

Java 面试题（最新）

我的建议：把出现的程序代码编写的尝试做下。

试试自己的 CPU 到底有多快

1. 点开始——附件——计算器
2. 在计算器的“查看”处，选择“科学型”
3. 打开任务管理器（Ctrl+Alt+Del），并切换到“进程”，然后点“查看”——“选择列”
4. 在“选择列”里，将“CPU时间”选上，确定

然后就可以开始测试了~~~~

计算器里输入 3 万，然后点 n!

再测测 10 万的 n!

（测 10 万的时候，要把计算器关闭后重新打开，不然任务管理器会累计你上次的测试时间。遇到提示 XXXXX 很慢，点继续就行）

一. 最基本题型（说明：此类题型比较简单）

1. 烧一根不均匀的绳，从头烧到尾总共需要1个小时。现在有若干条材质相同的绳子，问如何用烧绳的方法来计时一个小时十五分钟呢？（10分钟-20分钟）
2. 你有一桶果冻，其中有黄色、绿色、红色三种，闭上眼睛抓取同种颜色的两个。抓取多少个就可以确定你肯定有两个同一颜色的果冻？（5秒-1分钟）
3. 如果你有无穷多的水，一个3公升的提桶，一个5公升的提桶，两只提桶形状上下都不均匀，问你如何才能准确称出4公升的水？（40秒-3分钟）
4. 一个岔路口分别通向诚实国和说谎国。来了两个人，已知一个是诚实国的，另一个是说谎国的。诚实国永远说实话，说谎国永远说谎话。现在你要去说谎国，但不知道应该走哪条路，需要问这两个人。请问应该怎么问？（20秒-2分钟）
5. 12个球一个天平，现知道只有一个和其它的重量不同，问怎样称才能用三次就找到那个球。13个呢？（注意此题并未说明那个球的重量是轻是重，所以需要仔细考虑）（5分钟-1小时）
6. 在9个点上画10条直线，要求每条直线上至少有三个点？（3分钟-20分钟）
7. 在一天的24小时之中，时钟的时针、分针和秒针完全重合在一起的时候有几次？都分别是什么时间？你怎样算出来的？（5分钟-15分钟）

二. 没有答案型（说明：这些题显然不是考你智力。而考的是你的反应能力。这种题大多数没有答案，但是要看你的反应喽！）

1. 为什么下水道的盖子是圆的？
2. 中国有多少辆汽车？
3. 将汽车钥匙插入车门，向哪个方向旋转就可以打开车锁？
4. 如果你要去掉中国的34个省（含自治区、直辖市和港澳特区及台湾省）中的任何一个，你会去哪一个，为什么？
5. 多少个加油站才能满足中国的所有汽车？
6. 想象你站在镜子前，请问，为什么镜子中的影像可以颠倒左右，却不能颠倒上下？
7. 为什么在任何旅馆里，你打开热水，热水都会瞬间倾泻而出？
8. 你怎样将Excel的用法解释给你的奶奶听？

9.你怎样重新改进和设计一个ATM银行自动取款机？

10.如果你不得不重新学习一种新的计算机语言，你打算怎样着手来开始？

11.如果你的生涯规划中打算在5年内受到奖励，那获取该项奖励的动机是什么？观众是谁？

12.如果微软告诉你，我们打算投资五百万美元来启动你的投资计划，你将开始什么样商业计划？为什么？

13.如果你能够将全世界的电脑厂商集合在一个办公室里，然后告诉他们将被强迫做一件事，那件事将是什么？

三．难题（说明：这类题有一定难度，如果得不到答案，也不能说明什么。如果你想到了解题思路，那么答案马上就能出来。如果想不到思路，那么.....就别想解出来了。）

1.你让工人为你工作7天，回报是一根金条，这个金条平分成相连的7段，你必须在每天结束的时候给他们一段金条。如果只允许你两次把金条弄断，你如何给你的工人付费？

2.有一辆火车以每小时15公里的速度离开北京直奔广州，同时另一辆火车每小时20公里的速度从广州开往北京。如果有一只鸟，以30公里每小时的速度和两辆火车同时启动，从北京出发，碰到另一辆车后就向相反的方向返回去飞，就这样依次在两辆火车之间来回地飞，直到两辆火车相遇。请问，这只鸟共飞行了多长的距离？

3.你有四个装药丸的罐子，每个药丸都有一定的重量，被污染的药丸是没被污染的药丸的重量+1。只称量一次，如何判断哪个罐子的药被污染了？

4.门外三个开关分别对应室内三盏灯，线路良好，在门外控制开关时候不能看到室内灯的情况，现在只允许进门一次，确定开关和灯的对应关系？

5.人民币为什么只有1、2、5、10的面值？

6.你有两个罐子以及50个红色弹球和50个蓝色弹球，随机选出一个罐子，随机选出一个弹球放入罐子，怎么给出红色弹球最大的选中机会？在你的计划里，得到红球的几率是多少？

四．超难题（说明：如果你是第一次看到这种题，并且以前从来没有见过类似的题型，并且能够在半个小时之内做出答案。只能说明你的智力超常.....）

第一题．五个海盗抢到了100颗宝石，每一颗都一样大小和价值连城。他们决定这么分：

抽签决定自己的号码（1、2、3、4、5）

首先，由1号提出分配方案，然后大家表决，当且仅当超过半数的人同意时，按照他的方案进行分配，否则将被扔进大海喂鲨鱼

如果1号死后，再由2号提出分配方案，然后剩下的4人进行表决，当且仅当超过半数的人同意时，按照他的方案进行分配，否则将被扔入大海喂鲨鱼

依此类推

条件：每个海盗都是很聪明的人，都能很理智地做出判断，从而做出选择。

问题：第一个海盗提出怎样的分配方案才能使自己的收益最大化？

第二题．一道关于飞机加油的问题，已知：

每个飞机只有一个油箱，

飞机之间可以相互加油（注意是相互，没有加油机）

一箱油可供一架飞机绕地球飞半圈，

问题：

为使至少一架飞机绕地球一圈回到起飞时的飞机场，至少需要出动几架飞机？（所有飞机从同一机场起飞，而且必须安全返回机场，不允许中途降落，中间没有飞机场）

五. 主观题（说明：在以后的工作过程中，我们可定会犯这样那样的错误。既然错误已经酿成，损失在所难免，我们只能想办法把损失减少到最小。如果能巧妙地回答出这些问题，再发生错误的情况下。能让客户有最少的抱怨，公司有最少的损失。）

1.某手机厂家由于设计失误，有可能造成电池寿命比原来设计的寿命短一半（不是冲放电时间），解决方案就是免费更换电池或给50元购买该厂家新手机的折换券。请给所有已购买的用户写信告诉解决方案。

2.一高层领导在参观某博物馆时，向博物馆馆员小王要了一块明代的城砖作为纪念，按国家规定，任何人不得将博物馆收藏品变为私有。博物馆馆长需要如何写信给这位领导，将城砖取回。

3.营业员小姐由于工作失误，将2万元的笔记本电脑以1.2万元错卖给李先生，王小姐的经理怎么写信给李先生试图将钱要回来？

六.算法题（说明：这些题就不是什么花样了，考的是你的基础知识怎么样。再聪明而没有实学的人都会被这些题所淘汰。）

- 1.链表和数组的区别在哪里？
- 2.编写实现链表排序的一种算法。说明为什么你会选择用这样的方法？
- 3.编写实现数组排序的一种算法。说明为什么你会选择用这样的方法？
- 4.请编写能直接实现`strstr()`函数功能的代码。
- 5.编写反转字符串的程序，要求优化速度、优化空间。
- 6.在链表里如何发现循环链接？
- 7.给出洗牌的一个算法，并将洗好的牌存储在一个整形数组里。
- 8.写一个函数，检查字符是否是整数，如果是，返回其整数值。（或者：怎样只用4行代码编写出一个从字符串到长整形的函数？）
- 9.给出一个函数来输出一个字符串的所有排列。
- 10.请编写实现`malloc()`内存分配函数功能一样的代码。
- 11.给出一个函数来复制两个字符串A和B。字符串A的后几个字节和字符串B的前几个字节重叠。
- 12.怎样编写一个程序，把一个有序整数数组放到二叉树中？
- 13.怎样从顶部开始逐层打印二叉树结点数据？请编程。
- 14.怎样把一个链表掉个顺序（也就是反序，注意链表的边界条件并考虑空链表）？

北京微软面试题在微软的经历-微软面试

小明和小强都是张老师的学生，张老师的生日是 M 月 N 日，
2 人都知道张老师的生日是下列 10 组中的一天，
张老师把 M 值告诉了小明，把 N 值告诉了小强，
张老师问他们知道他的生日是那一天吗？

3 月 4 日 3 月 5 日 3 月 8 日

6 月 4 日 6 月 7 日

9 月 1 日 9 月 5 日

12月1日 12月2日 12月8日

小明说：如果我不知道的话，小强肯定也不知道

小强说：本来我也不知道，但是现在我知道了

小明说：哦，那我也知道了

我在网上看了有好几种答案，我自己当时做的是这个，越想越迷糊

这是我自己做的：

3月4日 3月5日 3月8日

6月4日 6月7日

9月1日 9月5日

12月1日 12月2日 12月8日

小明思考的小强该拿到什么

“如果我不知道的话”

只有6月7日和12月2日这两组日期的日数是唯一的。由此可知，如果小强得知的是7或者2，那么他必定知道了老师的生日。N值不能为7和2

“小强肯定也不知道”

小强知道的是4、5、8、1时都是重复的，所以他才肯定这样说，

“本来我也不知道”

如果小强拿到7和2时就知道了，所以小强排除M值6和12，此时他只在M值为3和9中考虑N的值1、4、5、8

只有5为重复所以N值不能为5

“但是现在我知道了”

排除N值4和8，因为M值为6和12中对应应有4和8

所以答案就是9月1号

网上还有这样一种说法：

3月4日 3月5日 3月8日

6月4日 6月7日

9月1日 9月5日

12月1日 12月2日 12月8日中只有6月7日和12月2日这两组日期的日数是唯一的。由此可知，如果小强得知的是7或者2，那么他必定知道了老师的生日。

所以排除这两个数据，则数据剩下了

3月4日 3月5日 3月8日

6月4日

9月1日 9月5日

12月1日 12月8日

在这组数据中只有六月是唯一的，而小明说我不知道。

可见6月4日由此排除。则数据剩下：

3月4日 3月5日 3月8日

9月1日 9月5日

12月1日 12月8日

此刻小强说我原本不知道，但现在我知道了。由于小强知道的是日子，这里有唯一的日子既是3月4日。所以小明马上说那我也知道了。可见日子的确是3月4日

---5.28.2008

使用 trac 调试服务器端代码

```
import java.io.File;
import java.io.PrintStream;
import java.io.FileNotFoundException;

private void printDebug(String msg){
    PrintStream ps = null;
    try {
        throw new Exception(msg);
    } catch(Exception e) {
        try {
            String filename = "trac.log";
            File f= new File(filename);
            if(!f.exists()){
                f.createNewFile();
            }
            ps = new PrintStream(new java.io.FileOutputStream(f,true));
            e.printStackTrace(ps);
        } catch (Exception e1) {
            e1.printStackTrace();
        } finally{
            if(ps != null) ps.close();
        }
    }
}
```

最全的华为面试最新的华为面试题---华为面试经历

首先声明这事是我在华为北研所面试的亲身经历

华为的面试是我认为最可笑的面试。

首先，华为主动给我打电话请我参加面试，就让我觉得奇怪，因为我的专业和工作经验跟他们要招聘的职位丝毫没有关系，他们却说看中我的学历和英语水平，觉得我可以胜任，并且把华为夸的跟朵花似的，向我介绍工作内容的时候基本上是含糊不清，内容简化。既然这样的话，我决定参加这个面试看看。

之前就看过关于华为的帖子，多数是说华为累，管理军事化之类，还有垃圾般的企业文化。在参加完 1，2 面之后，感觉华为的人真的没有什么面带笑容的，跟 机器人一样，而且华为北研所人满为患。1，2 面的人倒是还算和气，我的笔试也算顺利通过，2 面的那个叔叔还请我在

华为食堂吃了饭，而 1 面的那个人则是在吃完饭的时候逃之夭夭，也许他在想请我吃了这顿饭，他的这个找人指标得到钱就要打折扣了吧。

主要问题出现在 3，4 面的时候，使我对这个拥有名企声誉的民营企业产生了极大的失望，使我认定这个民营企业缺乏大家风范，缺乏对人的起码尊重。

事情是这样的，通知我上午 9：40 面试，10：10 分我才被叫到坐到了面试人面前。刚刚坐下，两个面试人之一就跟我要所有的本科学位证书，然后对一些证书进行反复的阅读，鉴定，还拿走我证件在一个台灯下面作判断，另外一个面试人在这个空隙一个劲地跟我说华为干活很累，而且问我是不是像我写的那样可以到全国各地工作，这时我低头一看，那个可以到全国各地工作的选项，根本不是我填的，而是由其他人代笔填写的，如果不是这个面试人特意问我一下，估计我还被蒙在鼓里。那个在台灯下仔细研究我证书的人回来后告诉我我的学位证书是假的，我听完马上问他，是凭什么判断的，他告诉我根据很多来面试的其他学生的证书上都有防伪标志，而我的没有。可是这位先生你有没有想过，我可不是刚毕业的本科学生，这是我 3 年前的证书，我现在是一个研究生毕业并且工作了一年的人，我们一届的本科学位证书跟现在的当然会不一样。但是你这样轻易的心口开河，断定我的证书是假的，你连最起码的调查都没作，凭什么妄下结论。不仅这样，这个人还跟我说，没关系，我们知道有些学生毕业了但是没获得学位证书。这是我有史以来听过的最可笑的话，如果这个证书有问题，你可以打电话到学校调查阿，这时这位仁兄又说了，打电话到学校只能知道有没有你这么个学生，并不能证明你的学士学位。

本来我可以把证书留给华为去做鉴定，或者是开个鉴定说明书，但是这段谈话建立在不平等的基础上，我觉得没有必要再继续给他们机会浪费我自己的时间，于是我结束了这次谈话并头也不回的走了出去。作为华为的面试官，你们缺乏对人的起码尊重和信任，你们的心态不平衡，把自己当成大老爷，摆出各种姿态来对付前来面试的人，这不是一个大企业应该具有的“品质”，你们即使拼命宣传你们的企业文化，也掩盖不住你们狭隘本质。通过这次面试谈话，我彻底放弃了对华为的那一丝希望。作为一个正在求职的人，我想说永别了华为，我对这个拥有垃圾般文化的企业有了一次真实彻底的认识。

奉劝将要到华为面试的人，千万要三思而后行。不要给这个企业任何机会去欺负应试者。

从毕业出来工作,一直到去华为面试前,能够去华为工作一直是自己一个职业生涯中的梦想,不管是在书店里还是在网上,太多关于华为的介绍了,关于华为的具体事情我就不想多说了,在这里只是将华为的一个面试经历讲出来,供大家对华为的一个认识.前段时间有个关系不错的同事从我的公司辞职后,然后就去了华为,在我们偶尔的一次聊天中,我将我也想去华为的想法告诉了他,这位同事挺仗义,说没问题,他帮我投简历,后来我就发了一个求职简历的邮件,第二天就收到了华为面试通知的电话,在通电话中,一位女士问了几个关于 C 语言方面的问题,然后就告诉我在下周五去华为面试.下周五如约来到华为,华为的面试地点在华为科研大楼的东门,具体的咱也搞不清楚,由于去的早一些,有幸看到华为上班的人群,那个是多啊,光大巴车都不下于几十辆,好像是从深圳各个方向来的人群走来,光从这点就感觉到华为的人气真是旺.接下来看到和我一样来面试的人也不少,约摸着得有个几百人,听朋友讲,华为每周五都是面试日,如果每周都按这个数来算的话,每个月通过华为面试给深圳带来的消费都是一笔不少的数目,然后排队拿身份证换入门贴进入华为,华为具体的面试地点在食堂,感觉食堂很大,比大学上学时的食堂还要大,本人面试的是测

试工作,接着把简历交给华为的人事工作人员,然后就是华为的第一轮笔试,笔试内容都是一些基本的东西,虽说考试题量不大,但题目中含盖了操作系统,数据结构,数据库,C语言多方面,而且这些题看起来不难,但实际上又都是些平时不太注意的知识点,大公司就是大公司,做起来还真不容易,很轻松的做完了试卷,感觉应该六十分不成问题,然后大家就坐在那里等下一轮技术面试的通知,很幸运,第一个技术面试的人就是我,当时过来一个主考官,接着就问了一些关于测试方面的问题,感觉主要是考查是否是真正的工作经验,然后就问了一些关于C语言和数据结构方面的问题,和那位主考官谈的很开心,也很顺利,最后他提出让我做两道题,一个关于排序的问题,一个是关于查找的问题,还好这两道题也不是很难,很快就搞定了.他告诉我很快就有下一轮的面试通知,OK,感觉第一关过了心里舒服多了,心想下一轮应该是综合面试了吧!谁知大约过了半小时以后,又过来一位主考官,坐下来问的还是技术方面的问题,心想这可能是第二轮技术面试,这位主考官问的问题更偏重于工作经验方面的一些问题,如果没有实际的工作经验很容易被问的卡壳,和这位主考官谈的也算顺利,最后他告诉我,他这一关是过了,等下面的综合面试,第二轮技术面试完后,已经到了中午吃饭的时间,由华为的人事工作人员给留下来的人一人发了一个小票,然后带我们去二楼的自助餐餐厅吃饭,饭挺好,菜,饮料,水果什么都有,吃完饭后,又到楼下去等,在等下午综合面试的这个时候,发现下午面试的人换了,换成了一男一女,这两个人就在那里翻我们剩下这几个人的简历,这时其中的一个男的过来了,走到我的前面,对我讲:"你的简历不符合我们的要求,你是专科生,我们要的是本科生以上",当时我一听这话蒙了,是啊,我在简历上写的很清楚,我是专科毕业,成人函授本科还在读,前两轮的面试都过了,为什么到这里说出这样的话来,我告诉他:"我的确是专科生,现在正在读本科,但我的工作经验足够胜任该项工作",然后又和他讲了一些关于我工作经验的事情,说那么多也就是为了能够给我一个机会,谁知这小子眉头一皱,就是说不行,绝对不可能,当时就要请我离开,气得我不行了,现在想想那小子当时的眼神就来气(如果是在外面的话,非和他打上一架),我看这样也没有意义了,只好起身离开,离开时我告诉他"我对华为很失望",接着就走出了食堂.至此,我的华为面试就结束了,后来我那位同事又帮忙问了一下,也没有什么结果.怎么说呢,这次面试的经历,一方面自己的学历真的不够高,这怪不得别人;另一方面,感觉华为的这种制度有些不合理,但这也怪不得华为,每个公司都有自己的一些硬性规定;再一个就是感觉华为的面试人员的素质个别的实在太差了,其实结束情况了,再讲什么也就没有意思了,在这里写出来,只是供大家参阅一下.谢谢!

