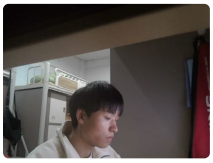
# 比特就业课105期Java方向笔试强训48天day10_10月20日-王世国-测评结果

## 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 |

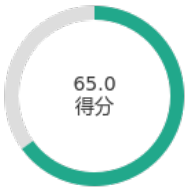
参考区域: 湖北省武汉市（183.94.121.184） | 做题用时：02:15:37(2022-10-19 19:28:50开始答题，21:44:34交卷) |

作答设备： PC | 已同意诚信声明和隐私协议

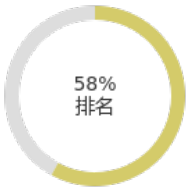
65.0 分 / 100分

在本次考试中，考生总成绩为65.0分/100分，评级为C（排名前58%），编程能力良好（1题通过，分数排名前63%），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，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







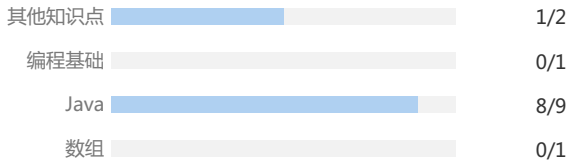

| 题型 | 得分 | 正确题数 | 排名 | 用时 | 是否阅卷 |
|---|---|---|---|---|---|
| 单选 | 40.0 | 8 | 10 | 00:14:18 | 已阅 |
| 编程 | 25.0 | 1 | 68 | 01:23:33 | 已阅 |

## 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

## 知识点技能图谱

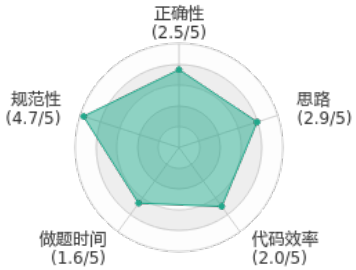



| 知识点 | 得分 | 正确题数 |
|---|---|---|
| 其他知识点 | 25.0 | 1 |
| 编程基础 | 0.0 | 0 |
| Java | 40.0 | 8 |
| 数组 | 0.0 | 0 |

## 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 |

## 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 | 0% | 1 | 00:21:41 | Java | 0ms | 0K | 差 | 优 | - |
| 编程题2 | 100% | 3 | 01:01:52 | Java | 48ms | 10872K | 良 | 良 | 1% |

1 [ 单选题 | 平均分1.19分 | 24人正确/101人做题 | 用时：<1分 得分： 0.0 / 5.0

结构型模式中最体现扩展性的模式是（ ）

A 装饰模式

B 合成模式

C 桥接模式

D 适配器

他的回答： C (错误)

正确答案： A

2 [ 单选题 | 平均分2.99分 | 61人正确/102人做题 | 用时：2分 得分： 5.0 / 5.0

下面代码运行结果是（ ）

```
public class Test{
    public int add(int a,int b){
        try {
            return a+b;
        }
        catch (Exception e) {
            System.out.println("catch语句块");
        }
        finally{
            System.out.println("finally语句块");
        }
        return 0;
    }
     public static void main(String argv[]){
        Test test =new Test();
        System.out.println("和是："+test.add(9, 34));
     }
}
```

A catch语句块 和是：43

B 编译异常

C finally语句块 和是：43

D 和是：43 finally语句块

他的回答： C (正确)

正确答案： C

3 [ 单选题 | 平均分1.52分 | 31人正确/102人做题 | 用时：2分 得分： 5.0 / 5.0

下列Java代码中的变量a、b、c分别在内存的____存储区存放。

```
class A {
    private String a = “aa”;
    public boolean methodB() {
        String b = “bb”;
        final String c = “cc”;
    }
}
```

成员变量，在堆上

b,c两个变量都是局部变量，所以是在栈上，只不过它们的值是字符串，是在字符串常量池中

A 堆区、堆区、堆区

B 堆区、栈区、堆区

C 堆区、栈区、栈区

D 堆区、堆区、栈区

E 静态区、栈区、堆区

F 静态区、栈区、栈区

他的回答： C (正确)

