

The screenshot shows a Java development environment with the following details:

- Toolbar:** Standard icons for file operations (New, Open, Save, Print, Find, Copy, Paste, etc.).
- Status Bar:** Displays "274.8/490.0MB" and two small circular icons.
- Search Bar:** "Search (⌘+F)"
- Project Bar:** Shows multiple open files: Start Page, Product.java, ElectronicProduct.java, BookProduct.java, ClothingProduct.java, Customer.java, Cart.java, Order.java, and Ecommerce... (partially visible).
- Source Tab:** Active tab, showing the code for `Product.java`.
- Code Editor:** The content of `Product.java` is displayed, defining a class `Product` with fields `productId`, `name`, and `price`, and methods for setting and getting each.

```
1 package assiment;
2 public class Product {
3     int productId;
4     String name;
5     double price;
6
7     public Product(int productId, String name, double price) {
8         this.productId = Math.abs(productId);
9         this.name = name;
10        this.price = Math.abs(price);
11    }
12    public int getProductId() {
13        return productId;
14    }
15    public void setProductId(int productId) {
16        this.productId= Math.abs(productId);
17    }
18    public String getName() {
19        return name;
20    }
21    public void setName(String name) {
22        this.name = name;
23    }
24    public double getPrice() {
25        return price;
26    }
27    public void setPrice(double price) {
28        this.price= Math.abs(price);
29    }
30    }
31 }
32 }
```

- Output Tab:** Shows two run configurations: "assiment (run)" and "assiment (run) #2".

The screenshot shows a Java development environment with the following details:

- Toolbar:** Includes icons for file operations (New, Open, Save, Print), search, and navigation.
- Status Bar:** Shows disk usage (389.0/490.0MB) and system status (Search (⌘+I)).
- Project Bar:** Lists multiple Java files: Product.java, ElectronicProduct.java, BookProduct.java, ClothingProduct.java, Customer.java, Cart.java, Order.java, and Ecommerce... (partially visible).
- Source Editor:** Displays the code for `ElectronicProduct.java`. The code defines a class `ElectronicProduct` that extends `Product`. It includes methods for setting and getting the brand name and warranty period, and methods for setting and getting the product ID. The code uses `Math.abs()` to ensure the warranty period is always positive.
- Output Bar:** Shows two tabs: "assiment (run)" and "assiment (run) #2".
- Bottom Status:** Shows the current run status as "running..." and other system information like resolution (2:1) and terminal type (INS Unix (LF)).

```
1 package assiment;
2 public class ElectronicProduct extends Product{
3     private String brand;
4     private int warrantyPeriod;
5
6     public ElectronicProduct( int productId, String name, double price, String brand, int warrantyPeriod) {
7         super(productId, name, price);
8         this.brand = brand;
9         this.warrantyPeriod = Math.abs( warrantyPeriod);
10    }
11
12    public String getBrand() {
13        return brand;
14    }
15
16    public void setBrand(String brand) {
17        this.brand = brand;
18    }
19
20    public int getWarrantyPeriod() {
21        return warrantyPeriod;
22    }
23
24    public void setWarrantyPeriod(int warrantyPeriod) {
25
26        this.warrantyPeriod = Math.abs(warrantyPeriod);
27    }
28
29    public int getProductId() {
30        return productId;
31    }
32
33    public void setProductId(int productId) {
34        this.productId = productId;
35    }
36}
```

The screenshot shows a Java code editor interface with the following details:

- Toolbar:** Includes icons for file operations (New, Open, Save, Print), configuration (Default Config), system status (CPU, RAM, Disk), and search.
- Project Bar:** Lists multiple Java files: Start Page, Product.java (selected), ElectronicProduct.java, BookProduct.java, ClothingProduct.java, Customer.java, Cart.java, Order.java, and Ecommerce... (partially visible).
- Navigator:** Shows the current file structure, with "Source" and "History" tabs selected.
- Code Editor:** Displays the code for `Product.java`. The code defines a class with methods for product ID, name, and price.

```
28  
29     public int getProductId() {  
30         return productId;  
31     }  
32  
33     public void setProductId(int productId) {  
34         this.productId = productId;  
35     }  
36  
37     public String getName() {  
38         return name;  
39     }  
40  
41     public void setName(String name) {  
42         this.name = name;  
43     }  
44  
45     public double getPrice() {  
46         return price;  
47     }  
48  
49     public void setPrice(float price) {  
50         this.price = price;  
51     }  
52  
53  
54  
55  
56 }  
57
```

- Output Bar:** Shows two tabs: "assiment (run)" and "assiment (run) #2". The first tab is currently active and shows the status "running...".