--5.29.2008

华为面试题1

- Q1: 请你分别划划 OSI 的七层网络结构图, 和 TCP/IP 的五层结构图?
- Q2: 请你详细的解释一下 IP 协议的定义, 在哪个层上面, 主要有什么作用? TCP 与 UDP 呢?
- Q3: 请问交换机和路由器分别的实现原理是什么? 分别在哪个层次上面实现的?
- Q4: 请问 C++ 的类和 C 里面的 struct 有什么区别?
- Q5: 请讲一讲析构函数和虚函数的用法和作用?
- Q6: 全局变量和局部变量有什么区别? 实怎么实现的? 操作系统和编译器是怎么知道的?
- Q7: 一些寄存器的题目, 主要是寻址和内存管理等一些知识。
- Q8: 8086 是多少位的系统? 在数据总线上是怎样实现的?
- 1、局部变量能否和全局变量重名?

答: 能, 局部会屏蔽全局。要用全局变量, 需要使用 "global"

局部变量可以与全局变量同名, 在函数内引用这个变量时, 会用到同名的局部变量, 而不会

用到全局变量。对于有些编译器而言，在同一个函数内可以定义多个同名的局部变量，比如在两个循环体内都定义一个同名的局部变量，而那个局部变量的作用域就在那个循环体内。

2、如何引用一个已经定义过全局变量？

答：extern

可以用引用头文件的方式，也可以用 extern 关键字，如果用引用头文件方式来引用某个在头文件中声明的全局变理，假定你将那个变写错了，那么在编译期间会报错，如果你用 extern 方式引用时，假定你犯了同样的错误，那么在编译期间不会报错，而在连接期间报错。

3、全局变量可不可以定义在可被多个 .C 文件包含的头文件中？为什么？

答：可以，在不同的 C 文件中以 static 形式来声明同名全局变量。

可以在不同的 C 文件中声明同名的全局变量，前提是其中只能有一个 C 文件中对此变量赋初值，此时连接不会出错

4、语句 for(;1 ;)有什么问题？它是什么意思？

答：和 while(1)相同。

5、do..... while 和 while..... do 有什么区别？

答：前一个循环一遍再判断，后一个判断以后再循环

6、请写出下列代码的输出内容

以下是引用片段：

```
#include
main()
{
int a,b,c,d;
a=10;
b=a++;
c=++a;
d=10*a++;
printf("b, c, d: %d, %d, %d", b, c, d);
return 0;
}
```

答：10， 12， 120

7、static 全局变量与普通的全局变量有什么区别？static 局部变量和普通局部变量有什么区

别?static 函数与普通函数有什么区别?

全局变量(外部变量)的说明之前再冠以 `static` 就构成了静态的全局变量。全局变量本身就是静态存储方式, 静态全局变量当然也是静态存储方式。这两者在存储方式上并无不同。这两者的区别虽在于非静态全局变量的作用域是整个源程序, 当一个源程序由多个源文件组成时, 非静态的全局变量在各个源文件中都是有效的。而静态全局变量则限制了其作用域, 即只在定义该变量的源文件内有效, 在同一源程序的其它源文件中不能使用它。由于静态全局变量的作用域局限于一个源文件内, 只能为该源文件内的函数公用, 因此可以避免在其它源文件中引起错误。

从以上分析可以看出, 把局部变量改变为静态变量后是改变了它的存储方式即改变了它的生存期。把全局变量改变为静态变量后是改变了它的作用域, 限制了它的使用范围。

`static` 函数与普通函数作用域不同。仅在本文件。只在当前源文件中使用的函数应该说明为内部函数(`static`), 内部函数应该在当前源文件中说明和定义。对于可在当前源文件以外使用的函数, 应该在一个头文件中说明, 要使用这些函数的源文件要包含这个头文件

`static` 全局变量与普通的全局变量有什么区别: `static` 全局变量只初使化一次, 防止在其他文件单元中被引用;

`static` 局部变量和普通局部变量有什么区别: `static` 局部变量只被初始化一次, 下一次依据上一次结果值;

`static` 函数与普通函数有什么区别: `static` 函数在内存中只有一份, 普通函数在每个被调用中维持一份拷贝

8、程序的局部变量存在于(堆栈)中, 全局变量存在于(静态区)中, 动态申请数据存在于(堆)中。

9、设有以下说明和定义:

```
typedef union {long i; int k[5]; char c;} DATE;
```

```
struct data { int cat; DATE cow; double dog;} too;
```

```
DATE max;
```

则语句 `printf("%d",sizeof(struct data)+sizeof(max));` 的执行结果是: 52

答: DATE 是一个 union, 变量公用空间. 里面最大的变量类型是 `int[5]`, 占用 20 个字节. 所以它的大小是 20

data 是一个 struct, 每个变量分开占用空间. 依次为 `int4 + DATE20 + double8 = 32`.

所以结果是 `20 + 32 = 52`.

当然...在某些 16 位编辑器下, int 可能是 2 字节,那么结果是 int2 + DATE10 + double8 = 20

10、队列和栈有什么区别?

队列先进先出, 栈后进先出

11、写出下列代码的输出内容

以下是引用片段:

```
#include
int inc(int a)
{
    return(++a);
}
int multi(int*a,int*b,int*c)
{
    return(*c=*a**b);
}
typedef int(FUNC1)(int in);
typedef int(FUNC2) (int*,int*,int*);
void show(FUNC2 fun,int arg1, int*arg2)
{
    INCp=&inc;
    int temp =p(arg1);
    fun(&temp,&arg1, arg2);
    printf("%d\n",*arg2);
}
main()
{
    int a;
    show(multi,10,&a);
    return 0;
}
```

答: 110

12、请找出下面代码中的所以错误

说明: 以下代码是把一个字符串倒序, 如“abcd”倒序后变为“dcba”

以下是引用片段:

```
1、 #include"string.h"
2、 main()
3、 {
```

```

4、 char*src="hello,world";
5、 char* dest=NULL;
6、 int len=strlen(src);
7、 dest=(char*)ma lloc(len);
8、 char* d=dest;
9、 char* s=src[len];
10、 while(len--!=0)
11、 d++=s--;
12、 printf("%s",dest);
13、 return 0;
14、 }

```

答：

方法 1:

以下是引用片段：

```

int main()
{
char* src = "hello,world";
int len = strlen(src);
char* dest = (char*)malloc(len+1);// 要为\0 分配一个空间
char* d = dest;
char* s = &src[len-1];//指向最后一个字符
while( len-- != 0 )
*d++=*s--;
*d = 0;//尾部要加\0
printf("%s\n",dest);
free(dest);// 使用完，应当释放空间，以免造成内存汇泄露
return 0;
}

```

方法 2:

以下是引用片段：

```

#include
#include
main()
{
char str[]="hello,world";
int len=strlen(str);
char t;
for(int i=0; i
{

```

```

t=str[i];
str[i]=str[len-i-1]; str[len-i-1]=t;
}
printf("%s",str);
return 0;
}

```

-----5.30, 2008-5-30

华为面试题 2

1.-1,2,7,28,,126 请问 28 和 126 中间那个数是什么?为什么?

第一题的答案应该是 $4^3-1=63$

规律是 n^3-1 (当 n 为偶数 0, 2, 4) n^3+1 (当 n 为奇数 1, 3, 5)

答案: 63

2.用两个栈实现一个队列的功能?要求给出算法和思路!

设 2 个栈为 A,B, 一开始均为空.

入队:

将新元素 push 入栈 A;

出队:

(1)判断栈 B 是否为空;

(2)如果不为空, 则将栈 A 中所有元素依次 pop 出并 push 到栈 B;

(3)将栈 B 的栈顶元素 pop 出;

这样实现的队列入队和出队的平摊复杂度都还是 $O(1)$, 比上面的几种方法要好。3.在 c 语言库函数中将一个字符转换成整型的函数是 atool()吗, 这个函数的原型是什么?

函数名: atol

功 能: 把字符串转换成长整型数

用 法: long atol(const char *nptr);

程序例：

以下是引用片段：

```
#include
#include
int main(void)
{
    long l;
    char *str = "98765432";
    l = atol(lstr);
    printf("string = %s integer = %ld\n", str, l);
    return(0);
}
```

13. 对于一个频繁使用的短小函数,在 C 语言中应用什么实现,在 C++中应用什么实现?

c 用宏定义, c++用 inline

14. 直接链接两个信令点的一组链路称作什么?

PPP 点到点连接

15. 接入网用的是什么接口?

16. voip 都用了那些协议?

17. 软件测试都有那些种类?

黑盒：针对系统功能的测试

白合：测试函数功能，各函数接口

18. 确定模块的功能和模块的接口是在软件设计的那个阶段完成的?

概要设计阶段

19.

enum string

{x1,x2,x3=10,x4,x5,} x;

问 x= 0x801005, 0x8010f4 ;

20.

```
unsigned char *p1;
```

```
unsigned long *p2;
```

```
p1=(unsigned char *)0x801000;
```

```
p2=(unsigned long *)0x810000;
```

请问 $p1+5=$;

$p2+5=$;

选择题:

21.Ethternet 链接到 Internet 用到以下那个协议?

A.HDLC;B.ARP;C.UDP;D.TCP;E.ID

22.属于网络层协议的是:

A.TCP;B.IP;C.ICMP;D.X.25

23.Windows 消息调度机制是:

A.指令队列;B.指令堆栈;C.消息队列;D.消息堆栈;

24.

```
unsigned short hash(unsigned short key)
```

```
{
```

```
return (key>>)%256
```

```
}
```

请问 hash(16),hash(256)的值分别是:

A.1.16;B.8.32;C.4.16;D.1.32

找错题:

25. 请问下面程序有什么错误？

```
int a[60][250][1000], i, j, k;
```

```
for(k=0; k<=1000; k++)
```

```
for(j=0; j<250; j++)
```

```
for(i=0; i<60; i++)
```

```
a[i][j][k]=0;
```

把循环语句内外换一下

26.

以下是引用片段：

```
#define Max_CB 500
void LmiQueryCSmd(Struct MSgCB * pmsg)
{
    unsigned char ucCmdNum;
    .....
    for(ucCmdNum=0; ucCmdNum
    {
        .....;
    }
}
```

死循环

27. 以下是求一个数的平方的程序, 请找出错误：

```
#define SQUARE(a)((a)*(a))
```

```
int a=5;
```

```
int b;
```

```
b=SQUARE(a++);
```

28.

以下是引用片段：

```
typedef unsigned char BYTE
int examply_fun(BYTE gt_len; BYTE *gt_code)
```

```

{
BYTE *gt_buf;
gt_buf=(BYTE *)MALLOC(Max_GT_Length);
.....
if(gt_len>Max_GT_Length)
{
return GT_Length_ERROR;
}
.....
}

```

问答题:

29.IP Phone的原理是什么？

IPV6

30.TCP/IP 通信建立的过程怎样，端口有什么作用？

三次握手，确定是哪个应用程序使用该协议

31.1 号信令和 7 号信令有什么区别，我国某前广泛使用的是那一种？

32.列举 5 种以上的电话新业务？

---5.31.2008

正方形不是长方形的终极解决办法 -----java 面试题

之前人们讨论的正方形长方形的问题的关键在哪里？我觉得就在于改动长方形的边的长度。我们可以这么考虑一下，一个长方形的 `instance` 的边长应该是可变的吗？我觉得一旦一个长方形的边长改变之后它就成了另一个长方形了（一个新的 `instance`）。所以长方形类里面不应该有改变其边长的方法，一个长方形实例各个的边长应当在 `new` 它的时候确定下来，并且它们应当是 `immutable` 的。基于这种考虑，我设计的长方形和正方形的类如下所示：

//长方形

```

public class Rectangle {
    private final int width;
    private final int height;

    public Rectangle(int width, int height) {
        this.width = width;
        this.height = height;
    }
}

```



```

    }

    public int getWidth() {
        return width;
    }

    public int getHeight() {
        return height;
    }

    public int getArea() {
        return width*height;
    }
}

//正方形
public class Square extends Rectangle{
    private final int side;

    public Square(int side) {
        super(side, side);
        this.side = side;
    }

    public int getSide() {
        return side;
    }
}

```

这种继承关系就既符合现实中的父子关系也遵循 LSP。之所以这么设计，我的想法是一个类所具有的方法不应当能够改变其本质。比如有一个 **Men** 类，它可以有 `eat()`,`sleep()`,`work()`,`make Lovewith(Person p)`方法，但是如果你在里面定义 `denatureToWomen ()`,`denatureToEunuch()`就很不恰当了，因为这改变了其本质，导致这个 **Men** 的实例不再属于 **Men** 类（至少已经和现实不吻合）了。除非这两个方法不能改变该实例本质，否则在 **Men** 里面定义这两个方法本身就是有问题的。不过如果用下面这种方式定义也许可行：

```

public Women denatureToWomen() {
    Women w = new Women();
    //set attributes here
    return w;
}

public Eunuch denatureToEunuch() {
    Eunuch e = new Eunuch();
    //set attributes here
}

```

```
return e;
}
```

这样一来，调用 `denatureToWomen()` 会产生一个新的实例，原来的那个 `Men` 实例依然存在，这和现实生活依然不吻合，现实生活中一个实例不光可以上型 (upcast)，还可以平行型

总之一句话，一个类的方法不应该改变其实例的本质。

Web 开发门户网站建设方案 ---java125

什么是 Portal （门户） ？

Portal 的组成可以分为三部份 (1) Portal Server (2) Portlet Container (3) Portlet

1) Portal Server 的定义

一个 Portal（门户网站）就是指一个 Web-based 的系统，通常都会提供个人化设置、单一登陆、以及由各种不同来源或不同网站取得各式各样的信息，并且将这些信息放在网页之中组合而成的呈现平台，门户网站会有精巧的个人化设置去提供定制的网页，当不同等级的使用者来浏览该页面将获得不同的信息内容。

2) Portlet Container 的定义

portlet container 是提供 portlets 执行的环境，包含了许多 portlets 并且管理他们的生命周期，他也会永远保存着 portlets 的喜好设置，一个 portlet container 接收到来自 portal 的请求后，接着将这个请求传递给存在 container 的 portlet 执行。portlet container 没有义务去组合 portlets 产生的信息内容，这个工作必须由 portal 来处理。portal 和 portlet container 可以放在一起视为同一个系统的组件，或者分开成为两个独立的组件。

什么是 Portlet?

