

比特就业课105期Java方向笔试强训48天day13_10月24日-王世国-测评结果

考生信息 存在作弊行为



王世国
投递编号：91 | 学校：武汉轻工大学 | 邮箱：1477649017@qq.com | 职位：比特就业课105期Java2班
参考区域: 湖北省武汉市 (183.94.121.252) | 做题用时：01:36:06(2022-10-23 20:11:01开始答题，23:15:22交卷)
作答设备：PC | 已同意诚信声明和隐私协议

65.0 分 / 100分

在本次考试中，考生总成绩为65.0分/100分，评级为**B（排名前31%）**，编程能力**良好（1题通过，分数排名前40%）**，编程思路**部分正确**，编程规范性**高**。该考生在本次考试中**存在**作弊行为，和较早通过的代码相似度高（93.0%），视频监控图片数目过少，有遮挡或关闭摄像头的嫌疑。

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	13	00:12:15	已阅
编程	25.0	1	42	01:20:59	已阅

作弊风险

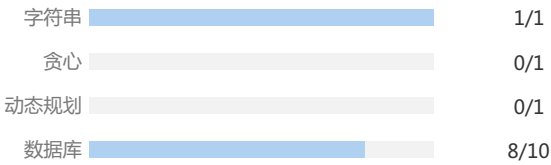
高风险 未开启摄像头

考生可能关闭摄像头，导致视频截图较少

高风险 代码相似度

编程题1 **93.00%**

知识点技能图谱



知识点	得分	正确题数
字符串	25.0	1
贪心	0.0	0
动态规划	0.0	0
数据库	40.0	8

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	比特就业课105期+2022寒假班C1考试	20.0%	48.8/60	单选:30.0分 编程:18.75分	否	2022-03-29 11:16:18	2022-03-31 18:51:27
2	比特就业课105期+2022寒假班C2考试	66.0%	24.0/60	单选:24.0分 编程:0.0分	否	2022-04-11 14:12:23	2022-04-11 20:12:16
3	比特就业课 105期JavaSE考试	11.0%	56.0/60	单选:26.0分 编程:30.0分	否	2022-07-12 16:00:16	2022-07-13 15:48:42
4	比特就业课 105期java方向 数据结构考试	47.0%	50.0/60	单选:20.0分 编程:30.0分	否	2022-07-23 12:49:22	2022-07-25 09:56:28
5	比特就业课105期Java方向笔试强训48天day01_10月10日	36.0%	80.0/100	单选:40.0分 编程:40.0分	是，摄像头监控异常	2022-10-09 17:29:16	2022-10-09 21:57:25
6	比特就业课105期Java方向笔试强训48天day02_10月11日	13.0%	90.0/100	单选:40.0分 编程:50.0分	是，代码抄袭	2022-10-10 10:43:48	2022-10-10 21:13:15
7	比特就业课105期Java方向笔试强训48天day03_10月12日	1.0%	95.0/100	单选:45.0分 编程:50.0分	是，摄像头监控异常	2022-10-11 10:40:53	2022-10-12 10:03:09
8	比特就业课105期Java方向笔试强训48天day04_10月13日	2.0%	95.0/100	单选:30.0分 不定项选择:15.0分 编程:50.0分	是，摄像头监控异常	2022-10-12 10:31:10	2022-10-12 21:14:05
9	比特就业课105期Java方向笔试强训48天day05_10月14日	12.0%	85.0/100	单选:35.0分 不定项选择:0.0分 编程:50.0分	是，摄像头监控异常 代码抄袭	2022-10-13 11:41:43	2022-10-14 11:33:24
10	比特就业课105期Java方向笔试强训48天day06_10月15日	13.0%	86.7/100	单选:25.0分 不定项选择:11.67分 编程:50.0分	是，摄像头监控异常	2022-10-14 10:59:38	2022-10-14 21:18:11
11	比特就业课105期Java方向笔试强训48天day07_10月17日	1.0%	100.0/100	单选:50.0分 编程:50.0分	是，摄像头监控异常	2022-10-16 16:46:53	2022-10-16 20:38:39
12	比特就业课105期Java方向笔试强训48天day08_10月18日	33.0%	87.5/100	单选:40.0分 编程:47.5分	是，摄像头监控异常	2022-10-17 16:18:42	2022-10-17 20:16:45
13	比特就业课105期Java方向笔试强训48天day09_10月19日	3.0%	95.0/100	单选:45.0分 编程:50.0分	是，摄像头监控异常	2022-10-18 17:07:17	2022-10-19 11:02:55
14	比特就业课105期Java方向笔试强训48天day10_10月20日	58.0%	65.0/100	单选:40.0分 编程:25.0分	是，摄像头监控异常	2022-10-19 15:29:54	2022-10-19 21:44:34
15	比特就业课105期Java方向笔试强训48天day11_10月21日	2.0%	95.0/100	单选:45.0分 编程:50.0分	是，摄像头监控异常	2022-10-20 16:46:09	2022-10-20 23:36:15
16	比特就业课105期Java方向笔试强训48天day12_10月22日	15.000001%	90.0/100	单选:40.0分 编程:50.0分	是，摄像头监控异常	2022-10-21 16:44:18	2022-10-21 22:49:24