Start Page x Product.java x ElectronicProduct.java x BookProduct.java x ClothingProduct.java x Customer.java x Cart.java x Order.java x Ecommerce... < > ▾

Source History

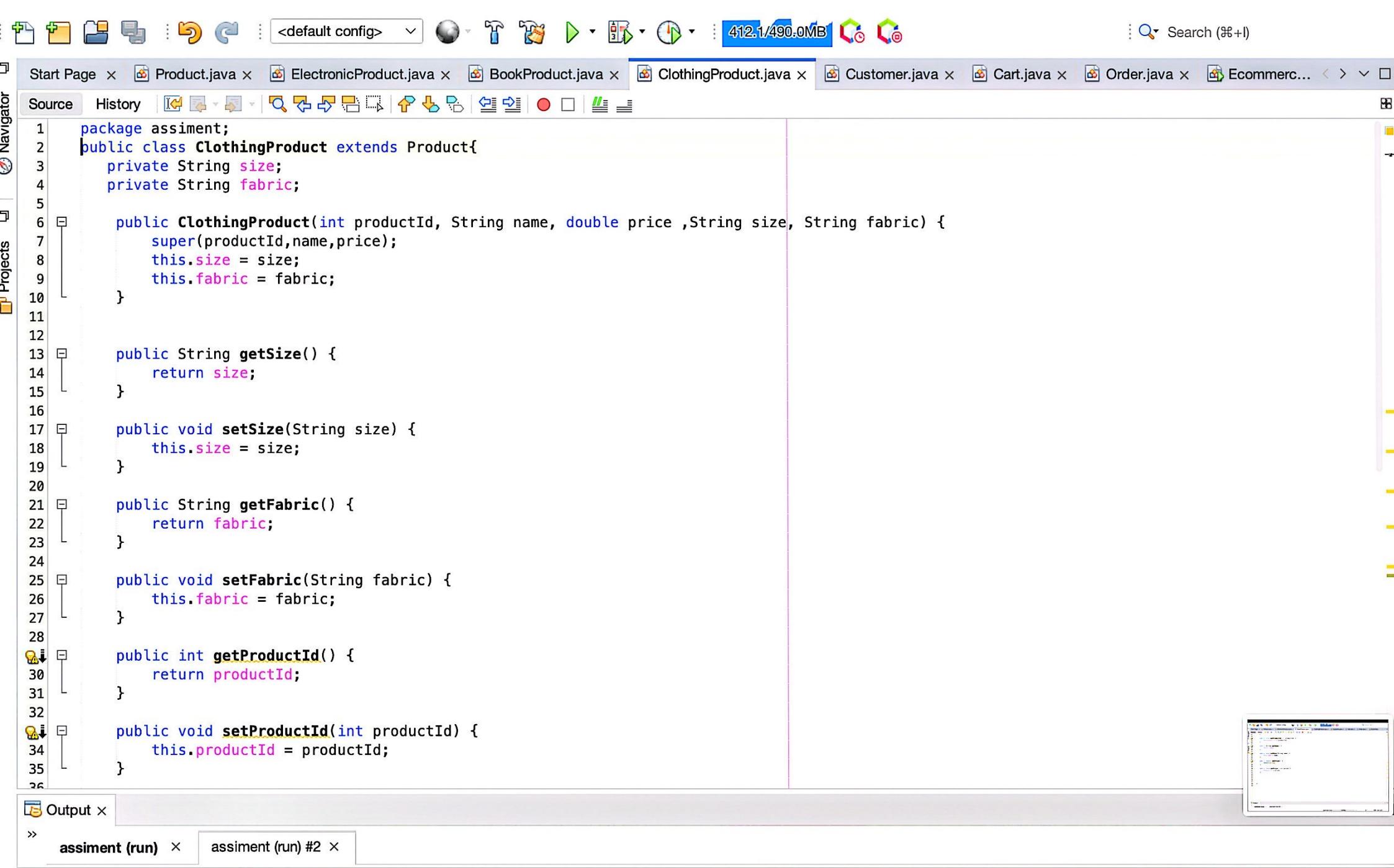
```
1 package assiment;
2 public class BookProduct extends Product {
3     private String author ;
4     private String publisher;
5
6     public BookProduct(int productId, String name, double price ,String author, String publisher) {
7         super(productId,name,price);
8         this.author = author;
9         this.publisher = publisher;
10    }
11
12
13
14     public String getAuthor() {
15         return author;
16     }
17
18     public void setAuthor(String author) {
19         this.author = author;
20     }
21
22     public String getPublisher() {
23         return publisher;
24     }
25
26     public void setPublisher(String publisher) {
27         this.publisher = publisher;
28     }
29
30
31     public int getProductId() {
32         return productId;
33     }
34
35     public void setProductId(int productId) {
36         this.productId = productId;
37     }
38 }
```

