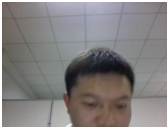


Java方向每日一题day08_12月1日-任栋-测评结果

考生信息



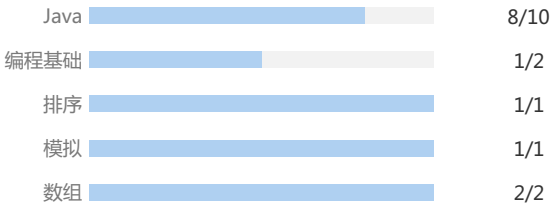
任栋
投递编号：2 | 学校：西安理工大学 | 邮箱：1104580363@qq.com | 职位：2020大四春招冲刺班 |
参考区域: 陕西省西安市 (221.11.20.102) |
做题用时：05:13:06(2020-11-30 22:18:58开始答题，2020-12-01 22:29:01交卷) | 作答设备：PC |
已同意诚信声明和隐私协议

考生成绩



题型	得分	正确题数	排名	用时	是否阅卷
单选	40.0	8	51	00:24:34	--
编程	50.0	2	1	04:35:13	--

知识点技能图谱



知识点	得分	正确题数
Java	40.0	8
编程基础	5.0	1
排序	25.0	1
模拟	25.0	1
数组	30.0	2

历史笔试记录

序号	试卷名称	排名	总得分	得分详情	作弊嫌疑	安排笔试时间	交卷时间
1	大四春招冲刺班JavaSE考试	31.0%	26.0/60	单选:26.0分	否	2020-10-28 17:33:16	2020-10-29 10:30:13
2	大四春招冲刺班数据结构考试	77.0%	22.0/60	单选:22.0分 编程:0.0分	否	2020-11-23 11:55:15	2020-11-24 10:40:01
3	Java方向每日一题day02_11月24日	70.0%	55.0/100	单选:30.0分 编程:25.0分	否	2020-11-23 12:10:19	2020-11-25 15:36:05
4	Java方向每日一题day03_11月25日	79.0%	55.0/100	单选:30.0分 编程:25.0分	否	2020-11-24 15:19:25	2020-11-24 22:21:11
5	Java方向每日一题day04_11月26日	61.0%	85.0/100	单选:35.0分 编程:50.0分	否	2020-11-25 14:58:20	2020-11-25 23:17:45

序号	试卷名称	排名	总得分	得分详情	作弊嫌疑	安排笔试时间	交卷时间
6	Java方向每日一题day05_11月27日	16.0%	85.0/100	单选:35.0分 编程:50.0分	否	2020-11-25 15:54:38	2020-11-26 23:32:02
7	Java方向每日一题day06_11月28日	43.0%	77.5/100	单选:30.0分 编程:47.5分	否	2020-11-27 14:19:26	2020-11-27 22:53:24
8	Java方向每日一题day07_11月30日	1.0%	100.0/100	单选:50.0分 编程:50.0分	否	2020-11-29 13:58:20	2020-11-30 22:17:40

编码能力

题号	正确性	提交次数	做题用时	使用语言	运行时间	占用内存	编程思路	代码规范	成绩排名
编程题1	100%	5	01:58:17	Java	26ms	10680K			1%
编程题2	100%	13	02:36:56	Java	28ms	10528K			1%

1 [平均分3.4分 | 80人正确/119人做题 | 用时 : <1分 | 得分 : 5.0 / 5.0

java中关于内存回收的正确说法是

- A 程序员必须创建一个线程来释放内存
- B 内存回收程序负责释放无用内存 **程序回收内存**
- C 内存回收程序允许程序员直接释放内存
- D 内存回收程序可以在指定的时间释放内存对象

他的回答 : B (正确)
正确答案 : B

2 [平均分4.5分 | 109人正确/120人做题 | 用时 : <1分 | 得分 : 5.0 / 5.0

A派生出子类B，B派生出子类C，并且在java源代码中有如下声明：

```
A a0=new A();
A a1=new B();
A a2=new C();
```

以下哪个说法是正确的？

- A 第1行，第2行和第3行的声明都是正确的
- B 第1,2,3行都能通过编译，但第2,3行运行时出错
- C 第1,2行能通过编译，但第3行编译出错
- D 只有第1行能通过编译

他的回答 : A (正确)
正确答案 : A

3 [平均分4.0分 | 96人正确/119人做题 | 用时 : <1分 | 得分 : 5.0 / 5.0

以下说法错误的是（ ）

- A 数组是一个对象
B 数组不是一种原生类
C 数组的大小可以任意改变 **不可以，系统会分配固定的空间**
D 在Java中，数组存储在堆中连续内存空间里

他的回答： C (正确)

