

比特就业课105期Java方向笔试强训48天day11_10月21日-王世国-测评结果

考生信息  存在作弊行为

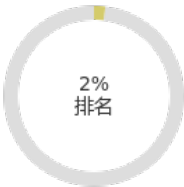


王世国
投递编号：91 | 学校：武汉轻工大学 | 邮箱：1477649017@qq.com | 职位：比特就业课105期Java2班
参考区域: 湖北省武汉市 (183.94.121.184) | 做题用时：02:27:30(2022-10-20 21:08:40开始答题，23:36:15交卷) |
作答设备：PC | 已同意诚信声明和隐私协议

95.0 分 / 100分

在本次考试中，考生总成绩为95.0分/100分，评级为A（排名前2%），编程能力优秀（2题完全通过，分数排名前1%），编程思路完全正确，编程规范性高。该考生在本次考试中存在作弊行为，无视频监控截图，未开启摄像头。

考生成绩



题型	得分	正确题数	排名	用时	是否阅卷
单选	45.0	9	3	00:11:17	已阅
编程	50.0	2	1	01:24:22	已阅

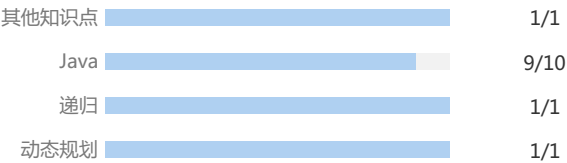
作弊风险

高风险

未开启摄像头

考生考试过程中未开启摄像头

知识点技能图谱



知识点	得分	正确题数
其他知识点	25.0	1
Java	45.0	9
递归	25.0	1
动态规划	25.0	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	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

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%	9	01:05:48	Java	17ms	9948K	良	良	1%
编程题2	100%	1	00:18:34	Java	41ms	10988K	良	良	1%

1

[单选题 | 平均分3.7分 | 71人正确/96人做题 | 用时：2分 | 得分：5.0 / 5.0]

下面哪个标识符是合法的？

- A HelloWorld
- B _Hello World
- C Hello*World
- D Hello\$World

java合法标识符：以数字，字母，下划线，美元符号组成
不过不能以数字开头

他的回答： D (正确)

正确答案： D

2 [单选题 | 平均分4.22分 | 81人正确/96人做题 | 用时：<1分 | 得分：5.0 / 5.0

以下java程序代码，执行后的结果是（ ）

```
java.util.HashMap map=new java.util.HashMap();
map.put("name",null);
map.put("name","Jack");
System.out.println(map.size());
```

- A 0
- B null
- C 1
- D 2

他的回答： C (正确)

正确答案： C

参考答案：

size() 方法用于返回在此映射中的键 - 值映射关系的数量。

HashMap在插入新元素时，会使用equals()方法判断集合中是否已经存在与新元素key相同的元素，如果存在，新元素会覆盖掉和他key相同的对象。

HashMap允许null作为key/value，因此map.put("name",null);会在map集合中存入key为"name"，value为null的对象。map.put("name","Jack");会在map集合中存入key为"name"，value为"Jack"的对象。前后两次存入的对象的key都为"name"，因此刚存入的key为"name"，value为null的对象会被覆盖掉，即map中只有一个key为"name"，value为"Jack"的对象。map.size()为1。

3 [单选题 | 平均分4.69分 | 91人正确/97人做题 | 用时：<1分 | 得分：5.0 / 5.0

Java中的集合类包括ArrayList、LinkedList、HashMap等类，下列关于集合类描述错误的是

- A ArrayList和LinkedList均实现了List接口
- B ArrayList的访问速度比LinkedList快
- C 添加和删除元素时，ArrayList的表现更佳
- D HashMap实现Map接口，它允许任何类型的键和值对象，并允许将null用作键或值

他的回答： C (正确)

正确答案： C

4 [单选题 | 平均分2.03分 | 39人正确/96人做题 | 用时：<1分 | 得分：0.0 / 5.0

以下描述正确的是

- A CallableStatement是PreparedStatement的父接口
 - B PreparedStatement是CallableStatement的父接口
 - C CallableStatement是Statement的父接口
 - D PreparedStatement是Statement的父接口
- PreparedStatement是CallableStatement的父接口
PreparedStatement继承于Statement

他的回答： D (错误)

正确答案： B

5 [单选题 | 平均分1.77分 | 34人正确/96人做题 | 用时：<1分 | 得分：5.0 / 5.0

下列程序的运行结果

```
public static void main(String args[]) {
```

```
Thread t = new Thread() {
    public void run() {
        pong();
    }
};
t.run();
System.out.print("ping");
}
static void pong() {
    System.out.print("pong");
}
```

- A pingpong
- B pongping
- C pingpong和pongping都有可能
- D 都不输出

他的回答： B (正确)

正确答案： B

6 [单选题 | 平均分1.09分 | 21人正确/96人做题 | 用时：<1分 | 得分：5.0 / 5.0]

下列关于容器集合类的说法正确的是？

- A LinkedList继承自List **List, Set都是接口，接口是用来实现的**
- B AbstractSet继承自Set
- C HashSet继承自AbstractSet
- D WeakMap继承自HashMap

他的回答： C (正确)

正确答案： C

7 [单选题 | 平均分2.16分 | 42人正确/97人做题 | 用时：<1分 | 得分：5.0 / 5.0]

ArrayList list = new ArrayList(20);中的list扩充几次

- A 0
- B 1
- C 2
- D 3

他的回答： A (正确)

正确答案： A

8 [单选题 | 平均分4.07分 | 79人正确/97人做题 | 用时：<1分 | 得分：5.0 / 5.0]