一个 Portlet 是以 Java 技术为技术的 Web 组件，由 Portlet Container 所管理，专门处理客户的 request 以及产生各种动态的信息内容。Portlets 为可插式 (pluggable) 的客户界面组件，提供呈现层成为一个信息系统。

这些由 portlet 产生的内容也被称为片段 (fragment)，而片段是具有一些规则的 Markup(HTML、XHTML、WML)，而且可以和其他的片段组合而成一个复杂的文件。而 Portlet 中的内容正常来说是与其它 Portlet 的内容聚合而成为一个 Portal 网页。而 Portlet 的生命周期是被 Portlet Container 所管理控制的。

客户端和 portlets 的互动是由 portal 通过典型的 request/response 方式实现，正常来说，客户会和 portlets 所产生的内容互动，举例来说，根据下一步的连接或者是确认送出的表单，结果 portal 将会接收到 portlet 的动作，将这个处理状况转向到目标 portlet。这些 portlet 内容的产生可能会因为不同的使用者而有不同的变化，完全是根据客户对于这个 portlet 的设置。

门户网站（PORTAL）的作用

单点登录

Portal 的单点登录是基于应用集成系统的，统一身份认证系统实现系统的单点登录：用户只需要进行一次登录，就可以访问到所有的授权服务；所有的应用系统可以提供统一的用户管理。

Portal 的用户权限管理能够控制到页面或 Portlet

个性化定制

针对不同的用户登录，可以展示不同的内容。可以根据用户的关注点不同来为用户提供定制桌面的功能。

资源整合

完成应用整合，以及业务整合的功能。不同的 web 系统，web 业务平台可以通过 portlet 的 web 剪切功能直接放入门户里面来。

门户网站可选方案分析

厂商的门户平台：

1、IBM WebSphere Portal

2、BEA PORTAL

厂商的门户平台，一半都有自己的开发标准，现在都要求用 JSR168 的标准进行开发，厂商的门户成本比较高，稳定性好，服务好产品的后期维护以及开发都有比较成功的案例；

开源的门户平台

JetSpeed

JetSpeed 是 Apache（[http:// jakarta.apache.org/jetspeed/](http://jakarta.apache.org/jetspeed/)）组织的一个开源项目，IBM 的 WebSphere Portal Server 正是基于 JetSpeed 进行二次开发。

在满足门户可定制性的需求上，引入了门户组件概念 Portlet，Portlet 是一个可插拔的组件，关于 Portlet 有各种定义，JetSpeed 在提交的 JSR-168 portlet API specifications 对 Portlet 的定义如下：

Portlet 是一个 Web 组件，可以被容器管理，可以产生动态内容。Portlet 可以很容易地被插入并且运行于一个 Web 应用中，Portlet 是被设计成聚合大量内容的组合页面，举例：同样一个 Portlet，根据不同的用户，可以产生不同的实例，这些实例中是根据用户设置包含不同的内容，也就是满足用户的可定制性。

为了实现门户内容的有效管理和监控，JetSpeed 的重要内容组件 Slide 引入了 Domain 和 Namespace 方面的概念，对所有资源进行树形结构的控制，在 Slide 中，Domain 是一系列 Namespace 的聚合，它类似文件系统中的“/”根目录，domain 可以控制在其中登记的 Namespace 的访问权限以及执行 Namespace 的初始化和连接等管理工作。

Namespace 是个自我独立的有实体内容的容器，它不能包含指向或连接到其它 namespace，每个应用系统可以是一个 NameSpace，Namespace 包含独立的数据和这些数据的安全访问机制。

JetSpeed 虽然提供了丰富的 Portal 技术功能，但是其可定制性比较差，其重要的内容组件 Slide 也比较难以让人理解，它的安全控制框架与通用的 J2EE 使用 JAAS 不相融合，所有这些都造成了在其基础上进行二次开发的难度。

Liferay

Liferay（<http://www.liferay.com>）代表了完整的 J2EE 应用，使用了 Web、EJB 以及 JMS 等技术，特别是其前台界面部分使用 Struts 框架技术，基于 XML 的 portlet 配置文件可以自由地动态扩展，使用了 Web Services 来支持一些远程信息的获取，使用 Apache Lucene 实现全文检索功能。

Liferay Portal 是一套相当成熟的 Java/J2EE portal system，遵循 Portlet API JSR168 的标准。

作为一个开源 Portal 产品，Liferay Portal 提供对多个独立系统的内容集成，帮助多个组织实现更有效的合作。与其他商业的 Portal 产品相比，Liferay Portal 有着一系列的优良特性，而且不需要付费。

Liferay 的缺点是它缺乏一个简单清晰可拓展的架构设计，portlet 设计显得比较凌乱，进行二次开发有一定的难度。

Pluto

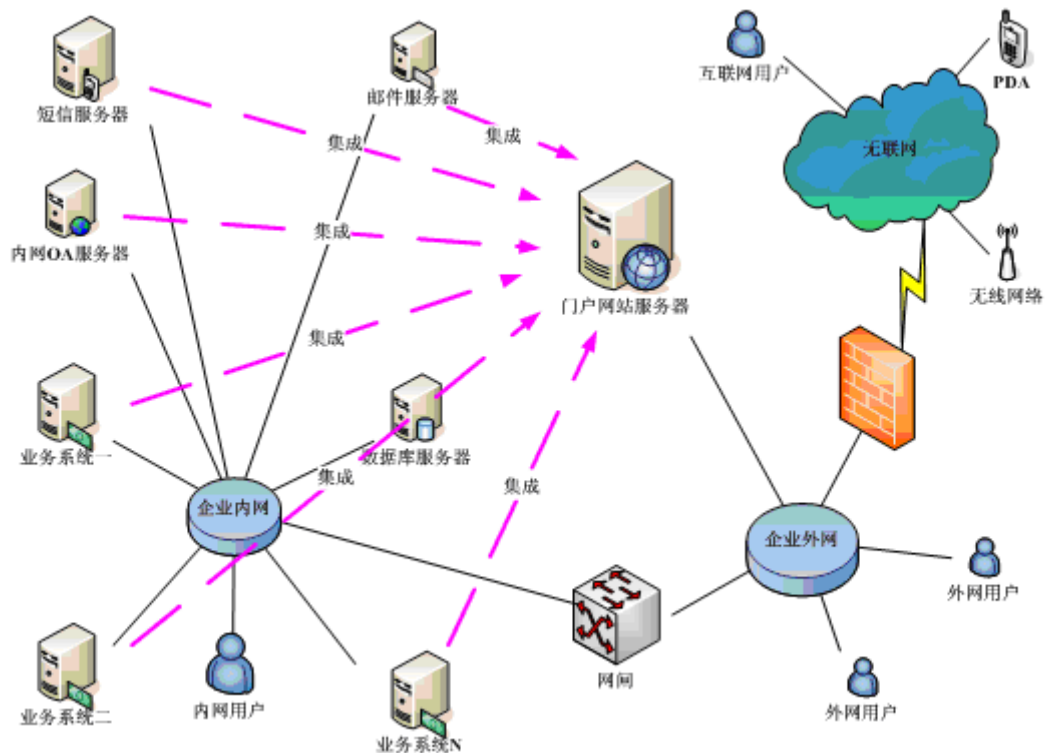
Pluto 是一个满足 Portlet API 规范的 Portlet 容器的实现，它为开发者提供了一个运行 portlets 的工作平台。然而，如果没有一个驱动器（driver），也就是 Portal，的支持的话，运行和测试 Portlet 容器将非常之麻烦。Pluto 本身也提供了一个简单的 Portal 模块，该模块仅仅是为了满足 Portlet 容器和 JSR 168 的需要而写的。

JBoss Portal

JBoss Portal 提供了一个开源符合标准的门户平台，可以用其搭建 / 布局一个门户网站的 web 界面，发布和管理内容以及定制它的用户体验。JBoss Portal 最新版本中在个性化设置，身份验证，工作流都进行重大更进。此外还集成了 Google Gadgets，以提高用户工作效率。

门户网站发展规划

总体规划图



外网的用户通过门户网站可以使用所有的业务系统以及短信、邮件、OA 等原有的系统，实现了应用整合。

由此可见门户的重要地位，公司初步的选定采用 LifeRay 作为公司门户产品的研发基础。

在 LifeRay 的基础上研发出公司自己的门户平台。

实施步骤

平台的学习使用

开发一个平台，首先要能熟练的使用这个平台，分析这个平台的不足与优点，在学习使用的过程中应该能够提出自己对将要研发平台的建议和规划以及平台的组成要素，开发的工具以及环境配置。

Portlet 的学习开发部署

平台会使用以后，下一步就要开发自己的公用的插件了，依据 JSR168portlet 开发的标准，进行 portlet 的开发，这一个阶段首要的任务是开发出自己的插件。然后开始组织培训，如何搭建开发环境如何调试 Portlet，以及平台如何使用，还有就是如何部署自己开发的 portlet。这一步需要开发人员有对新知识渴望学习的兴趣。

产品的包装命名

公司美工在开发人员的指导下，一起对产品的插件以及平台进行界面美化，让美工掌握如何对插件进行美化，以及设计自己的产品的皮肤样式。

产品的测试销售

产品必须经过严格的测试然后由销售人员针对不同的用户群销售。

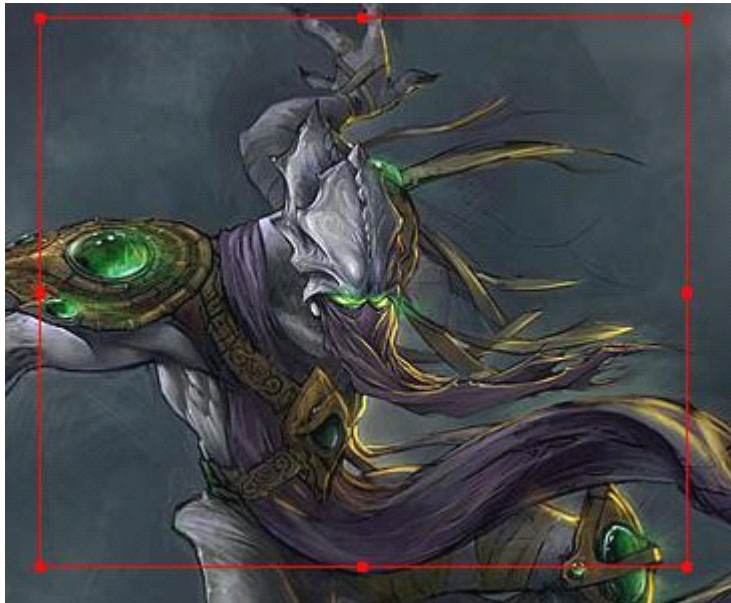
产品维护

作为平台的维护，主要是美工对皮肤的美化，还有就是开发新的插件；这一块可以有一个既懂美工又熟悉 portlet 编程的专人维护就可以了，因为维护的内容很少，所以后期的维护成本得以控制。

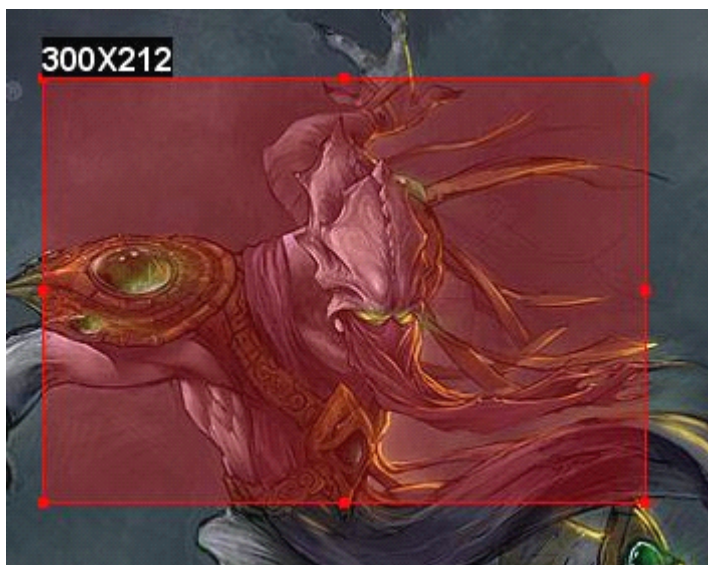
----6.2.2008

用 java 编写的截屏程序改进

☀️ SUNNY ★★☆☆



这是原来的图片，下面是改进后的



和改进的代码部分：

这部分代码插入 Temp 类的 paintComponent 方法中的 if (showTip) 这句的前面

```
Graphics2D g2d = (Graphics2D) g.create();
```

```

g2d.setRenderingHint(RenderingHints.KEY_ANTIALIASING,
    RenderingHints.VALUE_ANTIALIAS_ON);
g2d.setComposite(AlphaComposite.getInstance(
    AlphaComposite.SRC_OVER, 0.3F));
g2d.setColor(Color.RED.brighter().brighter());
int sX = Math.min(startX, endX);
int sY = Math.min(endY, startY);
g2d.fillRect(sX, sY, Math.abs(endX - startX), Math.abs(endY
    - startY));
g2d.setComposite(AlphaComposite.getInstance(
    AlphaComposite.SRC_OVER, 1F));
boolean drawCTip = endX - startX != 0 && endY - startY != 0;
if (drawCTip) {
    String cTip = String.format("%dX%d", Math.abs(endX - startX),
        Math.abs(endY - startY));
    int cTipH = 20;
    Font cTipFont = new Font("system", Font.BOLD, 16);
    g2d.setFont(cTipFont);
    int cTipW = SwingUtilities.computeStringWidth(
        getFontMetrics(cTipFont), cTip);
    g2d.setPaint(Color.BLACK);
    int cStartY = sY - cTipH > 0 ? sY - cTipH : sY;
    g2d.fillRect(sX, cStartY, cTipW, cTipH);
    g2d.setPaint(Color.WHITE);
    g2d.drawString(cTip, sX, cStartY == sY ? sY + cTipH - 3
        : sY - 3);
}
g2d.dispose();

```

怎么样，比起 QQ 的截图程序，我们又近一步了

容易犯错的 java 面试题

```

public class JustTestStatic {
    private static int i=0;
    public static void print(){
        System.out.println("value1:"+i);
        i++;
        System.out.println("value2:"+i);
        add(i);
        System.out.println("value3:"+i);
        add();
        System.out.println("value4:"+i);
    }
}

```

```

private static void add(int j){
    j++;
}
private static void add(){
    i++;
}
public static void main(String[] args){
    JustTestStatic.print();
}
}

```

输出结果多少？

【value1:0 value2:1 value3:1 value4:2】 这是答案

6.3.2008

今天碰到的 2 道 java 面试题

今天碰到的 2 道 java 面试题拿出来与大家分享下

一个是多线程的，一个 IO 流的，多线程的是 NIO 是什么东西啊？这个好象没学过啊？IO 流的是如何把一个 int[] 类型转成 byte 的啊？这个我不会，我说可以去网上查，面试官说我等的就是你这句话，郁闷！

面试官说的没错，人不可能所有知识点都知道的，就算你工作 5 年还是如此。所以学会用 Google 来查资料，是一门实际的技能。

Google 搜: java nio 教程

很多结果可以用。

1. 将 int[] 数组类型转换成 byte[]

```

public byte[] intToBytes(int number, byte[] source)
{
    byte [] ret=null;
    if(source.length<2)
        ret = new byte[2];
    else
        ret = source;
    //converter
    ret[0]=(byte)(number&0xff);
    ret[1]=(byte)((number>>8)&0xff);//还可以转换成 4 个字节的。

    return ret;
}

```


2.将 String 转换成 byte[]类型

```
String s = "面试题";
byte[] b = s.getBytes("GBK");
```

3.将上面的 byte[]转换成 String 类型!6

```
String s = new String(b,"GBK");}
```

Java hibernate 面试题

hibernate 的联合主键怎么设置? inverse 设置成 false 对 sql 语句有什么影响

首先需要建立一个主键类: bqMy

```
public class SCMapping implements Serializable{3\
    private Integer sno;      //学号 A4
    private Integer cno;      //课程号 MC_dS
    private Integer grade;    //成绩 60
    ...F*o
}
```

使用映射文件映射联合主键时,持久化类必须满足三个条件: M6gy?

1. 实现 java.io.Serializable 接口; :1
2. 覆盖 hashCode() 方法; p
3. 覆盖 equals() 方法。 S{#<

接着写映射文件: ?8Lg

```
<hibernate-mapping auto-import="falst">·RCA&x
    <class name="com.tarena.SCMapping" table="scmapping" catalog="jo
blog">T?vP#)
        <composite-id>y\
            <key-property name="sno" type="integer"> 8;0Y
                <column name="Sno" />/Y&4u
            </key-property>u]0M
            <key-property name="cno" type="integer">/t5
                <column name="Cno" />u.
            </key-property>
        </composite-id>#Uq
        <property name="grade" type="integer">XT
        <column name="Grede"/>ZY\b5
        </property>PS!
```

©www.java125.cn N/o'

</class>

</hibernate-mapping>))

©www.java125.cn mdu

然后调用 Hibernae Api 测试。nkc36?

