

|  |
| --- |
| /\*\*  \*  \*双向链表实现  \*  \*/  public class MyLinkedList {  private Node first;//链表的第一个节点  private Node last;//链表的最后一个节点  private int size;//节点的数量    //使用到内部类 描述节点  class Node{  Node prev;//上一个节点对象  Node next;//下一个节点对象  Object ele; // 当前节点中存储的值  //提供一个构造用来初始化节点值  public Node(Object ele) {  this.ele = ele;  }  }  /\*\*  \* 添加节点 前  \* @param ele 节点中的值  \*/  public void addFirst(Object ele) {  //1.创建节点对象  Node node = new Node(ele);  //2.判断是否是第一次插入节点  if(size == 0) { //第一次插入节点  this.first = node;//设置为头节点  this.last = node;//设置为尾节点  }else {//若执行else说明节点已经存在,需要将节点进行串联  //进行节点连接,当前节点即node应该在原有节点的前面  node.next = this.first;  //原有来节点需要连接当前节点即node  this.first.prev = node;  //更新节点将头节点赋值给node  this.first = node;  }  size++;//节点数+1  }  /\*\*  \* 添加节点 后  \* @param ele 元素值  \*/  public void addLast(Object ele) {  //1.先创建节点对象  Node node = new Node(ele);  //需要判断是不是第一次添加节点  if(size == 0) {  this.first = node;//添加头节点  this.last = node;//添加尾节点  }else {//当前已经不是第一次插入节点,需要串联节点  //原节点的下一位是当前节点即node  this.last.next = node;  //当前节点即node的上一位是原有节点  node.prev = this.last;  //将原有的尾部设置给当前节点即node  this.last = node;    }  //节点数自增  size ++;  }    /\*\*  \*节点删除  \*/  public void remove(Object ele) {  //知道被删除的节点  Node current = this.first;//获取头节点进行遍历  for(int i = 0; i<size;i++) {  if(!current.ele.equals(ele)) {//current.ele.equals(ele)不相等--> false !false --> true  if(current.next == null) { //判断是否还有下一个节点,return 结束  return;  }  current = current.next; //下一个节点  }    }  //删除具体节点  if(current == first) {//头  this.first = current.next;  this.first.prev = null;  }else if(current == last) {//尾  this.last = current.prev;  this.last.next = null;  }else {//某个位置  //把删除节点的上一个节点的下一位置连接到删除节点的下一个节点  current.prev.next = current.next;  //把删除节点的下一个节点的上个位置连接到删除节点的上一个节点  current.next.prev = current.prev;  }  //大小需要-1操作  size--;  }    @Override  public String toString() {  if(size == 0) {  return "[]";  }  StringBuilder bs = new StringBuilder();  //第一个节点  Node current = this.first;  bs.append("[");  for(int i = 0 ;i<size;i++) {  bs.append(current.ele);//取值  if(i != size-1) {  bs.append(",");  }else {  bs.append("]");  }  current = current.next;//获取下一个节点  }  return bs.toString();  }    } |