

Java方向每日一题day10_12月3日-辛蒙号-测评结果

考生信息



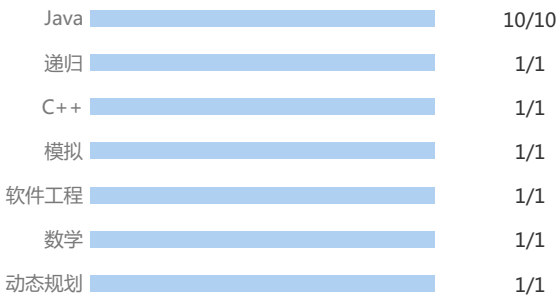
辛蒙号
投递编号：53 | 学校：西安工业大学 | 邮箱：2535342895@qq.com | 职位：2020大四春招冲刺班 |
参考区域: 陕西省西安市 (111.21.177.20) | 做题用时：00:06:35(2020-12-03 08:39:40开始答题，08:46:25交卷) |
作答设备：PC | 已同意诚信声明和隐私协议

考生成绩



题型	得分	正确题数	排名	用时	是否阅卷
单选	50.0	10	1	00:02:34	--
编程	50.0	2	1	00:03:28	--

知识点技能图谱



知识点	得分	正确题数
Java	50.0	10
递归	25.0	1
C++	5.0	1
模拟	25.0	1
软件工程	5.0	1
数学	25.0	1
动态规划	25.0	1

历史笔记记录

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	2.0%	60.0/60	单选:30.0分 编程:30.0分	否	2020-10-28 17:33:17	2020-10-29 10:01:52
2	Java方向每日一题day01_11月23日	7.0%	95.0/100	单选:50.0分 编程:45.0分	否	2020-11-22 11:26:51	2020-11-22 20:36:21
3	大四春招冲刺班数据结构考试	17.0%	54.0/60	单选:24.0分 编程:30.0分	否	2020-11-23 11:55:16	2020-11-24 09:49:48
4	Java方向每日一题day02_11月24日	7.0%	95.0/100	单选:45.0分 编程:50.0分	是，代码抄袭	2020-11-23 12:10:19	2020-11-24 11:06:41

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
5	Java方向每日一题day03_11月25日	1.0%	100.0/100	单选:50.0分 编程:50.0分	否	2020-11-24 15:19:25	2020-11-25 14:47:17
6	Java方向每日一题day04_11月26日	1.0%	100.0/100	单选:50.0分 编程:50.0分	否	2020-11-25 14:58:20	2020-11-26 08:57:04
7	Java方向每日一题day05_11月27日	5.0%	95.0/100	单选:45.0分 编程:50.0分	否	2020-11-25 15:54:38	2020-11-27 09:30:57
8	Java方向每日一题day06_11月28日	1.0%	100.0/100	单选:50.0分 编程:50.0分	否	2020-11-27 14:19:26	2020-11-28 19:26:31
9	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 13:58:19	2020-11-30 16:31:11
10	Java方向每日一题day08_12月1日	1.0%	100.0/100	单选:50.0分 编程:50.0分	否	2020-11-30 10:48:02	2020-12-01 15:51:46
11	Java方向每日一题day09_12月2日	1.0%	100.0/100	单选:50.0分 编程:50.0分	否	2020-12-01 10:43:40	2020-12-02 14:28:36

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%	2	00:02:25	Java	30ms	10740K			1%
编程题2	100%	1	00:01:03	Java	11ms	9404K			1%

1

[平均分2.3分 | 50人正确/110人做题 | 用时：<1分] 得分：5.0 / 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：A (正确)

正确答案：A

2 [平均分2.8分 | 63人正确/112人做题 | 用时：<1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 [平均分2.1分 | 47人正确/112人做题 | 用时：<1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" ;
    }
}
```

- 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 [平均分2.4分 | 54人正确/112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 [平均分4.8分 | 107人正确/112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 [平均分2.8分 | 63人正确/112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 [平均分2.3分 | 51人正确/112人做题 | 用时：<1分] 得分：5.0 / 5.0

代码片段：

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

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：C (正确)

正确答案：C

8 [平均分3.8分 | 86人正确/112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

9 [平均分4.1分 | 92人正确/112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 [平均分4.4分 | 99人正确/112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 Array_a[] = new int[10];
System.out.println(Array_a[10]);
```