©www.java125.cn FE)@q4

-----umNTU:

关于 inverse 设置成 false 对 sql 语句有什么影响: 81o

inverse 属性默认是 false, 就是说关系的两端都来维护关系。@Pi

比如 Student 和 Teacher 是多对多关系, 用一个中间表 TeacherStudent 维护。

Gp)i

如果 Student 这边 inverse="true", 那么关系由另一端 Teacher 维护, 就是说当插入 Student 时, 不会操作 TeacherStudent 表 (中间表)。只有 Teacher 插入或删除时才会触发对中间表的操作。 [xH]

所以两边都 inverse="true" 是不对的, 会导致任何操作都不触发对中间表的影响; MFP

当两边都 inverse="false" 或默认时, 会导致在中间表中插入两次关系。

数据库 sql 面试题最全

一是关于怎样找出和去除重复数据, 这在另一个帖子利已有详细介绍。

二是关于找出某一列里最大或最小的前几个, 或是大于或小于某一个值 (最大值或平均值) 的数据。针对这种情况, 再此做一个介绍。

1: 找出公司里收入最高的前三名员工:

```
SQL> select rownum, last_name, salary
```

```
2 from (select last_name, salary
```

```
3      from s_emp
```

```
4      order by salary desc)
```

```
5 where rownum<=3;
```

ROWNUM	LAST_NAME	SALARY
1	Velasquez	4750
2	Ropeburn	2945
3	Nguyen	2897.5

注意: 请大家分析一下一下语句为什么不对:

```
SQL> select rownum, last_name, salary
```

```
2 from s_emp
```

```
3 where rownum<=3
```

4 order by salary desc;

ROWNUM	LAST_NAME	SALARY
1	Velasquez	4750
3	Nagayama	2660
2	Ngao	2000

2: 找出表中的某一行或某几行的数据:

(1): 找出表中第三行数据:

用以下方法是不行的, 因为 rownum 后面至可以用 <或<=号, 不可以用 =, >号和其它的比较符号。

```
SQL> select * from s_emp
2 where rownum=3;
```

no rows selected

```
SQL> select * from s_emp
2 where rownum between 3 and 5;
```

no rows selected

正确的方法如下:

```
SQL> 1
2 select last_name, salary
3 from (select rownum a, b.*
4 from s_emp b)
5* where a=3
SQL> /
```

LAST_NAME	SALARY
Nagayama	2660

(2): 找出第三行到第五行之间的数据:

```
SQL> 1
2 select last_name, salary
3 from (select rownum a, b.*
4 from s_emp b)
5* where a between 3 and 5
```

SQL> /

LAST_NAME	SALARY
Nagayama	2660
Quick-To-See	2755
Ropeburn	2945

3: 找出那些工资高于他们所在部门的平均工资的员工。

(1): 第一种方法:

```
SQL> select last_name, dept_id, salary
2  from s_emp a
3  where salary > (select avg(salary)
4                  from s_emp
5                  where dept_id = a.dept_id);
```

LAST_NAME	DEPT_ID	SALARY
Velasquez	50	4750
Urguhart	41	2280
Menchu	42	2375
Biri	43	2090
Catchpole	44	2470
Havel	45	2483.3
Nguyen	34	2897.5
Maduro	41	2660
Nozaki	42	2280
Schwartz	45	2090

10 rows selected.

(2): 第二种方法:

```
SQL> 1
1  select a.last_name, a.salary, a.dept_id, b.avg_sal
2  from s_emp a, (select dept_id, avg(salary) avg_sal
3                  from s_emp
4                  group by dept_id) b
5  where a.dept_id = b.dept_id
6* and a.salary > b.avg_sal
SQL> /
```

LAST_NAME	SALARY	DEPT_ID	AVGSAL
-----------	--------	---------	--------

Velasquez	4750	50	3847.5
Urguhart	2280	41	2181.5
Menchu	2375	42	2055.16667
Biri	2090	43	1710
Catchpole	2470	44	1995
Havel	2483.3	45	2069.1
Nguyen	2897.5	34	2204
Maduro	2660	41	2181.5
Nozaki	2280	42	2055.16667
Schwartz	2090	45	2069.1

10 rows selected.

4: 找出那些工资高于他们所在部门的 manager 的工资的员工。

SQL> 1

```

1  select id, last_name, salary, manager_id
2  from s_emp a
3  where salary > (select salary
4                  from s_emp
5                  where id = a.manager_id)
SQL> /

```

ID	LAST_NAME	SALARY	MANAGER_ID
6	Urguhart	2280	2
7	Menchu	2375	2
8	Biri	2090	2
9	Catchpole	2470	2
10	Havel	2483.3	2
12	Giljum	2831	3
13	Sedeghi	2878.5	3
14	Nguyen	2897.5	3
15	Dumas	2755	3
16	Maduro	2660	6

10 rows selected.

找出部门工资排名第二，三的员工

```

1  select name, salary, deptno from (
2  select concat(last_name, first_name) name, salary, department_id deptno,
3  rank() over (partition by department_id order by salary desc) rn

```

4* from employees) where rnk=2 or rnk=3

SQL> /

NAME

	SALARY	DEPTNO

FayPat		
	6000	20
KhooAlexander		
	3100	30
BaidaShelli		
	2900	30

NAME

	SALARY	DEPTNO

WeissMatthew		
	8000	50
KauflingPayam		
	7900	50
ErnstBruce		
	6000	60

NAME

	SALARY	DEPTNO

AustinDavid		
	4800	60
PataballaValli		
	4800	60
PartnersKaren		
	13500	80

NAME

SALARY	DEPTNO	

ErrazurizAlberto		
12000	80	
KochharNeena		
17000	90	
De HaanLex		
17000	90	

NAME

SALARY	DEPTNO	

FavietDaniel		
9000	100	
ChenJohn		
8200	100	
GietzWilliam		
8300	110	

15 rows selected.

SQL>

找出部门工资排名第二，三的员工

```
1 select name,salary,deptno from (
2 select concat(last_name,first_name) name,salary,department_id deptno,
3 rank() over (partition by department_id order by salary desc) rnk
4* from employees) where rnk=2 or rnk=3
SQL> /
```

NAME

SALARY	DEPTNO	

FayPat
6000 20

KhooAlexander
3100 30

BaidaShelli
2900 30

NAME

SALARY DEPTNO

WeissMatthew
8000 50

KauflingPayam
7900 50

ErnstBruce
6000 60

NAME

SALARY DEPTNO

AustinDavid
4800 60

PataballaValli
4800 60

PartnersKaren
13500 80

NAME

SALARY DEPTNO

ErrazurizAlberto

12000	80
-------	----

KochharNeena

17000	90
-------	----

De HaanLex

17000	90
-------	----

NAME

SALARY	DEPTNO
--------	--------

FavietDaniel

9000	100
------	-----

ChenJohn

8200	100
------	-----

GietzWilliam

8300	110
------	-----

15 rows selected.

SQL>

又是一道面试题:

原表:

id	proid	praname
----	-------	---------

1	1	M
---	---	---

1	2	F
---	---	---

2	1	N
---	---	---

2	2	G
---	---	---

3	1	B
---	---	---

3	2	A
---	---	---

查询后的表:

id	pro1	pro2
----	------	------

1	M	F
---	---	---

2	N	G
---	---	---

3	B	A
---	---	---

写出查询语句

又是一道面试题:

原表:

id	proid	praname
----	-------	---------

1	1	M
---	---	---

1	2	F
---	---	---

2	1	N
---	---	---

2	2	G
---	---	---

3	1	B
---	---	---

3	2	A
---	---	---

查询后的表:

id	pro1	pro2
----	------	------

1	M	F
---	---	---

2	N	G
---	---	---

3	B	A
---	---	---

写出查询语句

又是一道面试题:

原表:

id	proid	praname
----	-------	---------

1	1	M
---	---	---

1	2	F
---	---	---

2	1	N
---	---	---

2	2	G
---	---	---

3	1	B
---	---	---

3	2	A
---	---	---

查询后的表:

id	pro1	pro2
----	------	------

1	M	F
---	---	---

2	N	G
---	---	---

3	B	A
---	---	---

写出查询语句

又是一道面试题:

原表:

id proid pronaime

1 1 M

1 2 F

2 1 N

2 2 G

3 1 B

3 2 A

查询后的表:

id pro1 pro2

1 M F

2 N G

3 B A

写出查询语句

解决方案可有以下三种作参考:

1: 使用 pl/sql 代码实现, 但要求你组合后的长度不能超出 oracle varchar2 长度的限制。

下面是一个例子

```
create or replace type strings_table is table of varchar2(20);
```

```
/
```

```
create or replace function merge (pv in strings_table) return varchar2
```

```
is
```

```
ls varchar2(4000);
```

```
begin
```

```
for i in 1..pv.count loop
```

```
    ls := ls || pv(i);
```

```
end loop;
```

```
return ls;
```

```
end;
```

```
/
```

```
create table t (id number,name varchar2(10));
```

```
insert into t values(1,'Joan');
```

```
insert into t values(1,'Jack');
```

```
insert into t values(1,'Tom');
```

```
insert into t values(2,'Rose');
```

```
insert into t values(2,'Jenny');
```

```
column names format a80;
```

```
select t0.id,merge(cast(multiset(select name from t where t.id = t0.id) as strings_table)) names
```

```
from (select distinct id from t) t0;
```

```
drop type strings_table;
```

```
drop function merge;
drop table t;
```

2: 用 sql:

Well if you have a theoretical maximum, which I would assume you would given the legibility of listing hundreds of employees in the way you describe then yes. But the SQL needs to use the LAG function for each employee, hence a hundred emps a hundred LAGs, so kind of bulky.

This example uses a max of 6, and would need more cut n pasting to do more than that.

```
SQL> select deptno, dname, emps
2 from (
3 select d.deptno, d.dname, rtrim(e.ename ||', '||
4 lead(e.ename,1) over (partition by d.deptno
5 order by e.ename) ||', '||
6 lead(e.ename,2) over (partition by d.deptno
7 order by e.ename) ||', '||
8 lead(e.ename,3) over (partition by d.deptno
9 order by e.ename) ||', '||
10 lead(e.ename,4) over (partition by d.deptno
11 order by e.ename) ||', '||
12 lead(e.ename,5) over (partition by d.deptno
13 order by e.ename)', ' ') emps,
14 row_number () over (partition by d.deptno
15 order by e.ename) x
16 from emp e, dept d
17 where d.deptno = e.deptno
18 )
19 where x = 1
20 /
```

DEPTNO DNAME EMPS

```
-----
10 ACCOUNTING CLARK, KING, MILLER
20 RESEARCH ADAMS, FORD, JONES, ROONEY, SCOTT, SMITH
30 SALES ALLEN, BLAKE, JAMES, MARTIN, TURNER, WARD
```

3: 先用 pl/sql 创建一个函数 (create function get_a2) ;
create or replace function get_a2(tmp_a1 number)

```

return varchar2
is
Col_a2 varchar2(4000);
begin
Col_a2:="";
for cur in (select a2 from unite_a where a1=tmp_a1)
loop
Col_a2=Col_a2||cur.a2;
end loop;
return Col_a2;
end get_a2;

```

```

select distinct a1 ,get_a2(a1) from unite_a
1 ABC
2 EFG
3 KMN
----6.4.2008

```

Clone 和 new 哪个更快

Clone 和 new 哪个更快呢，这个问题的答案不是一定的，要根据实际情况决定：

情况一：对象简单，这个时候 new 更快，测试代码如下：

```

class TestObj implements Cloneable{
    public Object clone(){
        Object obj = null ;
        try{
            obj = super.clone();
        }catch(Exception e){
        }
        return obj;
    }
}

public class CloneVsNew {
    static void cloneTest(int time){
        TestObj obj=new TestObj();
        for(int i=0;i<time;i++){
            obj.clone();
        }
    }
}

```

```

static void newTest(int time) {
    TestObj obj=new TestObj();
    for(int i=0;i<time;i++){
        obj=new TestObj();
    }
}
/** **/**
 * @param args
 */
public static void main(String[] args) {
    long start;
    long stop;
    int times=1000000;

    System.gc();

    start=System.currentTimeMillis();
    newTest(times);
    stop=System.currentTimeMillis();
    System.out.println("newTest Time:"+ (stop-start));

    System.gc();

    start=System.currentTimeMillis();
    cloneTest(times);
    stop=System.currentTimeMillis();
    System.out.println("cloneTest Time:"+ (stop-start));

}
}

```

情况二：对象复杂，例如一个包括集合类的类的对象。而且这个对象的 Clone 使用的浅拷贝。（其实快主要是快在这个地方）

不用例子了，浅拷贝只是引用的复制，肯定比复制快。

还有一些其它的情况，但总体来说，随着对象的复杂，clone 越来越快，new 越来越慢。不过在使用 clone 的时候

一定要想清楚再用，浅拷贝使用不当会出现很多问题。

经常在面试的时候,有很多公司问到 hibernate 的事务管理和延时加载问题.

hibernate 的事务管理是说共同操作数据库数据和需要对数据库的原子操作的概念,
延迟加载是一个类中有另一个类属性(他们不在一张表中)没有 join fetch 情况下他不会自动得到
如果你访问的时候他才会去取 而在 session 关闭了 他就无法得到了 即延迟加载,所以你如果要
这数据的话可以在 Hql 中加 join fetch 或是在 session 关闭之前取.

6.5

ASP | PHP | JSP | .NET | 比较

ASP,PHP,JSP,ASP.NET 这是当前比较流行的四种 WEB 网站编程语言,现在做网站大部分都是使用这几种语言中的其一。

ASP 是基于 WINDOWS 平台的,简单易用,由于是运行在 WINDOWS 平台上,所以移植性不好,不能跨平台运行.但是国内现在大部分的网站都是使用它来开发的.

PHP 是当前兴起备受推崇的一种 WEB 编程语言,开源且跨平台,在欧美都比较的流行,最近在国内也很受许多网站开发者的欢迎.开发效率高,成本低!

JSP 是了 SUN 公司推出的一种网络编程语言,比较难学(是网上公认的),跨平台运行.安全性比较高,运行效率也比较的快.备受企业级的公司及银行金融机构使用.

ASP.NET 从某种意义上说应该是 ASP 版本的升级,但是它又不完全是从 ASP 上升级来的,ASP.NET 的推出是微软为了与 SUN 公司的 JSP 在网络上的迅猛发展抵抗的,.NET 架构现在有许多人都在学习,主要是开发的效率比较高.但是还是基于 WIN 平台运行,不能跨平台.

第 I 条 ASP 简介

ASP, 英文全称为 Active Server Pages。

ASP 是一种动态网页, 文件后缀名为 .asp。

ASP 网页是包含有服务器端脚本 (server-side script) 的 HTML 网页。Web 服务器会处理这些脚本, 将其转换成 HTML 格式, 再传到客户的浏览器端。

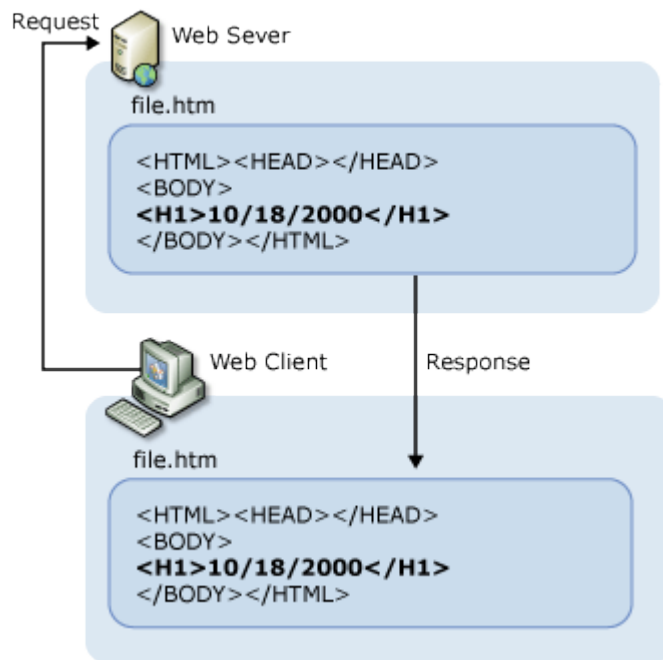
下面的图示, 你可以很直观地理解 HTML 和 ASP 的区别。

第 II 条 HTML 和 ASP 的区别

HTML 是编写网页的最简单的语言。HTML 只能用于编写静态的网页。

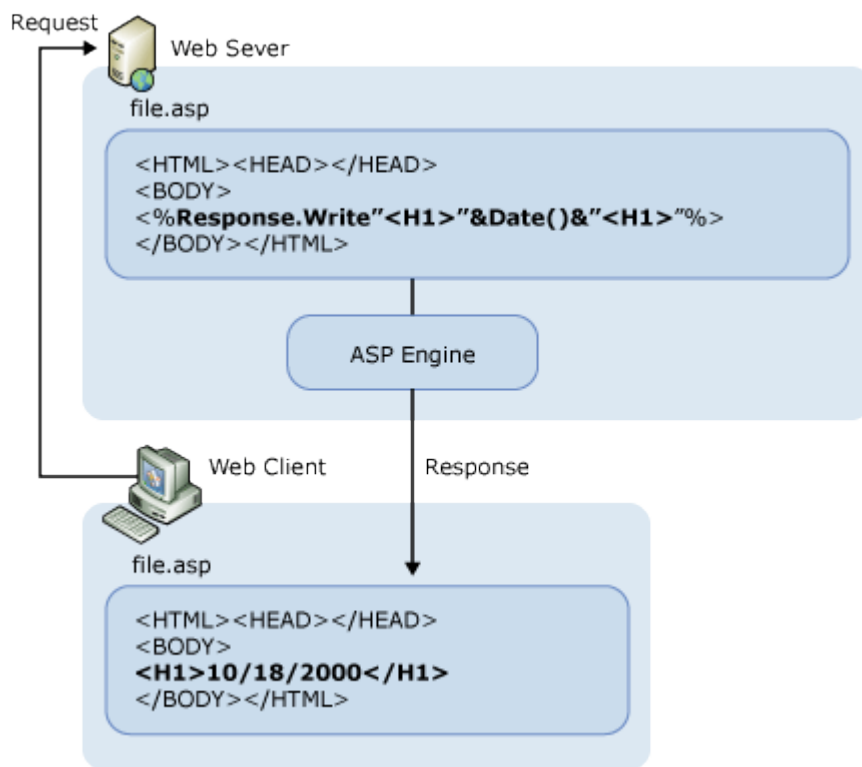
当一个用户浏览器(下图所示: Web Client) 从 Web 服务器(下图所示: Web Server) 要求一个 HTML 网页时, Web 服务器就将这个网页直接发送给用户浏览器, 不经过计算处理。然后用户浏览器会处理该网页的 HTML 代码, 然后将结果显示出来。

见下图所示:



而 ASP 的处理过程则更复杂些。当一个用户浏览器(下图所示: Web Client) 从 Web 服务器(下图所示: Web Server) 要求一个 ASP 网页时, Web 服务器会将这个 ASP 文件发送给 Web 服务器的 ASP 引擎(下图所示: ASP Engine), ASP 引擎则将该 ASP 网页中所有的服务器端脚本(下图所示: <%和%>之间的代码)转换成 HTML 代码, 然后将所有 HTML 代码发送给用户浏览器。

见下图所示:



第 III 条 ASP 能干什么？

网页最合适的脚本语言。有关 VBScript。

用 HTML，CSS 已经能够编写非常漂亮的静态网页，但是这些网页缺乏和用户的互动性。

比如，你需要为你的网站增加用户注册和登录的功能；你可能还需要做一些网上调查，了解用户对于某个事件或者产品的反馈；你还可能需要一个电子商务网站，及时发布产品信息和满足用户在线订购的需要.....

