Java方向编程题答案

day31

[编程题]24544-说反话

https://www.nowcoder.com/questionTerminal/aced908691df4ebca6744f9fbd437749

【题目解析】：

这题应该是挺简单的，就是考察几个 String 类的方法

【解题思路】：

按空格分割后，逆序输出即可

【示例代码】：

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;

public class Main {
    public static void main(String[] args) throws IOException {
        BufferedReader reader = new BufferedReader(
                new InputStreamReader(System.in)
        );
        String line = reader.readLine();
        String[] words = line.split(" ");
        for (int i = 0; i < words.length - 1; i++) {
            System.out.format("%s ", words[words.length - 1 - i]);
        }
        System.out.println(words[0]);
    }
}
```

[编程题]25368-简单错误记录

https://www.nowcoder.com/questionTerminal/67df1d7889cf4c529576383c2e647c48

【题目解析】：

题目看起来很复杂，但其实仔细分析下并不难，就是考察一个 Map 的使用。

【解题思路】：

我们只需要把文件名和行号作为 key，出现次数作为 value 即可。但有几个地方要特别注意。

【示例代码】：

```
import java.util.Collections;
import java.util.Comparator;

import java.util.LinkedHashMap;
```

```
import java.util.LinkedList;
import java.util.List;
import java.util.Map;
import java.util.Map.Entry;
import java.util.Scanner;


public class Main {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        // 这里必须用，LinkedHashMap，按插入顺序排序。随后之后会按照出错次数再排序，但如果出错次数一样，还是要按照插入的顺序来
        // 所以这里必须用 LInkedHashMap
        Map<String, Integer> map = new LinkedHashMap<String, Integer>();
        while(in.hasNext()){
            String path = in.next();
            int id = path.lastIndexOf('\\');
            String filename = id == -1 ? path : path.substring(id + 1);
            int line = in.nextInt();
            //统计频率
            String key = filename + " " + line;
            if(map.containsKey(key)){
                map.put(key, map.get(key) + 1);
            }else{
                map.put(key, 1);
            }
        }

        // 对记录进行排序，这里有个前提，就是 java 中的排序用的是归并排序，是稳定排序
        // 这样，如果出错次数一样多，仍然保持插入顺序
        List<Map.Entry<String, Integer>> list = new LinkedList<Map.Entry<String, Integer>>(map.entrySet());
        Collections.sort(list, new Comparator<Map.Entry<String, Integer>>(){
            @Override
            public int compare(Entry<String, Integer> a, Entry<String, Integer> b) {
                return b.getValue() - a.getValue();
            }
        });

        //只输出前8条
        int m = 0;
        for(Map.Entry<String, Integer> mapping : list){
            if (m >= 8) {
                break;
            }

            String[] str = mapping.getKey().split(" ");
            String filename = str[0];
            if (filename.length() > 16) {
                filename = filename.substring(filename.length() - 16);
            }
            String n = str[1];
            Integer count = mapping.getValue();

            System.out.printf("%s %s %d%n", filename, n, count);
```

```
        m++;
      }
    }
}
```