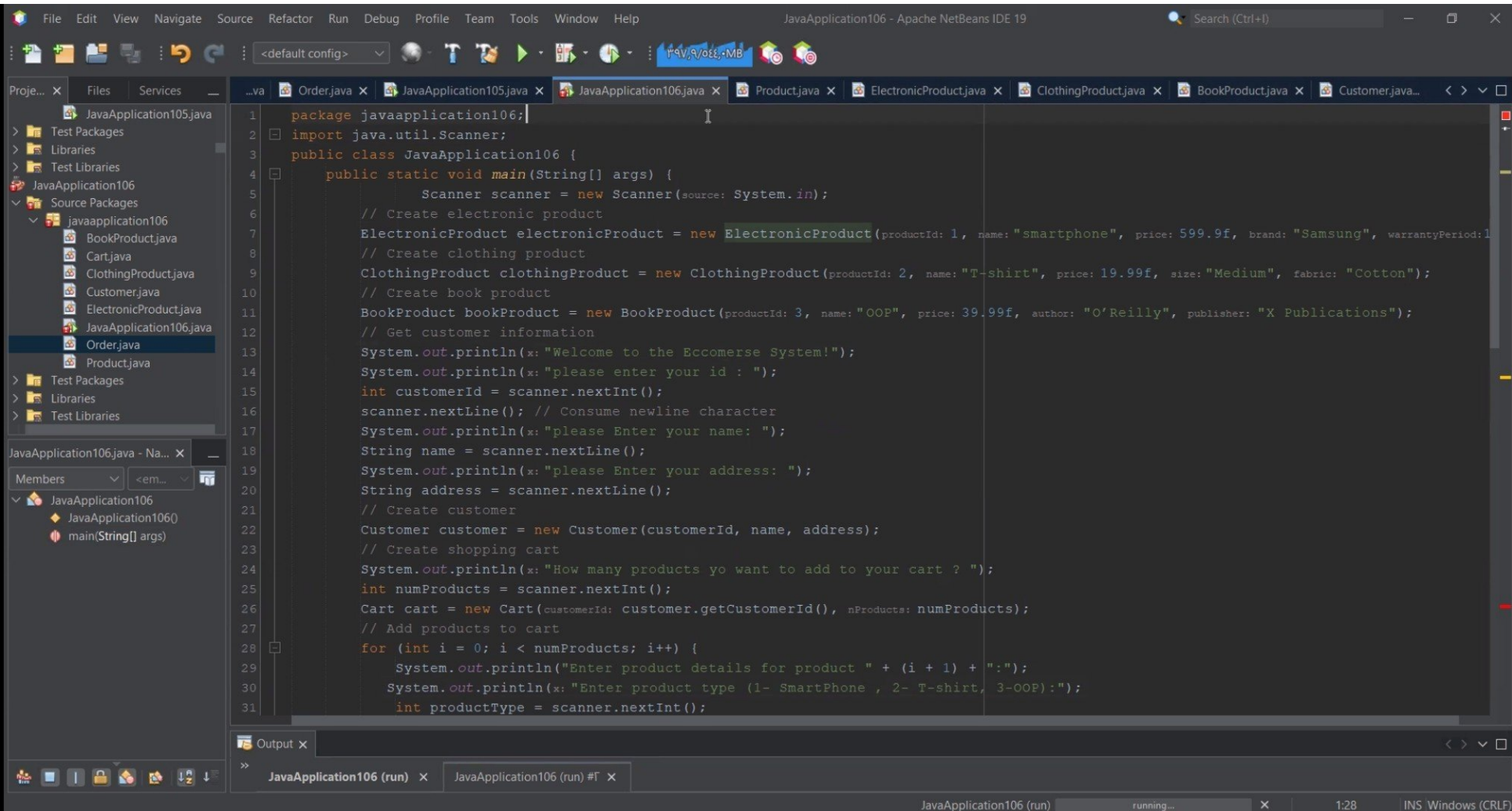


Name : jana wael gaber  
ID : 23012089



The screenshot displays the Apache NetBeans IDE interface. The main editor window shows the source code for `JavaApplication106.java`. The code is as follows:

```
1 package javaapplication106;
2 import java.util.Scanner;
3 public class JavaApplication106 {
4     public static void main(String[] args) {
5         Scanner scanner = new Scanner(System.in);
6         // Create electronic product
7         ElectronicProduct electronicProduct = new ElectronicProduct(productId: 1, name: "smartphone", price: 599.9f, brand: "Samsung", warrantyPeriod: 1);
8         // Create clothing product
9         ClothingProduct clothingProduct = new ClothingProduct(productId: 2, name: "T-Shirt", price: 19.99f, size: "Medium", fabric: "Cotton");
10        // Create book product
11        BookProduct bookProduct = new BookProduct(productId: 3, name: "OOP", price: 39.99f, author: "O'Reilly", publisher: "X Publications");
12        // Get customer information
13        System.out.println(x: "Welcome to the Eccomerse System!");
14        System.out.println(x: "please enter your id : ");
15        int customerId = scanner.nextInt();
16        scanner.nextLine(); // Consume newline character
17        System.out.println(x: "please Enter your name: ");
18        String name = scanner.nextLine();
19        System.out.println(x: "please Enter your address: ");
20        String address = scanner.nextLine();
21        // Create customer
22        Customer customer = new Customer(customerId, name, address);
23        // Create shopping cart
24        System.out.println(x: "How many products yo want to add to your cart ? ");
25        int numProducts = scanner.nextInt();
26        Cart cart = new Cart(customerId: customer.getCustomerId(), nProducts: numProducts);
27        // Add products to cart
28        for (int i = 0; i < numProducts; i++) {
29            System.out.println("Enter product details for product " + (i + 1) + ":");
30            System.out.println(x: "Enter product type (1- SmartPhone , 2- T-shirt, 3-OOP):");
31            int productType = scanner.nextInt();
```

The IDE's left sidebar shows the project structure for `JavaApplication106`, including source packages and individual Java files. The bottom status bar indicates the application is running.

Main

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help JavaApplication106 - Apache NetBeans IDE 19 Search (Ctrl+I)