和 ASP 的区别。HTML 代码，然后将结果显示出来。英文全称为！

编写 ASP 动态网页能够帮助你实现这些和用户互动的功能。

第 IV 条 一个简单的 ASP 示例

```
<html>
```

```
<head >
```

```
<title>ASP 代码：ASP 输出语法</title>
```

```
</head>
```

```
<body>

<p ><% = "布啦布啦 www .blabla .cn" %></p>

</body>

</html>
```

及时发布产品信息和满足用户在线订购的需要 ...?

是一种动态网页，文件后缀名为 .asp?

上面这个 ASP 网页代码示例中，你可以看到，这个 ASP 文件和一个普通的 HTML 网页基本一样，除了里面有<%= "布啦布啦 www.blabla.cn" %>这段代码。

<% %>表示在里面的代码是 ASP 代码。

<%= 表示需要输出 ASP 代码的结果。

PHP 介绍

如果您从事基于 Web 的开发工作，那么可能已经听说过 PHP。您也许不太确切地知道 PHP 是什么、如何工作或者为什么如此热门，但现在该是进一步了解 PHP 的时候了。因此本文简要介绍了关于 PHP 基础的基本概念。

一点背景知识

PHP 是作为一个小开放源码，随着越来越多的人意识到它的实用性从而逐渐发展起来。Rasmus Lerdorf 在 1994 年发布了 PHP 的第一个版本。从那时起它就飞速发展，并在原始发行版上经过无数的改进和完善现在已经发展到版本 4.0.3。

PHP 是一种嵌入在 HTML 并由服务器解释的脚本语言。它可以用于管理动态内容、支持数据库、处理会话跟踪，甚至构建整个电子商务站点。它支持许多流行的数据库，包括 MySQL、PostgreSQL、Oracle、Sybase、Informix 和 Microsoft SQL Server。

动态与静态内容

动态内容为什么这么热门？假设您正在管理有 10 个产品的电子商务站点。只要产品不是经常变动或者预料到它不会有太大的变动，那么手工编写 10 个带有必要的信息、表单和诸如此类内容的静态产品页面是不困难的。但是，假设您在本月再要添加 10 个或更多产品，然后在下个月要更多，而且价格有时会变动或者想改变站点的观感。那么您就会陷入用手工重新编写数十个，也许上百个静态页面的困境中。

另一方面，假设您从创建 `product.php` 页面开始。它没有静态信息，而是编码成可以从产品数据库中提取信息并动态地构建一个页面。然后您就拥有了一个元数据页面，它可以根据存储在数据库中的信息 提供一个、一百个、甚至十万个单独页面。现在网站管理员不再整天都简单重复更新静态页面的工作，因为在更新公司数据库中的信息同时就可以更新页面上的信息。这样就消除了令人头疼的时间延迟（在数据库中更改信息和在网站上显示信息之间的时间间隔）。

总体来说,PHP 非常适合 Web 上的工作。但它并不是唯一的方法; 如 Perl、Java、JavaScript、ASP、Python、Tcl、CGI 以及其它许多方法都可以生成动态的内容。但是，PHP 的优点是：它是专为基于 Web 的问题而设计的以及它是开放源码。

如果您正在为文字处理或 3D 游戏寻找程序设计语言，那 PHP 可能不是您需要的语言。如果您需要运行一个具有动态内容、数据库交互和电子贸易的网站，那么就请继续读下去，因为 PHP 在这方面确实是非常有用的。

PHP 适用的平台

大多数常规 PHP 的安装通常是与 Linux 或各种 UNIX 上的 Apache 一起运行的 PHP 模块。但是如果正在使用其它平台，不要担心。PHP 可以在 Windows NT 和 9x 以及其它许多 Web 服务器上运行。可以在[主要介绍 Apache/Linux/PHP 组合的一些网站上找到更多有关 PHP 的文档](#)，但它并不是支持 PHP 的唯一平台。

许可证和使用

购买具有全部功能的嵌入式 Web 脚本语言要花多少钱？一分钱不花？PHP 是一个开放源码项目，所以没有购买许可证的费用或限制使用的问题。您可以使用 PHP 来运行小的、非赢利性站点，或者运行十亿美元的电子商务网站，而且成本是一样的：零。不仅如此，如果想要或需要修改 PHP，可以修改它。

PHP 并没有得到 GPL 的许可，但它自己的许可证允许重新分发代码和 / 或二进制文件。

使用 PHP

好，现在您已经确信要真正地尝试一下 PHP 了吧。[我们先看一些简单的例子](#)，这样您对 PHP 就有个大概了解。记住决这不是深入了解 PHP 的途径，仅仅是个快速入门而已。

"Hello, World!"

为了对 PHP 有个了解，让我们来看一下几个非常简单的 PHP 脚本。既然 "Hello, World!" 是个常用的示例，那我们就编写一个友好的小 "Hello, World!" 脚本。

如早些时候所提到的，PHP 是嵌入在 HTML 中的。（可能您的文件几乎没有包含 HTML，但是通常这个文件是 PHP 与 HTML 的混合体。）这意味着在您正常的 HTML 中（或 XHTML，如果您处在比较前沿的位置），会有类似这样的 PHP 语句：

```
<body bgcolor="white">
    <strong>How to say "Hello, World!"</strong>
    <?php echo "Hello, World!";?>
    <br>
    Simple, huh?
</body>
```

很简单，不是吗？这仅仅是一个“echo”语句，就这样。当然，仅仅这样是没有多大用处的。但是它确实告诉我们关于语言的一些东西。（顺便说一下，如果检查 HTML 输出，就会注意到 PHP 的代码并没有出现在从服务器送到您 Web 浏览器的文件中。所有出现在 Web 页面中的 PHP 都会被处理并从页面中剥离；从 WEB 服务器返回给客户机的仅仅是纯 HTML 输出。）

在 Web 页面上打印日期和时间

现在我们做一些稍微实用的事情。这个示例将在 Web 页面上打印日期和时间。

```
<body bgcolor="white">
    <strong>An Example of PHP in Action</strong>
    <?php echo "The Current Date and Time is:<br>";
    echo date("g:i A l, F j Y.");?>
    // g = the hour, in 12-hour format
    // i = minutes
    // A = print AM or PM, depending...
    // l = print the day of the week
    // F = print the month
    // j = print the day of the month
    // Y = print the year - all four digits
```

此代码生成以下输出：

```
The Current Date and Time is:
11:00 AM Friday, October 20 2000.
```

请注意，这里揉和了 PHP 和 HTML。假设您已经了解 HTML，所以这里仅解释 PHP 代码。在 PHP.net （请参阅[参考资料](#)）上可以找到完整的 PHP 参考。

PHP 代码是以标记 <?php 开始并以 ?> 结束的。这就告诉服务器在 <?php 和 ?> 之间的所有内容需要用 PHP 指令进行语法分析，如果发现它们，就需要执行它们。请注意，当处理和服务您的文件时，客户机会收到普通的 HTML 文件。

浏览您站点的人看不到任何一个的 PHP 指令，除非您犯了错误，服务器把这些 PHP 代码分割开而没有先处理它们。

一般情况下会处理在 `<?php` 和 `?>` 之间的常规 HTML 标记。请注意上面这个简单的脚本中包含了 `
` 这个分行标记。如果不能很好地利用 HTML 格式，那么 PHP 就不会非常有用。

如果想和其它东西一起使用，或者您和我一样是一个健忘的人，您可能会想到要注释代码。`//` 字符表明是注释，服务器不会处理被 `//` 标记的内容，也不会象 HTML 中的注释一样，把内容传给客户机。如果在 `<?php` 和 `?>` 标记之间有标准的 `<!-- comment -->`，那么当服务器对它进行语法分析时，有可能引起错误。显然，您可能不会象我对这个基本功能一样注释您的代码，但它确是一个很好的示例。

最后要注意的是，每个 PHP 函数都封闭在圆括号内，并以分号结束，这和 C 或 Perl 相似。由于一个简单的印刷错误而遗漏一个结束的圆括号或分号，造成一些语法错误是很常见的，所以要确保检查代码。在象 Vim 或 Emacs 这样的编辑器（可以突出显示语法）中编写 PHP 是有助于消除此类错误。它使您能立刻捕捉到错误。

`date` 函数仅是内置 PHP 函数之一。PHP 附带了许多功能可以用于数据库连接、创建 PDF、Shockwave、JPG、GIF、PNG 和其它图象文件、发送电子邮件、阅读和书写文件、语法分析 XML、会话处理、经由 HTTP 直接与浏览器对话，以及许多其它功能。

PHP 也允许用户定义自己的函数。这使 PHP 语言能够经由 Web 提供大量的解决方案。而不是从一开始就把所有的事情都编写好了。在您编写函数之前，确保已经查看诸如 Zend.com、PHP Wizard 当然还有 Freshmeat，来看是否已经有您正在尝试编写的函数（请参阅[参考资料](#)）。

对于提供标题、自动更新新闻网站、基于 Web 的电子邮件客户机、数据库管理和其它方面，已经有许多开放源码的 PHP 解决方案。再从头来做这些是毫无意义的。相反，应该从已经构建好的基础开始，并把它定制到您自己的解决方案中。如果您仅仅是初步了解和学习 PHP，在心中并没有具体的项目，那么这些项目对于使用 PHP 仍然是很好的示例和学习资料。

JSP 简介

[jsp](#) (JavaServer Pages) 是由 Sun Microsystems 公司倡导、许多公司参与一起建立的一种动态网页技术标准，其网址为 <http://www.javasoft.com/prodUcts/jsp>。该技术为创建显示动态生成内容的 Web 页面提供了一个简捷而快速的方法。JSP 技术的设计目的是使得构造基于 Web 的应用程序更加容易和快捷，而这些应用程序能够与各种 Web [服务器](#)，应用服

务器，[浏览器](#)和开发工具共同工作。JSP 规范是 Web 服务器、应用服务器、交易系统、以及开发工具供应商间广泛合作的结果。在传统的网页 [Html](#) 文件 (*.htm, *.html) 中加入 Java 程序片段 (Scriptlet) 和 JSP 标记 (tag)，就构成了 JSP 网页 (*.jsp)。Web 服务器在遇到访问 JSP 网页的请求时，首先执行其中的程序片段，然后将执行结果以 HTML 格式返回给客户。程序片段可以操作数据库、重新定向网页以及发送 email 等等，这就是建立动态网站所需要的功能。所有程序操作都在服务器端执行，网络上传送给客户端的仅是得到的结果，对客户浏览器的要求最低，可以实现无 Plugin，无 ActiveX，无 Java Applet，甚至无 Frame。

JSP 技术在多个方面加速了动态 Web 页面的开发：

一．将内容的生成和显示进行分离

使用 JSP 技术，Web 页面开发人员可以使用 HTML 或者 [XML](#) 标识来设计和格式化最终页面。使用 JSP 标识或者小脚本来生成页面上的动态内容（内容是根据请求来变化的，例如请求帐户信息或者特定的一瓶酒的价格）。生成内容的逻辑被封装在标识和 JavaBeans 组件中，并且捆绑在小脚本中，所有的脚本在服务器端运行。如果核心逻辑被封装在标识和 Beans 中，那么其他人，如 Web 管理人员和页面设计者，能够编辑和使用 JSP 页面，而不影响内容的生成。在服务器端，JSP 引擎解释 JSP 标识和小脚本，生成所请求的内容（例如，通过访问 JavaBeans 组件，使用 JDBC 技术访问数据库，或者包含文件），并且将结果以 HTML（或者 XML）页面的形式发送回浏览器。这有助于作者保护自己的代码，而又保证任何基于 HTML 的 Web 浏览器的完全可用性。

二．强调可重用的组件

绝大多数 JSP 页面依赖于可重用的，跨平台的组件（JavaBeans 或者 Enterprise JavaBeans™ 组件）来执行应用程序所要求的更为复杂的处理。开发人员能够共享和交换执行普通操作的组件，或者使得这些组件为更多的使用者或者客户团体所使用。基于组件的方法加速了总体开发过程，并且使得各种组织在他们现有的技能和优化结果的开发努力中得到平衡。

三．采用标识简化页面开发

Web 页面开发人员不会都是熟悉脚本语言的[编程](#)人员。JavaServer Page 技术封装了许多功能，这些功能是在易用的、与 JSP 相关的 XML 标识中进行动态内容生成所需要的。标准的 JSP 标识能够访问和实例化 JavaBeans 组件，设置或者检索组件属性，[下载](#) Applet，以及执行用其他方法更难于编码和耗时的功能。

通过开发定制化标识库，JSP 技术是可以扩展的。今后，第三方开发人员和其他人员可以为常用功能创建自己的标识库。这使得 Web 页面开发人员能够使用熟悉的工具和如同标识一样的执行特定功能的构件来工作。

JSP 技术很容易整合到多种应用体系结构中，以利用现存的工具和技巧，并且扩展到能够支持企业级的分布式应用。作为采用 Java 技术家族的一部分，以及 Java 2（企业版体系结构）的一个组成部分，JSP 技术能够支持高度复杂的基于 Web 的应用。

由于 JSP 页面的内置脚本语言是基于 Java 编程语言的，而且所有的 JSP 页面都被编译成为 Java Servlet，JSP 页面就具有 Java 技术的所有好处，包括健壮的[存储](#)管理和[安全](#)性。作为 Java 平台的一部分，JSP 拥有 Java 编程语言“一次编写，各处运行”的特点。随着越来越多的供应商将 JSP 支持添加到他们的产品中，您可以使用自己所选择的服务器和工具，更改工具或服务器并不影响当前的应用。当与 Java 2 平台，企业版(J2EE)和 Enterprise JavaBean 技术整合时，JSP 页面将提供企业级的扩展性和性能，这对于在虚拟企业中部署基于 Web 的应用是必需的。

四. 技术分析

Microsoft 公司的 [ASP](#) 技术也是动态网页开发技术。JSP 和 ASP 从形式上非常相似，ASP 程序员一眼就能认出<% %>以及<%= %>。但是深入探究下去会发现它们很多的差别，其中最主要的有以下三点：

1、 JSP 的效率和安全性更高

ASP 以源码形式存放，以解释方式运行，每次 ASP 网页调用都需要对源码进行解释，运行效率不高。另外，IIS 的漏洞曾使得许多网站源程序大曝光，包括笔者以前用 ASP 开发的网站，ASP 程序全部被人下载了去。JSP 在执行以前先被编译成字节码 (byte code)，字节码由 Java 虚拟机 (Java Virtual Machine) 解释执行，比源码解释的效率高；服务器上还有字节码的 Cache 机制，能提高字节码的访问效率。第一次调用 JSP 网页可能稍慢，因为它被编译成 Cache，以后就快得多了。同时，JSP 源程序不大可能被下载，特别是 JavaBean 程序完全可以放到不对外的目录中。

