**JAVA Code**

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package newlist;

import java.util.List;

import java.util.ArrayList;

import java.util.Arrays;

import java.util.HashMap;

import java.util.HashSet;

import java.util.Map;

import java.util.Set;

/\*\*

\*

\* @author Minhao.Jin17

\*/

public class NewList<E> extends ArrayList<E> {

private Object[] data; //’data’ is used to store the input array

private int manyItems; //’manyItems’ is to record the data’ s length

public NewList(E[] data) { //the constructor

this.data = data;

manyItems = data.length;

}

public <E> int remove1(E target) { //task 1

for (int index = 0; index < manyItems; index++) {

if (data[index] == target) {

data[index] = data[manyItems - 1]; //if the target is found,then move the last item into the current position

manyItems--; //the whole length of the array decreases 1

index--; //to ensure that the last item is not the target which should be removed

}

}

data = Arrays.copyOf(data, manyItems); //refresh 'data', copy the new array to 'data'

return data.length;

}

public <E> int remove2() { //task 2

Map<Object, Integer> helper = new HashMap();

for (Object o : data) {

helper.put(o, helper.getOrDefault(o, 0) + 1);

//if the current item does not exist in the map, then set itself as key and value equals 0.

//else if the current item has already existed in the map, the add 1 to its value.

}

List<Object> target = new ArrayList();

for (Map.Entry<Object, Integer> entry : helper.entrySet()) {

if (entry.getValue() == 3) { //if one item appears 3 times

target.add(entry.getKey()); //then add this item to the target list, all the item in this list will be removed later

}

}

for (int index = 0; index < manyItems; index++) {

if (target.contains(data[index])) { //if current item is in the target list

for (int j = index; j < manyItems - 1; j++) { //then from this index, move the rest part of the array forward 1 position

data[j] = data[j + 1];

}

manyItems--; //the whole length of the array decreases 1

index--; //to ensure that next item is not the target which should be removed

}

}

data = Arrays.copyOf(data, manyItems);

System.out.println("The new List: " + Arrays.toString(data)); //print out the new array

return data.length;

}

public <E> List<List<String>> anagrams() { //task 3

String[] str = new String[data.length];

for (int i = 0; i < data.length; i++) {

str[i] = data[i].toString(); //convert the Object array to String array

}

if (data.length == 0) { //to ensure that the array's length>0, or return the empty list

return new ArrayList();

}

Map<String, Set> helper = new HashMap<String, Set>(); //create a map which key is String type and value is Set

for (String s : str) {

char[] ca = s.toCharArray(); //convert the current String item to a character array for sort

Arrays.sort(ca); //sort this character array

String K = String.valueOf(ca); //merge this character array to a String and stored in 'K' as the key

if (!helper.containsKey(K)) { //if the map does not contain the key

helper.put(K, new HashSet()); //then put this new key into the map and create an empty HashSet as the value

}

helper.get(K).add(s); //add the current String item into the Set according to its key

}

List<List<String>> res = new ArrayList(); //create a nested List to store the result

for (Set set : helper.values()) {

List<String> list = new ArrayList(set); //convert the Set to List

if (list.size() > 1) { //if this list contains more than one items

res.add(list); //the add this list to the result list

}

}

return res;

}

public static void main(String[] args) {

Integer[] A = {6, 2, 1, 3, 2};

NewList test\_A = new NewList(A);

System.out.println("teat A result: " + test\_A.remove1(2));

String[] B = {"accordion", "clarinet", "banjo", "clarinet", "clarinet", "cLarinet"};

NewList test\_B = new NewList(B);

System.out.println("teat B result: " + test\_B.remove1("clarinet"));

Integer[] C = {1, 1, 1, 2, 2, 3};

NewList test\_C = new NewList(C);

System.out.println("teat C result: " + test\_C.remove2());

String[] D = {"accordion", "banjo", "clarinet", "banjo", "clarinet", "clarinet"};

NewList test\_D = new NewList(D);

System.out.println("teat D result: " + test\_D.remove2());

String[] E = {"tea", "eat", "acde", "ate", "adec", "cdf", "cdf", "tea", "adecc"};

NewList test\_E = new NewList(E);

System.out.println("teat E result: " + test\_E.anagrams());

}

}