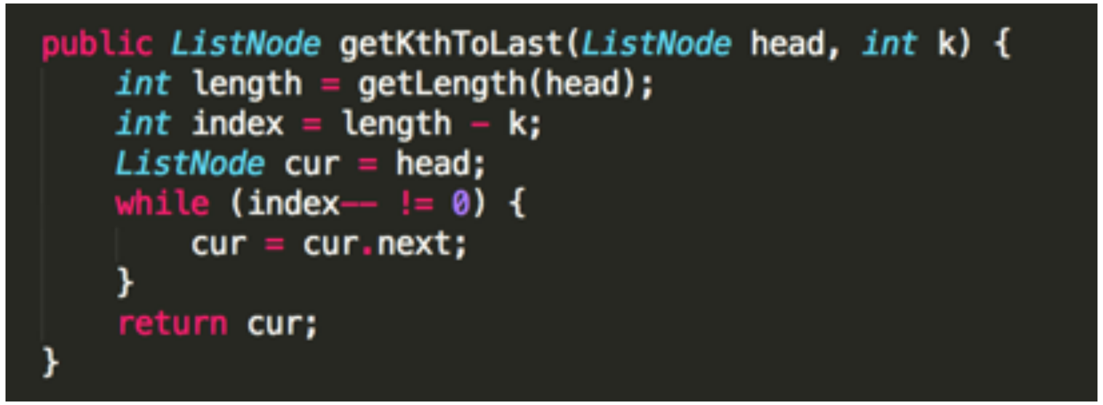
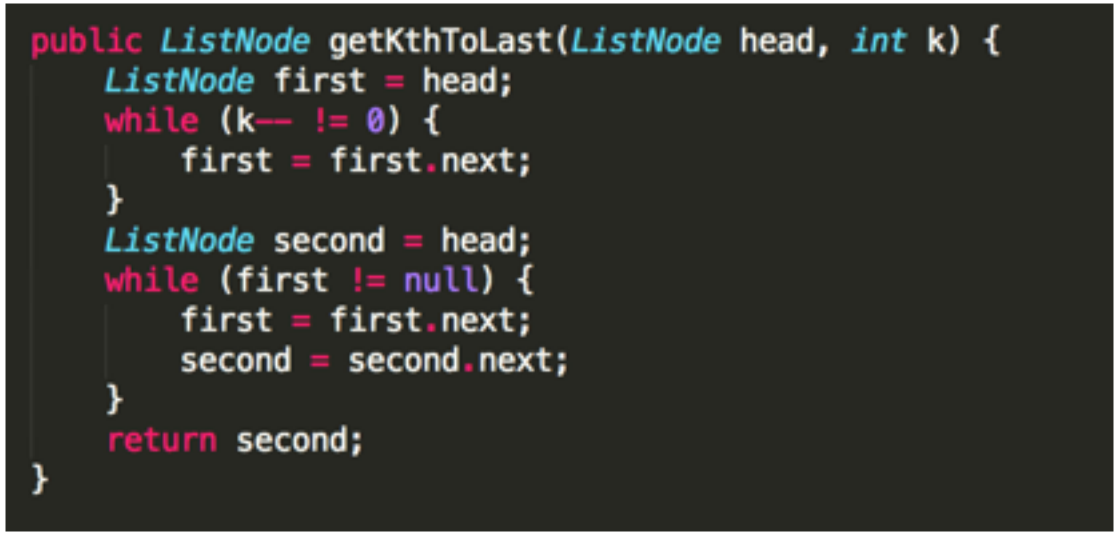
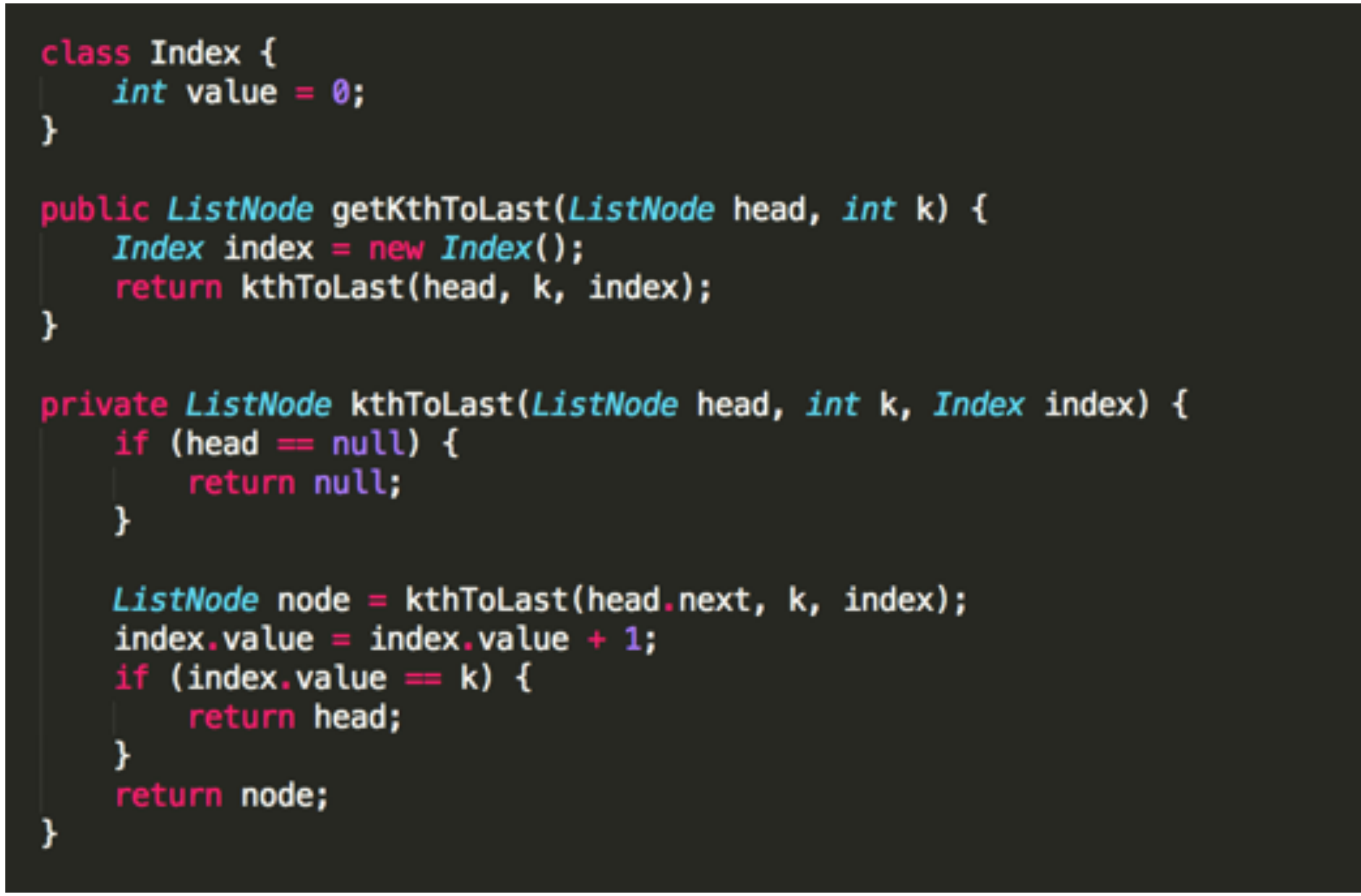


```java
public ListNode getKthToLast(ListNode head, int k) {
    int length = getLength(head);
    int index = length - k;
    ListNode cur = head;
    while (index-- != 0) {
        cur = cur.next;
    }
    return cur;
}
```



```java
public ListNode getKthToLast(ListNode head, int k) {
    ListNode first = head;
    while (k-- != 0) {
        first = first.next;
    }
    ListNode second = head;