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%	6	00:52:53	Java	37ms	10712K	良	良	1%
编程题2	0%	5	00:28:06	Java	48ms	11068K	-	良	-

1 [单选题 | 平均分1.63分 | 32人正确/98人做题 | 用时 : <1分 | 得分 : 5.0 / 5.0

一个关系数据库文件中的各条记录 ()。

- A 前后顺序不能任意颠倒，一定要按照输入的顺序排列
- B 前后顺序可以任意颠倒，不影响库中的数据关系
- C 前后顺序可以任意颠倒，但排列顺序不同，统计处理的结果就可能不同
- D 前后顺序不能任意颠倒，一定要按照关键字段值的顺序排列

他的回答 : B (正确)

正确答案 : B

关系型数据库：逻辑性强而物理性弱。数据库文件中的各条记录的顺序是可以颠倒的，因为我们在进行查询的时候，是按照你的查询语句的要求去进行操作的，所以顺序并不影响数据库中的关系。也不会影响查询的结果

2 [单选题 | 平均分4.08分 | 80人正确/98人做题 | 用时 : <1分 | 得分 : 5.0 / 5.0

一名员工可以使用多台计算机，每台计算机只能由一名员工使用，则实体员工和计算机间的联系是 ()

- A 一对多
- B 多对多
- C 多对一
- D 一对一

他的回答 : A (正确)

正确答案 : A

参考答案：

因为一名员工可以使用多台计算机，而一台计算机只能被一名员工使用，所以员工和计算机两个实体之间是一对多的关系。故本题答案为 A 选项。

3 [单选题 | 平均分2.96分 | 58人正确/98人做题 | 用时 : <1分 | 得分 : 0.0 / 5.0

下列关于视图的说法错误的是：

- A 视图是从一个或多个基本表导出的表，它是虚表
- B 视图一经定义就可以和基本表一样被查询、删除和更新
- C 某一用户可以定义若干个视图
- D 视图可以用来定义新的视图

他的回答 : D (错误)

正确答案 : B

视图是一个虚表，它的内容来自于查询结果

create view 视图名 as select

视图在经过定义之后可以和基本表一样进行查询，删除。但是不能进行更新，因为基本表中是没有修改的，你可以查询，可以是删除掉一部分(也就是视图中查不到)，但是不能更新。

一个用户可以定义若干个视图，一个视图可以作为另一个视图的查询结果来源，例如：
create view 视图名 as select ... from 视图

4 [单选题 | 平均分2.5分 | 49人正确/98人做题 | 用时 : <1分 | 得分 : 0.0 / 5.0

在Mysql中，订单表 (订单号，雇员代号，地区代号，订购日期) 中订单号为主键，要删除订单中前三年以前的信息，SQL 为：

- A delete from 订单表 where 订购日期<getdate()+3
- B delete from 订单表 where 订购日期<DATEADD(yy,3,getdate())
- C delete from 订单表 where 订购日期<getdate()-3
- D delete from 订单表 where 订购日期<DATEADD(yy,-3,getdate())

当前时间往前移三年
getdate() 获取到当前系统时间

他的回答 : C (错误)

