

# Java方向编程题答案

####

## day8

[编程题]26011-删数

<https://www.nowcoder.com/practice/f9533a71aada4f35867008be22be5b6e?tpId=49&&tqId=29296&rp=1&ru=/activity/oj&qu=/ta/2016test/question-ranking>

**【题目解析】：**这个题目我相信童鞋们应该可以读的懂，其实就是相当于这样的一个游戏：n个人围城一圈，开始报数，报数为3的退出。问存活的是谁？虽然这两个题不能等价，但是考的知识点差不多。

**【解题思路】：**这里我们用到了集合ArrayList，当然你也可以使用其他的集合。存放n个数，且都是从0到n-1的数字。因为每次删除的都是当前位置的后面第2个，那么我们就可以这样写： $(i+2)\%size$ ；当 $i=0$ 的时候，我们移除的就是2号下标，接着需要移除4号下标，那么新的 $i = (i+2)\%size$ ；那么结束条件应该是什么？我们要找到存活的，显而易见当集合的size等于1的时候停止删除。

**【示例代码】：**

```
import java.util.*;

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        while (sc.hasNext()) {
            int n = sc.nextInt();
            if (n > 1000) {
                n = 999;
            }
            List<Integer> list = new ArrayList<Integer>();
            for (int i = 0; i < n; i++) {
                list.add(i);
            }
            int i = 0;
            while (list.size() > 1) {
                i = (i + 2) % list.size();
                list.remove(i);
            }
            System.out.println(list.get(0));
        }
    }
}
```

[编程题]69386-n个数里最小的k个

链接：<https://www.nowcoder.com/questionTerminal/cc727473d1e248ccb674eb31bd8683dc>

**【题目解析】：**

这道题有很多种解法，但是经过咱们前几天的训练，相信大家看到题目就知道要考啥了吧。没错！还是快排的partition思想！

**【解题思路】：**

排序+分区：

找到下标为k的位置，则可知，前k小的都在左边，排序输出即可。

【示例代码】：

```
import java.util.*;
public class Main{
    public static void main(String[] args){
        Scanner sc = new Scanner(System.in);
        while(sc.hasNext()){
            String str = sc.nextLine();
            String[] s = str.split(" ");
            int[] num = new int[s.length-1];
            for(int i=0;i<s.length-1;i++){
                num[i] = Integer.parseInt(s[i]);
            }

            int k = Integer.parseInt(s[s.length-1]);
            int start = 0;
            int end = num.length-1;
            int index = qSort(num, start,end);

            while(index != k){
                if(index>k){
                    end = index - 1;
                    index = qSort(num, start, end);
                }else{
                    start = index + 1;
                    index = qSort(num, start, end);
                }
            }
            Arrays.sort(num, 0, k);
            for(int i=0;i<k;i++){
                System.out.print(i==k-1?num[i]:num[i] + " ");
            }
        }
    }
    public static int qSort(int[] num, int start, int end){
        int dig = num[start];
        while(start<end){
            while(start<end && num[end]>=dig)end--;
            num[start] = num[end];
            while(start<end && num[start]<dig)start++;
            num[end] = num[start];
        }
        num[start] = dig;
        return start;
    }
}
```

#####