Output x
assiment (run) x assiment (run) #2 x assiment (run) running... x 2:1 INS Unix (LF)

The screenshot shows a Java code editor interface with the following details:

- Toolbar:** Includes icons for file operations (New, Open, Save, Print), configuration (Default Config), system status (CPU, RAM, Disk), and search.
- Project Bar:** Lists files: Start Page, Product.java, ElectronicProduct.java, BookProduct.java, ClothingProduct.java, Customer.java, Cart.java, Order.java, and Ecommerce... (partially visible).
- Navigator:** Shows the current file is Product.java.
- Source Editor:** Displays the code for the Product class. The code includes methods for setting and getting product ID, name, and price, along with their respective accessors and mutators. Lines 32 through 59 are shown, ending with a closing brace for the class definition.
- Output Bar:** Shows two tabs: assiment (run) and assiment (run) #2. The assiment (run) tab is active and shows the status "running...".

```
32     }
33
34     public void setProductId(int productId) {
35         this.productId = productId;
36     }
37
38     public String getName() {
39         return name;
40     }
41
42     public void setName(String name) {
43         this.name = name;
44     }
45
46     public double getPrice() {
47         return price;
48     }
49
50     public void setPrice(float price) {
51         this.price = price;
52     }
53
54
55
56
57
58 }
59 }
```



Scanned with CamScanner

The screenshot shows a Java code editor within an IDE. The code is for a `Product` class, which has methods for setting and getting product ID, name, and price. The code is annotated with line numbers from 32 to 55. The IDE interface includes a toolbar at the top, a navigation bar with tabs for various files, and a bottom status bar.

```
32     public void setProductId(int productId) {
33         this.productId = productId;
34     }
35
36     public String getName() {
37         return name;
38     }
39
40     public void setName(String name) {
41         this.name = name;
42     }
43
44     public double getPrice() {
45         return price;
46     }
47
48     public void setPrice(float price) {
49         this.price = price;
50     }
51
52
53
54 }
55
```

Output:

- assiment (run) ×
- assiment (run) #2 ×

Bottom Status Bar:

- assiment (run) running... ×
- 2:1
- INS Unix (LF)

Start Page x Product.java x ElectronicProduct.java x BookProduct.java x ClothingProduct.java x Customer.java x Cart.java x Order.java x Ecommerce... < > ▾

Source History ▾

```
1 package assiment;
2 public class Customer {
3     int customerId;
4     String name;
5     String address;
6
7     public Customer(int customerId, String name, String address) {
8         this.customerId = Math.abs(a: customerId);
9         this.name = name;
10        this.address = address;
11    }
12
13
14
15    public int getCustomerId() {
16        return customerId;
17    }
18
19    public void setCustomerId(int customerId) {
20
21        this.customerId = Math.abs(a: customerId);
22    }
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 String getAddress() {
34        return address;
35    }
36
```

Output x assiment (run) x assiment (run) #2 x assiment (run) running... x 2:1 INS Unix (LF)

The screenshot shows a Java IDE interface with the following details:

- Toolbar:** Includes icons for file operations (New, Open, Save, Print), configuration (Default Config), system status (CPU, RAM, Disk), and search.
- Menubar:** Contains "File", "Edit", "View", "Project", "Tools", "Help".
- Project Explorer (Projects):** Shows a single project structure with a "Customer" folder containing "Customer.java".
- Navigator:** Shows the current file "Customer.java" is selected.
- Code Editor:** Displays the "Customer.java" source code. The code defines a class "Customer" with attributes "customerId" (private long) and "name" (private String), and methods "getCustomerId" (long), "getName" (String), "setName" (void), "getAddress" (String), and "setAddress" (void). Lines 21 through 44 are visible.
- Output:** Shows two tabs: "assiment (run)" and "assiment (run) #2".
- Bottom Status Bar:** Shows "assiment (run) running..." and other system information like "2:1" and "INS Unix (LF)".

