setName(String name) {
this.name = name;
}
public void setPrice(float price) {
this.price = Math.abs(price); // to be sure it is positive
}
// Getters
public int getProductId() {
return productId;
}
public String getName() {
return name;
}
public float getPrice() {
return price;
}
}
main+
0 3
0
Java: Ready
Ln 1, Col 23
Spaces: 4
UTF-8
CRLF
Java


File Edit Selection View Go Run ···
Project
Product.java A
ElectronicProduct.java A
ClothingProduct.java A
BookProduct.java A
Customer.java A
Cart.java 1, A
Order.java A
Econ
src > ElectronicProduct.java > ElectronicProduct
public class ElectronicProduct extends Product {
private String brand ;
private int warrantyPeriod ;
public ElectronicProduct(int productId , String name , float price , String brand , int warrantyPeriod){
super(productId, name, price); // invoking the super-class's constructor
this.brand = brand ;
this.warrantyPeriod =warrantyPeriod ;
}
// setters
public void setBrand(String brand) {
this.brand = brand;
}
public void setWarrantyPeriod(int warrantyPeriod) {
this.warrantyPeriod = warrantyPeriod;
}
// Getters
public String getBrand() {
return brand;
}
public int getWarrantyPeriod() {
return warrantyPeriod;
}
}
main+ 0 3 0 Java: Ready
Ln 30, Col 2 Spaces: 4 UTF-8 CRLF Java


File Edit Selection View Go Run ···
Project
Product.java A
ElectronicProduct.java A
ClothingProduct.java A
BookProduct.java A
Customer.java A
Cart.java 1, A
Order.java A
Econ
src > ClothingProduct.java > ClothingProduct
public class ClothingProduct extends Product {
private String size ;
private String fabric ;
public ClothingProduct(int productId , String name , float price , String size , String fabric){
super(productId, name, price); // invoking the super-class's constructor
this.size = size ;
this.fabric =fabric ;
}
// setters
public void setSize(String size) {
this.size = size;
}
public void setFabric(String fabric) {
this.fabric = fabric;
}
// Getters
public String getSize() {
return size;
}
public String getFabric() {
return fabric;
}
}
main+ 0 3 0 Java: Ready
Ln 1, Col 47 Spaces: 4 UTF-8 CRLF Java


File Edit Selection View Go Run ···
Project
Product.java A
ElectronicProduct.java A
ClothingProduct.java A
BookProduct.java A
Customer.java A
Cart.java 1, A
Order.java A
Econ
src > BookProduct.java > BookProduct
public class BookProduct extends Product {
private String author ;
private String publisher ;
public BookProduct(int productId , String name , float price , String author , String publisher){
super(productId, name, price) ; // invoking the super-class's constructor
this.author = author ;
this.publisher = publisher ;
}
// setters
public void setAuthor(String author) {
this.author = author;
}
public void setPublisher(String publisher) {
this.publisher = publisher;
}
// Getters
public String getAuthor() {
return author;
}
public String getPublisher() {
return publisher;
}
}
main+ 0 3 0 Java: Ready
Ln 2, Col 5 Spaces: 4 UTF-8 CRLF Java


File Edit Selection View Go Run ···
Project
Product.java A
ElectronicProduct.java A
ClothingProduct.java A
BookProduct.java A
Customer.java A
Cart.java 1, A
Order.java A
Econ
src > Customer.java > Customer > setCustomerId(int)
public class Customer {
private int customerId ;
private String name ;
private String address ;
public Customer(int customerId , String name , String address){
this.customerId = Math.abs(customerId) ; // to be sure it is positive
this.name = name ;
this.address =address ;
}
public void setCustomerId(int customerId) {
this.customerId = Math.abs(customerId); // to be sure it is positive
}
// setters
public void setName(String name) {
this.name = name;
}
public void setAddress(String address) {
this.address = address;
}
// Getters
public int getCustomerId() {
return customerId;
}
public String getName() {
main+ 0 3 0 Java: Ready
Ln 16, Col 10 Spaces: 4 UTF-8 CRLF Java


File Edit Selection View Go Run ···
Project
Product.java A
ElectronicProduct.java A
ClothingProduct.java A
BookProduct.java A
Customer.java A
Cart.java 1, A
Order.java A
Econ
src > Customer.java > Customer > setCustomerId(int)
public class Customer {
// setters
public void setName(String name) {
this.name = name;
}
public void setAddress(String address) {
this.address = address;
}
// Getters
public int getCustomerId() {
return customerId;
}
public String getName() {
return name;
}
public String getAddress() {
return address;
}
}
main+ 0 3 0 Java: Ready
Ln 16, Col 10 Spaces: 4 UTF-8 CRLF Java