2、 JSP 的组件 (Component) 方式更方便

ASP 通过 COM 来扩充复杂的功能，如文件上载、发送 email 以及将业务处理或者复杂计算分离出来成为独立可重复利用的模块。JSP 通过 JavaBean 实现了同样的功能扩充。在开发方面，COM 的开发远比 JavaBean 复杂和繁琐，学会 ASP 不难，但学会开发 COM 可不简单。而 JavaBean 就简单多了，从本文上述示例中可以看出开发 JavaBean 很方便。在维护方面，COM 必须在服务器上注册，如果[修改](#)了 COM 程序，就必须重新注册，甚至必须关机和重新启动。JavaBean 则不需要注册，放在 CLASSPATH 包含的目录中就行了。如果 JavaBean 进行了修改，则 JSDK 和 Tomcat 现在还需要关闭和重新运行（但不是关机），但开发者已经许诺将在以后的版本中做到不需要关闭服务器。另外 JavaBean 是完全的 OOP，可以针对不同的业务处理功能方便地建立一整套可重复利用的对象库，例如用户权限控制、email 自动回复等等。

3、 JSP 的适应平台更广

ASP 目前仅适用于 NT 和 IIS。虽然 Unix 下有 ChiliSoft 的插件来支持 ASP，但是 ASP 本身的功能有限，必须通过 ASP+COM 的组合来扩充，Unix 下的 COM 实现起来非常困难。

ASP. NET

1.1 什么是 ASP.net

ASP.net 是一种建立在通用语言上的程序构架，能被用于一台 Web 服务器来建立强大的 Web 应用程序。ASP.net 提供许多比现在的 Web 开发模式强大的优势。

执行效率的大幅提高

ASP.net 是把基于通用语言的程序在服务器上运行。不像以前的 ASP 即时解释程序，而是将程序在服务器端首次运行时进行编译，这样的执行效果，当然比一条一条的解释强很多。

世界级的工具支持

ASP.net 构架是可以用 Microsoft (R) 公司最新的产品 Visual Studio.net 开发环境进行开发，WYSIWYG (What You See Is What You Get 所见即为所得) 的编辑。这些仅是 ASP.net 强大化软件支持的一小部分。

强大性和适应性

因为 ASP.net 是基于通用语言的编译运行的程序，所以它的强大性和适应性，可以使它运行在 Web 应用软件开发者的几乎全部的平台（笔者到现在为止只知道它只能用在 Windows 2000 Server 上）。通用语言的基本库，消息机制，数据接口的处理都能无缝的整合到 ASP.net 的 Web 应用中。ASP.net 同时也是 language-independent 语言独立化的，所以，你可以选择一种最适合你的语言来编写你的程序，或者把你的程序用很多种语言来写，现在已经支持的有 C# (C++ 和 Java 的结合体)，VB, Jscript。将来，这样的多种程序语言协同工作的能力保护您现在的基于 COM+ 开发的程序，能够完整的移植向 ASP.net。

简单性和易学性

ASP.net 是运行一些很平常的任务如表单的提交客户端的身份验证、分布系统和网站配置变得非常简单。例如 ASP.net 页面构架允许你建立你自己的用户分界面，使其不同于常见的 VB-Like 界面。另外，通用语言简化开发使把代码结合成软件简单的就像装配电脑。

高效可管理性

ASP.net 使用一种字符基础的，分级的配置系统，使你服务器环境和应用程序的设置更加简单。因为配置信息都保存在简单文本中，新的设置有可能都不需要启动本地的管理工具就可以实现。这种被称为“Zero Local Administration”的哲学观念使 Asp.net 的基于应用的开发更加具体，和快捷。一个 ASP.net 的应用程序在一台服务器系统的安装只需要简单的拷贝一些必须得文件，不需要系统的重新启动，一切就是这么简单。

多处理器环境的可靠性

ASP.net 已经被刻意设计成为一种可以用于多处理器的开发工具，它在多处理器的环境下用特殊的无缝连接技术，将很大的提高运行速度。即使你现在的 ASP.net 应用软件是为一个处理器开发的，将来多处理器运行时不需要任何改变都能提高他们的效能，但现在的 ASP 确做不到这一点。

自定义性，和可扩展性

ASP.net 设计时考虑了让网站开发人员可以在自己的代码中自己定义“plug-in”的模块。这与原来的包含关系不同，ASP.net 可以加入自己定义的任何组件。网站程序的开发从来没有这么简单过。

安全性

基于 Windows 认证技术和每应用程序配置，你可以确性你的原程序时绝对安全的。（但是 Asp 的经验告诉我们，不能这么自信，M\$总是会出 Bug 的）

1.2 本篇教程的主要内容

ASP.net 的语法

ASP.net 的语法有可能对老的 ASP 程序员来说很熟悉，但也有一些是全新的东西，QuickStart 教程将在细节上全部谈到它们。

ASP.net 的结构和特征

介绍 ASP.net 的所有特性，是 ASP 开发者可以编写交互性的世界级的应用软件，用比以前少的多时间和精力。

最佳话的练习

引用了一些练习题和如何回避在程序开发中潜在地容易犯的错误的方法，是开发更加顺利。

为那些人编写的

如果你从来没有开发过网站程序，那么这不适合你，你应该至少掌握一些 HTML 语言和简单的 Web 开发术语。你不需要先前的 ASP 开发经验（当然有经验更好），但是你必须了解交互式 Web 程序开发的概念，包含窗体，脚本，和数据接口的概念。

1.3 安装 ASP.net

跟基督山一起检查你们的计算机哦

CPU Pentium II 450 以上，推荐 733

内存 256M 推荐 512M

硬盘空间 1G 或者更多

操作系统 Windows 2000 Pro Server AD Server 都可以，并安装 IIS 5 和 Windows 2000 SP1

但基督山的机器配置 C266 超 333 内存 128M 硬盘 2G 还是安装成功了。

检查一下大家是不是都用 IE 5.5 了？

我们还需要 MDAC (ADO) 2.6 以上版本（可以在微软网站下面下载。）

最后，就是我们讲座的主角 Microsoft .Net Framework SDK 诶，在 <http://msdn.microsoft.com/net> 下可以下载，绝对地址在 <http://download.microsoft.com/download/VisualStudioNET/Install/2204/N15/EN-US/setup.exe> 111M，还好带宽不小 ADSL 的朋友可以达到全速（如果 down 不了的朋友，基督山可以帮着刻一张光盘，包括 MDAC i IE5.5 SP1，.net SDK。）

注意：教育网络上的朋友么，呵呵，在北京科技大学的服务器上面，有完整的文件可以下载，速度还不错，地址是 <ftp://bbs.ustb.edu.cn/incoming/newsoft>

终于 down 完了，运行安装程序，.net 安装到你的计算机里面去。

1.4 安装 .NET Framework 的快速入门示例

打开 SDK Overview 链接 (.NET Framework SDK Overview)，该链接已在 SDK 安装过程中添加至您的桌面。

HTML 页面打开后，单击 .NET Framework Samples 链接。

按照安装步骤来安装示例。

安装完成后，您可以在 <http://localhost/quickstart/default.htm> 访问示例。

最后，在浏览默认文档时，您可以选择页面顶端的 ASP.NET 链接。此链接包含了 ASP.NET 示例的列表。这个教程是目前国际上面唯一的完全教程，也是 Microsoft 提供给我们学习 ASP.net 的唯一教程，可惜是全英文的，ASP.net 的先行者，全靠它了。

1.5 学习 ASP.net 的过程中如何求助——加入 ASPNG 讨论列表

Charles Carroll 作为不断壮大的 ASP.NET 社团的领导者之一，一直尽心尽力地主持着基于电子邮件的 ASP.NET 讨论列表。事实上，从七月初专业开发人员会议（PDC）的第一天开始，这个讨论列表就一直存在。

此讨论列表是学习和发布日益丰富的 ASP.NET 开发人员知识的最佳选择。Charles 会检查和筛选所有的消息，确保发布的文章都与此列表相关，从而使讨论不脱离主题。

除了社区方面的列表以外，Microsoft .NET Framework 小组还定期参加讨论并发布示例代码。最近一次是 Scott Guthrie 发布了一个动态生成图像的 ASP.NET 示例。

要加入讨论列表，只需访问 <http://www.asplists.com/asplists/aspng.asp>（英文）。在此页面上，您可以找到与 ASP.NET 相关的不同主题的电子邮件讨论列表的完整列表。最活跃的列表是 ASPNGBeta，我们推荐您加入这个列表。请从列表中选择 ASPNGBeta。您也可以直接访问 <http://www.asplists.com/asplists/aspngbeta.asp>（英文），选择所需的接收电子邮件分发的选项。

1.6 完整的.NET 解决方案演示

IBuySpy Microsoft ASP.NET 示例应用程序

IBuySpy（英文）是功能全面的电子商务应用程序，它详细介绍了如何使用 ASP.NET 创建高性能、可伸缩的 Web 应用程序。您可以联机浏览示例应用程序（包括完整的源代码和文档），也可以下载完整的 C# 或 Visual Basic.NET 版本的应用程序。

<http://www.ibuyspy.com/>

1.7 免费的.net 空间

Brinkster Inc.（英文）是最大的免费 ASP Web 主机服务提供商，它提供免费 ASP.NET 主机服务。加入的每个成员可获得 5 MB 空间用于测试和学习 ASP.NET。用于测试和学习的成员身份，给开发人员提供了了解这项新技术的机会。Brinkster 提供免费 ASP 主机服务已经有一年多的时间，在 ASP/ASP.NET 主机服务市场处于领先地位。

<http://www.brinkster.com/>（速度不是很理想）

1.8 国外相关的技术支持网站

<http://www.aspfree.com> (英文) 是比较全面的 ASP.NET 社区站点之一, 由 Steve Schofield 管理。您可以在这里找到十多篇关于 ASP.NET 的文章, 以及联机的 ASP.NET 快速入门示例。如果您不能安装 NET. SDK, 这是您学习 ASP.NET 的不错的选择。作为奖励, 您还可以得到 Wrox Press 出版的 A Preview of ASP.NET 第 1 章的联机版本。

<http://www.aspnet.com> (英文) 是另外一个比较全面的 ASP.NET 相关站点。ASPNET 包括至以下内容的链接: Charles Carroll 上载的 ASP.NET 类和 Stephen Walther 上载的 ASP.NET Workshop。最后, ASPNET 还包括一个比较活跃的电子邮件讨论列表。相关信息请参见 <http://www.asplists.com> (英文)。

<http://www.ASP101.com> (英文) 几篇 ASP.NET 相关文章, 以及按线索组织的、基于 Web 讨论的 ASP.NET [论坛](#)。

<http://www.asptoday.com> (英文) Wrox Press 的联机杂志已经有 10 篇关于 ASP.NET 的文章, 范围从 ADO+ 到生成服务器控件。

<http://www.4guysfromrolla.com> (英文) 4GuysFromRolla 包括一些文章和其他相关资源。文章内容涉及 ADO+ 和 ASP.NET 本身的概述。

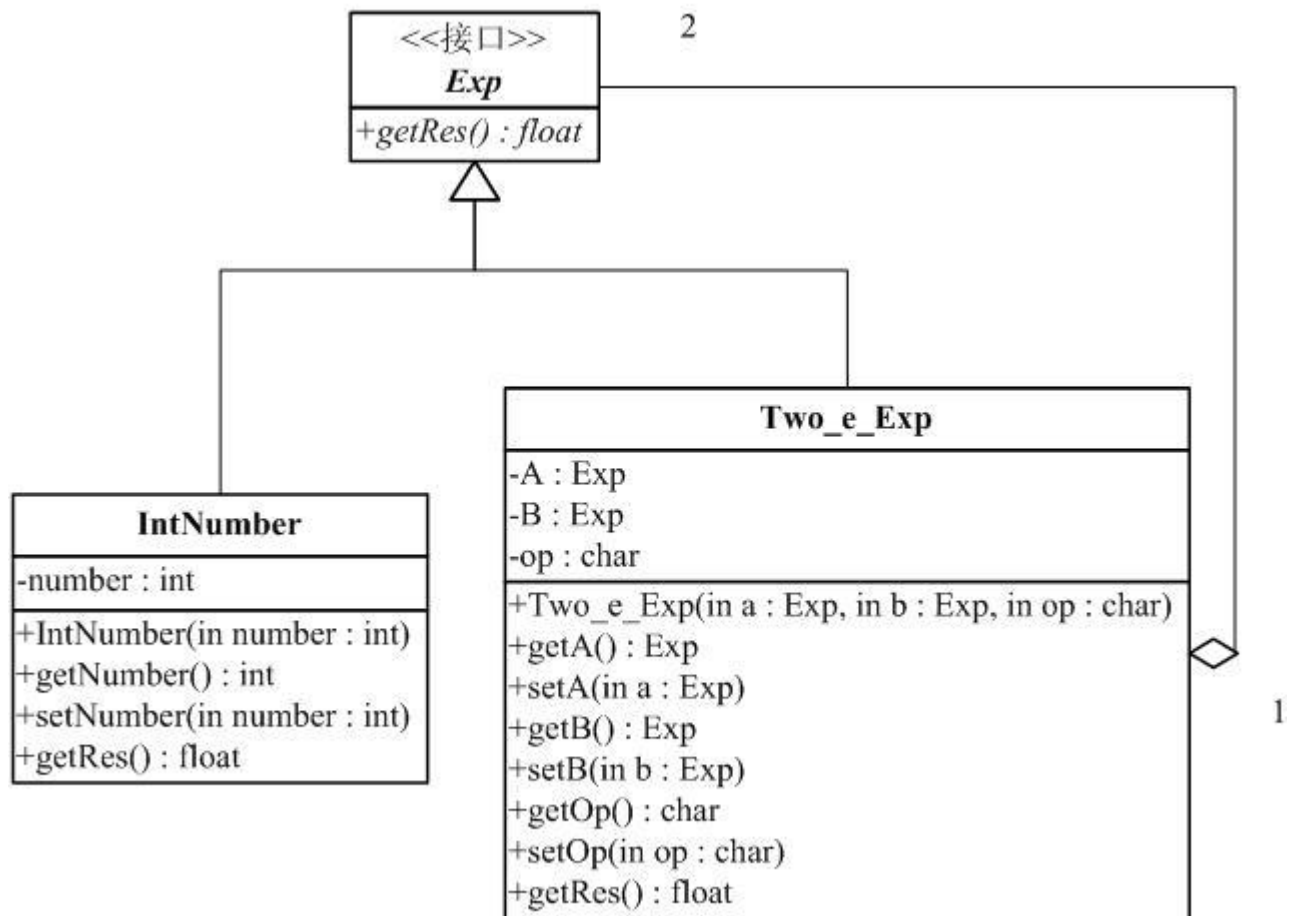
<http://www.devx.com> (英文) 最后, devx.com 包含比较全面的 ASP.NET 列表和可用的 .NET 相关资源。从书籍、Web 站点到活动, 您都可以在这里找到感兴趣的东西。

6.7

某公司 java 面试题 uml

题目是要求描述算术表达式, 操作符仅限于+, -, *, /四种二元操作符, 操作数仅限于整数。

回来想了一下这个题其实是可以利用 composite 模式来做的, UML 静态类图如下:



相应的 Java 代码如下（没有考虑除数为零的异常）：

```

1 interface Exp {
2     float getRes();
3 }
4
5 public class Two_e_Exp implements Exp {
6     private Exp A;
7
8     private Exp B;
9
10    private char op;
11
12    /**
13     * @param a
14     * @param b
15     * @param op
16     */
17    public Two_e_Exp(Exp a, Exp b, char op) {
18        A = a;
19        B = b;
20        this.op = op;
21    }
22 }
  
```

```
18
19 /** **/**
20  * @return a
21  */
22 public Exp getA() {
23     return A;
24 }
25
26 /** **/**
27  * @param a
28  *      要设置的 a
29  */
30 public void setA(Exp a) {
31     A = a;
32 }
33
34 /** **/**
35  * @return b
36  */
37 public Exp getB() {
38     return B;
39 }
40
41 /** **/**
42  * @param b
43  *      要设置的 b
44  */
45 public void setB(Exp b) {
46     B = b;
47 }
48
49 /** **/**
50  * @return op
51  */
52 public char getOp() {
53     return op;
54 }
55
56 /** **/**
57  * @param op
58  *      要设置的 op
59  */
60 public void setOp(char op) {
61     this.op = op;
```

```

62  }
63
64  public float getRes() {
65      // case '+'
66      float res = A.getRes() + B.getRes();
67      switch (op) {
68          case '-': {
69              res = A.getRes() - B.getRes();
70              break;
71          }
72          case '*': {
73              res = A.getRes() * B.getRes();
74              break;
75          }
76          case '/': {
77              res = A.getRes() / B.getRes();
78              break;
79          }
80      }
81      return res;
82  }
83
84}

1 public class IntNumber implements Exp {
2     private int number;
3
4     /** **/**
5      * @param number
6      */
7     public IntNumber(int number) {
8         this.number = number;
9     }
10
11     /** **/**
12      * @return number
13      */
14     public int getNumber() {
15         return number;
16     }
17
18     /** **/**
19      * @param number 要设置的 number
20      */
21     public void setNumber(int number) {