正确答案 : D

参考答案：

DATEADD() 函数在日期中添加或减去指定的时间间隔。

语法

DATEADD(datepart,number,date)

datepart参数的取值范围是{yy/yyyy,qq/q,mm/m,dy/y,dd/d,wk/ww,dw/w,hh,mi/n,ss/s,ms,mcs,ns} 决定你移动的单位是年，月，日等等

number 是您希望添加的间隔数

date 参数是合法的日期表达式

5 [单选题 | 平均分2.81分 | 55人正确/98人做题 | 用时: <1分 | 得分: 5.0 / 5.0

负责数据库中查询操作的数据库语言是()。

- A 数据定义语言
 - B 数据管理语言
 - C 数据操纵语言
 - D 数据控制语言
- 数据定义语言: DDL, 主要用来创建数据库中的各种对象, 例如表, 视图, 索引等等
数据操纵语言: DML 负责数据的操作, 也就是增删查改数据控制语言
数据控制语言: DCL 负责数据的授权等等

他的回答: C (正确)

正确答案: C

参考答案:

数据定义语言: 负责数据的模式定义与数据的物理存取构建; 数据操纵语言: 负责数据的操纵, 包括查询及增、删、改等操作; 数据控制语言: 负责数据完整性、安全性的定义与检查以及并发控制、故障恢复等功能。

6 [单选题 | 平均分3.93分 | 77人正确/98人做题 | 用时: <1分 | 得分: 5.0 / 5.0

数据库管理系统是()。

- A 操作系统的一部分
- B 在操作系统支持下的系统软件
- C 一种编译系统
- D 一种操作系统

他的回答: B (正确)

正确答案: B

参考答案:

数据库管理系统是数据库的机构, 它是一种系统软件, 负责数据库中数据组织、数据操纵、数据维护、控制及保护和数据服务等。是一种在操作系统之上的系统软件。

7 [单选题 | 平均分2.7分 | 53人正确/98人做题 | 用时: <1分 | 得分: 5.0 / 5.0

SQL 语句中修改表结构的命令是()

- A MODIFY TABLE
 - B MODIFY STRUCTURE
 - C ALTER TABLE
 - D ALTER STRUCTURE
- 添加表字段: alter table 表名 add 字段名称 字段属性
删除表字段: alter table 表名 drop 字段名
修改表字段:
alter table 表名 change 旧字段名称 新字段名称 字段类型
alter table 表名 modify 字段名称 字段类型(修改类型的)

他的回答: C (正确)

正确答案: C

8 [单选题 | 平均分3.62分 | 71人正确/98人做题 | 用时: <1分 | 得分: 5.0 / 5.0

在sql数据库中, 哪个语句能校验整数列 i 的值不小于 1 不大于 10 ?

- A i BETWEEN 1 AND 10
 - B i BETWEEN 0 AND 11
 - C i IN INTERVAL(0,11)
 - D i IN INTERVAL(1,10)
- 【1, 10】

他的回答: A (正确)

正确答案: A

9 [单选题 | 平均分3.99分 | 79人正确/99人做题 | 用时: <1分 | 得分: 5.0 / 5.0

在学生表 Student 的系别(Sdept) 属性中查询信息系(IS)、数学系(MA)和计算机系(CS)的学生姓名(Sname)和性别(Ssex), 正确的命名格式应为:

- A SELECT Student FROM Sname, Ssex WHERE Sdept IN ('IS' , ' MA' , 'CS')
- B SELECT Sname, Ssex FROM Student WHERE Sdept IN ('IS' , ' MA' , 'CS')
- C SELECT Sname, Ssex FROM Student WHERE Sdept (IS, MA, CS)
- D SELECT Sname, Ssex FROM Student WHERE Sdept LIKE IS, MA, CS

他的回答： B (正确)

正确答案： B

10 [单选题 | 平均分1.99分 | 39人正确/98人做题 | 用时：<1分 | 得分： 5.0 / 5.0

SQL 查询语句中 WHERE、GROUP BY、HAVING 这些关键字区别和用法总结错误的是（ ）

- A HAVING在查询语句中必须依赖于GROUP BY
- B WHERE子句用来限制SELECT语句从表中指定选取的行
- C GROUP BY子句用来分组WHERE子句的输出结果集
- D HAVING子句用来从分组的結果中筛选列 **筛选的是记录，是行**

他的回答： D (正确)