正确答案： C

4 [平均分1.9分 | 46人正确/120人做题 | 用时：12分] 得分：0.0 / 5.0

Test.main() 函数执行后的输出是 ()

```
public class Test {  
    public static void main(String [] args){  
        System.out.println(new B().getValue());  
    }  
  
    static class A{  
        protected int value;  
        public A(int v) {  
            setValue(v);  
        }  
        public void setValue(int value){  
            this.value = value;  
        }  
        public int getValue(){  
            try{ 1  
                value++;  
                return value;  
            } catch (Exception e){  
                System.out.println(e.toString());  
            } finally {  
                this.setValue(value);  
                System.out.println(value);  
            }  
            return value; 2  
        }  
    }  
  
    static class B extends A{  
        public B(){  
            super(5);  
            setValue(getValue() - 3);  
        }  
        public void setValue(int value){  
            super.setValue(2 * value);  
        }  
    }  
}
```

1. value=5

2.value = (5+1) 返回6
print :

- A 11 17 34
B 22 74 74
C 6 7 7
D 22 34 17

他的回答： C (错误)

正确答案： D

5 [平均分4.8分 | 115人正确/120人做题 | 用时：<1分] 得分：5.0 / 5.0

关于Java的异常处理机制的叙述哪些正确？

- A 如果程序发生错误及捕捉到异常情况了，才会执行finally部分
B 其他选项都不正确
C 当try区段的程序发生异常且被catch捕捉到时，才会执行catch区段的程序
D catch部分捕捉到异常情况时，才会执行finally部分

一般情况下finally绝对会执行
除非前面加了System.exit(0)

他的回答： C (正确)

正确答案： C

6 [平均分4.0分 | 95人正确/120人做题 | 用时：<1分 | 得分：5.0 / 5.0]

如何跳出Array的forEach循环？（ ）

- A break
B return true
C return false
D 以上都不是

他的回答： A (正确)

正确答案： A

7 [平均分4.5分 | 108人正确/119人做题 | 用时：<1分 | 得分：5.0 / 5.0]

一个以“.java”为后缀的源文件

- A 只能包含一个类，类名必须与文件名相同
B 只能包含与文件名相同的类以及其中的内部类
C 只能有一个与文件名相同的类，可以包含其他类
D 可以包含任意类

可以包含多个类，但只能有一个与文件名相同的类

他的回答： C (正确)

正确答案： C

8 [平均分4.5分 | 107人正确/120人做题 | 用时：<1分 | 得分：0.0 / 5.0]

如下Java语句

```
double x= 3.0;  
int y=5;  
x/--y;
```

<https://baike.baidu.com/item/%E8%BF%90%E7%AE%97%E7%AC%A6%E4%BC%98%E5%85%88%E7%BA%A7/4752611?fr=aladdin>

--运算符优先级大于/=赋值运算符

执行后，x的值是（ ）

- A 3
B 0.6
C 0.4
D 0.75

他的回答： B (错误)

正确答案： D

9 [平均分4.7分 | 112人正确/120人做题 | 用时：3分 | 得分：5.0 / 5.0]

```
public class Test{
    static int cnt = 6;
    static{
        cnt += 9;
    }
    public static void main(String[] args ) {
        System.out.println ( "cnt ="  + cnt);
    }
    static{
        cnt /=3;
    };
}
```

9+6后 除3

cnt的值是

- A cnt=5
- B cnt=2
- C cnt=3
- D cnt=6

他的回答： A (正确)
正确答案： A

10 [平均分4.3分 | 103人正确/120人做题 | 用时： <1分 | 得分： 5.0 / 5.0

JUnit主要用来完成什么

get post

- A 发送HTTP请求
- B 建立TCP连接
- C 集成测试
- D 单元测试

他的回答： D (正确)
正确答案： D

11 ACM编程题 语言限制 [平均分19.9分 | 82人正确/103人做题 | 提交: 5 次 | 得分： 25.0 / 25.0

标题：删数 | 时间限制：1秒 | 内存限制：32768K | 语言限制： 不限

【删数】有一个数组a[N]顺序存放0~N-1，要求每隔两个数删掉一个数，到末尾时循环至开头继续进行，求最后一个被删掉的数的原始下标位置。以8个数(N=7)为例：{ 0 , 1 , 2 , 3 , 4 , 5 , 6 , 7 }，0->1->2(删除)->3->4->5(删除)->6->7->0(删除),如此循环直到最后一个数被删除。