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

```
1 package assiment;
2 public class Cart {
3     int customerId;
4     int nproducts;
5     int i=0;
6     Product[]products;
7     private double totalPrice;
8
9     public Cart(int customerId, int nproducts) {
10         this.customerId = Math.abs(a: customerId) ;
11         this.nproducts = Math.abs(a: nproducts);
12         this.products = new Product[nproducts];
13     }
14
15
16
17     public int getCustomerId() {
18         return customerId;
19     }
20
21     public void setCustomerId(int customerId) {
22
23         this.customerId = Math.abs(a: customerId);
24     }
25
26     public int getNproducts() {
27         return nproducts;
28     }
29
30     public void setNproducts(int nproducts) {
31
32         this.nproducts = Math.abs(a: nproducts);
33     }
34
35     public Product[] getProducts() {
36
37     }
38 }
```

Output x
assiment (run) x assiment (run) #2 x

The screenshot shows a Java development environment with the following details:

- Toolbar:** Includes icons for file operations (New, Open, Save, etc.), search, and navigation.
- Status Bar:** Shows memory usage (338.0/490.0MB) and system icons.
- Project Explorer:** On the left, shows a single project named "Ecommerce".
- Code Editor:** The main window displays the `Cart.java` file with the following code:

```
34     }
35
36     public Product[] getProducts() {
37         return products;
38     }
39
40     public void setProducts(Product[] products) {
41         this.products = products;
42     }
43     public void addProduct(Product product){
44
45         products[i]=product;
46         i++;
47     }
48
49
50     public void removeProduct(Product product){
51         for(int i=0; i< products.length;i++){
52             if(products[i]!= null && products[i].equals(obj:product)){
53                 products[i]=null;
54                 return;
55             }
56         }
57         System.out.println("Cannot find this product");
58     }
59
60     public double calculatePrice(){
61         double totalPrice =0;
62         for(int i=0; i<nproducts ;i++){
63
64             totalPrice+= products[i].getPrice();
65
66         }
67
68     }
69 }
```

- Output View:** At the bottom, there are two tabs: "assiment (run)" and "assiment (run) #2".

The screenshot shows a Java development environment with the following details:

- Toolbar:** Includes icons for file operations (New, Open, Save, etc.), project management, and system status (CPU, RAM, Disk).
- Top Bar:** Shows the current configuration as <default config>, the number of files open (436.0/490.0MB), and a search bar labeled "Search (⌘+I)".
- Project Explorer:** On the left, it lists "Start Page", "Product.java", "ElectronicProduct.java", "BookProduct.java", "ClothingProduct.java", "Customer.java", "Cart.java", "Order.java", and "Ecommerce...".
- Code Editor:** The main area displays the code for the `Cart.java` class. The code handles adding products to a cart, removing products, calculating the total price, and placing an order.
- Terminal:** A small window on the right shows the command line interface with the text "assiment (run) running...".
- Output Log:** At the bottom, the "Output" tab shows two entries: "assiment (run)" and "assiment (run) #2".

```
48     }
49
50    public void removeProduct(Product product){
51        for(int i=0; i< products.length;i++){
52            if(products[i] != null && products[i].equals(obj:product)){
53                products[i]=null;
54                return;
55            }
56        }
57        System.out.println(x: "Cannot find this product");
58    }
59
60    public double calculatePrice(){
61        double totalPrice =0;
62        for(int i=0; i<products ;i++){
63
64            totalPrice+= products[i].getPrice();
65
66        }
67
68
69        return totalPrice;
70    }
71
72    public Order place_order(int orderId){
73        if(products.length == 0){
74            System.out.println(x: "The cart is empty");
75            return null;
76        }
77        Order order=new Order(customerId ,orderId,products);
78        return order;
79    }
80}
```

assiment (run) running... x 2:1 INS Unix (LF)

```
1 package assiment;
2 import assiment.Product;
3 public class Order {
4     int customerId;
5     int orderId;
6     double totalPrice;
7     Product[] products;
8
9     public Order(int customerId, int orderId, Product[] products) {
10         this.customerId = Math.abs(a: customerId);
11         this.orderId = Math.abs(a: orderId);
12         this.products = products;
13     }
14
15     public int getCustomerId() {
16         return customerId;
17     }
18
19     public void setCustomerId(int customerId) {
20         this.customerId = customerId;
21     }
22
23     public int getOrderId() {
24         return orderId;
25     }
26
27     public void setOrderId(int orderId) {
28         this.orderId = orderId;
29     }
30
31     public double getTotalPrice() {
32         return totalPrice;
33     }
34
35     public void setTotalPrice(double totalPrice) {
```

