**JAVA笔试题（社）**

**姓名\_\_\_\_\_\_\_\_\_\_\_ 应聘职位\_\_\_\_\_\_\_\_\_\_\_\_\_ 日期\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

1. 中瑞壹好服务节举办的一次比武大赛中，来自东北战区、西北战区、西南战区、东南战区和华东战区的5名服务顾问(每战区1名)相遇在一起，有张三、李四、王五、马六、洪七，5名服务顾问之间的比赛信息满足以下条件：  
   (1)张三仅与2名选手比赛过。  
   (2)华东战区的选手和3名选手比赛过。  
   (3)李四不是西北战区的，也没有和西北战区的选手对阵过。  
   (4)西南战区的选手和王五比赛过。  
   (5)西北战区，西南战区，东南战区的选手互相都交过手。  
   (6)马六仅与1名选手比赛过。  
   根据以上条件，请问王五来自哪个战区（ ）
2. Java的四个基本特性，包括继承、\_\_\_\_\_\_\_\_、\_\_\_\_\_\_\_\_\_\_\_、\_\_\_\_\_\_\_\_\_\_\_。

继承是\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。

1. Java重载\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;

重写\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。

1. Java访问控制符：

public\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_；

private\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_；

protected\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。

1. Java”==”和equal的区别

”==”\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_；

equal\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。

1. Java实例变量和static静态变量区别，程序运行时的区别

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Java自动装箱和拆箱：

装箱\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;

拆箱\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。

1. 什么情况会造成内存泄漏
2. Java集合类：

List\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;

set\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;

map\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;

1. Java中ArrayList和linkedlist区别，

ArrayList：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;

linkedlist：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;

1. Java多线程的实现方法

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;

1. Java多线程wait()和sleep()区别，

wait():\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;

sleep():\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;

1. Spring IOC的理解

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;

IOC注入方式：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;

1. Spring的AOP理解
2. #{}和${}的区别（）
3. 快速排序算法

*/\*\*  
 \* 递归方法-快速排序  
 \*  
 \** ***@param*** *array 原数组  
 \** ***@param*** *startIndex 开始下标  
 \** ***@param*** *endIndex 结束下标  
 \*/*public static void quickSort(int[] array, int startIndex, int endIndex) {  
 // 递归结束条件：startIndex大于或等于endIndex时  
 if (startIndex >= endIndex) {  
 return;  
 }  
  
 // 得到基准元素位置  
 int pivotIndex = *partition2*(array, startIndex, endIndex);  
 // 根据基准元素，分成两部分进行递归排序  
 *quickSort*(array, startIndex, pivotIndex - 1);  
 *quickSort*(array, pivotIndex + 1, endIndex);  
}

*/\*\*  
 \* 分治算法（单边循环）  
 \*  
 \** ***@param*** *array 原数组  
 \** ***@param*** *startIndex 起始下标  
 \** ***@param*** *endIndex 结束下标  
 \** ***@return*** *\*/*public static int partition2(int[] array, int startIndex, int endIndex) {  
 // 选取基准元素  
 int pivot = array[startIndex];  
 int mark = startIndex;  
 for (int i = startIndex + 1; i <= endIndex; i++) {  
 if (array[i] < pivot) {  
 mark++;  
 int p = array[mark];  
 array[mark] = array[i];  
 array[i] = p;  
 }  
 }  
 array[startIndex] = array[mark];  
 array[mark] = pivot;  
 return mark;  
}

1. Java实现层级树形菜单