

```
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help JavaApplication6 - Apache NetBeans IDE 20
<default config> 138.1/229.0MB
JavaApplication6.java x Student.java x EcommerceSystem.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
1 package javaapplication6;
2 import java.util.Scanner;
3 public class EcommerceSystem {
4     public static void main(String[] args) {
5         Scanner input=new Scanner(System.in);
6
7         ElectronicProduct E1=new ElectronicProduct(1,"SmartPhone",599.99,"Samsung",1);
8         ClothingProduct C1=new ClothingProduct(2,"T-Shirt",19.99,"Medium","Cotton");
9         BookProduct B1=new BookProduct(3,"OOP",39.99,"O'Reilly","X Publications");
10
11         System.out.println("Welcome to the E-Commerce System!");
12
13         System.out.println("Please enter your id:");
14         int id=Integer.parseInt(input.nextLine());
15         System.out.println("Please enter your name:");
16         String name=input.nextLine();
17         System.out.println("Please enter your address:");
18         String address=input.nextLine();
19
20         Customer Cust1=new Customer(id,name,address);
21
22         System.out.println("How many products you want in your cart?");
23         int num=input.nextInt();
24
25         Cart Cal=new Cart(id,num);
26         Cal.setnproducts(num);
27         for(int i=0;i<num;i++){
28             System.out.println("Which product would you like to add? 1-SmartPhone 2-T-Shirt 3-OOP");
29             int choice=input.nextInt();
30             switch(choice){
31                 case 1:
32                     Cal.addProducts(E1,i);
33                     break;
34                 case 2:
35                     Cal.addProducts(C1,i);
36                     break;
37                 case 3:
38                     Cal.addProducts(B1,i);
39                     break;
40                 default:
41                     break;
42             }
43         }
44
45         int x=1;
46         Order O =new Order(id,num,x,(double)Cal.calculatePrice(),Cal.products);
47         System.out.println("Your Total is $"+Cal.calculatePrice()+" .Would you like to place the order? 1-Yes 2-No");
48         int YN=input.nextInt();
49
50         if(Cal.placeOrder(YN)==true){
51             O.printOrderInfo();
52             x++;
53         }
54     }
55 }
```

```
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
Output 5:46 INS Windows (CRLF)
```

The screenshot shows the NetBeans IDE with the `Product.java` file open. The code defines a `Product` class with attributes `productid`, `name`, and `price`. It includes a constructor and several getter and setter methods. The IDE interface includes a menu bar, a toolbar, and a project explorer on the left.

```
1 package javaapplication6;
2
3 public class Product {
4     int productid;
5     String name;
6     double price;
7
8     public Product() {
9     }
10
11     public Product(int productid, String name, double price) {
12         this.productid = Math.abs(productid);
13         this.name = name;
14         this.price = Math.abs(price);
15     }
16
17     public void setproductid(int productid) {
18         this.productid = Math.abs(productid);
19     }
20
21     public int getproductid() {
22         return productid;
23     }
24
25     public void setname(String name) {
26         this.name = name;
27     }
28
29     public String getname() {
30         return name;
31     }
32
33     public void setprice(double price) {
34         this.price = Math.abs(price);
35     }
36
37     public double getprice() {
38         return price;
39     }
40 }
```

The status bar at the bottom indicates line 352 and column INS.

The screenshot shows the NetBeans IDE with the `ElectronicProduct.java` file open. This class extends the `Product` class and adds attributes `brand` and `warrantyPeriod`. It includes a constructor and several getter and setter methods. The IDE interface is consistent with the previous screenshot.

```
1 package javaapplication6;
2
3 public class ElectronicProduct extends Product {
4     private String brand;
5     private int warrantyPeriod;
6
7     public ElectronicProduct() {
8     }
9
10    public ElectronicProduct(int productid, String name, double price, String brand, int warrantyPeriod) {
11        super(productid, name, price);
12        this.brand = brand;
13        this.warrantyPeriod = Math.abs(warrantyPeriod);
14    }
15
16    public void setbrand(String brand) {
17        this.brand = brand;
18    }
19
20    public String getbrand() {
21        return brand;
22    }
23
24    public void setwarrantyPeriod(int warrantyPeriod) {
25        this.warrantyPeriod = Math.abs(warrantyPeriod);
26    }
27
28    public int getwarrantyPeriod() {
29        return warrantyPeriod;
30    }
31 }
```

The status bar at the bottom indicates line 282 and column INS.

The screenshot shows the NetBeans IDE with the `ClothingProduct.java` file open. The code defines a `ClothingProduct` class that extends `Product`. It includes private attributes `size` and `fabric`, a no-argument constructor, a constructor taking `productId`, `name`, `price`, `size`, and `fabric`, and methods to set and get `size` and `fabric`.

```
1 package javaapplication6;
2
3 public class ClothingProduct extends Product {
4     private String size;
5     private String fabric;
6
7     public ClothingProduct() {
8     }
9
10    public ClothingProduct(int productId, String name, double price, String size, String fabric) {
11        super(productId, name, price);
12        this.size = size;
13        this.fabric = fabric;
14    }
15
16    public void setsize(String size) {
17        this.size = size;
18    }
19    public String getsize() {
20        return size;
21    }
22    public void setfabric(String fabric) {
23        this.fabric = fabric;
24    }
25    public String getfabric() {
26        return fabric;
27    }
28 }
```

The bottom status bar shows "282" and "INS".

