****

**面向对象程序设计上机实验报告**

实验题目2：Set/Map/List及文件处理

**学院名称 智能与计算学部**

**专 业 软件工程**

**学生姓名 陈昊昆**

**学 号 3021001196**

**年 级 2021级**

**班 级 3班**

**时 间 2022年 4月10日**

1. **实验目的**

熟悉Eclipse/MyEclipse；

熟悉Java中Set/Map/List File的使用；

1. **实验内容**

字符串content是一个超市的历次购物小票的合计，每次购物的明细之间用分号分割，每个商品之间用半角逗号分开。请开发程序完成如下功能:

找出哪两个商品总被同时购买的频率最高(不考虑商品的先后顺序)；

找出哪N个商品(N=2)总被同时购买的频率最高。（选做）

1. **程序实现**

**public** String getFrequentItem(String content) {

Map map = **new** HashMap();

// 按购买次数分割字符串

String part[] = content.split(";");

**for**(**int** u = 0; u < part.length; u++) {

String single[] = part[u].split(",");

**int** comnum = single.length \* (single.length - 1);

StringBuffer combine[] = **new** StringBuffer[comnum];

**for**(**int** e = 0; e < comnum; e++) {

combine[e] = **new** StringBuffer("");

}

**int** t = 0;

// 从前往后 构造两件商品的字符串

**for**(**int** w = 0; w < single.length; w++) {

String c = single[w] + ",";

**for**(**int** y = w + 1; y < single.length; y++) {

combine[t].append(c);

combine[t].append(single[y]);

t++;

}

}

// 从后往前 构造两件商品的字符串

**for**(**int** w = single.length-1; w > -1; w--) {

String c = single[w] + ",";

**for**(**int** y = w -1; y > -1; y--) {

combine[t].append(c);

combine[t].append(single[y]);

t++;

}

}

// 将两件商品作为key 使用map统计同时购买的次数

**for** (**int** i = 0; i < t; i++) {

**int** count = 1;

**if** (map.containsKey(combine[i].toString())) {

count = (Integer) map.get(combine[i].toString()) + 1;

}

map.put(combine[i].toString(), count);

}

}

// 找出购买次数前2的两件商品

StringBuffer frestr = **new** StringBuffer("");

**int** maxcnt = 0;

String fres = "";

Iterator it = map.keySet().iterator();

**while**(it.hasNext()) {

String key = (String)it.next();

**if** ((Integer)map.get(key) > maxcnt) {

maxcnt = (Integer)map.get(key);

fres = key;

}

}

frestr.append(fres);

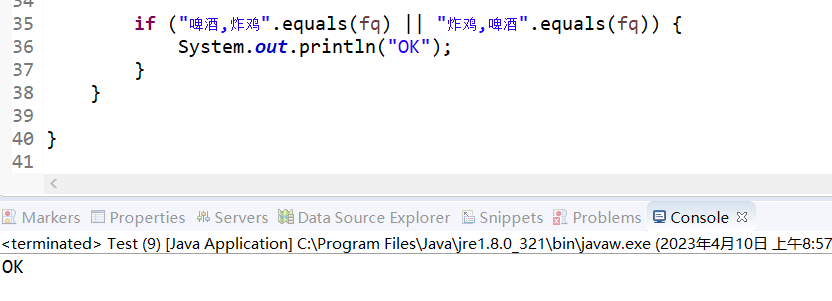
map.remove(fres);

String s = frestr.toString();

**return** s;

}

1. **实验结果**



1. **实验中遇到的问题及解决方法**

这个实验中，我掌握了对字符串的处理，比如转化为char数组和通过某个符号分割字符串。还学会了如何创建map，已经将统计值写入和覆盖进入map。

问题：构造每次购买中同时购买的两件商品

解决：先按照购买次数分割，再将商品单个分离出，从前往后和从后往前分别遍历组合出同时购买的两件商品