猿圈网的几个 Java 挑战题评测题分享

作者：iTanken
原文链接：https://ld246.com/article/1490286107104
来源网站：链滴

先不管猿圈网是干什么的，反正我觉得那是一个可以做题，可以敲代码的网站
aughing ，昨天意外发现了这个网站，在上面做了几道 Java 的题，总感觉哪里做的不太对劲，分享来相互交流一下，帮忙看看我做的到底哪块是不对的。

相关说明：

代码中从注释 //No.1 开始到注释 //end_code 结束是需要自己写代码实现功能的地方，除此部分之的代码基本都是试题给定的代码。

1. 插入排序

用 java 代码实现插入排序

```
import java.util.Arrays; // 自己导入

public class InsertSortTest {

    public static void main(String[] args) {
        int[] array = { 49, 38, 65, 97, 76, 13, 27, 14, 10 };
        //No.1
        //开始写代码，用java实现插入排序
        if (array == null || array.length == 0) {    //判断数组不存在或为空
            return;
        }

        int i, j, tmp;
        int len = array.length;
        for (i = 1; i < len; i++) { // 从 1 开始遍历 array
            j = i;
            tmp = array[i];
            while (j > 0 && tmp < array[j - 1]) { // 如果 array[i] < array[i - 1]
                array[j] = array[j - 1]; // array[i] = array[i - 1]
                j--;
            }
            array[j] = tmp; // array[i - 1] = array[i]
        }
        System.out.println(Arrays.toString(array));
        //end_code
    }
}
```

2. ip 转化为整数
> #### `使用 java 代码实现将 IPV4 的 IP 地址转化为对应整数并输出，比如 192.168.199.1 对应整为 3232286465`
>
> ```` java
> public class IpToNumber {
>     public static void main(String[] args) {
>         IpToNumber ipToNumber = new IpToNumber();
>         System.out.println("IPV4的IP地址对应的整数为 : " + ipToNumber.ipToLong("192.168.199
1"));

>   }
>
>   public long ipToLong(String ipAddress) {
>   long result = 0;
>   String[] ipAddressInArray;
>
>   //No.1
>   //开始写代码，将IPV4的IP地址转化为相对应的整数
>   ipAddressInArray = ipAddress.split("\\."); // 将 ip 截为 4 个部分字符串
>   long[] ipLong = new long[4];
>   for (int i = 0; i < ipAddressInArray.length; i++) {
>   ipLong[i] = Long.parseLong(ipAddressInArray[i]); // 分别转为 long 型后存入 ipLong 中
>   }
>   result = (ipLong[0] << 24) + (ipLong[1] << 16) + (ipLong[2] << 8) + ipLong[3]; // 进位移操作
>   //end_code
>   return result;
>   }
> }
> ```

3. 找重复数字和未出现数字

> #### `假设 0-10000 数字中有 2 个数字相同，还有 1 个数字没有出现，仅遍历一次数组找出重复和未出现的数字。`
>
> ```java
> public class FindNumberTest {
>     public static void main(String[] args) {
>         int number[] = new int[10001];
>         int numCopy[] = new int[10001]; // 这个变量我没有用到...不知道是用来干嘛的
>         int repeat = 0, notAppear = 0;
>         int sumNumber = 0;
>         int i;
>
>         for (i = 0; i < 10001; i++) {//数组初始化
>             number[i] = i;
>         }
>         number[573] = 5236;//设定重复数字5236出现两次，573不出现
>
>         //No.1
>         //开始写代码，仅遍历一次数组找出重复数和未出现的数字。
>         for (int j = 0; j < number.length; j++) {
>             if(j != number[j] && number[j] == number[sumNumber]) {  // 因为数组 number[0] = 0, number[1] = 1... 所以这算不算是投机取巧 :joy:
>                 repeat = number[j];
>                 notAppear = j;
>             }
>             sumNumber++;
>         }
>         //end_code
>         System.out.println("重复数字："+repeat + " 未出现数字：" + notAppear);
>     }
> }
> ```

