Java方向编程题答案

day33

[编程题]24513-锤子剪刀布

https://www.nowcoder.com/questionTerminal/79db907555c24b15a9c73f7f7d0e2471

【题目解析】：

这题应该不难

【解题思路】：

考察几个常用方法的使用，本身没有什么特别难的地方

【示例代码】：

```
import java.util.HashMap;
import java.util.Scanner;
import java.util.TreeMap;

public class Main {
    private static class Record {
        public int win;
        public int tie;
        public int lose;
        TreeMap<String, Integer> winCount;

        Record() {
            this.win = 0;
            this.tie = 0;
            this.lose = 0;
            this.winCount = new TreeMap<>();
            this.winCount.put("B", 0);
            this.winCount.put("C", 0);
            this.winCount.put("J", 0);
        }
    }

    private static void win(Record[] records, int win, String gesture) {
        int lose = 1 - win;
        records[win].win++;
        records[lose].lose++;
        int winCount = records[win].winCount.get(gesture);
        records[win].winCount.put(gesture, winCount + 1);
    }

    private static void judge(String a, String b, Record[] records) {

        if (a.equals("B")) {
```

```
            if (b.equals("B")) {
                records[0].tie++;
                records[1].tie++;
            } else if (b.equals("C")) {
                win(records, 0, "B");
            } else {
                win(records, 1, "J");
            }
        } else if (a.equals("C")) {
            if (b.equals("B")) {
                win(records, 1, "B");
            } else if (b.equals("C")) {
                records[0].tie++;
                records[1].tie++;
            } else {
                win(records, 0, "C");
            }
        } else {
            if (b.equals("B")) {
                win(records, 0, "J");
            } else if (b.equals("C")) {
                win(records, 1, "C");
            } else {
                records[0].tie++;
                records[1].tie++;
            }
        }
    }

    private static String mostWinGesture(TreeMap<String, Integer> winCount) {
        if (winCount.get("C") >= winCount.get("J")) {
            if (winCount.get("B") >= winCount.get("C")) {
                return "B";
            } else {
                return "C";
            }
        } else {
            if (winCount.get("B") >= winCount.get("J")) {
                return "B";
            } else {
                return "J";
            }
        }
    }

    public static void main(String[] args) {
        Record[] records = new Record[2];
        records[0] = new Record();
        records[1] = new Record();
        Scanner scanner = new Scanner(System.in);
        int n = scanner.nextInt();
        for (int i = 0; i < n; i++) {

            String a = scanner.next();
```

```
            String b = scanner.next();
            judge(a, b, records);
        }

        System.out.format("%d %d %d\n",
                records[0].win, records[0].tie, records[0].lose);
        System.out.format("%d %d %d\n",
                records[1].win, records[1].tie, records[1].lose);
        System.out.format(
            mostWinGesture(records[0].winCount)
            + " "
            + mostWinGesture(records[1].winCount));
    }
}
```

[编程题]24964-上楼梯

https://www.nowcoder.com/questionTerminal/7f0661ace6df48d0af3f924950d57126

【题目解析】：

大家应该都做个可以 1、2 的上楼梯，那我们再换个 1、2、3 的练习下

【解题思路】：

动态规划上楼梯的变形

【示例代码】：

```
import java.util.*;

public class GoUpstairs {
    public int countWays(int n) {
        // 第一阶有一种方法
        // 第二阶有两种方法，一步一步 or 一次两步
        // 第三阶有四种方法，1+1+1 or 1 + 2 or 2 + 1 or 3
        long[] pre = {1, 2, 4};
        if(n<=0) return 0;
        else if(n<=3) return (int)pre[n-1];
        else{
            for(int i=4; i<=n; i++){
                long tmp = (pre[0] + pre[1] + pre[2]) % 1000000007;
                pre[0] = pre[1];
                pre[1] = pre[2];
                pre[2] = tmp;
            }
        }
        return (int)pre[2];
    }
}
```