Output x
assiment (run) x assiment (run) #2 x

assiment (run) running... x 2:1 INS | Unix (LF)

Scanned with CamScanner

The screenshot shows a Java development environment with the following details:

- Toolbar:** Includes icons for file operations (New, Open, Save, Print), search, and system status (Memory: 263.7/490.0MB, Battery: 100%, Network).
- Menu Bar:** Contains "File", "Edit", "View", "Tools", "Help".
- Project Explorer:** Shows a single project named "Ecommerce".
- Navigator:** Shows the current file "Order.java" is selected.
- Code Editor:** Displays the content of the "Order.java" file. The code implements a class with methods for setting and getting products, calculating total price, printing product info, and printing order info.
- Output:** Shows the output tab with two entries: "assiment (run)" and "assiment (run) #2".

```
34
35     public void setTotalPrice(double totalPrice) {
36         this.totalPrice = totalPrice;
37     }
38
39     public Product[] getProducts() {
40         return products;
41     }
42
43     public void setProducts(Product[] products) {
44         this.products = products;
45     }
46     public double calculateTotalPrice(){
47         double totalPrice=0;
48         for(int i=0;i<products.length;i++){
49             totalPrice += products[i].getPrice();
50
51         }
52         return totalPrice;
53     }
54     public void productInfo(Product[] p){
55         for(int i=0;i<products.length;i++){
56             System.out.println("Product [" + (i+1) + "] Name : "+ p[i].getName() + "\n" + "Product [" + (i+1) + "] Price : "+ p[i].getPrice() + "\n");
57         }
58     }
59     public void printOrderInfo(){
60         System.out.println("The Order Id : " + orderId);
61         System.out.println("The Customer Id : " + customerId);
62         productInfo(p: products);
63
64         double price = calculateTotalPrice();
65         System.out.println("Total price: " + price );
66
67
68 }
```

The screenshot shows a Java IDE interface with the following details:

- Toolbar:** Includes icons for file operations (New, Open, Save, etc.), search, and system status (Memory: 384.7/490.0MB).
- Project Explorer:** Shows files like ElectronicProduct.java, BookProduct.java, ClothingProduct.java, Customer.java, Cart.java, Order.java, EcommerceSystem.java, and Assiment... (partially visible).
- Source Editor:** Displays the code for `EcommerceSystem`. The code initializes three products (ElectronicProduct e1, ClothingProduct c1, BookProduct b1) and creates a Customer object. It then prompts the user to enter their ID, name, and address. The user is asked how many products they want to add to their cart. A switch statement handles three cases: adding e1, c1, or b1 to the cart. The code uses System.out.println for output and Scanner for input.
- Output Window:** Shows two tabs: "assiment (run)" and "assiment (run) #2".
- Bottom Status Bar:** Shows the current run configuration ("assiment (run)"), its status ("running..."), the build number (62:64), and the terminal type (INS Unix (LF)).

```
1 package assiment;
2 import java.util.Scanner;
3 public class EcommerceSystem {
4
5     public static void main(String[] args) {
6         Scanner input=new Scanner(source: System.in);
7         ElectronicProduct e1 = new ElectronicProduct(productId: 1, name: "smartphone", price: 599.9, brand: "samsung", warrantyPeriod: 2);
8         ClothingProduct c1 = new ClothingProduct(productId: 2, name: "Tshirt", price: 19.99, size: "medium", fabric: "cotton");
9         BookProduct b1 = new BookProduct(productId: 3, name: "OOP", price: 39.99, author: "OReilly", publisher: "X Publications");
10
11         System.out.println(x: "Welcome to the E-commerce System!");
12         System.out.println(x: "Enter your Id");
13         int Id = input.nextInt();
14         System.out.println(x: "Enter your name");
15         String name = input.next();
16         System.out.println(x: "Enter your address");
17         String address= input.next();
18         Customer customer1= new Customer(customerId: Id, name, address);
19         System.out.println(x: "How many products do you want?");
20         int nProducts= input.nextInt();
21         Cart cart=new Cart(customerId: Id, nproducts: nProducts);
22         for(int i=0;i<nProducts;i++){
23             System.out.println(x: "which products you want to add to your cart ?");
24             System.out.println("1-smartphone"+`\n2-Tshirt"+`\n3-OOP");
25             int number=input.nextInt();
26             switch(number){
27                 case 1:
28                     cart.addProduct(product: e1);
29                     System.out.println(x: "-----");
30                     break;
31                 case 2:
32                     cart.addProduct(product: c1);
33                     System.out.println(x: "-----");
34                     break;
35                 case 3:
36                     cart.addProduct(product: b1);
```