```

```

22     this.number = number;
23 }
24
25 public float getRes() {
26     return number;
27 }
28
29}

```

一个简单的测试程序：

```

1 public class Main {
2
3     /** */
4     * @param args
5     */
6     public static void main(String[] args) {
7         Exp e1 = new Two_e_Exp(new IntNumber(1), new IntNumber(2), '+');
8         Exp e2 = new Two_e_Exp(new IntNumber(3), new IntNumber(4), '*');
9         Exp e = new Two_e_Exp(e2, e1, '/');
10        // (3*4)/(1+2)=4
11        System.out.println(e.getRes());
12        ((Two_e_Exp)e1).setA(new IntNumber(2));
13        // (3*4)/(2+2)=3
14        System.out.println(e.getRes());
15        ((Two_e_Exp)e).setA(new Two_e_Exp(new IntNumber(12), new IntNumber(12), '+'));
16        // (12+12)/(2+2)=6
17        System.out.println(e.getRes());
18        // (12+12)/(2+(5-3))=6
19        ((Two_e_Exp)e1).setB(new Two_e_Exp(new IntNumber(5), new IntNumber(3), '-'));
20        System.out.println(e.getRes());
21    }
22
23}
6.8

```

今天遇到的没有答出来有两个矩形，边与坐标轴平行。写一个判断两个矩形是否有重复的算法

给那些已知条件阿？比如说给坐标吧。

我说一下过程，程序就不写了。

矩形 1 : a (a1,a2) b(b1,b2) c(c1,c2) d(d1,d2)

矩形 1 : A (A1,A2) B(B1,B2) C(C1,C2) D(D1,D2)

这是坐标。

如果有负值，则移动坐标，怎么移算法就不说了。移动后坐标会发生变化。这样，两个矩形就都在第一象限。

$(a1*a1+a2*a2)$ 每个点都这样计算,求出每个矩形的里离原点最近的点和最远的点。例如点分别是 $min1, max1, min2, max2$;
然后在求出 $mi=\min(min1,min2)$, $ma=\max(max1,max2)$
 $mi*ma$ 与 $min1*max1*2+min2*max2*2$ 比较。
前者大说明不重, 后者大说明重, 一样大说明有一点相连。
对于重复线的情况, 用坐标来看, 这里就不说了。

最全的 java 面试题 (一)

一、Java 基础知识

1. Java 有那些基本数据类型, String 是不是基本数据类型, 他们有何区别。
2. 字符串的操作:
写一个方法, 实现字符串的反转, 如: 输入 abc, 输出 cba
写一个方法, 实现字符串的替换, 如: 输入 bbbwlirbbb, 输出 bbbhhtccc。
3. 数据类型之间的转换
如何将数值型字符串转换为数字 (Integer, Double)
如何将数字转换为字符串
如何取小数点前两位, 并四舍五入。
4. 日期和时间
如何取得年月日, 小时分秒
如何取得从 1970 年到现在的毫秒数
如何获取某个日期是当月的最后一天
如何格式化日期
5. 数组和集合
6. 文件和目录 (I/O) 操作
如何列出某个目录下的所有文件
如何列出某个目录下的所有子目录
判断一个文件或目录是否存在
如何读写文件
7. Java 多态的实现 (继承、重载、覆盖)
8. 编码转换, 怎样实现将 GB2312 编码的字符串转换为 ISO-8859-1 编码的字符串。
9. Java 中访问数据库的步骤, Statement 和 PreparedStatement 之间的区别。
10. 找出下列代码可能存在的错误, 并说明原因:

二、JSP&Servlet 技术

1. 描述 JSP 和 Servlet 的区别、共同点、各自应用的范围
2. 在 Web 开发中需要处理 HTML 标记时, 应做什么样的处理, 要筛选那些字符 (<> & “”)
3. 在 JSP 中如何读取客户端的请求, 如何访问 CGI 变量, 如何确定某个 Jsp 文件的真实路径。
4. 描述 Cookie 和 Session 的作用, 区别和各自的应用范围, Session 工作原理。
5. 列出 Jsp 中包含外部文件的方式, 两者有何区别。
6. 说明 Jsp 中 errorPage 的作用, 应用范围。
7. 介绍在 Jsp 中如何使用 JavaBeans。
8. 简单介绍 JSP 的标记库
9. Jsp 和 Servlet 中的请求转发分别如何实现。

三、J2EE 相关知识

- 1.介绍 J2EE、J2SE、J2SE 的区别。
- 2.J2EE 是一种技术还是一种平台，他提供了那些技术。
- 3.什么是 Application Server，它有什么功能和优点。
- 4.简单介绍连接池的优点和原理。
- 5.Web.xml 的作用

四、其他

- 1.Web 安全性的考虑（表单验证、浏览器 Basic 方式的验证，应用程序的安全性，SSL，代码考虑）
- 2.简单介绍您所了解的 MVC。
- 3.简单介绍所了解的 XML。
- 4.文档和编码规范
- 5.Java 中的分页、效率考虑。
- 6.简单介绍您所了解的 struts。

最全的 java 面试题（二）

- 1.xml 在项目中的作用
- 2.s-EJB 与 e-EJB 的区别
- 3.会话面的作用
- 4.cmp 与 bmp 的优缺点
- 5.j2me 程序的必需的几个部分
- 6.c/s 与 b/s 的区别
- 7.构建一个 connect pool,然后再调用它，
- 8.j2ee 平台与 dotnet 平台的区别
- 9.ejb 的 life cycle
- 10.session bean 和 entity bean 的区别
- 11.ejb 中的 transaction 机制
- 12.synchronized (生产者和消费)
- 13.String 和 StringBuffer
- 14.Serializable
- 15.MVC（Struts 的工作流程）
- 16.什么是 MDA
- 17.tcp 与 udp 的区别
- 18.链表与散列表和数组的区别
- 19.堆和栈的区别
- 20.ejb 的分类及区别
- 21.你对现在软件业以及国内软件业的看法
- 22.谈谈 java 多线程

23. 谈谈文件加密技术
24. 软件开发生命周期
25. 路由协议种类及特点
26. java 的 awt 和 swing 组件的 GUI 设计的关键
27. 对于 java 流的认识

28. 简单描述一下 awt 与 swing 区别。
29. 简述 java 编程中事件处理模式。
30. 你编写过 applet 吗? applet 的安全权限如何? 试列举 java application 或者 applet 中与 servlet/jsp 通信可以采用的方式。

31. 简述逻辑操作 (如 &, |) 与条件操作 (如 &&, ||) 的区别。
32. 简述 Java Server Page 和 Servlet 的联系和区别。
33. 简述 synchronized 和 java.util.concurrent.locks.Lock 的异同?
34. EJB 规范规定 EJB 中禁止的操作有哪些?
35. java 除了 8 种基本类型外, 在虚拟机里还有哪一种, 有什么作用?
36. 除了使用 new 关键字创建对象意外, 试列举另外三种以上创建实例的方式?
37. classloader 中, JDK 的 API、Classpath 中的同 web-inf 中的 class 加载方式有什么区别?
38. 列举三种以上垃圾回收算法, 并比较其优缺点?
39. 编写代码实现一个线程池

40. 描述一下 JVM 加载 class 文件的原理机制?
41. 试举例说明一个典型的垃圾回收算法?
42. 请用 java 写二叉树算法, 实现添加数据形成二叉树功能, 并以先序的方式打印出来。
43. 请写一个 java 程序实现线程连接池功能?
44. 给定一个 C 语言函数, 要求实现在 java 类中进行调用。

45. 如何获得数组的长度?
46. 访问修饰符“public/private/protected/缺省的修饰符”的使用
47. 用关键字 final 修饰一个类或者方法时, 有何意义?
48. 掌握类和对象的概念, 掌握面向对象编程的本质
49. 静态变量和静态方法的意义, 如何引用一个类的静态变量或者静态方法?
50. JAVA 语言如何进行异常处理, 关键字: throws, throw, try, catch, finally
51. Object 类(或者其子类)的 finalize() 方法在什么情况下被调用?
52. 一个“.java”原文件中是否可以包括多个类(不是内部类)?
53. 掌握内部类和接口的概念
54. StringTokenizer 类的使用
55. 数据结构, 如何遍历 List 中的元素?
如果要按照键值保存或者访问数据, 使用什么数据结构?
要掌握 Collection 相关的接口和类的使用
56. 使用 StringBuffer 类与 String 类进行字符串连接时有何区别?
57. 调用 Thread 类的 destroy() 方法有什么后果?
58. 多线程, 用什么关键字修饰同步方法? stop() 和 suspend() 方法为何不推荐使用?
59. 使用 socket 建立客户端与服务器的通信的过程

- 60. JAVA 语言国际化应用, Locale 类, Unicode
- 61. 描述反射机制的作用
- 62. 如何读写一个文件?
- 63. 在图形界面中, 一个按钮如何处理鼠标点击事件?
- 64. 在图形界面中, 一个表格, 如何实现编辑单元格时弹出下拉框?
- 65. 如何加载图片?
- 66. 什么是模态对话框?
- 67. 阐述 MVC 的概念
- 68. GUI 布局管理器的使用, FlowLayout, BorderLayout, GridBagLayout
- 69. 如何构造一棵树? 选择树的一个节点时, 如何得到这个节点?
- 70. 向编辑框中输入字符时, 如何控制只输入整数?
- 71. 描述使用 JDBC 连接数据库的过程
- 72. EJB 分为几类? 什么是 BMP, CMP?
- 73. 什么是 JNDI?

- 74. ADO 是什么? ActiveX 数据对象, 是一个应用级程序接口.
- 75. 四种 JDBC 方式? 目前的版本?
- 76. EJB 有哪几种? 区别是什么?
- 77. JavaBean 与 EJB 有什么区别?
- 78. 软件开发生命周期有哪几个阶段?
- 79. 软件开发有哪些因素?
- 80. 软件开发中如何进行版本控制?
- 81. UML 中, 类视图如何表示类中的继承与聚合?
- 82. 客户端游标与服务器端游标的区别?
- 83. 动态游标与静态游标的区别?
- 84. dotnet 由哪几个基本框架组成?
- 85. Oracle 中 SGA 是什么?
- 86. web servers 是什么?
- 87. UNIX 中 QT 是什么意思?
- 88. 在软件开发生命周期中的哪个阶段开始测试?
- 89. dotnet 与 J2EE 的比较?
- 90. 什么是 ActiveX?
- 91. Java 中 IDL 是什么?
- 92. ISO9000 和 CMM 是什么? ISO9000 和 CMM(软件能力成熟度模型) 认证是国际上通用的软件质量评估方法. CMM 的五个成熟度等级。

最全的 java 面试题 (三)

第一, 谈谈 final, finally, finalize 的区别。

final? 修饰符(关键字) 如果一个类被声明为 final, 意味着它不能再派生出新的子类, 不能作为父类被继承。因此一个类不能既被声明为 abstract 的, 又 被声明为 final 的。将变量或方法声明为 final, 可以保证它们在使用中不被改变。被声明为 final 的变量必须在声明时给定初值, 而在

以后的引用中 只能读取，不可修改。被声明为 **final** 的方法也同样只能使用，不能重载 **finally**?再异常处理时提供 **finally** 块来执行任何清除操作。如果抛出一个异常，那么相匹配的 **catch** 子句就会执行，然后控制就会进入 **finally** 块（如果有的话）。

finalize? 方法名。Java 技术允许使用 **finalize()** 方法在垃圾收集器将对象从内存中清除出去之前做必要的清理工作。这个方法是由垃圾收集器在确定这个对象没有被引用时对这个对象调用的。它是在 **Object** 类中定义的，因此所有的类都继承了它。子类覆盖 **finalize()** 方法以整理系统资源或者执行其他清理工作。**finalize()** 方法是在垃圾收集器删除对象之前对这个对象调用的。

第二， **Anonymous Inner Class** (匿名内部类) 是否可以 **extends**(继承) 其它类，是否可以 **implements**(实现)**interface**(接口)?

匿名的内部类是没有名字的内部类。不能 **extends**(继承) 其它类，但一个内部类可以作为一个接口，由另一个内部类实现。

第三， **Static Nested Class** 和 **Inner Class** 的不同，说得越多越好(面试题有的很笼统)。

Nested Class (一般是 C++ 的说法)，**Inner Class** (一般是 JAVA 的说法)。Java 内部类与 C++ 嵌套类最大的不同就在于是否有指向外部的引用上。具体可见 <http://www.frontfree.net/articles/service/view.asp?id=704&page=1>

注：静态内部类 (**Inner Class**) 意味着 1 创建一个 **static** 内部类的对象，不需要一个外部类对象，2 不能从一个 **static** 内部类的一个对象访问一个外部类对象

第四，**&**和**&&**的区别。

&是位运算符。**&&**是布尔逻辑运算符。

第五，**HashMap** 和 **Hashtable** 的区别。

都属于 **Map** 接口的类，实现了将唯一键映射到特定的值上。

HashMap 类没有分类或者排序。它允许一个 **null** 键和多个 **null** 值。

Hashtable 类似于 **HashMap**，但是不允许 **null** 键和 **null** 值。它也比 **HashMap** 慢，因为它是同步的。

第六，**Collection** 和 **Collections** 的区别。

Collections 是个 **java.util** 下的类，它包含有各种有关集合操作的静态方法。

Collection 是个 **java.util** 下的接口，它是各种集合结构的父接口。

第七，什么时候用 **assert**。

断言是一个包含布尔表达式的语句，在执行这个语句时假定该表达式为 **true**。如果表达式计算为 **false**，那么系统会报告一个 **AssertionError**。它用于调试目的：

```
assert(a > 0); // throws an AssertionError if a <= 0
```

断言可以有两种形式：

```
assert Expression1 ;
```

```
assert Expression1 : Expression2 ;
```

Expression1 应该总是产生一个布尔值。

Expression2 可以是得出一个值的任意表达式。这个值用于生成显示更多调试信息的 **String** 消息。

断言在默认情况下是禁用的。要在编译时启用断言，需要使用 **source 1.4** 标记：

```
javac -source 1.4 Test.java
```

要在运行时启用断言，可使用 `-enableassertions` 或者 `-ea` 标记。

要在运行时选择禁用断言，可使用 `-da` 或者 `-disableassertions` 标记。

要系统类中启用断言，可使用 `-esa` 或者 `-dsa` 标记。还可以在包的基础上启用或者禁用断言。

可以在预计正常情况下不会到达的任何位置上放置断言。断言可以用于验证传递给私有方法的参数。不过，断言不应该用于验证传递给公有方法的参数，因为不管是否启用了断言，公有方法都必须检查其参数。不过，既可以在公有方法中，也可以在非公有方法中利用断言测试后置条件。另外，断言不应该以任何方式改变程序的状态。

第八，GC 是什么？为什么要有 GC？(基础)。

GC 是垃圾收集器。Java 程序员不用担心内存管理，因为垃圾收集器会自动进行管理。要请求垃圾收集，可以调用下面的方法之一：

`System.gc()`

`Runtime.getRuntime().gc()`

第九，`String s = new String("xyz");` 创建了几个 String Object？

两个对象，一个是“xyz”，一个是指向“xyz”的引用对象 s。

第十，`Math.round(11.5)` 等于多少？`Math.round(-11.5)` 等于多少？

`Math.round(11.5)` 返回 (long) 12，`Math.round(-11.5)` 返回 (long) -11；

第十一，`short s1 = 1; s1 = s1 + 1;` 有什么错？`short s1 = 1; s1 += 1;` 有什么错？

`short s1 = 1; s1 = s1 + 1;` 有错，s1 是 short 型，s1+1 是 int 型，不能显式转化为 short 型。可修改为 `s1 = (short)(s1 + 1)`。`short s1 = 1; s1 += 1` 正确。

第十二，`sleep()` 和 `wait()` 有什么区别？搞线程的最爱

`sleep()` 方法是使线程停止一段时间的方法。在 `sleep` 时间间隔期满后，线程不一定立即恢复执行。这是因为在那个时刻，其它线程可能正在运行而且没有被调度为放弃执行，除非(a)“醒来”的线程具有更高的优先级