File Edit Selection View Go Run ··· Project

Product.java A | ElectronicProduct.java A | ClothingProduct.java A ● | BookProduct.java A | Customer.java A | Cart.java 1, A ✕ | Order.java A | Econ

src > Cart.java > Cart > placeOrder()

```java
import java.util.Scanner;

public class Cart {

    private int customerId ;
    private int nProducts ;
    public Product products[] ;


        public Cart(int customerId , int nProducts){
            this.customerId = Math.abs(customerId) ; // to be sure it is positive
            this.nProducts =Math.abs(nProducts) ; // to be sure it is positive
            this.products =new Product[nProducts];  // object array to store the products
        }

                // setters

        public void setCustomerId(int customerId) {
            this.customerId = Math.abs(customerId); // to be sure it is positive
        }

        public void setnProducts(int nProducts) {
            this.nProducts = Math.abs(nProducts); // to be sure it is positive
        }
                // Getters

        public int getCustomerId() {
            return customerId;
        }

        public int getnProducts() {
            return nProducts;
```

main+ ⊗ 0 ⚠ 3 0 Java: Ready Ln 61, Col 48 Spaces: 4 UTF-8 CRLF { } Java

File Edit Selection View Go Run ··· Project

Product.java A | ElectronicProduct.java A | ClothingProduct.java A ● | BookProduct.java A | Customer.java A | Cart.java 1, A ✕ | Order.java A | Econ

src > Cart.java > Cart > placeOrder()

```
3   public class Cart {
30
31      public int getnProducts() {
32          return nProducts;
33      }
34
35      public void addProduct(Product obj,int index){  // method to add objects to the object array of products
36
37          if (index >= 0 && index < nProducts) {  // check the existence of the desired index
38              products[index] = obj;
39          }
40
41      }
42
43      public void removeProduct(int index){   // method to remove objects from the object array of products
44
45          if (index >= 0 && index < nProducts) { // check the existence of the desired index
46              products[index] = null;  // assign the value to be null
47          }
48      }
49
50      public float calculatePrice(){ // get the total price of the objects in the array
51
52          float total = 0f ;
53          for (int i = 0; i < products.length; i++) {
54              total += products[i].getPrice() ; // use the function getPrice with each object in the array
55          }
56          return total ;
57      }
58
59      public boolean placeOrder(){ // method to check whether to confirm the order or remove it
60
```

main+ ⊗ 0 ⚠ 3 0 Java: Ready Ln 61, Col 48 Spaces: 4 UTF-8 CRLF { } Java

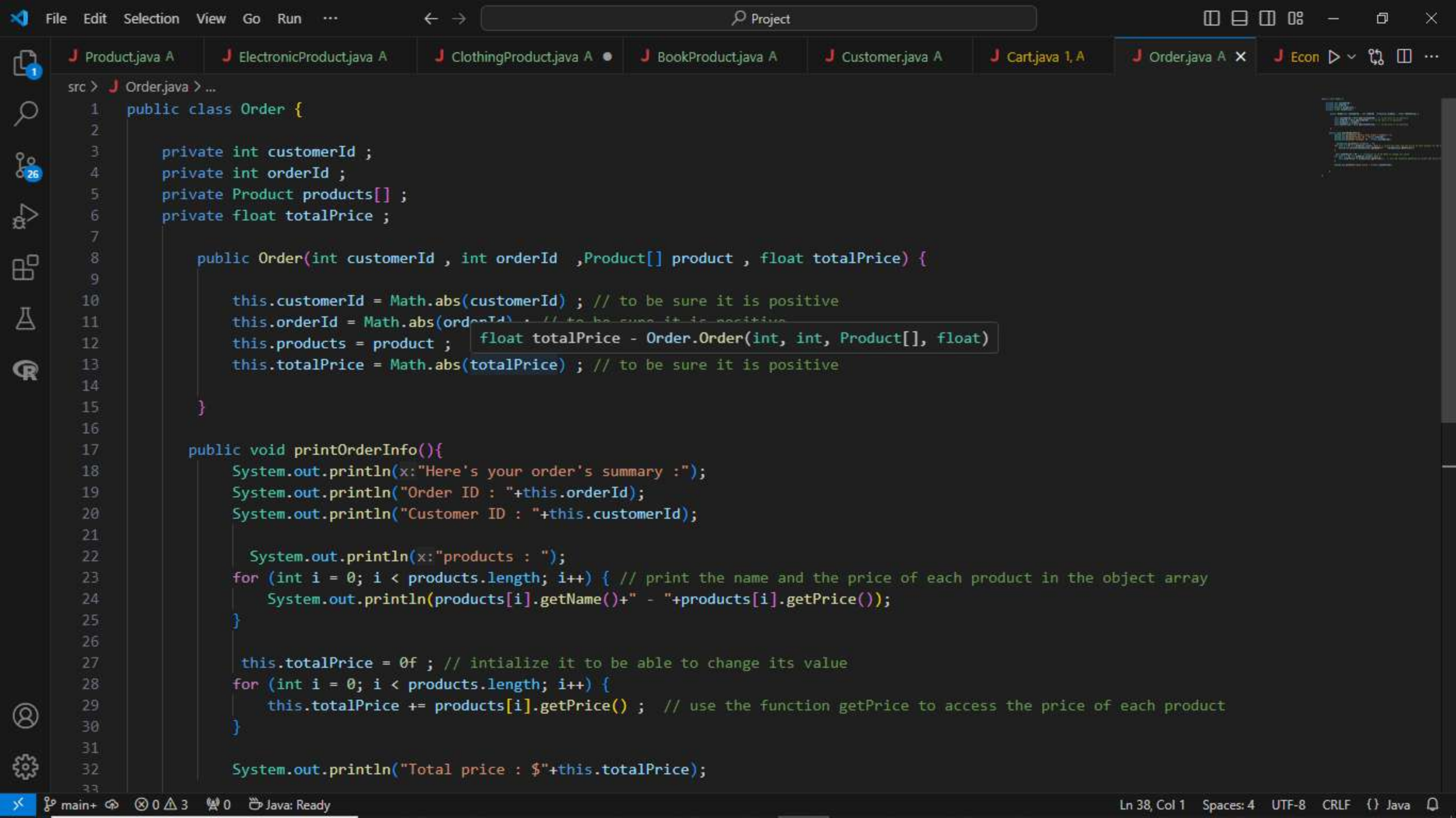
File Edit Selection View Go Run ··· Project