正确答案： C

4 [ 单选题 | 平均分2.28分 | 46人正确/101人做题 | 用时：<1分 得分： 5.0 / 5.0

以下声明合法的是

A default String s
B public final static native int w( )
C abstract double d
D abstract final double hyperbolicCosine( )

他的回答： B (正确)

正确答案： B

---

5 [ 单选题 | 平均分4.61分 | 94人正确/102人做题 | 用时：<1分 得分： 5.0 / 5.0

在使用super 和this关键字时，以下描述正确的是

A 在子类构造方法中使用super（ ）显示调用父类的构造方法，super（ ）必须写在子类构造方法的第一行，否则编译不通过
B super（ ）和this（ ）不一定要放在构造方法内第一行
C this（ ）和super（ ）可以同时出现在一个构造函数中
D this（ ）和super（ ）可以在static环境中使用，包括static方法和static语句块

他的回答： A (正确)

正确答案： A

---

6 [ 单选题 | 平均分3.09分 | 63人正确/102人做题 | 用时：<1分 得分： 5.0 / 5.0

下面代码的输出结果是什么？

```
public class ZeroTest {
   public static void main(String[] args) {
    try{
      int i = 100 / 0;
      System.out.print(i);
  }catch(Exception e){
      System.out.print(1);
      throw new RuntimeException();
  }finally{
      System.out.print(2);
  }
     System.out.print(3);
 }
 }
```

A 3
B 123
C 1
D 12

他的回答： D (正确)

正确答案： D

---

7 [ 单选题 | 平均分1.67分 | 34人正确/102人做题 | 用时：2分 得分： 0.0 / 5.0

有代码片段如下：

```
byte b1=1,b2=2,b3,b6;
final byte b4=4,b5=6;
b6=b4+b5;
b3=(b1+b2);
System.out.println(b3+b6);
```

重点注意这里的final修饰，修饰之后在运算的时候类型是不会做提升的
按照常理而言，b4+b5都会先提升为int后做计算，但是因为b4,b5都被final修饰了，所以不会进行提升，那么也就没有什么问题

关于上面代码片段叙述正确的是（ ）

A 输出结果：13
B 语句：b6=b4+b5编译出错
C 语句：b3=b1+b2编译出错
D 运行期抛出异常

他的回答： B (错误)

正确答案： C

8 [ 单选题 | 平均分3.88分 | 80人正确/103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（ ）

```
public class Test {
    public static void main(String[] args) {
        Object o = new Object() {
            public boolean equals(Object obj) {
                return true;
            }
        };
        System.out.println(o.equals("Fred"));
        }
}
```

A Fred
B true
C 编译错误
D 运行时抛出异常

他的回答： B (正确)

正确答案： B

**参考答案：**
考察的知识点：多态、重载
首先，创建了一个匿名内部类，并将所创建的匿名对象赋给 Object (多态：子类对象赋给超类引用)。同时，该匿名内部类重写了 Object 类的 equals 方法，根据多态及覆盖原则，会调用匿名内部类重写后的 equals 方法，所以选B。

9 [ 单选题 | 平均分4.5分 | 91人正确/101人做题 | 用时：<1分 得分： 5.0 / 5.0

执行以下程序后的输出结果是（ ）

```
public class Test {
    public static void main(String[] args) {
        StringBuffer a = new StringBuffer("A");
        StringBuffer b = new StringBuffer("B");
        operator(a, b);
        System.out.println(a + "," + b);
    }
    public static void operator(StringBuffer x, StringBuffer y) {
        x.append(y); y = x;
    }
}
```

A A,A
B A,B
C B,B
D AB,B

他的回答： D (正确)

正确答案： D

10 [ 单选题 | 平均分4.31分 | 88人正确/102人做题 | 用时：<1分 得分： 5.0 / 5.0

下面所示的java代码，运行时，会产生（ ）类型的异常

```
int Arry_a[] = new int[10];
System.out.println(Arry_a[10]);
```

A ArithmeticException

B NullPointException

C IOException

D ArrayIndexOutOfBoundsException

他的回答： D (正确)