    while (first != null) {
        first = first.next;
        second = second.next;
    }
    return second;
}
```

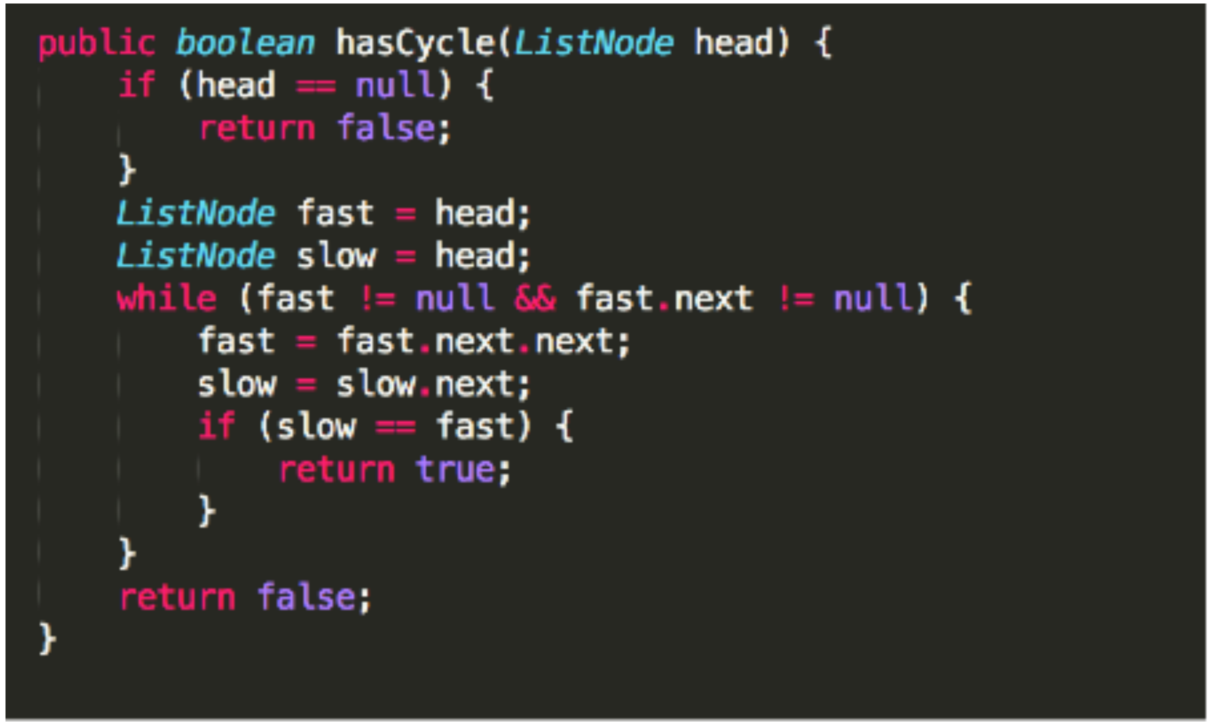


```java
class Index {
    int value = 0;
}

public ListNode getKthToLast(ListNode head, int k) {
    Index index = new Index();
    return kthToLast(head, k, index);
}

private ListNode kthToLast(ListNode head, int k, Index index) {
    if (head == null) {
        return null;
    }

    ListNode node = kthToLast(head.next, k, index);
    index.value = index.value + 1;
    if (index.value == k) {
        return head;
    }
    return node;
}
```

环形链表 I

给定一个链表，判断链表中否有环

（https://leetcode.com/problems/linked-list-cycle/description/）



```java
public boolean hasCycle(ListNode head) {
    if (head == null) {
        return false;
    }
    ListNode fast = head;
    ListNode slow = head;
    while (fast != null && fast.next != null) {
        fast = fast.next.next;
        slow = slow.next;
        if (slow == fast) {
            return true;
        }
    }
    return false;
}
```