输入描述：

每组数据为一行一个整数n(小于等于1000)，为数组成员数,如果大于1000，则对a[999]进行计算。

输出描述：

一行输出最后一个被删掉的数的原始下标位置。

示例1：

输入

8

输出

6

代码片段

功能实现

代码提交统计

代码执行统计

总通过率	TA的 100%	平均 79%	使用语言	TA的 Java	平均 01:58:17 00:48:34	答案错误 : 3
基本测试用例通过率	6/6 (100%)	79%	做题用时	5	5	答案正确 : 2
边缘测试用例通过率	4/4 (100%)	79%	提交次数			
代码效率			代码规范及可读性			
运行时间	TA的 26ms	参考 1s	代码规范得分			
占用内存	10680K	32768K	Line 2: 'CLASS_DEF' 前应有空行。 [EmptyLineSeparator]			
			Line 2: 顶级类 Node 应位于它自己的源文件中。 [OneTopLevelClass]			
			Line 10: 'CLASS_DEF' 前应有空行。 [EmptyLineSeparator]			
			Line 14:17: Local variable name 'n' must match pattern '[a-zA-Z0-9]*\$'. [LocalVariableName]			
			4.2			

他的代码：

做题用时: 118 分钟 语言：Java 运行时间：26ms 占用内存：10680K 程序状态：答案正确

```
import java.util.*;
class Node {
    int data;
    Node next;

    public Node(int data) {
        this.data = data;
    }
}
public class Main{
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        while (scanner.hasNextInt()){
            int n = scanner.nextInt();
            Node head = new Node(0);
            Node cur = head;
            if (n > 1000){
                n = 1000;
            }
            for (int i = 1; i < n; i++) {
                Node node = new Node(i);
                cur.next = node;
                cur = cur.next;
            }
            cur.next = head;
            cur = head;
            while (cur.next != cur){
                cur.next.next = cur.next.next.next;
                cur = cur.next.next;
            }
            System.out.println(cur.data);
        }
    }
}
```



点此或手机扫描二维码查看代码编写过程

12

ACM编程题

语言限制

平均分20.4分 | 91人正确/112人做题 | 提交: 13 次

得分 : 25.0 / 25.0

标题 : n个数里最小的k个 | 时间限制 : 1秒 | 内存限制 : 32768K | 语言限制 : 不限

【n个数里最小的k个】找出n个数里最小的k个

输入描述 :

每个测试输入包含空格分割的n+1个整数，最后一个整数为k值,n
不超过100。

输出描述 :

输出n个整数里最小的k个数。升序输出

示例1 :

输入

3 9 6 8 -10 7 -11 19 30 12 23 5

输出

-11 -10 3 6 7

代码片段									
功能实现			代码提交统计				代码执行统计		
总通过率	TA的	平均		TA的	平均		答案错误 : 7		
	100%	81%		使用语言	Java		返回非零 : 2		
	2/2	81%		做题用时	02:36:56	00:31:44	编译错误 : 3		
	(100%)			提交次数	13	4	答案正确 : 1		
边缘测试用例通过率	1/1	82%							
	(100%)								
代码效率						代码规范及可读性			
	TA的	参考	代码规范得分						4.5
运行时间	28ms	1s	Line 13:17: Local variable name 'k' must match pattern '[a-z][a-z0-9][a-zA-Z0-9]*\$'. [LocalVariableName]						
占用内存	10528K	32768K	Line 21:48: Parameter name 'k' must match pattern '[a-z][a-z0-9][a-zA-Z0-9]*\$'. [ParameterName]						

他的代码 :

做题用时: 156 分钟 语言 : Java 运行时间 : 28ms 占用内存 : 10528K 程序状态 : 答案正确

```
import java.util.*;

public class Main{
```

```

public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
    while (scanner.hasNextLine()){
        String str = scanner.nextLine().trim();
        String[] split = str.split(" ");
        int[] array = new int[split.length-1];
        for (int i = 0; i < split.length-1; i++) {
            array[i] = Integer.parseInt(split[i]);
        }
        int k = Integer.parseInt(split[split.length-1]);
        int[] func = func(array,k);
        for (int i :func) {
            System.out.print(i+ " ");
        }
    }
}

private static int[] func(int[] array, int k) {
    PriorityQueue<Integer> priorityQueue = new PriorityQueue<> (new Comparator<Integer>() {
        @Override
        public int compare(Integer o1, Integer o2) {
            return o2-o1;
        }
    });
    for (int i = 0; i < array.length; i++) {
        if (i < k) {
            priorityQueue.offer(array[i]);
        }else if (array[i] < priorityQueue.peek()){
            priorityQueue.poll();
            priorityQueue.offer(array[i]);
        }
    }
    int[] arr = new int[k];
    for (int i = 0; i < k; i++) {
        arr[k-1-i] = priorityQueue.poll();
    }
    return arr;
}
}

```



[点此或手机扫描二维码查看代码编写过程](#)