正确答案： D

11 完善核心代码 语言限制 [ 编程题 | 平均分19.01分 | 73人正确/104人做题 | 提交: 1 次 得分： 0.0 / 25.0

标题：井字棋 | 时间限制：3秒 | 内存限制：32768K | 语言限制： [Python, C++, C#, Java]

【井字棋】

给定一个二维数组**board**，代表棋盘，其中元素为1的代表是当前玩家的棋子，0表示没有棋子，-1代表是对方玩家的棋子。当一方棋子在横竖斜方向上有连成排的及获胜（及井字棋规则），返回当前玩家是否胜出。

测试样例：

```
[[1,0,1],[1,-1,-1],[1,-1,0]]
```

```
返回：true
```

**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 | 编译错误 ：1 |
| 总通过率 | 0% | 76% | 使用语言 | Java | | |
| 基本测试用例通过率 | 0/19 (0%) | 75% | 做题用时 | 00:21:41 | 00:19:52 | |
| 边缘测试用例通过率 | 0/12 (0%) | 77% | 提交次数 | 1 | 2 | |

| 代码效率 | | | 代码规范及可读性 |
|---|---|---|---|
| | TA的 | 参考 | 代码规范得分 5.0 |
| 运行时间 | 0ms | 3s | |
| 占用内存 | 0K | 32768K | |

他的代码：

做题用时: 21 分钟 语言：Java 运行时间：0ms 占用内存：0K 程序状态：编译错误

```
import java.util.*;

public class Board {
    public boolean checkWon(int[][] board) {
        // write code here

    }
}
```

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16.04分 | 59人正确/103人做题 | 提交: 3 次 得分： 25.0 / 25.0

标题：密码强度等级 | 时间限制：1秒 | 内存限制：32768K | 语言限制： 不限

【密码强度等级】密码按如下规则进行计分，并根据不同的得分为密码进行安全等级划分。

这个题没有什么特别的点，就是多写几个函数，分别来遍历统计字母，数字，符号的数量，对应计算分数。就是一个遍历字符串的题

一、密码长度:
5 分: 小于等于4 个字符
10 分: 5 到7 字符
25 分: 大于等于8 个字符

二、字母:
0 分: 没有字母
10 分: 密码里的字母全都是小（大）写字母
20 分: 密码里的字母符合”大小写混合“

三、数字:
0 分: 没有数字
10 分: 1 个数字
20 分: 大于1 个数字

四、符号:
0 分: 没有符号
10 分: 1 个符号
25 分: 大于1 个符号

五、奖励（只能选符合最多的那一种奖励）:
2 分: 字母和数字
3 分: 字母、数字和符号
5 分: 大小写字母、数字和符号

最后的评分标准:
>= 90: 非常安全
>= 80: 安全（Secure）
>= 70: 非常强
>= 60: 强（Strong）
>= 50: 一般（Average）
>= 25: 弱（Weak）
>= 0: 非常弱（Very_Weak）

对应输出为：

VERY_SECURE
SECURE
VERY_STRONG
STRONG
AVERAGE
WEAK

VERY_WEAK

请根据输入的密码字符串，进行安全评定。

注：
字母：a-z, A-Z
数字：0-9
符号包含如下：（ASCII码表可以在UltraEdit的菜单view->ASCII Table查看）
!"#$%&'()*+,-./ (ASCII码：0x21~0x2F)
:;<=>?@ (ASCII码：0x3A~0x40)
[\]^_` (ASCII码：0x5B~0x60)
{|}~ (ASCII码：0x7B~0x7E)

提示:
1 <= 字符串的长度<= 300
输入描述：

> 输入一个string的密码

输出描述：

> 输出密码等级

示例1：
输入

> 38$@NoNoNo

输出

> VERY_SECURE

**代码片段**

功能实现

| | TA的 | 平均 |
|---|---|---|
| 总通过率 | 100% | 64% |
| 基本测试用例通过率 | 18/18 (100%) | 64% |
| 边缘测试用例通过率 | 12/12 (100%) | 64% |

代码提交统计

| | TA的 | 平均 |
|---|---|---|
| 使用语言 | Java | |
| 做题用时 | 01:01:52 | 00:38:36 |
| 提交次数 | 3 | 2 |

