Java: Grouping Transactions by Item’s Names

import java.io.\*;

import java.math.\*;

import java.security.\*;

import java.text.\*;

import java.util.\*;

import java.util.concurrent.\*;

import java.util.function.\*;

import java.util.regex.\*;

import java.util.stream.\*;

import static java.util.stream.Collectors.joining;

import static java.util.stream.Collectors.toList;

class Result {

/\*

\* Complete the 'groupTransactions' function below.

\*

\* The function is expected to return a STRING\_ARRAY.

\* The function accepts STRING\_ARRAY transactions as parameter.

\*/

public static List<String> groupTransactions(List<String> transactions) {

// Write your code here

List<String> output = new ArrayList<String>(transactions.size());

Map<String, Integer> map = new HashMap<String, Integer>();

for (String temp : transactions) {

Integer count = map.get(temp);

map.put(temp, (count == null) ? 1 : count + 1);

}

for (Map.Entry<String, Integer> entry : map.entrySet()) {

output.add(entry.getKey() + " " + entry.getValue());

}

Collections.sort(output);

return output;

}

}

public class Solution {

public static void main(String[] args) throws IOException {

BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(System.in));

BufferedWriter bufferedWriter = new BufferedWriter(new FileWriter(System.getenv("OUTPUT\_PATH")));

int transactionsCount = Integer.parseInt(bufferedReader.readLine().trim());

List<String> transactions = IntStream.range(0, transactionsCount).mapToObj(i -> {

try {

return bufferedReader.readLine();

} catch (IOException ex) {

throw new RuntimeException(ex);

}

})

.collect(toList());

List<String> result = Result.groupTransactions(transactions);

bufferedWriter.write(

result.stream()

.collect(joining("\n"))

+ "\n"

);

bufferedReader.close();

bufferedWriter.close();

}

}