```java
import java.util.Date;

public class Bread extends Product{
    private Size size;
    private Date expDate;
    static public int amountBread;

    Bread(){
        super();
        amountBread++;
    }
    Bread(String productCode , double price , Date expDate ,Size size){
        super(productCode , price);
        this.size = size;
        this.expDate  = expDate;
        amountBread++;
    }

    public Size getSize() {
        return size;
    }
    public Date getExpDate() {
        return expDate;
    }

    public int getAmountBread() {
        return amountBread;
    }
    @Override
    public String toString() {
        return super.toString() + "Bread [size=" + size + "]";
    }

}
enum Size{
    S,M,L
}
```

```java
public class Calculator extends Product{
    private double warrantyYear;
    static public int amountCal;
    Calculator(){
        super();
        Calculator.amountCal++;
    }
    Calculator(String productCode , double price,double warrantyYear){
        super(productCode , price);
        this.warrantyYear = warrantyYear;
        Calculator.amountCal++;
    }

    public double getWarrantyYear() {
        return warrantyYear;
    }
    @Override
    public String toString() {
        return super.toString() + "Calculator [warrantyYear=" + warrantyYear +
"]";
    }

}
```

```java
import java.util.Date;

public class Main {
    public static void main(String[] args) {
        try{
            Bread bread1 = new Bread("BRE1", 20, new Date(2022,05,20),
Size.S);
            Calculator cal1 = new Calculator("CAL1", 100, 1);
            System.out.println("Bread amount: " + Bread.amountBread);
            System.out.println("Calculator amount: " + Calculator.amountCal);
            System.out.println("Total income: " + Product.totalIncome);
            Product.sale(bread1);
            System.out.println("Bread amount: " + Bread.amountBread);
            System.out.println("Calculator amount: " + Calculator.amountCal);
            System.out.println("Total income: " + Product.totalIncome);
            Product.sale(cal1);
            System.out.println("Bread amount: " + Bread.amountBread);
            System.out.println("Calculator amount: " + Calculator.amountCal);
            System.out.println("Total income: " + Product.totalIncome);
        }
        catch(Exception ex){
            System.out.println(ex);
        }
    }
}
```

```java
public class Product implements Comparable<Product>{
    private String productCode;
    private double price;
    static protected int amount;
    static protected double totalIncome;

    Product(){
        Product.amount++;
    }
    Product(String productCode , double price){
        this.price = price;
        this.productCode = productCode;
        Product.amount++;
    }
    public double getPrice() {
        return price;
    }
    public void setPrice(double price) {
        this.price = price;
    }
    public int getAmount() {
        return amount;
    }

    public String getProductCode() {
        return productCode;
    }
    static public void sale(Product product){
        if(product instanceof Bread){
           Bread bread = (Bread)product;
           switch(bread.getSize()){
               case S :
               Product.amount--;
               Bread.amountBread--;
               Product.totalIncome += bread.getPrice();
               System.out.println("sale the " + bread.toString());
               break;
               case M :
               Product.amount--;
               Bread.amountBread--;
               Product.totalIncome += bread.getPrice()+5;
               System.out.println("sale the " + bread.toString());
               break;
               case L :
               Product.amount--;
               Bread.amountBread--;
               Product.totalIncome += bread.getPrice()+8;
               System.out.println("sale the " + bread.toString());
```

```
                break;
            }
        }
        else if(product instanceof Calculator){
            Calculator cal = (Calculator)product;
            Product.amount--;
            Calculator.amountCal--;
            Product.totalIncome += cal.getPrice();
            System.out.println("sale the" + cal.toString());
        }
    }

    @Override
    public int compareTo(Product o) {
        if(o.productCode == this.productCode){
            return 1;
        }
        else{
            return 0;
        }
    }
    @Override
    public String toString() {
        return "Product [amount=" + amount + ", code=" + productCode + ",
price=" + price + "]";
    }

}
```