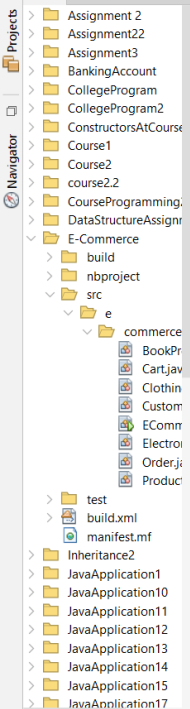




File... X Servic... Favi...



Start Page X ECommerce.java X Product.java X ElectronicProduct.java X ClothingProduct.java X BookProduct.java X Customer.java X Cart.java X Order.java X

Source History

```
5 package e.commerce;
6 import java.util.Scanner;
7 /**
8  *
9  * @author Mohamed khairy
10 */
11 public class ECommerce {
12
13     /**
14      * @param args the command line arguments
15      */
16     public static void main(String[] args) {
17         ElectronicProduct smartphone=new ElectronicProduct(WarrantyPeriod: 1,brand: "Smartphone",productPrice: 599.99f,productName: "Samsung",ProductID:1);
18         ClothingProduct tShirt=new ClothingProduct(size: "Medium",fabric:"Cotton",ProductID:2,productName: "T-Shirt",productPrice: 19.99f);
19         BookProduct oop=new BookProduct(author:"O'Reilly",publisher:"X Publications",ProductID:3,productName: "OOP",productPrice: 39.99f);
20
21         Scanner scanner=new Scanner(source:System.in);
22         System.out.println(x: "Welcome to the E-Commerce system.");
23         System.out.println(x: "Enter your ID:");
24         int CustomerID=scanner.nextInt();
25         scanner.nextLine();
26         System.out.println(x: "Enter your name:");
27         String customerName=scanner.nextLine();
28         System.out.println(x: "Enter your address:");
29         String address=scanner.nextLine();
30         Customer customer = new Customer(CustomerID,customerName,address);
31
32         System.out.println(x: "Total items to add to your cart:");
33         int Nproducts=scanner.nextInt();
34         scanner.nextLine();
35         Cart cart=new Cart(ID: customer.getCustomerID(),Nproducts);
36
37         for(int i=0;i<Nproducts;i++){
38             System.out.println(x: "Which product do you want?1)Smartphone 2)T-Shirt 3)OOP");
39             int choice=scanner.nextInt();
40             scanner.nextLine();
41             switch(choice){
```

Search (Ctrl+I)

<default config>

410.9/504.0MB

Files Services Favorites

Projects

Navigator

- Assignment 2
- Assignment22
- Assignment3
- BankingAccount
- CollegeProgram
- CollegeProgram2
- ConstructorsAtCourse
- Course1
- Course2
- course2.2
- CourseProgramming22
- DataStructureAssignment
- E-Commerce
 - build
 - nbproject
 - src
 - e
 - commerce
 - BookProduct.java
 - Cart.java
 - ClothingProduct.java
 - Customer.java
 - ECommerce.java
 - ElectronicProduct.java
 - Order.java
 - Product.java
 - test
 - build.xml
 - manifest.mf
- Inheritance2
- JavaApplication1
- JavaApplication10
- JavaApplication11
- JavaApplication12
- JavaApplication13
- JavaApplication14
- JavaApplication15
- JavaApplication17

Source History

```

1 public class Product {
12     int ProductID;
13     String productName;
14     float productPrice;
15
16     public Product(int ProductID, String productName, float productPrice) {
17         this.ProductID = ProductID;
18         this.productName = productName;
19         this.productPrice = productPrice;
20     }
21
22     public int getProductID() {
23         return ProductID;
24     }
25
26     public void setProductID(int ProductID) {
27         this.ProductID = ProductID;
28     }
29
30     public String getProductName() {
31         return productName;
32     }
33
34     public void setProductName(String productName) {
35         this.productName = productName;
36     }
37
38     public float getProductPrice() {
39         return productPrice;
40     }
41
42     public void setProductPrice(float productPrice) {
43         this.productPrice = productPrice;
44     }
45
46 }
47