4. 约瑟夫环问题
> #### `用 java 代码实现约瑟夫环问题，50 个人围成一圈报数，报到 3 的倍数的离开，求最后剩下那个人原来站的位置`
> ```` java
> // 在主方法中调用
> public static void YueSeFuLoop() {
>     LinkedList linkedlist = new LinkedList();
>     //No.1
>     //开始写代码，有50人围成一圈报数，报到3的倍数的人离开，求最后剩下的人原来站在第几位 实现removeFromList方法
>     for (int i = 1; i <= 50; i++) {
>         linkedlist.add(i);
>     }
>     int index = 0;
>     while(linkedlist.size() > 0) { // 最后一个人离开之前
>         for (int i = 0; i < (3 - 1); i++) { // 如果不是 3 的倍数
>             int num = linkedlist.remove(0); // 移除不是 3 的倍数的数，第一次开始为 1
>             linkedlist.add(num); // 将不是 3 的倍数的数放在最后，第一次开始是把 1 放在 50 之后
>         }
>         // 未进入 for 循环，说明是 3 的倍数，则从列表中移除
>         index = linkedlist.remove(0);
>         System.out.println("end : " + linkedlist.size() + " \t index: " + index);
>     }
>     // while 循环执行完毕，最后一个人已被移除
>     System.out.println(index); // 最后一个人开始所站的位置 11
> }
> ```

5. 阿姆斯特朗数问题
> #### `例如 153=1^3+5^3+3^3 的数叫做 Armstrong 数，用 java 代码实现输出三位数的 Armstrong 数`
> ```` java
// 同样还是在主方法里调用 :grin:
> public static void armstrongTest() {
>   int hundredsDigit,tensDigit,unitsDigit;//hundredsDigit表示数字的百位，tensDigit表示数字十位，unitsDigit表示数字的个位
>   System.out.println("寻找Armstrong数：");
>   for (int i = 100; i <= 999; i++) {
>   //No.1
>   //开始写代码，例如153可以满足1^3 + 5^3 + 3^3 = 153，这样的数称为Armstrong数，出所有三位数中的Armstrong数
>   hundredsDigit = i / 100; // 获取百位数
>   tensDigit = (i % 100) / 10; // 十位数
>   unitsDigit = i % 10; // 个位数
>   // Math.pow(double a, double b) 返回 a 的 b 次方
>   if ((Math.pow(hundredsDigit, 3) + Math.pow(tensDigit, 3) + Math.pow(unitsDigit, 3)) = i) // 如果是Armstrong数则输出
>   System.out.print(i + " ");
>   //end_code
>   }
>   System.out.println();
> }
> ```

6. 统计指定类型字符个数
> #### `用 java 代码实现输入一串字符串，统计其中的数字、英文、空格、其他字符个数`
> ```` java
public static void strLength() {
    int digital = 0;//数字个数
    int character = 0;//英文个数
    int other = 0;//其他字符个数
    int blank = 0;//空格个数
    char[] chars = null;
    System.out.println("这是任意一串字符：");
    String string = "djfiepqo ioghr4 8758495 7123hr37hfjW$@@$@^%!";
    chars = string.toCharArray();
    //No.1
    //开始写代码，计算任意一串字符中的数字个数、英文字母个数、空格个数和其他字符个数
    for (int i = 0; i < chars.length; i++) {
        if(chars[i] >= '0' && chars[i] <= '9') { // 数字
            digital++;
        } else if((chars[i] >= 'a' && chars[i] <='z') || (chars[i] >= 'A' && chars[i] <='Z')) { // 英文
母
            character++;
        } else if(chars[i] == ' ') { // 空格
            blank++;
        } else { // 其他字符
            other++;
        }
    }
    //end_code
    System.out.println("数字个数: " + digital);
    System.out.println("英文字母个数: " + character);
    System.out.println("空格个数: " + blank);
    System.out.println("其他字符个数:" + other);
}
> ```

7. 用递归方法计算一个数的阶乘
> #### `给定一个正整数，Java 实现用递归的方法计算它的阶乘`
> ```` java
> public class Test {
>     public static void main(String[] args) {
>         int number = 12;
>         Recursion factorialRecursion = new Recursion();
>         System.out.println(number + "! = " + factorialRecursion.recursion(number));
>     }
> }
> //No.1
> //开始写代码，给定一个正整数，用递归的方法计算它的阶乘.main函数已给出，实现Recursion类
> class Recursion {
>     public int recursion(int number) {
>         if(number < 0) {
>             System.out.println("请输入0或正整数!");
>             return 0;
>         } else if(number == 1 || number == 0) {
>             return 1;

>         } else {
>             return number * recursion(number - 1);
>         }
>     }
> }
> //end_code
> ```
>
> 艾玛呀，一不小心又到凌晨了....