1 . public class Product implements Comparable<Product>{

private int productID ;

public int getProductID() {

return productID;

}

public void setProductID(int productID) {

this.productID = productID;

}

private String productName ;

public String getProductName() {

return productName;

}

public void setProductName(String productName) {

this.productName = productName;

}

private float price ;

public float getPrice() {

return price;

}

public void setPrice(float price) {

this.price = price;

}

private int quanity ;

public int getQuanity() {

return quanity;

}

public void setQuanity(int quanity) {

this.quanity = quanity;

}

2 . public Product (int productID , String ProductName){

setProductID(productID);

setProductName(ProductName);

InstanceCount++;

}

3 .

public Product (int productID , String productName , float price ,int quanity){

this(productID,productName) ;

setPrice(price);

setQuanity(quanity);

}

4 . a . لا لاتمكننا لأنها تقارن بين المؤشرات وليس الأغراض

B . لا لأننا لم نقم بإعادة تعريف التابع بعد

وهوة يأخذ التابع من الأب object

وهوة يقارن بين المؤشرات ايضا

@Override

public boolean equals(Object obj) {

if (obj == null) {

return false;

}

if (getClass() != obj.getClass()) {

return false;

}

final Product other = (Product) obj;

if (this.productID != other.productID) {

return false;

}

if (!Objects.equals(this.productName, other.productName)) {

return false;

}

if (Float.floatToIntBits(this.price) != Float.floatToIntBits(other.price)) {

return false;

}

if (this.quanity != other.quanity) {

return false;

}

return true;

}

C . لا لايمكننا لأننا لم نحدد معيار المقارنة للترتيب من خلال اعادة تعريف التابع comperTo

D . public static Comparator<Product> getPoductNameCorparator(){

return new Comparator<Product>() {

@Override

public int compare(Product o1, Product o2) {

return o1.getProductName().compareTo(o2.getProductName()) ;

}

} ;

}

الطلب الخامس والسادس فقط نسخ كود