代码执行统计

编译错误 ：2
答案正确 ：1

代码效率

| | TA的 | 参考 |
|---|---|---|
| 运行时间 | 48ms | 1s |
| 占用内存 | 10872K | 32768K |

代码规范及可读性

代码规范得分 4.38144

Line 2: 'CLASS_DEF' should be separated from previous statement. [EmptyLineSeparator]

Line 4:13: Local variable name 'Upper' must match pattern '^[a-z][a-z0-9][a-zA-Z0-9]*$'. [LocalVariableName]

Line 5:13: Local variable name 'Lower' must match pattern '^[a-z][a-z0-9][a-zA-Z0-9]*$'. [LocalVariableName]

Line 24:23: Method name 'Num' must match pattern '^[a-z][a-z0-9][a-zA-Z0-9_]*$'. [MethodName]

Line 40:23: Method name 'Symbol' must match pattern '^[a-z][a-z0-9][a-zA-Z0-9_]*$'. [MethodName]

Line 44: Line is longer than 100 characters (found 133). [LineLength]

他的代码：
做题用时: 61 分钟 语言：Java 运行时间：48ms 占用内存：10872K 程序状态：答案正确

```
import java.util.*;
public class Main {
    public static int letter(String passWord){
        int Upper = 0;
        int Lower = 0;
        for(int i = 0;i < passWord.length();i++){
            if(passWord.charAt(i) >= 'a' && passWord.charAt(i) <= 'z'){
                Lower++;
            }else if(passWord.charAt(i) >= 'A' && passWord.charAt(i) <= 'Z'){
                Upper++;
            }

        }
        if(Upper == 0 && Lower == 0){
            return 0;
        }else if(Upper == 0 && Lower != 0 || Upper != 0 && Lower == 0){
            return 10;
        }else{
            return 20;
        }

    }

    public static int Num(String passWord){
        int count = 0;
        for(int i = 0;i < passWord.length();i++){
            if(passWord.charAt(i) >= 48 && passWord.charAt(i) <= 57){
                count++;
            }
        }
        if(count == 0){
            return 0;
        }
        if(count == 1){
            return 10;
        }
        return 20;
    }

    public static int Symbol(String passWord){
        int count = 0;
        for(int i = 0;i < passWord.length();i++){
            char ch = passWord.charAt(i);
            if((ch >= 0x21 && ch <= 0x2F) || (ch >= 0x3A && ch <= 0x40) || (ch >= 0x5B && ch <= 0x60) || (ch >= 0x7B && ch <= 0x
7E)){
                count++;
            }
        }
        if(count == 0){
            return 0;
        }else if(count == 1){
            return 10;
        }
        return 25;
    }

    public static void main(String[] args){
        Scanner scan = new Scanner(System.in);
        String passWord = scan.next();
        int score = 0;
```

```
        int len = passWord.length();
        if(len <= 4){
            score += 5;
        }else if(len > 4 && len <= 7){
            score += 10;
        }else{
            score += 25;
        }

        int ret1 = letter(passWord);
        int ret2 = Num(passWord);
        int ret3 = Symbol(passWord);
        score = score + ret1 + ret2 + ret3;
        if(ret1 != 0 && ret2 != 0 && ret3 == 0){
            score += 2;
        }else if(ret1 == 10 && ret2 != 0 && ret3 != 0){
            score += 3;
        }else if(ret1 == 20 && ret2 != 0 && ret3 != 0){
            score += 5;
        }

        if(score >= 90){
            System.out.println("VERY_SECURE");
        }else if(score >= 80){
            System.out.println("SECURE");
        }else if(score >= 70){
            System.out.println("VERY_STRONG");
        }else if(score >= 60){
            System.out.println("STRONG");
        }else if(score >= 50){
            System.out.println("AVERAGE");
        }else if(score >= 25){
            System.out.println("WEAK");
        }else if(score >= 0){
            System.out.println("VERY_WEAK");
        }
    }
}
```

点此或手机扫描二维码查看代码编写过程

**监控截图**