**✅ Implementing equals Method for SinglyLinkedList**

To correctly implement the equals method for a **SinglyLinkedList**, we need to check the following:

1. **Same Class Type:** Ensure the compared object is also a SinglyLinkedList.
2. **Same Size:** If lists have different sizes, they can't be equal.
3. **Same Elements in Order:** Traverse both lists and compare corresponding elements.

**Implementation:**

public class SinglyLinkedList<E> {

private static class Node<E> {

E element;

Node<E> next;

public Node(E element, Node<E> next) {

this.element = element;

this.next = next;

}

}

private Node<E> head = null;

private Node<E> tail = null;

private int size = 0;

public int size() {

return size;

}

public void addLast(E element) {

Node<E> newNode = new Node<>(element, null);

if (tail == null) {

head = tail = newNode;

} else {

tail.next = newNode;

tail = newNode;

}

size++;

}

@Override

public boolean equals(Object obj) {

if (this == obj) return true; // Same memory reference

if (!(obj instanceof SinglyLinkedList<?>)) return false; // Different class

SinglyLinkedList<?> other = (SinglyLinkedList<?>) obj;

if (this.size != other.size) return false; // Different sizes

Node<E> currentA = this.head;

Node<?> currentB = other.head;

while (currentA != null) { // Both lists should have same elements in order

if (!currentA.element.equals(currentB.element)) return false;

currentA = currentA.next;

currentB = currentB.next;

}

return true;

}

}

**🔍 Explanation of equals Method:**

1. **Check Reference (this == obj)**
   * If both references are the same object, return true.
2. **Check Type (obj instanceof SinglyLinkedList<?>)**
   * If obj is not a SinglyLinkedList, return false.
3. **Compare Sizes**
   * If the lists have different sizes, they can't be equal.
4. **Compare Elements One by One**
   * Iterate through both lists and check if all corresponding elements are equal.

**Example Usage:**

public class Main {

public static void main(String[] args) {

SinglyLinkedList<Integer> list1 = new SinglyLinkedList<>();

list1.addLast(1);

list1.addLast(2);

list1.addLast(3);

SinglyLinkedList<Integer> list2 = new SinglyLinkedList<>();

list2.addLast(1);

list2.addLast(2);

list2.addLast(3);

SinglyLinkedList<Integer> list3 = new SinglyLinkedList<>();

list3.addLast(1);

list3.addLast(2);

list3.addLast(4);

System.out.println(list1.equals(list2)); // true

System.out.println(list1.equals(list3)); // false

}

}

**Output:**

true

false