以下程序的输出结果是？

```
public class Example {
    String str = new String("good");
    char[] ch = { 'a', 'b', 'c' };

    public static void main(String args[]) {
        Example ex = new Example();
        ex.change(ex.str, ex.ch);
        System.out.print(ex.str + " and ");
        System.out.print(ex.ch);
    }

    public static void change(String str, char ch[])
```

```
{
    str = "test ok";
    ch[0] = 'g';
}
}
```

- A good and abc
- B good and gbc
- C test ok and abc
- D test ok and gbc

他的回答： B (正确)

正确答案： B

9 [单选题 | 平均分4.32分 | 83人正确/96人做题 | 用时：<1分 | 得分： 5.0 / 5.0

下面的方法，当输入为2的时候返回值是多少？

```
public static int getValue(int i) {
    int result = 0;
    switch (i) {
        case 1:
            result = result + i;
        case 2:
            result = result + i * 2;
        case 3:
            result = result + i * 3;
    }
    return result;
}
```

- A 0
- B 2
- C 4
- D 10

他的回答： D (正确)

正确答案： D

10 [单选题 | 平均分2.71分 | 52人正确/96人做题 | 用时：3分 | 得分： 5.0 / 5.0

下列关于Java并发的说法中正确的是（ ）

- A CopyOnWriteArrayList适用于写多读少的并发场景
- B ReadWriteLock适用于读多写少的并发场景
- C ConcurrentHashMap的写操作不需要加锁，读操作需要加锁
- D 只要在定义int类型的成员变量i的时候加上volatile关键字，那么多线程并发执行i++这样的操作的时候就是线程安全的了

copyOnWriteArrayList的精髓就在于写时拷贝，也就是在进行修改操作的时候，会拷贝一份新的数组出来进行修改，然后修改完成之后再吧引用赋值回去，这个过程不影响读操作。读操作不会加锁，但是写操作会加锁。读写之间不冲突。因为涉及到拷贝，所以数据量大和写操作频繁的情况下不适合用

他的回答： B (正确)

正确答案： B

ReadWriteLock读写分离锁
读与读之间不阻塞，读与写之间阻塞，写与写之间阻塞

11 完善核心代码 语言限制 [编程题 | 平均分18.04分 | 70人正确/97人做题 | 提交: 9 次 | 得分： 25.0 / 25.0

标题：最近公共祖先 | 时间限制：3秒 | 内存限制：32768K | 语言限制：[Python, C++, C#, Java]

【最近公共祖先】

将一棵无穷大满二叉树的结点按根结点一层一层地从左往右编号，根结点编号为1。现给定a，b为两个结点。设计一个算法，返回a、b最近的公共祖先的编号。注意其祖先也可能是结点本身。

测试样例：

2, 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		
总通过率		100%	72%	使用语言		Java		答案错误 : 7	
基本测试用例通过率		1/1 (100%)	72%	做题用时		01:05:48 00:19:07		编译错误 : 1	
				提交次数		9 2		答案正确 : 1	
代码效率							代码规范及可读性		
		TA的	参考					代码规范得分 4.0	
运行时间		17ms	3s					Line 3: Abbreviation in name 'LCA' must contain no more than '1' capital letters. [AbbreviationAsWordInName]	
占用内存		9948K	32768K					Line 4: Abbreviation in name 'getLCA' must contain no more than '1' capital letters. [AbbreviationAsWordInName]	
								Line 4:27: Parameter name 'a' must match pattern '^[a-z][a-z0-9][a-zA-Z0-9]*\$'. [ParameterName]	
								Line 4:34: Parameter name 'b' must match pattern '^[a-z][a-z0-9][a-zA-Z0-9]*\$'. [ParameterName]	
								Line 24: Comment has incorrect indentation level 0, expected is 8, indentation should be the same level as line 26. [CommentsIndentation]	

他的代码：

做题用时：65 分钟 语言：Java 运行时间：17ms 占用内存：9948K 程序状态：答案正确

```
import java.util.*;

public class LCA {
    public int getLCA(int a, int b) {
        // write code here
        // if(a == 1 || b == 1){
        //     return 1;
        // }
        // if(a + 1 == b || b + 1 == a){
        //     return a/2;
        // }

        // if(a%b == 0 || a%b == 1){
        //     return b;
        // }
        // if(b%a == 0 || b%a == 1){
        //     return a;
        // }

        // while(a != b){
        //     a /= 2;
        //     b /= 2;
        // }
        // return a;

        while(a != b){
            if(a > b){
                a /= 2;
            }else{
                b /= 2;
            }
        }
    }
}
```

[点此](#)或手机扫描二维码查看代码编写过程



2

代码规范及可读性

TA的	参考	代码规范得分	4.6
运行时间	41ms	1s	Line 2: 'CLASS_DEF' should be separated from previous statement. [EmptyLineSeparator]
占用内存	10988K	32768K	Line 6:17: 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: 18 分钟 语言: Java 运行时间: 41ms 占用内存: 10988K 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);
        while(scan.hasNextInt()){
            int n = scan.nextInt();
            int count = 0;
            int flag = 0;
            int max = 0;
            for(int i = 0;i < 32;i++){
                if(((n>>i)&1) == 1){
                    if(flag == 0){
                        flag = 1;
                        count = 0;
                    }
                    count++;
                    if(count > max){
                        max = count;
                    }
                }else{
                    flag = 0;
                }
            }
            System.out.println(max);
        }
    }
}
```



[点此](#)或手机扫描二维码查看代码编写过程

监控截图