Product.java A | ElectronicProduct.java A | ClothingProduct.java A ● | BookProduct.java A | Customer.java A | Cart.java 1, A ✕ | Order.java A | Econ

src > Cart.java > Cart

```
3   public class Cart {
50          public float calculatePrice(){ // get the total price of the objects in the array
53              for (int i = 0; i < products.length; i++) {
54                  total += products[i].getPrice() ; // use the function getPrice with each object in the array
55              }
56          return total ;
57      }
58
59      public boolean placeOrder(){ // method to check whether to confirm the order or remove it
60
61          System.out.println("your total is $"+calculatePrice()+". Would you like to place the order? 1-Yes 2-No");
62
63           Scanner scanner = new Scanner(System.in);
64                  int choice = scanner.nextInt();
65                      if (choice == 1) // means confirm the order
66                      {
67                          return true ;
68                      }
69
70                      else    // means cancel the order the emptys the cart
71                      {
72                          return false ;
73                      }
74
75      }
76
77
78
79  }
80
```

main+ ⊗ 0 ⚠ 3 0 Java: Ready | Ln 78, Col 1 | Spaces: 4 | UTF-8 | CRLF | {} Java




```java
public class Order {

    private int customerId ;
    private int orderId ;
    private Product products[] ;
    private float totalPrice ;

        public Order(int customerId , int orderId  ,Product[] product , float totalPrice) {

            this.customerId = Math.abs(customerId) ; // to be sure it is positive
            this.orderId = Math.abs(orderId) ; // to be sure it is positive
            this.products = product ;
            this.totalPrice = Math.abs(totalPrice) ; // to be sure it is positive

        }

       public void printOrderInfo(){
            System.out.println(x:"Here's your order's summary :");
            System.out.println("Order ID : "+this.orderId);
            System.out.println("Customer ID : "+this.customerId);

              System.out.println(x:"products : ");
            for (int i = 0; i < products.length; i++) { // print the name and the price of each product in the object array
                System.out.println(products[i].getName()+" - "+products[i].getPrice());
            }

             this.totalPrice = 0f ; // intialize it to be able to change its value
            for (int i = 0; i < products.length; i++) {
                this.totalPrice += products[i].getPrice() ;  // use the function getPrice to access the price of each product
            }

            System.out.println("Total price : $"+this.totalPrice);
```

File Edit Selection View Go Run ··· Project

oduct.java A | ClothingProduct.java A ● | BookProduct.java A | Customer.java A | Cart.java 1, A | Order.java A | EcommerceSystem.java 2, A ● | Settings