正确答案： D

11 ACM编程题 语言限制 [编程题 | 平均分16.01分 | 52人正确/97人做题 | 提交： 5 次 | 得分： 25.0 / 25.0

标题：参数解析 | 时间限制：1秒 | 内存限制：32768K | 语言限制：不限

【参数解析】

在命令行输入如下命令：

xcopy /s c:\ d:\e，

各个参数如下：

参数1：命令字xcopy

参数2：字符串/s

参数3：字符串c:\

参数4：字符串d:\e

请编写一个参数解析程序，实现将命令行各个参数解析出来。

解析规则：

- 1.参数分隔符为空格
- 2.对于用""包含起来的参数，如果中间有空格，不能解析为多个参数。比如在命令行输入xcopy /s "C:\program files" "d:\"时，参数仍然是4个，第3个参数应该是字符串C:\program files，而不是C:\program，注意输出参数时，需要将"去掉，引号不存在嵌套情况。
- 3.参数不定长
- 4.输入由用例保证，不会出现不符合要求的输入

$$1 \leq s \leq 1000$$

数据范围：字符串长度：

$$O(n) \qquad O(n)$$

进阶：时间复杂度：，空间复杂度：

输入描述：

输入一行字符串，可以有空格

输出描述：

输出参数个数，分解后的参数，每个参数都独占一行

示例1：

输入

xcopy /s c:\ d:\

输出

4
xcopy
/s
c:\
d:\

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 61%	使用语言	TA的 Java	平均 00:52:53 00:36:58	答案错误 : 2
基本测试用例通过率	5/5 (100%)	63%	做题用时	6	2	编译错误 : 1
边缘测试用例通过率	3/3 (100%)	58%	提交次数			答案正确 : 3
代码效率			代码规范及可读性			
运行时间	TA的 37ms	参考 1s	代码规范得分			4.58333
占用内存	10712K	32768K	Line 2: 'CLASS_DEF' should be separated from previous statement. [EmptyLineSeparator]			
			Line 13:17: '}' at column 17 should be alone on a line. [RightCurly]			

他的代码：

做题用时: 52 分钟 语言 : Java 运行时间 : 37ms 占用内存 : 10712K 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 str = scan.nextLine();
        //先求出参数的个数，这里注意，参数个数是空格数加1，当然这个空格不是双引号中的空格
        int count = 0;
        for(int i = 0;i < str.length();i++){
            if(str.charAt(i) == '"'){
                //如果这个字符是双引号，那就一直加加直到遇到第二个双引号
                do{
                    i++;
                }while(str.charAt(i) != '"');//遇到第二个"就会跳出循环
            }
            if(str.charAt(i) == ' '){
                //遇到了不是在双引号中的空格
                count++;
            }
        }
        System.out.println(count+1);
        //开始进行分割
        int flag = 0;//用来标记遇到了双引号
        for(int i = 0;i < str.length();i++){
            if(str.charAt(i) == '"'){
                if(flag == 0){
                    flag = 1;//说明是第一次遇到"
                }else if(flag == 1){
                    //说明这是遇到了与之前双引号匹配的第二个"
                    flag = 0;//重新置为0，方便下次寻找
                }
            }
            //双引号，以及不是双引号中的空格都是不需要输出的
            if(str.charAt(i) != '"' && str.charAt(i) != ' '){
                System.out.print(str.charAt(i));//先不输出换行
            }
            //等于空格还需要判断一下是双引号中的空格是需要输出的
            if(str.charAt(i) == ' ' && flag == 1){
                System.out.print(str.charAt(i));//把这个空格输出
            }
            if(str.charAt(i) == ' ' && flag == 0){
                //遇到空格。但是这个空格不是双引号中的，所以不输出，并且要换行
            }
        }
    }
}
```