The screenshot shows a Java IDE interface with the following details:

- Top Bar:** Includes icons for file operations (New, Open, Save, etc.), a search bar (<default config>), memory usage (329.5/490.0MB), and a search bar (Search (⌘+I)).
- Toolbar:** Contains icons for Undo, Redo, Cut, Copy, Paste, Find, Replace, and other common operations.
- Project Explorer (Projects):** Shows a single project named "Assiment".
- Code Editor:** Displays Java code for an E-commerce system. The code includes imports for `java.util.*` and `java.io.*`, and classes `ElectronicProduct.java`, `BookProduct.java`, `ClothingProduct.java`, `Customer.java`, `Cart.java`, and `Order.java`. The main logic involves adding products to a cart, calculating the total price, and prompting the user to place an order (Yes or No). If Yes, it places the order and prints a summary.
- Output Window (Output):** Shows two tabs: "assiment (run)" and "assiment (run) #2".
- Bottom Status Bar:** Shows the current run configuration ("assiment (run)"), the status ("running..."), and the time ("62:64").

```
32     cart.addProduct(product:c1);
33     System.out.println(x: "-----");
34     break;
35   case 3:
36     cart.addProduct(product:b1);
37     System.out.println(x: "-----");
38     break;
39
40   default:
41     System.out.println(x: "the number is invalid");
42
43 }
44
45 System.out.println("Your total is : "+cart.calculatePrice());
46 System.out.println("would you like to place your order"+ "\n1- Yes 2- No");
47 int order= input.nextInt();
48
49 int orderId=0;
50 switch(order){
51   case 1:
52     cart.place_order(orderId);
53     System.out.println(x: "-----");
54     orderId++;
55     break;
56   case 2:
57     break;
58   default:
59     System.out.println(x: "the number is invalid");
60 }
61 Order o1 = new Order(customerId: Id,orderId,products: cart.getProducts());
62 System.out.println(x: "Here is a summary of your order : |");
63 o1.printOrderInfo();
64
65 }
66
67 }
```

<default config> 188.5/490.0MB Search (⌘+I)

...va ElectronicProduct.java x BookProduct.java x ClothingProduct.java x Customer.java x Cart.java x Order.java x EcommerceSystem.java x Assiment... < > □

Source History | cart.addProduct(product,c1);

Output x assiment (run) x assiment (run) #2 x

```
run:  
Welcome to the E-commerce System!  
Enter your Id  
20231  
Enter your name  
Nour  
Enter your address  
address  
How many products do you want?  
4  
which products you want to add to your cart ?  
1-smartphone  
2-Tshirt  
3-00P  
2  
-----  
which products you want to add to your cart ?  
1-smartphone  
2-Tshirt  
3-00P  
3  
-----  
which products you want to add to your cart ?  
1-smartphone  
2-Tshirt  
3-00P  
2  
-----  
which products you want to add to your cart ?  
1-smartphone  
2-Tshirt  
3-00P  
1  
-----  
Your total is : 679.87
```

assiment (run) running... x 62:64 INS Unix (LF)

<default config> 199.3/490.0MB Search (⌘+I)

...va ElectronicProduct.java x BookProduct.java x ClothingProduct.java x Customer.java x Cart.java x Order.java x EcommerceSystem.java x Assiment... < > □

Source History □ cart_addProduct(product,c1).

Output x assiment (run) x assiment (run) #2 x

```
which products you want to add to your cart ?
1-smartphone
2-Tshirt
3-00P
2

which products you want to add to your cart ?
1-smartphone
2-Tshirt
3-00P
1

Your total is : 679.87
would you like to place your order
1- Yes 2- No
1

Here is a summary of your order :
The Order Id : 1
The Customer Id : 20231
Product [1] Name : Tshirt
Product [1] Price : 19.99

Product [2] Name : 00P
Product [2] Price : 39.99

Product [3] Name : Tshirt
Product [3] Price : 19.99

Product [4] Name : smartphone
Product [4] Price : 599.9

Total price: 679.87
BUILD SUCCESSFUL (total time: 24 seconds)
```

assiment (run) running... x 62:64 INS | Unix (LF)