```

Profiling Points Output

45:6 INS Windows (CRLF)

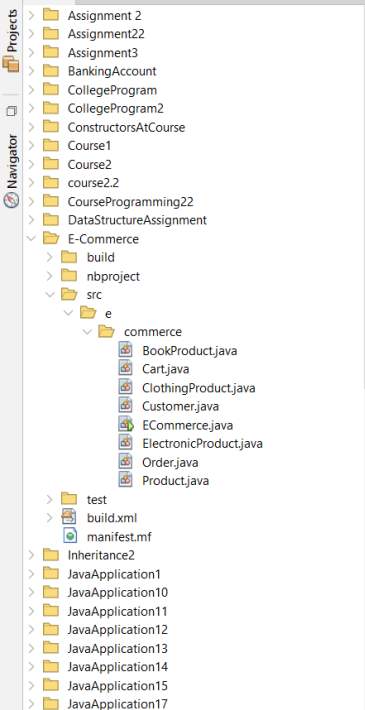
The screenshot displays the Apache NetBeans IDE interface. On the left, the **Projects** and **Navigator** panels show the project structure. The **Projects** panel lists various projects, including Assignment 2, Assignment22, Assignment3, BankingAccount, CollegeProgram, CollegeProgram2, ConstructorsAtCourse, Course1, Course2, course2.2, CourseProgramming22, DataStructureAssignment, and E-Commerce. The **Navigator** panel shows the **E-Commerce** project structure, including **build**, **nbproject**, **src**, and **test** folders. The **src** folder contains the **e** package, which includes the **commerce** package. The **commerce** package contains the following files: BookProduct.java, Cart.java, ClothingProduct.java, Customer.java, ECommerce.java, ElectronicProduct.java, Order.java, and Product.java.

The main editor window displays the **ClothingProduct.java** file. The code is as follows:

```
7  /**
8  *
9  * @author Mohamed khairy
10 * */
11 public class ClothingProduct extends Product {
12     String size;
13     String fabric;
14
15     public ClothingProduct(String size, String fabric, int ProductID, String productName, float productPrice) {
16         super(ProductID, productName, productPrice);
17         this.size = size;
18         this.fabric = fabric;
19     }
20
21     public String getSize() {
22         return size;
23     }
24
25     public void setSize(String size) {
26         this.size = size;
27     }
28
29     public String getFabric() {
30         return fabric;
31     }
32
33     public void setFabric(String fabric) {
34         this.fabric = fabric;
35     }
36
37 }
38
39 }
```



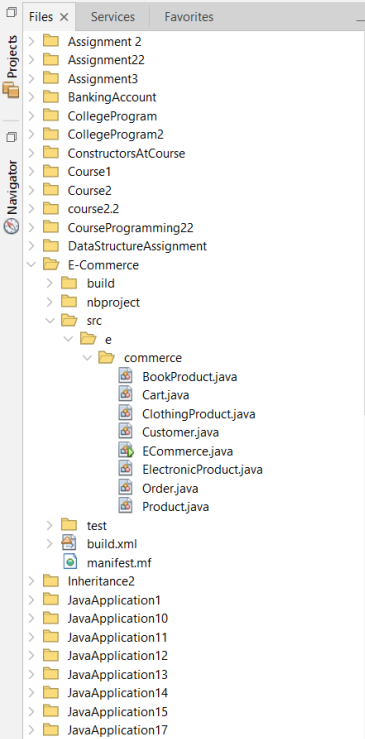
Files Services Favorites



ECommerce.java Product.java ElectronicProduct.java ClothingProduct.java BookProduct.java Customer.java Cart.java Order.java

