public class LinkedList implements List {

private static class Node {

Object data;

Node next;

Node(Object data) {

this.data = data;

this.next = null;

}

}

private Node head;

private int size;

public LinkedList() {

head = new Node(null);

size = 0;

}

public boolean isEmpty() {

return size == 0;

}

public int size() {

return size;

}

public void add(Object item) {

add(size + 1, item);

}

public void add(int index, Object item) {

if (index < 1 || index > size + 1) {

throw new IndexOutOfBoundsException("Invalid index");

}

Node newNode = new Node(item);

Node prev = head;

for (int i = 1; i < index; i++) {

prev = prev.next;

}

newNode.next = prev.next;

prev.next = newNode;

size++;

}

public void remove(int index) {

if (index < 1 || index > size) {

throw new IndexOutOfBoundsException("Invalid index");

}

Node prev = head;

for (int i = 1; i < index; i++) {

prev = prev.next;

}

prev.next = prev.next.next;

size--;

}

public void remove(Object item) {

Node prev = head;

Node current = head.next;

while (current != null) {

if (current.data.equals(item)) {

prev.next = current.next;

size--;

break;

}

prev = current;

current = current.next;

}

}

public List duplicate() {

LinkedList duplicateList = new LinkedList();

Node current = head.next;

while (current != null) {

duplicateList.add(current.data);

current = current.next;

}

return duplicateList;

}

public List duplicateReversed() {

LinkedList reversedList = new LinkedList();

Node current = head.next;

while (current != null) {

reversedList.add(1, current.data);

current = current.next;

}

return reversedList;

}

public String toString() {

StringBuilder sb = new StringBuilder("[ size: " + size + " - ");

Node current = head.next;

while (current != null) {

sb.append(current.data);

if (current.next != null) {

sb.append(", ");

}

current = current.next;

}

sb.append(" ]");

return sb.toString();

}

public static void main(String[] args) {

LinkedList list = new LinkedList();

list.add("Item1");

list.add("Item2");

list.add("Item3");

list.add("Item4");

System.*out*.println("Original List: " + list);

List duplicate = list.duplicate();

System.*out*.println("Duplicate List: " + duplicate);

List reversed = list.duplicateReversed();

System.*out*.println("Reversed List: " + reversed);

}

}