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

```java
import java.util.Scanner;

public class EcommerceSystem {
    Run | Debug
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        // Creating the first object (Electronic product)
        ElectronicProduct product_1 = new ElectronicProduct(productId:1, name:"smartphone",  price:599.99f, brand:"samsung", warrantyPeriod:1);

        // Creating the second object (Clothing product)
        ClothingProduct product_2 = new ClothingProduct(productId:2, name:"T-shirt",  price:19.99f, size:"Medium", fabric:"Cotton");

        // Creating the third object (Book product)
        BookProduct product_3 = new BookProduct(productId:3, name:"OOP", price:39.99f, author:"O'Reilly", publisher:"X Publications");

        System.out.println(x:"please enter your id"); // taking the customer ID
        int customerId = input.nextInt() ;

        input.nextLine();   // clean the \n which stored in the buffer

        System.out.println(x:"please enter your name ");  // taking the customer name
        String name = input.nextLine() ;

        System.out.println(x:"please enter your address");    // taking the customer address
        String address = input.nextLine() ;

        Customer customer = new Customer(customerId, name, address) ;   // create a customer object with the data entered by the user

        System.out.print(s:"How many products you want to add to your cart ? ");
        int numberOfProducts = input.nextInt() ;    // take the amount of the products to be the size of the array


        Cart cart = new Cart(customerId, Math.abs(numberOfProducts)); // Create a cart object with unique customer id and certain amount of products

        for (int i = 0; i < Math.abs(numberOfProducts); i++) { // take the order from the customer and storing the products in the array

            System.out.println(x:"Which product you would like to add ? 1-Smartphone 2-T-shirt 3-OOP");
            int choice = input.nextInt();

```

main+ ⊗ 0 ⚠ 3 0 Java: Ready Ln 16, Col 78 Spaces: 4 UTF-8 CRLF {} Java

File Edit Selection View Go Run ··· Project

oduct.java A | ClothingProduct.java A ● | BookProduct.java A | Customer.java A | Cart.java 1, A | Order.java A | EcommerceSystem.java 2, A ● | Settings

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

```
3   public class EcommerceSystem {
4       public static void main(String[] args) {
30          int numberOfProducts = input.nextInt() ;    // take the amount of the products to be the size of the array
31
32
33          Cart cart = new Cart(customerId, Math.abs(numberOfProducts)); // Create a cart object with unique customer id and certain amount of products
34
35          for (int i = 0; i < Math.abs(numberOfProducts); i++) { // take the order from the customer and storing the products in the array
36
37              System.out.println(x:"Which product you would like to add ? 1-Smartphone 2-T-shirt 3-OOP");
38              int choice = input.nextInt();
39
40
41              switch (Math.abs(choice)) {
42                  case 1:
43
44                      cart.addProduct(product_1, i);
45                          break;
46
47                  case 2 :
48
49                      cart.addProduct(product_2, i);
50                      break;
51
52                  case 3 :
53
54                      cart.addProduct(product_3, i);
55                      break;
56
57                  default :
58                      System.out.println(x:"invalid input");
59                      break ;
60              }
61
62          }
63
64          if(cart.calculatePrice() == 0) // if the total price is 0 that means that the cart is empty
65          {
66              System.out.println(x:"The cart is empty");
67          }
68
```

main+ ⊗ 0 ⚠ 3 0 Java: Ready Ln 16, Col 78 Spaces: 4 UTF-8 CRLF { } Java

File Edit Selection View Go Run ··· Project

oduct.java A | ClothingProduct.java A ● | BookProduct.java A | Customer.java A | Cart.java 1, A | Order.java A | EcommerceSystem.java 2, A ● | Settings

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

```
3   public class EcommerceSystem {
4       public static void main(String[] args) {
62          }
63
64          if(cart.calculatePrice() == 0) // if the total price is 0 that means that the cart is empty
65          {
66              System.out.println(x:"The cart is empty");
67          }
68
69          else
70          {
71
72              boolean confirmation = cart.placeOrder();  // check the state of the confirmation of the order
73
74
75              if (confirmation)   // if confirm == true --> save the order and print its details
76              {
77                  Order order = new Order(customerId, orderId:1, cart.products, cart.calculatePrice()) ;
78
79                  order.printOrderInfo();
80
81              }
82
83              else    // if confirm == false --> cancel the order and empty the array
84              {
85                  for (int i = 0; i < numberOfProducts; i++) {
86
87                      cart.removeProduct(i);
88                  }
89
90                  System.out.println(x:"The order has been cancelled.");
91
92              }
93
94          }
95
96      }
97
98
99  }
```

main+ ⊗ 0 ⚠ 3 0 Java: Ready | Ln 16, Col 78 | Spaces: 4 | UTF-8 | CRLF | {} Java

```
please enter your id
23011631
please enter your name
"Youssef Gad "
please enter your address
"  "
How many products you want to add to your cart ? 4
Which product you would like to add ? 1-Smartphone 2-T-shirt 3-OOP
1
Which product you would like to add ? 1-Smartphone 2-T-shirt 3-OOP
2
Which product you would like to add ? 1-Smartphone 2-T-shirt 3-OOP
3
Which product you would like to add ? 1-Smartphone 2-T-shirt 3-OOP
2
your total is $679.95996. Would you like to place the order? 1-Yes 2-No
1
Here's your order's summary :
Order ID : 1
Customer ID : 23011631
products :
smartphone - 599.99
T-shirt - 19.99
OOP - 39.99
T-shirt - 19.99
Total price : $679.95996
```