```
public class CycleList {
    private Node head;
    private Node current;
    private int size;
    //构造空循环链表
    CycleList() {
        //构造头结点
        this.head = new Node(null);
        //头结点指向自己,变成循环链表
        this.head.next = this.head;
        //当前节点指向头结点
        this.current = this.head;
        this.size = 0;
    }

    //定位
    private void index(int idx) {
        //头节点的下标
        if (idx==-1) {
            //如果位置为头结点, 则当前节点指向头结点
            current = head;
            return;
        }
        if (idx<-1) {
            throw new RuntimeException("下标越界");
        }
        idx = idx%size;
        //如果定位位置为非头结点, 则让current指向head.next
        current = head.next;
        for(int i=0; i<idx; i++) {
            current = current.next;
        }
    }

    //添加节点
    public void add (int idx, Object data) {
        index(idx-1);
        current.next = new Node(data, current.next);
```

```
            this.size++;
        }

        //删除节点
        public void remove (int idx) {
            index(idx-1);
            current.next = current.next.next;
            this.size--;
        }

        //返回节点的值
        public Object get (int idx) {
            index(idx);
            return current.data;
        }

        //判断链表是否为空
        public boolean isEmpty() {
            return this.size==0;
        }

        //链表的长度
        public int size() {
            return this.size;
        }
}
class Node {
        public Object data;
        public Node next;

        //构造头节点
        Node(Node next){
            this.data = null;
            this.next = next;
        }
        //非头节点
        Node(Object data, Node next){
            this.data = data;
            this.next = next;
```

```
    }
    public Object getData() {
        return data;
    }
    public void setData(Object data) {
        this.data = data;
    }
    public Node getNext() {
        return next;
    }
    public void setNext(Node next) {
        this.next = next;
    }
}
```