The screenshot shows the NetBeans IDE with the `BookProduct.java` file open. The code defines a `BookProduct` class that extends `Product`. It includes private attributes `author` and `publisher`, a no-argument constructor, a constructor taking `productId`, `name`, `price`, `author`, and `publisher`, and methods to set and get `author` and `publisher`.

```
1 package javaapplication6;
2
3 public class BookProduct extends Product {
4     private String author;
5     private String publisher;
6
7     public BookProduct() {
8     }
9
10    public BookProduct(int productId, String name, double price, String author, String publisher) {
11        super(productId, name, price);
12        this.author = author;
13        this.publisher = publisher;
14    }
15
16    public void setauthor(String author) {
17        this.author = author;
18    }
19    public String getauthor() {
20        return author;
21    }
22    public void setpublisher(String publisher) {
23        this.publisher = publisher;
24    }
25    public String getpublisher() {
26        return publisher;
27    }
28 }
```

The bottom status bar shows "282" and "INS".

The screenshot shows the Apache NetBeans IDE with the 'Customer.java' file open. The code defines a 'Customer' class with attributes 'customerID', 'customername', and 'address'. It includes a no-argument constructor, a three-argument constructor, and setter/getter methods for each attribute. The 'customerID' setter and getter methods use 'Math.abs()' to ensure non-negative values.

```
package javaapplication6;

public class Customer {
    int customerID;
    private String customername;
    private String address;

    public Customer() {
    }

    public Customer(int customerID, String customername, String address) {
        this.customerID = Math.abs(customerID);
        this.customername = customername;
        this.address = address;
    }

    public void setcustomerID(int customerID) {
        this.customerID = Math.abs(customerID);
    }

    public int getcustomerID() {
        return customerID;
    }

    public void setcustomername(String customername) {
        this.customername = customername;
    }

    public String getcustomername() {
        return customername;
    }

    public void setaddress(String address) {
        this.address = address;
    }

    public String getaddress() {
        return address;
    }
}
```

The IDE interface includes a menu bar, a toolbar, a project explorer on the left, and an output window at the bottom.

The screenshot shows the Apache NetBeans IDE with the 'Cart.java' file open. The code defines a 'Cart' class with attributes 'customerID', 'nProducts', and 'products'. It includes a no-argument constructor, a two-argument constructor, and methods for setting/getting 'customerID' and 'nProducts', adding/removing products, and getting the products array.

```
package javaapplication6;

public class Cart {
    int customerID;
    int nProducts;
    Product products[];

    public Cart() {
    }

    public Cart(int customerID, int nProducts) {
        this.customerID = Math.abs(customerID);
        this.nProducts = Math.abs(nProducts);
    }

    public void setcustomerID(int customerID) {
        this.customerID = Math.abs(customerID);
    }

    public int getcustomerID() {
        return customerID;
    }

    public void setnProducts(int nProducts) {
        this.nProducts = Math.abs(nProducts);
        products = new Product[nProducts];
    }

    public int getnProducts() {
        return nProducts;
    }

    public void addProducts(Product product, int index) {
        if (index >= 0 && index < nProducts) {
            products[index] = product;
        }
    }

    public void removeProduct(int empty) {
        products[empty] = null;
    }
}
```

The IDE interface is consistent with the previous screenshot, showing the same menu, toolbar, and project explorer.

```
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59

public double calculatePrice() {
    double sum=0;
    for(int i=0;i<nProducts;i++){
        Product p=products[i];
        if(p!=null){
            sum+=p.getPrice();
        }
    }
    return sum;
}

public boolean placeOrder(int choice){
    boolean option =true;
    if(choice==1){
        option=true;
    }else if(choice==2){
        option=false;
    }
    return option;
}
```

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help JavaApplication6 - Apache NetBeans IDE 20 Search (Ctrl+F)

JavaApplication6.java x Student.java x EcommerceSystem.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

```
1 package javaapplication6;
2
3 public class Order {
4     int customerId;
5     int nProducts;
6     int orderId;
7     Product products[]=new Product[nProducts];
8     double totalPrice;
9
10    public Order() {
11    }
12
13    public Order(int customerId,int nProducts,int orderId,double totalPrice,Product[] products){
14        this.customerId=Math.abs(customerId);
15        this.nProducts=Math.abs(nProducts);
16        this.orderId=Math.abs(orderId);
17        this.totalPrice=Math.abs(totalPrice);
18        this.products=products;
19    }
20
21    public void printOrderInfo(){
22        System.out.println("Here's your order's summary:");
23        System.out.println("Order ID: "+orderId);
24        System.out.println("Customer ID: "+customerId);
25        System.out.println("Products:");
26        for(int i=0;i<nProducts;i++){
27            Product p=products[i];
28            System.out.println(p.getName()+" - $"+p.getPrice());
29        }
30        System.out.println("Total Price: $"+totalPrice);
31    }
32 }
```

