

```
package singlelinkedlist;

public class SingleLinkedList {

    public static void main(String[] args) {

        LinkedList l1 = new LinkedList();
        l1.addFirst(1);
        l1.addFirst(2);
        l1.addFirst(33);
        l1.addFirst(22);
        l1.show();
        System.out.println();
        l1.removeFirst();
        l1.show();
        System.out.println();
        l1.removeLast();
        l1.show();

    }

}
```

```

package singlelinkedlist;

import java.util.NoSuchElementException;

public class LinkedList {

    private class Node {

        private int value;
        private Node next;

        public Node(int value) {
            this.value = value;
        }
    }

    private Node first;
    private Node last;
    private int size;

    public void addLast(int item) {
        var node = new Node(item);

        if (isEmpty()) {
            first = last = node;
        } else {
            last.next = node;
            last = node;
        }

        size++;
    }

    public void addFirst(int item) {
        var node = new Node(item);

        if (isEmpty()) {
            first = last = node;
        } else {
            node.next = first;
            first = node;
        }

        size++;
    }

    private boolean isEmpty() {
        return first == null;
    }

    public void removeFirst() {
        if (isEmpty()) {
            throw new NoSuchElementException();
        }

        if (first == last) {
            first = last = null;
        } else {
            var second = first.next;
            first.next = null;
            first = second;
        }

        size--;
    }

    public void removeLast() {
        if (isEmpty()) {
            throw new NoSuchElementException();
        }

        if (first == last) {
            first = last = null;
        } else {
            var previous = getPrevious(last);

```

```

        last = previous;
        last.next = null;
    }

    size--;
}

private Node getPrevious(Node node) {
    var current = first;
    while (current != null) {
        if (current.next == node) {
            return current;
        }
        current = current.next;
    }
    return null;
}

public int size() {
    return size;
}

public void show() {
    Node temp;
    temp = first;
    while (temp != null) {
        System.out.print(temp.value + "-->");
        temp = temp.next;
    }
}
}

```