class Node {

int data;

Node next;

Node prev;

Node(int data) {

this.data = data;

this.next = null;

this.prev = null;

}

}

class Question7 {

private Node head;

// Insert at the end of the list

public void insert(int data) {

Node newNode = new Node(data);

if (head == null) {

head = newNode;

} else {

Node temp = head;

while (temp.next != null) {

temp = temp.next;

}

temp.next = newNode;

newNode.prev = temp;

}

}

// Delete a node by its value

public void deleteByValue(int value) {

if (head == null) return;

// If the head node holds the value to be deleted

if (head.data == value) {

head = head.next;

if (head != null) {

head.prev = null;

}

return;

}

Node current = head;

while (current != null && current.data != value) {

current = current.next;

}

// If the value was found

if (current != null) {

if (current.next != null) {

current.next.prev = current.prev;

}

if (current.prev != null) {

current.prev.next = current.next;

}

}

}

// Traverse the list and return the elements as a string

public String traverse() {

StringBuilder result = new StringBuilder();

Node current = head;

result.append("[");

while (current != null) {

result.append(current.data);

current = current.next;

if (current != null) {

result.append(", ");

}

}

result.append("]");

return result.toString();

}

public static void main(String[] args) {

// Test Case 1

Question7 list1 = new Question7();

list1.insert(10);

list1.insert(20);

list1.insert(30);

list1.deleteByValue(20);

System.out.println("Output: List = " + list1.traverse());

// Test Case 2

Question7 list2 = new Question7();

list2.insert(1);

list2.insert(2);

list2.insert(3);

list2.deleteByValue(1);

System.out.println("Output: List = " + list2.traverse());

}

}  
  
