**import** jdk.internal.org.objectweb.asm.commons.StaticInitMerger;  
  
**import** javax.naming.event.ObjectChangeListener;  
**import** javax.xml.soap.Node;  
  
*/\*\*  
 \* Created by acer on 3/18/2022.  
 \*/***public class** DoublyLinkedList<E> {  
**private** Node<E> **header**;  
 **private** Node<E> **trailer**;  
  
 **public boolean** isEmoty(){  
 **return header**==**null**;  
 }  
 **public** E getFrist(){  
 **return header** !=**null**? **header**.getElement():**null**;  
 }  
 **public** E getLast(){  
 **return trailer** !=**null**? **trailer**.getElement():**null**;  
 }  
 **public void** addFrist(E element){  
 Node<E> newNod= **new** Node<E>(element ,**null**,**header**);  
 **header**=newNod;  
 **if**(**trailer**==**null**){  
 **trailer**=newNod;  
 }  
 }  
 **public boolean** equals(Object that) {  
 **if** (!(that **instanceof** DoublyLinkedList))  
 **return false**;  
 DoublyLinkedList<E> other = (DoublyLinkedList<E>) that;  
 **if** (**header** == **null**) {  
 **return** other.**header** == **null** ? **true** : **false**;  
 }  
 **if** (!**header**.equals(other.**header**))  
 **return false**;  
 **if** (**header** == **trailer**) {  
 **return true**;  
 }  
 **if** (!**trailer**.equals(other.**trailer**))  
 **return false**;  
 Node<E> thisNode = **header**;  
 Node<E> otherNode = other.**header**;  
 **while** (thisNode.getNext() != **trailer**) {  
 thisNode = thisNode.getNext();  
 otherNode = otherNode.getNext();  
 **if** (!(thisNode.equals(otherNode))) {  
 **return false**;  
 }  
 }  
 **return true**;  
 }  
 **public** String toString() {  
 StringBuilder sb = **new** StringBuilder();  
 sb.append(**"Dobel"**);  
 Node<E> finger = **header**;  
 **while** (finger != **null**) {  
 sb.append(finger.toString());  
 **if** (finger.getNext() != **null**) {  
 sb.append(**"-"**);  
 }  
 finger = finger.getNext();}  
 **return** sb.toString();  
  
  
  
 }  
  
 **public class** Node<E>{  
 **private** E **element**;  
 **private** Node<E> **previous**;  
 **private** Node<E> **next**;  
  
 **public** E getElement() {  
 **return element**;  
 }  
  
 **public** Node<E> getPrevious() {  
 **return previous**;  
 }  
  
 **public** Node<E> getNext() {  
 **return next**;  
 }  
  
 **public** Node(E element, Node<E> previous, Node<E> next) {  
 **this**.**element** = element;  
 **this**.**previous** = previous;  
 **this**.**next** = next;  
 }  
 **public boolean** equals (Object that){  
 **if**(!(that **instanceof** Node)) {  
 **return false**;  
  
 }  
  
 Node<E> other =(Node<E>) that;  
 **if**(**element**==**null**){  
 **return** other.**element**==**null**?**true** :**false**;  
 }  
 **return element**.equals(other.**element**);  
 }  
 **public** String toString(){  
 **return element**.toString();  
 }  
 }  
}

//////////////////////////////////////////////

*/\*\*  
 \* Created by acer on 3/18/2022.  
 \*/***public class** TEST {  
 **public static void** main(String[] args) {  
 DoublyLinkedList<Integer>dll1=**new** DoublyLinkedList<Integer>();  
  
 dll1.addFrist(100);  
 dll1.addFrist(100);  
  
 DoublyLinkedList<Integer>dll2=**new** DoublyLinkedList<Integer>();  
 dll2.addFrist(100);  
 dll2.addFrist(200);  
  
 DoublyLinkedList<Integer>dll3=**new** DoublyLinkedList<Integer>();  
 dll1.addFrist(42);  
 DoublyLinkedList<String>blankList1=**new** DoublyLinkedList<String>();  
  
 DoublyLinkedList<String>blankList2=**new** DoublyLinkedList<String>();  
**if**(blankList1.equals(blankList2)){  
 System.***out***.println(blankList1+**"="**+blankList2);  
}  
**if**(!dll1.equals(dll3)){  
 System.***out***.println(dll1+**" !="**+dll3);  
}  
**if** (dll1.equals(dll2)){  
 System.***out***.println(dll1+**" ="**+dll2);  
}  
 }  
}