```
package exp6;
import java.util.Arrays;
import java.util.Iterator;
import java.util.List;
import javax.jdo.JDOHelper;
import javax.jdo.PersistenceManager;
import javax.jdo.PersistenceManagerFactory;
import javax.jdo.Query;
import javax.jdo.Transaction;
public class MyApp {
public static void main(String[] args)
{//TEST is the persistent unit name
PersistenceManagerFactory pmf = JDOHelper.getPersistenceManagerFactory("TEST");
PersistenceManager pm = pmf.getPersistenceManager();
Transaction tx=pm.currentTransaction();
try
{
tx.begin();

Inventory inv = new Inventory("My Inventory");
Product product = new Product("Sony Discman", "Best", 549.99);
Product product1 = new Product("Sony xperia z1", "Good", 49.99);
Product product2 = new Product("Sony bravia", "Best", 458.99);
Product product3 = new Product("LG Refrigerator", "good", 569.99);
Product product4 = new Product("LG Microwave Oven", "High quality", 225.99);
Product product5 = new Product("Dell Inspiron", "OK", 178.99);
Product product6 = new Product("Micromax canvas", "OK", 80.99);
Product product7 = new Product("Lenova note3", "A tablet", 170.99);
Product product8 = new Product("Apple AirMac", "Best", 160.99);
Product product9 = new Product("Samsung s6", "Good", 50.99);
Product product10 = new Product("Sony xperia z6", "High quality", 10.99);

inv.getProducts().add(product);
inv.getProducts().add(product1);
inv.getProducts().add(product2);
inv.getProducts().add(product3);
inv.getProducts().add(product4);
inv.getProducts().add(product5);
inv.getProducts().add(product6);
inv.getProducts().add(product7);
inv.getProducts().add(product8);
inv.getProducts().add(product9);
inv.getProducts().add(product10);

pm.makePersistent(inv);


//display all products
System.out.println("\n All Products");
Query q1 = pm.newQuery("SELECT FROM " + Product.class.getName());
List<Product> products = (List<Product>)q1.execute();
Iterator<Product> iter = products.iterator();
```

```
while (iter.hasNext())
{
Product p = iter.next();
System.out.println("Name: "+p.name+"\t Description:"+p.description+"\tPrice: "+p.price);
}

// display all products with price less than 150.00
System.out.println("\n Products with Price < 150.00");
Query q2 = pm.newQuery("SELECT FROM " + Product.class.getName() +" WHERE price <
150.00");
products = (List<Product>)q2.execute();
iter = products.iterator();
while (iter.hasNext())
{
Product p = iter.next();
System.out.println("Name: "+p.name+"\t Description:"+p.description+"\tPrice: "+p.price);
}

//display all products with price less than 150.00 ordered by Price
System.out.println("\n Products with Price < 150.00 ordered by Price");
Query q3= pm.newQuery("SELECT FROM " + Product.class.getName() +" WHERE price <
150.00 ORDER BY price ASC");
products = (List<Product>)q3.execute();
iter = products.iterator();
while (iter.hasNext())
{
Product p = iter.next();
System.out.println("Name: "+p.name+"\t Description:"+p.description+"\tPrice: "+p.price);
}

// display all products alphabetically in ascending order by name of product
Query q4 = pm.newQuery(Product.class);
q4.setOrdering("name asc");
products = (List<Product>)q4.execute();
iter = products.iterator();
System.out.println("\nAll Products in ascending order by name of product");
while (iter.hasNext())
{
Product p = iter.next();
System.out.println("Name: "+p.name+"\t Description:"+p.description+"\tPrice: "+p.price);
}

// delete products with price > 500.00
Query q5 = pm.newQuery(Product.class);
q5.setFilter("price > max");
q5.declareParameters("int max");
q5.deletePersistentAll(500);
products = (List<Product>)q1.execute();
iter = products.iterator();
System.out.println("\nAfer deleting products with price > 500.00");
while (iter.hasNext())
{
```

```
Product p = iter.next();
System.out.println("Name: "+p.name+"\t Description:"+p.description+"\tPrice: "+p.price);
}

//display all products with name equal to Sony xperia z6 or Dell Inspiron
System.out.println("\ndisplay all products with name equal to Sony xperia z6 or Dell Inspiron");
Query q6 = pm.newQuery(Product.class, ":p.contains(name)");
products = (List<Product>)q6.execute(Arrays.asList("Sony xperia z6", "Dell Inspiron"));
iter = products.iterator();
while (iter.hasNext())
{
Product p = iter.next();
System.out.println("Name: "+p.name+"\t Description:"+p.description+"\tPrice: "+p.price);
}

//display all products with setRange
System.out.println("\n All Products with setRange");
Query q7 = pm.newQuery("SELECT FROM " + Product.class.getName());
/* The range indicates which results in the complete result set should be the first and last returned.
Results are identified by their numeric indices,
with 0 denoting the first result in the set.
For example, a range of 5, 10 returns the 6th through 10th results. */
q7.setRange(5, 10);
products = (List<Product>)q7.execute();
iter = products.iterator();
while (iter.hasNext())
{
Product p = iter.next();
System.out.println("Name: "+p.name+"\t Description:"+p.description+"\tPrice: "+p.price);
}

//display all products alphabetically in descending order by name of product
Query q8 = pm.newQuery(Product.class);
q8.setOrdering("name desc");
products = (List<Product>)q8.execute();
iter = products.iterator();
System.out.println("\nAll Products in descending order by name of product");
while (iter.hasNext())
{
Product p = iter.next();
System.out.println("Name: "+p.name+"\t Description:"+p.description+"\tPrice: "+p.price);
}

//display all products with price less than 150.00 ordered by Description
System.out.println("\n Products with Price < 150.00 ordered by Description");
Query q9= pm.newQuery("SELECT FROM " + Product.class.getName() +" WHERE price <
150.00 ORDER BY description ASC");
products = (List<Product>)q9.execute();
iter = products.iterator();
while (iter.hasNext())
{
Product p = iter.next();
```

```
System.out.println("Name: "+p.name+"\t Description:"+p.description+"\tPrice: "+p.price);
}

//display all products ordered by Description
System.out.println("\n Products - ordered by Description");
Query q10= pm.newQuery("SELECT FROM " + Product.class.getName() +" ORDER BY
description ASC");
products = (List<Product>)q10.execute();
iter = products.iterator();
while (iter.hasNext())
{
Product p = iter.next();
System.out.println("Name: "+p.name+"\t Description:"+p.description+"\tPrice: "+p.price);
}


tx.commit();
}
catch(Exception e)
{
        e.printStackTrace();
}
finally
{
if (tx.isActive())
{
tx.rollback();
}
pm.close();
}
}
}
```