

Java方向每日一题day04_11月26日-任栋-测评结果

考生信息



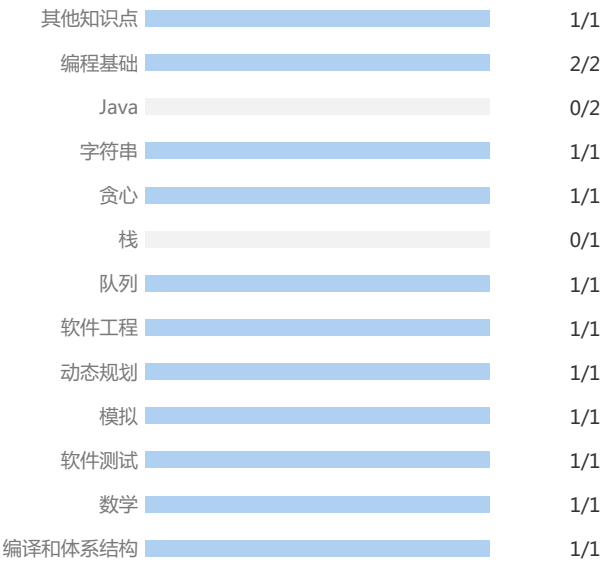
任栋
投递编号：2 | 学校：西安理工大学 | 邮箱：1104580363@qq.com | 职位：2020大四春招冲刺班 |
参考区域: 陕西省西安市 (221.11.20.102) | 做题用时：01:13:18(2020-11-25 22:04:22开始答题，23:17:45交卷) |
作答设备：PC | 已同意诚信声明和隐私协议

考生成绩



题型	得分	正确题数	排名	用时	是否阅卷
单选	35.0	7	103	00:07:32	--
编程	50.0	2	1	01:03:56	--

知识点技能图谱



知识点	得分	正确题数
其他知识点	5.0	1
编程基础	10.0	2
Java	0.0	0
字符串	25.0	1
贪心	25.0	1
栈	0.0	0
队列	5.0	1
软件工程	5.0	1
动态规划	25.0	1
模拟	25.0	1
软件测试	5.0	1
数学	25.0	1
编译和体系结构	5.0	1

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

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%	8	00:41:27	Java	30ms	10664K			1%
编程题2	100%	2	00:22:29	Java	23ms	10524K			1%

1 [平均分4.6分 | 115人正确/126人做题 | 用时：<1分] 得分：5.0 / 5.0

下列与队列结构有关联的是（ ）

- A 函数的递归调用
- B 数组元素的引用
- C 多重循环的执行
- D 先到先服务的作业调度

他的回答： D (正确)

正确答案： D

参考答案：
队列的修改是依先进先出的原则进行的， D 正确。

2 [平均分4.0分 | 101人正确/125人做题 | 用时：<1分] 得分：5.0 / 5.0

下面属于黑盒测试方法的是（ ）

- A 语句覆盖
- B 逻辑覆盖
- C 边界值分析
- D 路径覆盖

他的回答： C (正确)

正确答案： C

参考答案：
黑盒测试不关心程序内部的逻辑，只是根据程序的功能说明来设计测试用例。在使用黑盒测试法时，手头只需要有程序功能说明就可以了。黑盒测试法分等价类划分法、边界值分析法和错误推测法，答案为 C。而 A、 B、 D 均为白盒测试方法。

3 [平均分4.9分 | 124人正确/126人做题 | 用时：<1分] 得分：5.0 / 5.0

下列不能用作存储容量单位的是（ ）。

- A Byte
- B GB
- C MIPS
- D KB

他的回答： C (正确)

正确答案： C

参考答案：

计算机存储信息的最小单位，称之为位（ bit ，又称比特），存储器中所包含存储单元的数量称为存储容量，其计量基本单位是字节（ Byte ，简称 B ）， 8 个二进制位称为 1 个字节，此外还有 KB 、 MB 、 GB 、 TB 等。 MIPS 即 Million Instruction Per Second 的简写 - - 计算机每秒钟执行的百万指令数，是衡量计算机速度的指标。