JavaApplication105.java Order.java JavaApplication105.java JavaApplication106.java Product.java ElectronicProduct.java ClothingProduct.java BookProduct.java Customer.java...

```
31 int productType = scanner.nextInt();
32 //scanner.nextLine(); // Consume newline character
33 Product product;
34 switch (productType) {
35     case 1:
36         product = electronicProduct;
37         break;
38     case 2:
39         product = clothingProduct;
40         break;
41     case 3:
42         product = bookProduct;
43         break;
44     default:
45         System.out.println(x: "Invalid product type. Defaulting to electronic product.");
46         product = electronicProduct;
47 }
48 cart.addProduct(product, index: i);
49 }
50 // Ask user if they want to place order
51 System.out.println(x: "Do you want to place the order? (yes/no)");
52 String choice = scanner.next();
53 if (choice.equalsIgnoreCase(anotherString: "yes")) {
54     // Place order
55     Order order = cart.placeOrder();
56     // Print order info
57     System.out.println(x: "Here's your order summary : ");
58     order.printOrderInfo();
59 } else {
60     System.out.println(x: "Order not placed. Exiting...");
61 }
```

Output x

JavaApplication106 (run) x JavaApplication106 (run) #1 x

JavaApplication106 (run) running... 64:1 INS Windows (CRLF)

# Product Class

The screenshot displays the Apache NetBeans IDE interface. The main editor window shows the `Product.java` file with the following code:

```
1 package javaapplication106;
2 public class Product {
3     private int productId;
4     private String name;
5     private float price;
6
7     public Product(int productId, String name, float 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 = productId;
17    }
18    public String getName() {
19        return name;
20    }
21    public void setName(String name) {
22        this.name = name;
23    }
24    public float getPrice() {
25        return price;
26    }
27    public void setPrice(float price) {
28        this.price = price;
29    }
30 }
31
```

The left sidebar shows the Project Explorer with the following structure:

- JavaApplication105.java
- Test Packages
- Libraries
- Test Libraries
- JavaApplication106
  - Source Packages
    - javaapplication106
      - BookProduct.java
      - Cart.java
      - ClothingProduct.java
      - Customer.java
      - ElectronicProduct.java
      - JavaApplication106.java
      - Order.java
      - Product.java
- Test Packages
- Libraries
- Test Libraries

The bottom status bar shows the application is running: `JavaApplication106 (run)` and `JavaApplication106 (run) #1`.