Source

```
11 public class Customer {
12     int CustomerID;
13     String customerName;
14     String address;
15
16     public Customer(int CustomerID, String customerName, String address) {
17         this.CustomerID = CustomerID;
18         this.customerName = customerName;
19         this.address = address;
20     }
21
22     public int getCustomerID() {
23         return CustomerID;
24     }
25
26     public void setCustomerID(int CustomerID) {
27         this.CustomerID = CustomerID;
28     }
29
30     public String getCustomerName() {
31         return customerName;
32     }
33
34     public void setCustomerName(String customerName) {
35         this.customerName = customerName;
36     }
37
38     public String getAddress() {
39         return address;
40     }
41
42     public void setAddress(String address) {
43         this.address = address;
44     }
45
46 }
47
```



```
Source History
1  /*
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4   */
5   package e.commerce;
6
7   /**
8    *
9    * @author Mohamed khairy
10   */
11   public class BookProduct extends Product {
12       String author;
13       String publisher;
14
15       public BookProduct(String author, String publisher, int ProductID, String productName, float productPrice) {
16           super(ProductID, productName, productPrice);
17           this.author = author;
18           this.publisher = publisher;
19       }
20
21       public String getAuthor() {
22           return author;
23       }
24
25       public void setAuthor(String author) {
26           this.author = author;
27       }
28
29       public String getPublisher() {
30           return publisher;
31       }
32
33       public void setPublisher(String publisher) {
34           this.publisher = publisher;
35       }
36
37   }
```



Projects

Services

Favorites

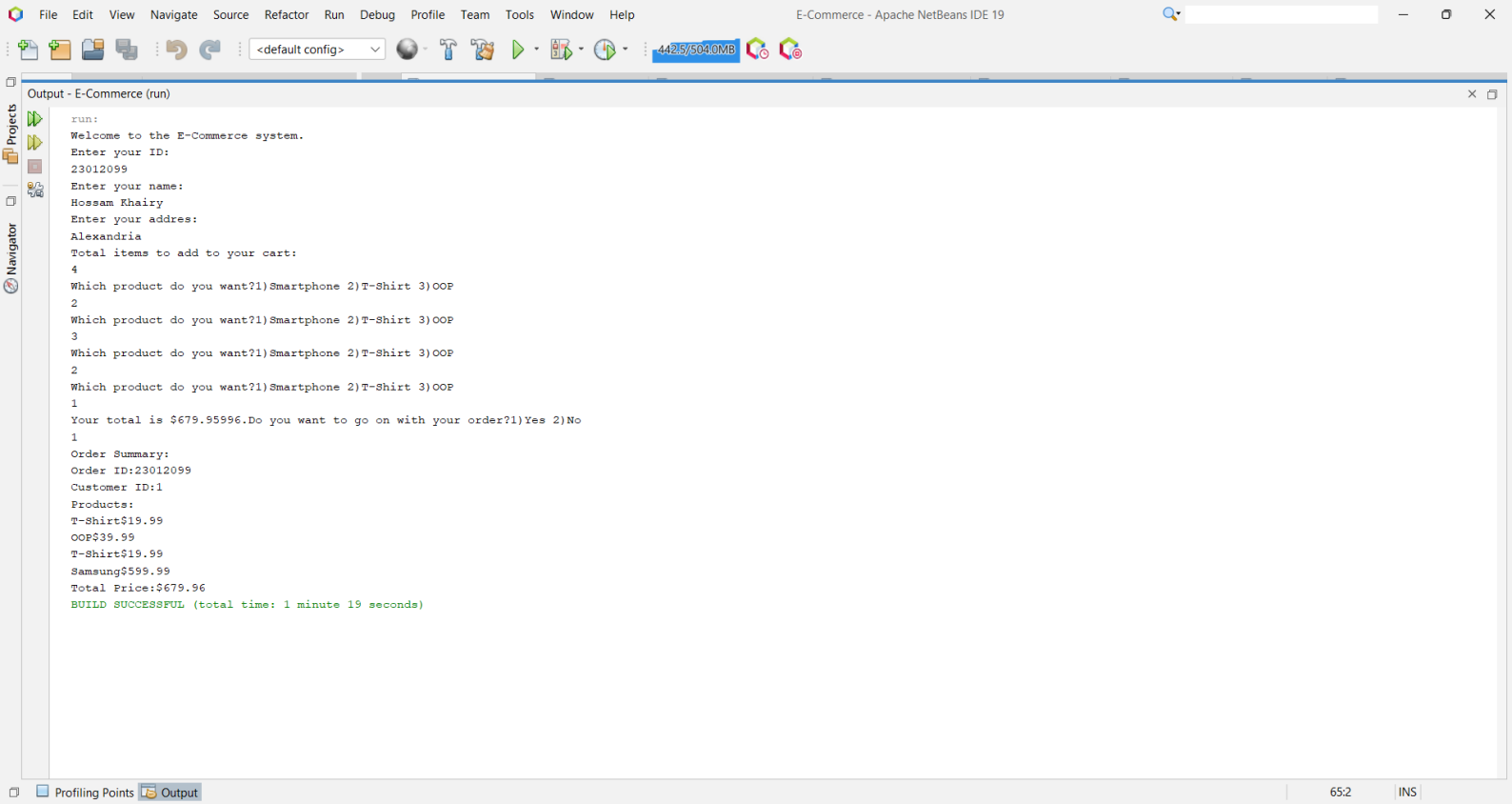
- Assignment 2
- Assignment22
- Assignment3
- BankingAccount
- CollegeProgram
- CollegeProgram2
- ConstructorsAtCourse
- Course1
- Course2
- course2.2
- CourseProgramming22
- DataStructureAssignment
- E-Commerce
 - build
 - nbproject
 - src
 - e
 - commerce
 - BookProduct.java
 - Cart.java
 - ClothingProduct.java
 - Customer.java
 - ECommerce.java
 - ElectronicProduct.java
 - Order.java
 - Product.java
 - test
 - build.xml
 - manifest.mf
- Inheritance2
- JavaApplication1
- JavaApplication10
- JavaApplication11
- JavaApplication12
- JavaApplication13
- JavaApplication14
- JavaApplication15
- JavaApplication17