4 [平均分4.6分 | 117人正确/126人做题 | 用时： <1分 | 得分： 5.0 / 5.0

一个教师可讲授多门课程，一门课程可由多个教师讲授。则实体教师和课程间的联系是（ ）。

- A 1：1联系
- B 1：m联系
- C m：1联系
- D m：n联系

他的回答： D (正确)

正确答案： D

参考答案：

因为一个教师可讲授多门课程，而一门课程又能由多个老师讲授所以他们之间是多对多的关系，可以表示为 m：n。

5 [平均分4.9分 | 122人正确/125人做题 | 用时： <1分 | 得分： 0.0 / 5.0

下列关于栈叙述正确的是（ ）。

- A 算法就是程序 **算法是解决问题的步骤**
- B 设计算法时只需要考虑数据结构的设计
- C 设计算法时只需要考虑结果的可靠性
- D 以上三种说法都不对

他的回答： B (错误)

正确答案： D

参考答案：

算法是指解题方案的准确而完整的描述，算法不等于程序，也不等于计算方法，所以 A 错误。设计算法时不仅要考虑对数据对象的运算和操作，还要考虑算法的控制结构。

System.arraycopy > clone > Arrays.copyOf > for循环

6 [平均分3.1分 | 77人正确/126人做题 | 用时： <1分 | 得分： 0.0 / 5.0 **abc都是浅复制 d深复制**

java语言的下面几种数组复制方法中，哪个效率最高？

- A for 循环逐一复制 **for性能最差**
- B System.arraycopy **将指定源数组中的数组从指定位置复制到目标数组的指定位置：本地方法**
- C Array.copyOf **Arrays.copyOf 不属于本地方法（但本质上调用了System.arraycopy）**
- D 使用clone方法 **2.先实现cloneable接口，再重写 再调用：本地方法**

他的回答： C (错误)

正确答案： B https://blog.csdn.net/weixin_43392489/article/details/103288308

7 [平均分4.2分 | 107人正确/126人做题 | 用时： <1分 | 得分： 5.0 / 5.0

结构化程序的三种基本结构是（ ）

- A 递归、迭代和回溯
- B 过程、函数和子程序
- C 顺序、选择和循环

为了提升性能,在JVM里对该注解的进行了手写,这里要提一个叫JNI(Java Native Interface)的东西,普通的native方法通俗的讲就是编译后还要通过JNI再次编译成.cpp文件才能执行.而有@HotSpotIntrinsicCandidate这个注解的方法在JVM里就是用.cpp文件写好的,所以就跳过了JNI阶段,所以速度就能提升,这也是System.arraycopy()速度冠绝群雄的原因.
clone方法：属于本地方法它并没有被手工写在JVM里面，所以它不得不走JNI的路子，所以它就成了2哥。

D 调用、返回和选择

他的回答：C (正确)

正确答案：C

参考答案：

仅使用顺序、选择、循环三种基本控制结构就足以表达出各种其他形式结构的程序设计方法。故本题答案为 C 选项。

8 [平均分4.6分 | 115人正确/125人做题 | 用时：<1分 | 得分：5.0 / 5.0

下面属于软件定义阶段任务的是（ ）

- A 需求分析
- B 软件测试
- C 详细设计
- D 系统维护

他的回答：A (正确)

正确答案：A

参考答案：

软件生命周期分为 3 个阶段：软件定义阶段，任务是确定软件开发工作必须完成的目标，确定工程的可行性；软件开发阶段，任务是具体完成设计和实现定义阶段所定义的软件，通常包括总体设计、详细设计、编码和测试；软件维护阶段，任务是使软件在运行中持久地满足用户的需要。需求分析属于软件定义阶段的任务。故本题答案为 A 选项。

9 [平均分4.7分 | 118人正确/126人做题 | 用时：<1分 | 得分：5.0 / 5.0

1GB 的准确值是（ ）。

- A 1024×1024 Bytes
- B 1024 KB
- C 1024 MB
- D 1000×1000 KB

他的回答：C (正确)