```
        System.out.println();
    }

}

}

}
```



[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

[编程题 | 平均分8.37分 | 20人正确/95人做题 | 提交: 5 次

🕒 得分 : 0.0 / 25.0

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

【跳石板】小易来到了一条石板路前，每块石板上从1挨着编号为：1、2、3.....

这条石板路要根据特殊的规则才能前进：对于小易当前所在的编号为K的 石板，小易单次只能往前跳K的一个约数(不含1和K)步，即跳到K+X(X为K的一个非1和本身的约数)的位置。 小易当前处在编号为N的石板，他想跳到编号恰好为M的石板去，小易想知道最少需要跳跃几次可以到达。

例如：

N = 4，M = 24：

4->6->8->12->18->24

于是小易最少需要跳跃5次，就可以从4号石板跳到24号石板

输入描述：

输入为一行，有两个整数N，M，以空格隔开。（ $4 \leq N \leq 100000$ ）（ $N \leq M \leq 100000$ ）

输出描述：

输出小易最少需要跳跃的步数,如果不能到达输出-1

示例1：

输入

4 24

输出

5

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	使用语言	TA的	平均	答案错误：5
	0%	32%		Java		
	0/6	35%		做题用时	00:28:06 00:30:10	
	(0%)			提交次数	5 2	
基本测试用例通过率	0/4					
边缘测试用例通过率	(0%)	27%				
代码效率			代码规范及可读性			

TA的	参考	代码规范得分	4.4
运行时间	48ms	1s	Line 2: 'CLASS_DEF' should be separated from previous statement. [EmptyLineSeparator]
占用内存	11068K	32768K	Line 5:13: Local variable name 'N' must match pattern '^[a-z][a-z0-9][a-zA-Z0-9]*\$'. [LocalVariableName]
			Line 6:13: Local variable name 'M' must match pattern '^[a-z][a-z0-9][a-zA-Z0-9]*\$'. [LocalVariableName]

他的代码：

做题用时: 28 分钟 语言：Java 运行时间：48ms 占用内存：11068K 程序状态：答案错误

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

    }
}
```



[点此](#)或手机扫描二维码查看代码编写过程

监控截图



代码相似度

编程题1相似代码			
相似代码来源考生：黄子健	试卷：比特就业课105期Java方向笔试强训48天day13_10月24日	相似度：93.00%	通过时间提前：03:43:51

本代码

```
import java.util.*;
public class Main{
    public static void main(String[] args){
        Scanner scan = new Scanner(System.in);
        String str = scan.nextLine();
        //先求出参数的个数，这里注意，参数个数是空格数加1，当然这
        //个空格不是双引号中的空格
        int count = 0;
        for(int i = 0;i < str.length();i++){
            if(str.charAt(i) == '"'){
                //如果这个字符是双引号，那就一直加加直到遇到第二个双
                //引号
                do{
                    i++;
                }while(str.charAt(i) != '"');//遇到第二个"就会跳出循环
            }
            if(str.charAt(i) == ' '){
                //遇到了不是在双引号中的空格
                count++;
            }
        }
        System.out.println(count+1);
        //开始进行分割
        int flag = 0;//用来标记遇到了双引号
        for(int i = 0;i < str.length();i++){
            if(str.charAt(i) == '"'){
                if(flag == 0){
                    flag = 1;//说明是第一次遇到"
                }else if(flag == 1){
                    //说明这是遇到了与之前双引号匹配的第二个"
                    flag = 0;//重新置为0，方便下次寻找
                }
            }
            //双引号，以及不是双引号中的空格都是不需要输出的
            if(str.charAt(i) != '"' && str.charAt(i) != ' '){
                System.out.print(str.charAt(i));//先不输出换行
            }
            //等于空格还需要判断一下是双引号中的空格是需要输出的
            if(str.charAt(i) == ' ' && flag == 1){
                System.out.print(str.charAt(i));//把这个空格输出
            }
            if(str.charAt(i) == ' ' && flag == 0){
                //遇到空格。但是这个空格不是双引号中的，所以不输出，
                //并且要换行
                System.out.println();
            }
        }
    }
}
```

相似代码

```
import java.util.*;
public class Main{
    public static void main(String[] args){
        Scanner sc = new Scanner(System.in);
        String str = sc.nextLine();
        int count = 0;
        for(int i = 0;i<str.length();i++){
            if(str.charAt(i)=='""){
                do{
                    i++;
                }while(str.charAt(i)!='"');
            }
            if(str.charAt(i)==' '){
                count++;
            }
        }
        System.out.println(count+1);
        int flg = 1;

        for(int i = 0;i<str.length();i++){
            if(str.charAt(i)=='""){
                flg^=1;
            }
            if(str.charAt(i)!=""&&str.charAt(i)!=' '){
                System.out.print(str.charAt(i));
            }
            if(str.charAt(i)==' ' && flg==1){
                System.out.println();
            }
            if(str.charAt(i)==' ' && flg==0){
                System.out.print(' ');
            }
        }
    }
}
```