Source

History

388.9/504.0MB

```
10 //
11 public class Cart {
12     int ID;
13     int Nproducts;
14     Product[] products;
15     int orderID=1;
16
17     public Cart(int ID, int Nproducts) {
18         this.ID = Math.abs(a: ID);
19         this.Nproducts = Math.abs(a: Nproducts);
20         this.products=new Product[Nproducts];
21     }
22     public float priceOfProducts(){
23         float sum=0.0f;
24         for(Product product:products){
25             if(product!=null){
26                 sum+=product.getProductPrice();
27             }
28         }
29         return sum;
30     }
31     public void addProduct(Product product){
32         for(int i=0;i<Nproducts;i++){
33             if(products[i]==null){
34                 products[i]=product;
35                 return;
36             }
37         }
38         System.out.println("Cart is full.");
39     }
40     public void OrderPlacement(Customer customer){
41         float sum=priceOfProducts();
42         Order order=new Order(orderID++,orderID: customer.getCustomerID(),Nproducts:this.Nproducts,products: this.products,sum: sum);
43         order.printInformation(customer);
44     }
45 }
46 }
```



Projects

Services

Favorites

- > Assignment 2
- > Assignment22
- > Assignment3
- > BankingAccount
- > CollegeProgram
- > CollegeProgram2
- > ConstructorsAtCourse
- > Course1
- > Course2
- > course2.2
- > CourseProgramming22
- > DataStructureAssignment
- > E-Commerce
 - > build
 - > nbproject
 - > src
 - > e
 - > commerce
 - BookProduct.java
 - Cart.java
 - ClothingProduct.java
 - Customer.java
 - ECommerce.java
 - ElectronicProduct.java
 - Order.java
 - Product.java
 - > test
 - > build.xml
 - > manifest.mf
- > Inheritance2
- > JavaApplication1
- > JavaApplication10
- > JavaApplication11
- > JavaApplication12
- > JavaApplication13
- > JavaApplication14
- > JavaApplication15
- > JavaApplication17

Source History

```

package e.commerce;

/**
 *
 * @author Mohamed khairy
 */
public class Order {
    private int iD;
    private int orderID;
    private float Sum;
    private Product[] products;

    public Order(int iD, int orderID, int Nproducts, Product[] products, float Sum) {
        this.iD = iD;
        this.orderID = orderID;
        this.Sum = Sum;
        this.products = products;
    }

    public void printInformation(Customer customer){
        System.out.println(x: "Order Summary:");
        System.out.println("Order ID:"+this.orderID);
        System.out.println("Customer ID:"+this.iD);
        System.out.println(x: "Products:");
        for(Product product:this.products){
            if(product!=null){
                System.out.println(product.getProductPrice()+product.getProductPrice());
            }
        }
        System.out.printf(format:"Total Price:$.2f%n",args: this.Sum);
    }
}
    
```