# Electronic Product Class

The screenshot displays the Apache NetBeans IDE interface. The main editor window shows the `ElectronicProduct.java` file, which is a Java class extending the `Product` class. The code includes private fields for `brand` and `warrantyPeriod`, a constructor, and several getter and setter methods. The `setWarrantyPeriod` method uses `Math.abs` to ensure the warranty period is non-negative.

```
1 package javaapplication106;
2 public class ElectronicProduct extends Product {
3     private String brand;
4     private int warrantyPeriod;
5
6     public ElectronicProduct(int productId, String name, float price, String brand, int warrantyPeriod) {
7         super(productId, name, price);
8         this.brand = brand;
9         this.warrantyPeriod = Math.abs(warrantyPeriod);
10    }
11
12    // Setters and getters
13    public String getBrand() {
14        return brand;
15    }
16
17    public void setBrand(String brand) {
18        this.brand = brand;
19    }
20
21    public int getWarrantyPeriod() {
22        return warrantyPeriod;
23    }
24
25    public void setWarrantyPeriod(int warrantyPeriod) {
26        this.warrantyPeriod = Math.abs(warrantyPeriod);
27    }
28 }
29
```

The left sidebar shows the project structure with `ElectronicProduct.java` selected under the `javaapplication106` package. The bottom status bar indicates the application is running.



# Clothing Product Class

The screenshot displays the Apache NetBeans IDE interface. The main editor window shows the `ClothingProduct.java` file, which is part of the `javaapplication106` package. The code defines a `ClothingProduct` class that extends the `Product` class. The class has two private attributes: `String size` and `String fabric`. The constructor `ClothingProduct(int productId, String name, float price, String size, String fabric)` calls `super(productId, name, price)` and initializes `this.size` and `this.fabric`. The class also includes four methods: `getSize()` (returns `size`), `setSize(String size)` (sets `this.size`), `getFabric()` (returns `fabric`), and `setFabric(String fabric)` (sets `this.fabric`).

The left sidebar shows the Project Explorer with the following structure:

- JavaApplication105.java
- Test Packages
- Libraries
- Test Libraries
- JavaApplication106
  - Source Packages
    - javaapplication106
      - BookProduct.java
      - Cart.java
      - ClothingProduct.java
      - Customer.java
      - ElectronicProduct.java
      - JavaApplication106.java
      - Order.java
      - Product.java
  - Test Packages
  - Libraries
  - Test Libraries

The bottom status bar indicates that the application is running: `JavaApplication106 (run) running...` at 1:27.

# Book Product Class

The screenshot displays the Apache NetBeans IDE interface. The main editor window shows the `BookProduct.java` file, which is part of the `javaapplication106` package. The code defines a `BookProduct` class that extends the `Product` class. It includes private attributes `author` and `publisher`, a constructor, and getter/setter methods for these attributes.

```
1 package javaapplication106;
2 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    // Setters and getters
13    public String getAuthor() {
14        return author;
15    }
16
17    public void setAuthor(String author) {
18        this.author = author;
19    }
20
21    public String getPublisher() {
22        return publisher;
23    }
24
25    public void setPublisher(String publisher) {
26        this.publisher = publisher;
27    }
28 }
29
30
```

The left sidebar shows the project structure, with `BookProduct.java` selected under the `javaapplication106` package. The bottom status bar indicates that the application is running.

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help JavaApplication106 - Apache NetBeans IDE 19 Search (Ctrl+I)

JavaApplication105.java x JavaApplication106.java x Product.java x ElectronicProduct.java x ClothingProduct.java x BookProduct.java x Customer.java x Cart.java...

Test Packages Libraries Test Libraries JavaApplication106 Source Packages javaapplication106 BookProduct.java Cart.java ClothingProduct.java Customer.java ElectronicProduct.java JavaApplication106.java Order.java Product.java Test Packages Libraries Test Libraries

Members <em...>

Customer

- Customer(int customerId, String name, String address)
- getAddress() : String
- getCustomerId() : int
- getName() : String
- setAddress(String address)
- setCustomerId(int customerId)
- setName(String name)
- address : String
- customerId : int
- name : String

```
1 package javaapplication106;
2 class Customer {
3     private int customerId;
4     private String name;
5     private String address;
6     public Customer(int customerId, String name, String address) {
7         this.customerId = Math.abs(a: customerId);
8         this.name = name;
9         this.address = address;
10    }
11    // Setters and getters
12    public int getCustomerId() {
13        return customerId;
14    }
15    public void setCustomerId(int customerId) {
16        this.customerId = Math.abs(a: customerId);
17    }
18    public String getName() {
19        return name;
20    }
21    public void setName(String name) {
22        this.name = name;
23    }
24    public String getAddress() {
25        return address;
26    }
27    public void setAddress(String address) {
28        this.address = address;
29    }
30 }
31
```

Output x

JavaApplication106 (run) x JavaApplication106 (run) #1 x

JavaApplication106 (run) running... 32:1 INS Windows (CRLF)

# Cart Class

The screenshot displays the Apache NetBeans IDE interface with the following components:

- Project Explorer (Left):** Shows the project structure for 'JavaApplication106'. The 'Source Packages' folder is expanded, revealing the 'javaapplication106' package which contains several Java files, including 'Cart.java'.
- Members View (Bottom Left):** Displays the members of the 'Cart' class, including the constructor 'Cart(int customerId, int nProducts)' and methods like 'addProduct', 'calculatePrice', 'getCustomerId', 'getProducts', 'getnProducts', 'placeOrder', 'removeProduct', 'setCustomerId', 'setProducts', and 'setnProducts'.
- Code Editor (Center):** Shows the implementation of the 'Cart' class in 'Cart.java'. The code includes:
  - `addProduct`: Adds a product to the 'products' array at a specific index, with a check for valid index range.
  - `removeProduct`: Removes a product from the 'products' array at a specific index, with a check for valid index range.
  - `calculatePrice`: Iterates through the 'products' array, summing up the prices of all non-null products.
  - `placeOrder`: Returns a new 'Order' object, passing the 'customerId', 'nProducts', 'products', and the result of 'calculatePrice()'.
- Output Window (Bottom):** Shows the execution of 'JavaApplication106 (run)'.



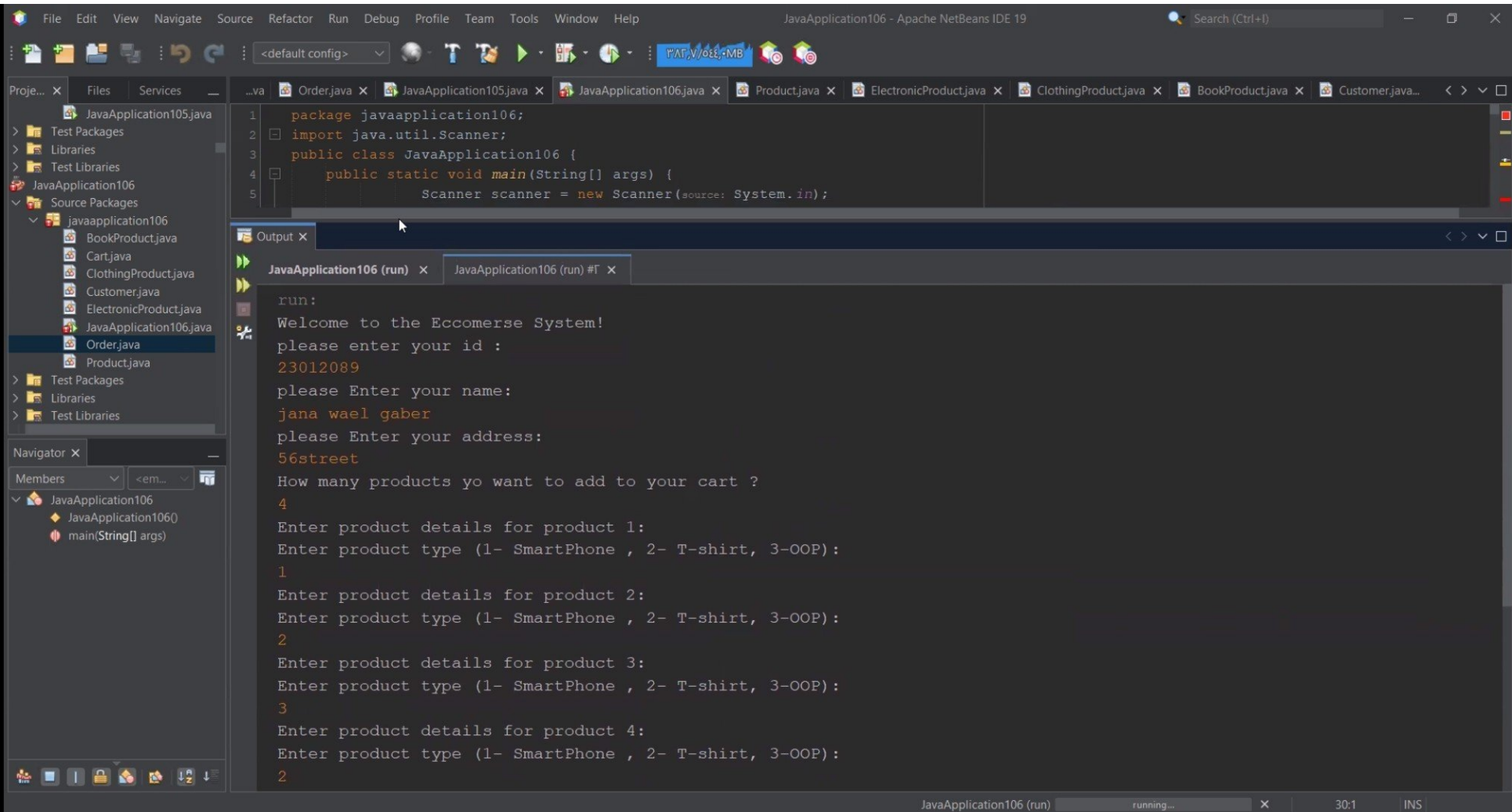
# Order Class

The screenshot displays the Apache NetBeans IDE interface. The main editor window shows the `Order.java` file, which is part of the `javaapplication106` package. The class `Order` contains the following code:

```
1 package javaapplication106;
2 class Order {
3     private int customerId;
4     private int orderId;
5     private Product[] products;
6     private float totalPrice;
7
8     public Order(int customerId, int orderId, Product[] products, float totalPrice) {
9         this.customerId = Math.abs(a: customerId);
10        this.orderId = Math.abs(a: orderId);
11        this.products = products;
12        this.totalPrice = Math.abs(a: totalPrice);
13    }
14
15    // Print order information
16    public void printOrderInfo() {
17        System.out.println("Order ID: " + orderId);
18        System.out.println("Customer ID: " + customerId);
19        System.out.println("Products:");
20
21        for (int i = 0 ; i < products.length ; i++) {
22            if (products[i] != null) {
23                System.out.println("- " + products[i].getName() + ": $" + products[i].getPrice());
24            }
25        }
26
27        System.out.println("Total Price: $" + totalPrice);
28    }
29 }
30
```

The left sidebar shows the project structure, including the `Order.java` file. The bottom status bar indicates the application is running.