- A ArithmeticException
- B NullPointerException
- C IOException
- D ArrayIndexOutOfBoundsException

他的回答： D (正确)

正确答案： D

11 ACM编程题 语言限制 [平均分23.2分 | 85人正确/97人做题 | 提交: 2 次] 得分：25.0 / 25.0

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

【Fibonacci数列】Fibonacci数列是这样定义的：

F[0] = 0
F[1] = 1
for each i ≥ 2: F[i] = F[i-1] + F[i-2]
因此，Fibonacci数列就形如：0, 1, 1, 2, 3, 5, 8, 13, ...，在Fibonacci数列中的数我们称为Fibonacci数。给你一个N，你想让其变为一个Fibonacci数，每一步你可以把当前数字X变为X-1或者X+1，现在给你一个数N求最少需要多少步可以变为Fibonacci数。
输入描述：

输入为一个正整数N(1 ≤ N ≤ 1,000,000)

输出描述：

输出一个最小的步数变为Fibonacci数"

示例1：

输入
15
输出
2

代码片段					
功能实现			代码提交统计		
	TA的	平均		TA的	平均
总通过率	100%	92%	使用语言	Java	
基本测试用例通过率	6/6 (100%)	92%	做题用时	00:02:25	00:15:14
边缘测试用例通过率	4/4 (100%)	93%	提交次数	2	2
代码效率			代码规范及可读性		
	TA的	参考	代码规范得分		
运行时间	30ms	1s	Line 2: 'CLASS_DEF' should be separated from previous statement. [EmptyLineSeparator]		
占用内存	10740K	32768K	Line 6:17: Local variable name 'm' must match pattern '^ [a-z][a-z0-9][a-zA-Z0-9]*\$'. [LocalVariableName]		
			4.6		

他的代码：
做题用时: 2 分钟 语言：Java 运行时间：30ms 占用内存：10740K 程序状态：答案正确

```
import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        while (in.hasNextInt()) {
            int m = in.nextInt();
            int f0 = 0;
            int f1 = 1;
            while(f1 < m){
                int temp = f0 + f1;
                f0 = f1;
                f1 = temp;
            }
            int count = (f1- m) < (m - f0) ? (f1 - m) : (m - f0);
            System.out.println(count);
        }
    }
}
```

```
}  
}
```



点此或手机扫描二维码查看代码编写过程

12 完善核心代码 语言限制 [平均分22.0分 | 80人正确/91人做题 | 提交: 1 次] 得分 : 25.0 / 25.0

标题：机器人走方格I | 时间限制：3秒 | 内存限制：32768K | 语言限制：[Python, C++, C#, Java]

【机器人走方格I】

有一个XxY的网格，一个机器人只能走格点且只能向右或向下走，要从左上角走到右下角。请设计一个算法，计算机器人有多少种走法。给定两个正整数int x,int y，请返回机器人的走法数目。保证x+y小于等于12。

测试样例：

2,2

返回：2

输入描述：

输出描述：

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	答案正确：1
	100%	87%		使用语言	Java	
	1/1 (100%)	87%		做题用时	00:01:03 00:20:55	
基本测试用例通过率				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 4.2			
运行时间	11ms	3s	Line 4:30: Parameter name 'x' must match pattern '^[a-z][a-z0-9][a-zA-Z0-9]*\$'. [ParameterName]			
占用内存	9404K	32768K	Line 4:37: Parameter name 'y' must match pattern '^[a-z][a-z0-9][a-zA-Z0-9]*\$'. [ParameterName]			
			Line 5: 'if' construct must use '{}'. [NeedBraces]			
			Line 6: 'if' construct must use '{}'. [NeedBraces]			

他的代码：

做题用时: <1 分钟 语言：Java 运行时间：11ms 占用内存：9404K 程序状态：答案正确

```
import java.util.*;
```

```
public class Robot {  
    public int countWays(int x, int y) {  
        if(x == 0 && y == 0)return 0;  
        if(x == 1 || y == 1)return 1;  
        return countWays(x,y-1) + countWays(x-1,y);  
    }  
}
```



[点此](#)或手机扫描二维码查看代码编写过程