```java
package LinkedList;

class Node {

    Node next;
    int val;

    public Node(int val) {

        this.val = val;
        this.next = null;
    }

    public Node() {
    }
}


class BasicOperationsOfNode {

    Node head;

    public BasicOperationsOfNode(Node headNode) {
        this.head = headNode;
    }

    public BasicOperationsOfNode() {

    }

    //在末尾添加节点
    public void addNodeToTail(int val) {

        if (head == null) {
            head = new Node(val);

        } else {

            Node temp = head;
            while (temp.next != null) {
                temp = temp.next;
            }
            temp.next = new Node(val);
        }
    }


    //删除第index个节点
    public boolean deleteNodeOfIndex(int index) {

        Node slow = null;
        Node fast = head;
        int indexCount = 0;

        while (fast != null) {
            if (indexCount == index) {
                if (fast == head) {
                    head = head.next;
                } else {
                    slow.next = fast.next;
                }
                return true;
            }
            indexCount++;
            slow = fast;
            fast = fast.next;
        }

        return false;
    }


    //返回链表的长度
```

```
74      public int lengthOfLinkedList() {
75
76          int length = 0;
77          Node temp = head;
78
79          while (temp != null) {
80              length++;
81              temp = temp.next;
82          }
83
84          return length;
85      }
86
87
88      //对链表进行排序(升序)
89      public void orderLinkedList() {
90
91          // 冒泡排序法
92          Node outerNode = head;
93          Node innerNode;
94
95          while ( outerNode != null ) {
96
97              innerNode = outerNode.next;
98              while ( innerNode != null ) {
99
100                 if ( outerNode.val > innerNode.val ) {
101                     int temp = outerNode.val;
102                     outerNode.val = innerNode.val;
103                     innerNode.val = temp;
104                 }
105                 innerNode = innerNode.next;
106             }
107             outerNode = outerNode.next;
108         }
109
110      }
111
112
113     public static void main(String[] args) {
114
115         Node headNode = new Node(7);
116         BasicOperationsOfNode basicOperationsOfNode = new
            BasicOperationsOfNode(headNode);
117         headNode.next = new Node(2);
118         headNode.next.next = new Node(3);
119         headNode.next.next.next = new Node(1);
120
121         /*BasicOperationsOfNode basicOperationsOfNode1 = new BasicOperationsOfNode();
122         basicOperationsOfNode1.orderLinkedList();
123
124         BasicOperationsOfNode basicOperationsOfNode2 = new
            BasicOperationsOfNode(headNode);
125         basicOperationsOfNode2.orderLinkedList();*/
126
127         //测试在末尾添加节点
128         basicOperationsOfNode.addNodeToTail(6);
129
130         int length = basicOperationsOfNode.lengthOfLinkedList();
131         basicOperationsOfNode.orderLinkedList();
132
133         System.out.println("haha");
134     }
135 }
```