# Output



The screenshot displays the Apache NetBeans IDE interface. The top menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, and Help. The toolbar contains various icons for file operations, running, and debugging. The left sidebar shows the Project Explorer with a tree view of the project structure, including packages like Test Packages, Libraries, and Source Packages. The main editor window displays the source code for `JavaApplication106.java`, which is a Java application using `java.util.Scanner` to collect user input. The code is as follows:

```
1 package javaapplication106;
2 import java.util.Scanner;
3 public class JavaApplication106 {
4     public static void main(String[] args) {
5         Scanner scanner = new Scanner(System.in);
```

Below the editor, the Output window is open, showing the execution output for `JavaApplication106 (run)`. The output text is:

```
run:
Welcome to the Eccomerse System!
please enter your id :
23012089
please Enter your name:
jana wael gaber
please Enter your address:
56street
How many products yo want to add to your cart ?
4
Enter product details for product 1:
Enter product type (1- SmartPhone , 2- T-shirt, 3-OOP):
1
Enter product details for product 2:
Enter product type (1- SmartPhone , 2- T-shirt, 3-OOP):
2
Enter product details for product 3:
Enter product type (1- SmartPhone , 2- T-shirt, 3-OOP):
3
Enter product details for product 4:
Enter product type (1- SmartPhone , 2- T-shirt, 3-OOP):
2
```

The status bar at the bottom indicates that `JavaApplication106 (run)` is running, with a progress bar and a time display of 30:1.

JavaApplication106 - Apache NetBeans IDE 19

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl+I)

Project: JavaApplication105.java  
Test Packages  
Libraries  
Test Libraries  
JavaApplication106  
Source Packages  
javaapplication106  
BookProduct.java  
Cart.java  
ClothingProduct.java  
Customer.java  
ElectronicProduct.java  
JavaApplication106.java  
Order.java  
Product.java  
Test Packages  
Libraries  
Test Libraries

Navigator  
Members  
JavaApplication106  
JavaApplication106()  
main(String[] args)

```
1 package javaapplication106;  
2 import java.util.Scanner;  
3 public class JavaApplication106 {  
4     public static void main(String[] args) {  
5         Scanner scanner = new Scanner(System.in);
```

Output

JavaApplication106 (run) x JavaApplication106 (run) #1 x

```
Enter product details for product 2:  
Enter product type (1- SmartPhone , 2- T-shirt, 3-OOP):  
2  
Enter product details for product 3:  
Enter product type (1- SmartPhone , 2- T-shirt, 3-OOP):  
3  
Enter product details for product 4:  
Enter product type (1- SmartPhone , 2- T-shirt, 3-OOP):  
2  
Do you want to place the order? (yes/no)  
yes  
Here's your order summary :  
Order ID: 4  
Customer ID: 23012089  
Products:  
- smartphone: $599.9  
- T-shirt: $19.99  
- OOP: $39.99  
- T-shirt: $19.99  
Total Price: $679.87  
BUILD SUCCESSFUL (total time: 1 minutes 0 seconds)
```

JavaApplication106 (run) running... 30:1 INS