代码4

GitHub地址：https://github.com/HuaZhouyang/Course\_JavaProgramming

———————————————————————————————————————

**Unit 7:**

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

***7-3 CountOccurrenceOfNumbers.java:***

package Unit\_7;

import java.util.\*;

public class \_03\_CountOccurrenceOfNumbers {

public static void main(String[] args) {

Map<Integer, Integer> map = new TreeMap<>();

Scanner sc = new Scanner(System.in);

int n;

while ((n = sc.nextInt()) != 0) {

if (map.containsKey(n)) {

map.put(n, map.get(n) + 1);

} else {

map.put(n, 1);

}

}

Set<Integer> keys = map.keySet();

for (Integer key : keys) {

int t = map.get(key);

System.out.println(key + " occurs " + t + (t == 1 ? " time" : " times"));

}

}

}

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

***7-5 PrintDistinctNumbers.java:***

package Unit\_7;

import java.util.Scanner;

import java.util.StringJoiner;

import java.util.TreeSet;

public class \_05\_PrintDistinctNumbers {

public static void main(String[] args) {

TreeSet<Integer> set = new TreeSet<>();

System.out.print("Enter 10 numbers: ");

Scanner sc = new Scanner(System.in);

for (int i = 0; i < 10; i++) {

set.add(sc.nextInt());

}

System.out.println("The number of distinct numbers is " + set.size());

System.out.print("The distinct numbers are: ");

StringJoiner sj = new StringJoiner(" ");

for (Integer n : set) {

sj.add(n.toString());

}

System.out.println(sj);

}

}

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

***7-18 BubbleSort.java:***

package Unit\_7;

import java.util.Arrays;

import java.util.Scanner;

public class \_18\_BubbleSort {

public static void main(String[] args) {

System.out.println("Please enter 10 numbers: ");

Scanner sc = new Scanner(System.in);

double[] arr = new double[10];

for (int i = 0; i < 10; i++) {

arr[i] = sc.nextDouble();

}

bubbleSort(arr);

System.out.println("Result: " + Arrays.toString(arr));

}

private static void bubbleSort(double[] a) {

if (a.length <= 1) return;

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

// 提前退出冒泡循环的标志位

boolean flag = false;

for (int j = 0; j < a.length - i - 1; ++j) {

if (a[j] > a[j+1]) { // 交换

double tmp = a[j];

a[j] = a[j+1];

a[j+1] = tmp;

flag = true; // 表示有数据交换

}

}

if (!flag) break; // 没有数据交换，提前退出

}

}

}