
1.单链表的缺点

1. 单链表智能单向遍历（双链表可以双向查找）
2. 单链表不能自我删除，需要靠辅助节点（双链表可以自我删除）

2.双链表的实现

1.节点特点:

pre：指向前一个节点

next：指向下一个节点

2.双链表的增加，插入，遍历，删除

尾部增加节点

```
public void addNode(Node node) {
    if (headNode.next == null) {
        headNode.next = node;
        node.pre = headNode;
        footNode = node;
    } else {
        footNode.next = node;
        node.pre = footNode;
        footNode = node;
    }
}
```

按照顺序插入节点

```
public void insertNodeById(Node node) {
    if (headNode.next == null) {
        headNode.next = node;
        footNode = node;
        return;
    }
    Node temp = headNode.next;
    while (temp != null) {
        int no = temp.getNo();
        int newNo = node.getNo();
        if (newNo == no) {
            System.out.println("添加失败");
            return;
        } else if (newNo > no) {
            if (temp.next == null) {
                temp.next = node;
                node.pre=temp;
                footNode = node;
                return;
            }
            temp = temp.next;
        } else {
            temp.pre.next = node;
            node.pre=temp.pre;
            node.next = temp;
            return;
        }
    }
}
```

双链表的遍历

```
public void list() {
    Node node = headNode.next;
    if (node == null) {
        System.out.println("空链表");

    } else {
        while (node != null) {
            System.out.println(node);
            node = node.next;
        }
    }
}
```

链表的删除

```
public void deleteNode(Node node) {
    if (headNode.next == null) {
```

```
            System.out.println("空链表");
            return;
        }
        Node temp = headNode.next;
        while (temp != null) {
            int n = temp.getNo();
            if (n == node.getNo()) {
                if (temp == footNode) {
                    footNode = temp.pre;
                    temp.pre.next=null;
                    return;
                }
                temp.pre.next=temp.next;
                temp.next.pre=temp.pre;
                return;
            }
            temp = temp.next;
        }
        System.out.println("没有该节点");
    }
```