正确答案：C

参考答案：

1GB=1024MB=1024 × 1024KB=1024 × 1024 × 1024B。

10 [平均分2.9分 | 71人正确/124人做题 | 用时：<1分 | 得分：0.0 / 5.0

关于 Socket 通信编程，以下描述错误的是：（ ）

- A 服务器端通过new ServerSocket()创建TCP连接对象
- B 服务器端通过TCP连接对象调用accept()方法创建通信的Socket对象
- C 客户端通过new Socket()方法创建通信的Socket对象
- D 客户端通过new ServerSocket()创建TCP连接对象

他的回答：A (错误)

正确答案：D

11 ACM编程题 语言限制 [平均分22.6分 | 96人正确/118人做题 | 提交：8 次 | 得分：25.0 / 25.0

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

【买苹果】小易去附近的商店买苹果，奸诈的商贩使用了捆绑交易，只提供6个每袋和8个每袋的包装(包装不可拆分)。可是小易现在只想购买恰好n个苹果，小易想购买尽量少的袋数方便携带。如果不能购买恰好n个苹果，小易将不会购买。

输入描述：

输入一个整数n，表示小易想购买n(1 ≤ n ≤ 100)个苹果

输出描述：

输出一个整数表示最少需要购买的袋数，如果不能买恰好n个苹果则输出-1

示例1：

输入

20

输出

3

代码片段

功能实现			代码提交统计			代码执行统计	
	TA的	平均		TA的	平均	返回非零 : 1	
总通过率	100%	90%	使用语言	Java		编译错误 : 4	
基本测试用例通过率	6/6 (100%)	91%	做题用时	00:41:27	00:41:05	答案正确 : 3	
边缘测试用例通过率	4/4 (100%)	88%	提交次数	8	5		

代码效率			代码规范及可读性		
	TA的	参考	代码规范得分		4.6
运行时间	30ms	1s	Line 2: 'CLASS_DEF' should be separated from previous statement. [EmptyLineSeparator]		
占用内存	10664K	32768K	Line 5:13: Local variable name 'n' must match pattern '^[a-z][a-z0-9][a-zA-Z0-9]*\$'. [LocalVariableName]		

他的代码：

做题用时: 41 分钟 语言：Java 运行时间：30ms 占用内存：10664K 程序状态：答案正确

```
import java.util.Scanner;
public class Main{
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int n = scanner.nextInt();
        int sum = 0;
        if (n%2 == 1 || n <6 || n == 10) {
            sum = -1;
        }else {
            sum = (n % 8==0) ? n/8: n/8+1;
        }
        System.out.println(sum);
    }
}
```



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22.1分 | 99人正确/115人做题 | 提交: 2 次] 得分 : 25.0 / 25.0

标题：删除公共字符 | 时间限制：1秒 | 内存限制：32768K | 语言限制：不限

【删除公共字符】输入两个字符串，从第一字符串中删除第二个字符串中所有的字符。例如，输入” They are students.” 和” aeiou” ，则删除之后的第一个字符串变成” Thy r stdnts.”

输入描述：

每个测试输入包含2个字符串

输出描述：

输出删除后的字符串

示例1：

输入

They are students. aeiou

输出

Thy r stdnts.

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	返回非零 : 1 答案正确 : 1		
	100%	88%	使用语言	Java				
	基本测试用例通过率	2/2 (100%)	做题用时	00:22:29	00:27:06			
	边缘测试用例通过率	2/2 (100%)	提交次数	2	4			
代码效率			代码规范及可读性					
	TA的	参考	代码规范得分					4.8
运行时间	23ms	1s	Line 2: 'CLASS_DEF' 前应有空行。 [EmptyLineSeparator]					
占用内存	10524K	32768K						

他的代码：

做题用时: 22 分钟 语言：Java 运行时间：23ms 占用内存：10524K 程序状态：答案正确

```
import java.util.*;
public class Main{
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
```

```
String str1 = scan.nextLine();
String str2 = scan.nextLine();
HashSet<Character> hashSet = new HashSet<>();
for (int i = 0; i < str2.length(); i++) {
    hashSet.add(str2.charAt(i));
}
StringBuffer str = new StringBuffer();
for (int i = 0; i < str1.length(); i++) {
    if (!hashSet.contains(str1.charAt(i))) {
        str.append(str1.charAt(i));
    }
}
System.out.println(str);
}
```



[点此或手机扫描二维码查看代码编写过程](#)