421.8/504.0MB

Start Page x ECommerce.java x Product.java x ElectronicProduct.java x ClothingProduct.java x BookProduct.java x Customer.java x Cart.java x Order.java x

Source History

```

37 for (int i=0;i<Nproducts;i++){
38     System.out.println(x: "Which product do you want?1)Smartphone 2)T-Shirt 3)OOP");
39     int choice=scanner.nextInt();
40     scanner.nextLine();
41     switch (choice) {
42     case 1:
43         cart.addProduct(product: smartphone);
44         break;
45     case 2:
46         cart.addProduct(product: tShirt);
47         break;
48     case 3:
49         cart.addProduct(product: oop);
50         break;
51     default:
52         System.out.println(x: "invalid choice.");
53         break;
54     }
55
56     System.out.printf(format:"Your total is $%.5f.Do you want to go on with your order?1)Yes 2)No%n",args: cart.priceOfProducts());
57     int orderChoice=scanner.nextInt();
58     scanner.nextLine();
59     if (orderChoice == 1) {
60         cart.OrderPlacement(customer);
61     }else{
62         System.out.println(x: "Order is canceled.");
63     }
64 }
65 }
66
67

```

Projects

- Assignment 2
- Assignment22
- Assignment3
- BankingAccount
- CollegeProgram
- CollegeProgram2
- ConstructorsAtCourse
- Course1
- Course2
- course2.2
- CourseProgramming.
- DataStructureAssign
- E-Commerce
 - build
 - nbproject
 - src
 - e
 - commerce
 - BookPr
 - Cartjav
 - Clothin
 - Custom
 - EComm
 - Electroi
 - Order.j
 - Product
 - test
 - build.xml
 - manifest.mf
- Inheritance2
- JavaApplication1
- JavaApplication10
- JavaApplication11
- JavaApplication12
- JavaApplication13
- JavaApplication14
- JavaApplication15
- JavaApplication17

Profiling Points Output

65:2 INS Windows (CRLF)

Search (Ctrl+I)

<default config>

449.9/504.0MB

Files Services Favorites

Projects

Navigator

- Assignment 2
- Assignment22
- Assignment3
- BankingAccount
- CollegeProgram
- CollegeProgram2
- ConstructorsAtCourse
- Course1
- Course2
- course2.2
- CourseProgramming22
- DataStructureAssignment
- E-Commerce
 - build
 - nbproject
 - src
 - e
 - commerce
 - BookProduct.java
 - Cart.java
 - ClothingProduct.java
 - Customer.java
 - ECommerce.java
 - ElectronicProduct.java
 - Order.java
 - Product.java
 - test
 - build.xml
 - manifest.mf
- Inheritance2
- JavaApplication1
- JavaApplication10
- JavaApplication11
- JavaApplication12
- JavaApplication13
- JavaApplication14
- JavaApplication15
- JavaApplication17

Source History

```

1  /*
2  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3  * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4  */
5  package e.commerce;
6
7  /**
8   *
9   * @author Mohamed khairy
10  */
11 public class ElectronicProduct extends Product {
12     String brand;
13     int WarrantyPeriod;
14
15     public ElectronicProduct(int WarrantyPeriod, String brand, float productPrice, String productName, int ProductID) {
16         super(ProductID, productName, productPrice);
17         this.brand = brand;
18         this.WarrantyPeriod = WarrantyPeriod;
19     }
20
21     public String getBrand() {
22         return brand;
23     }
24
25     public void setBrand(String brand) {
26         this.brand = brand;
27     }
28
29     public int getWarrantyPeriod() {
30         return WarrantyPeriod;
31     }
32
33     public void setWarrantyPeriod(int WarrantyPeriod) {
34         this.WarrantyPeriod = WarrantyPeriod;
35     }
36
37

```

Profiling Points Output

36:5 INS Windows (CRLF)