public class DoublyLinkedList {

// تعريف الكلاس Node لتمثيل العقدة

private static class Node {

Object element; // العنصر المخزن في العقدة

Node prev; // المؤشر إلى العقدة السابقة

Node next; // المؤشر إلى العقدة التالية

public Node(Object element) {

this.element = element;

this.prev = null;

this.next = null;

}

}

private Node header; // العقدة الرأسية (Sentinel)

private Node trailer; // العقدة النهائية (Sentinel)

public DoublyLinkedList() {

header = new Node(null); // عقدة رأسية لا تحتوي على بيانات

trailer = new Node(null); // عقدة نهائية لا تحتوي على بيانات

header.next = trailer;

trailer.prev = header;

}

// دالة لإرجاع حجم القائمة

public int size() {

int count = 0;

Node current = header.next;

while (current != trailer) {

count++;

current = current.next;

}

return count;

}

// دالة equals لمقارنة قائمتين مرتبطتين مزدوجتين

public boolean equals(DoublyLinkedList other) {

// إذا كانت القائمة الأخرى غير موجودة أو الأحجام مختلفة، فهي ليست متساوية

if (other == null) {

return false;

}

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

return false;

}

// مقارنة العناصر في العقد المقابلة للقائمتين

Node currentSelf = this.header.next;

Node currentOther = other.header.next;

while (currentSelf != this.trailer) {

if (!currentSelf.element.equals(currentOther.element)) {

return false;

}

currentSelf = currentSelf.next;

currentOther = currentOther.next;

}

return true; // إذا كانت جميع العناصر متطابقة

}

// دالة لإضافة عنصر إلى نهاية القائمة

public void addLast(Object element) {

Node newNode = new Node(element);

newNode.prev = trailer.prev;

newNode.next = trailer;

trailer.prev.next = newNode;

trailer.prev = newNode;

}

// اختبار الدالة equals

public static void main(String[] args) {

DoublyLinkedList dll1 = new DoublyLinkedList();

dll1.addLast(1);

dll1.addLast(2);

dll1.addLast(3);

DoublyLinkedList dll2 = new DoublyLinkedList();

dll2.addLast(1);

dll2.addLast(2);

dll2.addLast(3);

DoublyLinkedList dll3 = new DoublyLinkedList();

dll3.addLast(4);

dll3.addLast(5);

dll3.addLast(6);

System.out.println(dll1.equals(dll2)); // Output: true

System.out.println(dll1.equals(dll3)); // Output: false

}

}