(b)正在运行的线程因为其它原因而阻塞。

`wait()` 是线程交互时，如果线程对一个同步对象 x 发出一个 `wait()` 调用，该线程会暂停执行，被调对象进入等待状态，直到被唤醒或等待时间到。

第十三，Java 有没有 goto？

Goto? java 中的保留字，现在没有在 java 中使用。

第十四，数组有没有 `length()` 这个方法？String 有没有 `length()` 这个方法？

数组没有 `length()` 这个方法，有 `length` 的属性。

String 有 `length()` 这个方法。

第十五，Overload 和 Override 的区别。Overloaded 的方法是否可以改变返回值的类型？

方法的重写 Overriding 和重载 Overloading 是 Java 多态性的不同表现。重写 Overriding 是父类

与子类之间多态性的一种表现，重载 Overloading 是一个类中多态性的一种表现。如果在子类中定义某方法与其父类有相同的名称和参数，我们说该方法被重写 (Overriding)。子类的对象使用这个方法时，将调用子类中的定义，对它而言，父类中的定义如同被“屏蔽”了。如果在一个类中定义了多个同名的方法，它们或有不同的参数个数或有不同的参数类型，则称为方法的重载 (Overloading)。Overloaded 的方法是可以改变返回值的类型。

第十六，Set 里的元素是不能重复的，那么用什么方法来区分重复与否呢？是用 == 还是 equals()？它们有何区别？

Set 里的元素是不能重复的，那么用 iterator() 方法来区分重复与否。equals() 是判断两个 Set 是否相等。

equals() 和 == 方法决定引用值是否指向同一对象 equals() 在类中被覆盖，为的是当两个分离的对象的内容和类型相配的话，返回真值。

第十七，给我一个你最常见到的 runtime exception。

ArithmeticException, ArrayStoreException, BufferOverflowException, BufferUnderflowException, CannotRedoException, CannotUndoException, ClassCastException, CMMException, ConcurrentModificationException, DOMException, EmptyStackException, IllegalArgumentException, IllegalMonitorStateException, IllegalPathStateException, IllegalStateException, ImagingOpException, IndexOutOfBoundsException, MissingResourceException, NegativeArraySizeException, NoSuchElementException, NullPointerException, ProfileDataException, ProviderException, RasterFormatException, SecurityException, SystemException, UndeclaredThrowableException, UnmodifiableSetException, UnsupportedOperationException

第十八，error 和 exception 有什么区别？

error 表示恢复不是不可能但很困难的情况下的一种严重问题。比如说内存溢出。不可能指望程序能处理这样的情况。

exception 表示一种设计或实现问题。也就是说，它表示如果程序运行正常，从不会发生的情况。

第十九，List, Set, Map 是否继承自 Collection 接口？

List, Set 是

Map 不是

第二十，abstract class 和 interface 有什么区别？

声明方法的存在而不去实现它的类被叫做抽象类 (abstract class)，它用于要创建一个体现某些基本行为的类，并为该类声明方法，但不能在该类中实现该方法的情况。不能创建 abstract 类的实例。然而可以创建一个变量，其类型是一个抽象类，并让它指向具体子类的一个实例。不能有抽象构造函数或抽象静态方法。Abstract 类的子类为它们父类中的所有抽象方法提供实现，否则它们也是抽象类。取而代之，在子类中实现该方法。知道其行为的其它类可以在类中实现这些方法。

接口 (interface) 是抽象类的变体。在接口中，所有方法都是抽象的。多继承性可通过实现这样的接口而获得。接口中的所有方法都是抽象的，没有一个有程序体。接口只可以定义 static final 成员变量。接口的实现与子类相似，除了该实现类不能从接口定义中继承行为。当类实现特殊

接口时，它定义（即将程序体给予）所有这种接口的方法。然后，它可以在实现了该接口的类的任何对象上调用接口的方法。由于有抽象类，它允许使用接口名作为引用变量的类型。通常的动态联编将生效。引用可以转换到接口类型或从接口类型转换，`instanceof`运算符可以用来决定某对象的类是否实现了接口。

第二十一，`abstract`的 `method` 是否可同时是 `static`，是否可同时是 `native`，是否可同时是 `synchronized`？都不能

第二十二，接口是否可继承接口？抽象类是否可实现 (`implements`) 接口？抽象类是否可继承实体类 (`concrete class`)？

接口可以继承接口。抽象类可以实现 (`implements`) 接口，抽象类是否可继承实体类，但前提是实体类必须有明确的构造函数。

第二十三，启动一个线程是用 `run()` 还是 `start()`？

启动一个线程是调用 `start()` 方法，使线程所代表的虚拟处理机处于可运行状态，这意味着它可以由 JVM 调度并执行。这并不意味着线程就会立即运行。`run()` 方法可以产生必须退出的标志来停止一个线程。

第二十四，构造器 `Constructor` 是否可被 `override`？

构造器 `Constructor` 不能被继承，因此不能重写 `Overriding`，但可以被重载 `Overloading`。

第二十五，是否可以继承 `String` 类？

`String` 类是 `final` 类故不可以继承。

第二十六，当一个线程进入一个对象的一个 `synchronized` 方法后，其它线程是否可进入此对象的其它方法？

不能，一个对象的一个 `synchronized` 方法只能由一个线程访问。

第二十七，`try {}` 里有一个 `return` 语句，那么紧跟在这个 `try` 后的 `finally {}` 里的 `code` 会不会被执行，什么时候被执行，在 `return` 前还是后？

会执行，在 `return` 前执行。

第二十八，编程题：用最有效率的方法算出 2 乘以 8 等于几？

有 C 背景的程序员特别喜欢问这种问题。

`2 << 3`

第二十九，两个对象值相同 (`x.equals(y) == true`)，但却可有不同的 `hash code`，这句话对不对？

不对，有相同的 `hash code`。

第三十，当一个对象被当作参数传递到一个方法后，此方法可改变这个对象的属性，并可返回变

化后的结果，那么这里到底是值传递还是引用传递？

是值传递。Java 编程语言只由值传递参数。当一个对象实例作为一个参数被传递到方法中时，参数的值就是对该对象的引用。对象的内容可以在被调用的方法中改变，但对象的引用是永远不会改变的。

第三十一，switch 是否能作用在 byte 上，是否能作用在 long 上，是否能作用在 String 上？

switch (expr1) 中，expr1 是一个整数表达式。因此传递给 switch 和 case 语句的参数应该是 int、short、char 或者 byte。long、string 都不能作用于 switch。

第三十二，编程题：写一个 Singleton 出来。

Singleton 模式主要作用是保证在 Java 应用程序中，一个类 Class 只有一个实例存在。

一般 Singleton 模式通常有几种形式：

第一种形式：定义一个类，它的构造函数为 private 的，它有一个 static 的 private 的该类变量，在类初始化时实例化，通过一个 public 的 getInstance 方法获取对它的引用，继而调用其中的方法。

```
public class Singleton {
    private Singleton(){}
    //在自己内部定义自己一个实例，是不是很奇怪？
    //注意这是 private 只供内部调用
    private static Singleton instance = new Singleton();
    //这里提供了一个供外部访问本 class 的静态方法，可以直接访问
    public static Singleton getInstance() {
        return instance;
    }
}
```

第二种形式：

```
public class Singleton {
    private static Singleton instance = null;
    public static synchronized Singleton getInstance() {
        //这个方法比上面有所改进，不用每次都进行生成对象，只是第一次
        //使用时生成实例，提高了效率！
        if (instance==null)
            instance=new Singleton();
        return instance;
    }
}
```

其他形式：

定义一个类，它的构造函数为 private 的，所有方法为 static 的。

一般认为第一种形式要更加安全些

6.9

最全的 java 面试题（四）

Java 面试题和答案

JAVA 相关基础知识

1、面向对象的特征有哪些方面

1.抽象:

抽象就是忽略一个主题中与当前目标无关的那些方面，以便更充分地注意与当前目标有关的方面。抽象并不打算了解全部问题，而只是选择其中的一部分，暂时不用部分细节。抽象包括两个方面，一是过程抽象，二是数据抽象。

2.继承:

继承是一种联结类的层次模型，并且允许和鼓励类的重用，它提供了一种明确表述共性的方法。对象的一个新类可以从现有的类中派生，这个过程称为类继承。新类继承了原始类的特性，新类称为原始类的派生类（子类），而原始类称为新类的基类（父类）。派生类可以从它的基类那里继承方法和实例变量，并且类可以修改或增加新的方法使之更适合特殊的需要。

3.封装:

封装是把过程和数据包围起来，对数据的访问只能通过已定义的界面。面向对象计算始于这个基本概念，即现实世界可以被描绘成一系列完全自治、封装的对象，这些对象通过一个受保护的接口访问其他对象。

4. 多态性:

多态性是指允许不同类的对象对同一消息作出响应。多态性包括参数化多态性和包含多态性。多态性语言具有灵活、抽象、行为共享、代码共享的优势，很好的解决了应用程序函数同名问题。

2、String 是最基本的数据类型吗？

基本数据类型包括 byte、int、char、long、float、double、boolean 和 short。

java.lang.String 类是 final 类型的，因此不可以继承这个类、不能修改这个类。为了提高效率节省空间，我们应该用 StringBuffer 类

3、int 和 Integer 有什么区别

Java 提供两种不同的类型：引用类型和原始类型（或内置类型）。Int 是 java 的原始数据类型，Integer 是 java 为 int 提供的封装类。Java 为每个原始类型提供了封装类。

原始类型 封装类

boolean Boolean

char Character

byte Byte

short Short

int Integer

long Long

float Float

double Double

引用类型和原始类型的行为完全不同，并且它们具有不同的语义。引用类型和原始类型具有不同的特征和用法，它们包括：大小和速度问题，这种类型以哪种类型的数据结构存储，当引用类型和原始类型用作某个类的实例数据时所指定的缺省值。对象引用实例变量的缺省值为 null，而原始类型实例变量的缺省值与它们的类型有关。

4、String 和 StringBuffer 的区别

JAVA 平台提供了两个类：String 和 StringBuffer，它们可以储存和操作字符串，即包含多个字符的字符数据。这个 String 类提供了数值不可改变的字符串。而这个 StringBuffer 类提供的字符串进行修改。当你知道字符数据要改变的时候你就可以使用 StringBuffer。典型地，你可以使用 StringBuffer 来动态构造字符数据。

5、运行时异常与一般异常有何异同？

异常表示程序运行过程中可能出现的非正常状态，运行时异常表示虚拟机的通常操作中可能遇到的异常，是一种常见运行错误。java 编译器要求方法必须声明抛出可能发生的非运行时异常，但是并不要求必须声明抛出未被捕获的运行时异常。

6、说出 Servlet 的生命周期，并说出 Servlet 和 CGI 的区别。

Servlet 被服务器实例化后，容器运行其 init 方法，请求到达时运行其 service 方法，service 方法自动派遣运行与请求对应的 doXXX 方法（doGet，doPost）等，当服务器决定将实例销毁的时候调用其 destroy 方法。

与 cgi 的区别在于 servlet 处于服务器进程中，它通过多线程方式运行其 service 方法，一个实例可以服务于多个请求，并且其实例一般不会销毁，而 CGI 对每个请求都产生新的进程，服务完成后就销毁，所以效率上低于 servlet。

7、说出 ArrayList, Vector, LinkedList 的存储性能和特性

ArrayList 和 Vector 都是使用数组方式存储数据，此数组元素数大于实际存储的数据以便增加和插入元素，它们都允许直接按序号索引元素，但是插入元素要涉及数组元素移动等内存操作，所以索引数据快而插入数据慢，Vector 由于使用了 synchronized 方法（线程安全），通常性能上较 ArrayList 差，而 LinkedList 使用双向链表实现存储，按序号索引数据需要进行前向或后向遍历，但是插入数据时只需要记录本项的前后项即可，所以插入速度较快。

8、EJB 是基于哪些技术实现的？并说出 SessionBean 和 EntityBean 的区别，StatefulBean 和 StatelessBean 的区别。

EJB 包括 Session Bean、Entity Bean、Message Driven Bean，基于 JNDI、RMI、JAT 等技术实现。

SessionBean 在 J2EE 应用程序中被用来完成一些服务器端的业务操作，例如访问数据库、调用其他 EJB 组件。EntityBean 被用来代表应用系统中用到的数据。

对于客户机，SessionBean 是一种非持久性对象，它实现某些在服务器上运行的业务逻辑。

对于客户机，EntityBean 是一种持久性对象，它代表一个存储在持久性存储器中的实体的对象视图，或是一个由现有企业应用程序实现的实体。

Session Bean 还可以再细分为 Stateful Session Bean 与 Stateless Session Bean，这两种的 Session Bean 都可以将系统逻辑放在 method 之中执行，不同的是 Stateful Session Bean 可以记录呼叫者的状态，因此通常来说，一个使用者会有一个相对应的 Stateful Session Bean 的实体。Stateless Session Bean 虽然也是逻辑组件，但是他却不负责记录使用者状态，也就是说当使用者呼叫 Stateless Session Bean 的时候，EJB Container 并不会找寻特定的 Stateless Session Bean 的实体来执行这个 method。换言之，很可能数个使用者在执行某个 Stateless Session Bean 的 methods 时，会是同一个 Bean 的 Instance 在执行。从内存方面来看，Stateful Session Bean 与 Stateless Session Bean 比较，Stateful Session Bean 会消耗 J2EE Server 较多的内存，然而 Stateful Session Bean 的优势却在于他可以维持使用者的状态。

9、Collection 和 Collections 的区别。

Collection 是集合类的上级接口，继承与他的接口主要有 Set 和 List。

Collections 是针对集合类的一个帮助类，他提供一系列静态方法实现对各种集合的搜索、排序、线程安全化等操作。

10、&和&&的区别。

&是位运算符，表示按位与运算，&&是逻辑运算符，表示逻辑与（and）。

11、HashMap 和 Hashtable 的区别。

HashMap 是 Hashtable 的轻量级实现（非线程安全的实现），他们都完成了 Map 接口，主要区别在于 HashMap 允许空（null）键值（key），由于非线程安全，效率上可能高于 Hashtable。HashMap 允许将 null 作为一个 entry 的 key 或者 value，而 Hashtable 不允许。

HashMap 把 Hashtable 的 contains 方法去掉了，改成 containsvalue 和 containsKey。因为 contains 方法容易让人引起误解。

Hashtable 继承自 Dictionary 类，而 HashMap 是 Java1.2 引进的 Map interface 的一个实现。

最大的不同是，Hashtable 的方法是 Synchronize 的，而 HashMap 不是，在多个线程访问 Hashtable 时，不需要自己为它的方法实现同步，而 HashMap 就必须为之提供外同步。

Hashtable 和 HashMap 采用的 hash/rehash 算法都大概一样，所以性能不会有很大的差异。

12、final, finally, finalize 的区别。

final 用于声明属性，方法和类，分别表示属性不可变，方法不可覆盖，类不可继承。

finally 是异常处理语句结构的一部分，表示总是执行。

finalize 是 Object 类的一个方法，在垃圾收集器执行的时候会调用被回收对象的此方法，可以覆盖此方法提供垃圾收集时的其他资源回收，例如关闭文件等。

13、sleep() 和 wait() 有什么区别？

sleep 是线程类（Thread）的方法，导致此线程暂停执行指定时间，给执行机会给其他线程，但是监控状态依然保持，到时后会自动恢复。调用 sleep 不会释放对象锁。

wait 是 Object 类的方法，对此对象调用 wait 方法导致本线程放弃对象锁，进入等待此对象的等待锁定池，只有针对此对象发出 notify 方法（或 notifyAll）后本线程才进入对象锁定池准备获得对象锁进入运行状态。

14、Overload 和 Override 的区别。Overloaded 的方法是否可以改变返回值的类型？

方法的重写 Overriding 和重载 Overloading 是 Java 多态性的不同表现。重写 Overriding 是父类与子类之间多态性的一种表现，重载 Overloading 是一个类中多态性的一种表现。如果在子类中定义某方法与其父类有相同的名称和参数，我们说该方法被重写（Overriding）。子类的对象使用这个方法时，将调用子类中的定义，对它而言，父类中的定义如同被“屏蔽”了。如果在一个类中定义了多个同名的方法，它们或有不同的参数个数或有不同的参数类型，则称为方法的重载（Overloading）。Overloaded 的方法是可以改变返回值的类型。

15、error 和 exception 有什么区别？

error 表示恢复不是不可能但很困难的情况下的一种严重问题。比如说内存溢出。不可能指望程序能处理这样的情况。

exception 表示一种设计或实现问题。也就是说，它表示如果程序运行正常，从不会发生的情况。

16、同步和异步有何异同，在什么情况下分别使用他们？举例说明。

如果数据将在线程间共享。例如正在写的的数据以后可能被另一个线程读到，或者正在读的数据可能已经被另一个线程写过了，那么这些数据就是共享数据，必须进行同步存取。

当应用程序在对象上调用了一个需要花费很长时间来执行的方法，并且不希望让程序等待方法的返回时，就应该使用异步编程，在很多情况下采用异步途径往往更有效率。

17、abstract class 和 interface 有什么区别？

声明方法的存在而不去实现它的类被叫做抽象类（abstract class），它用于要创建一个体现某些基本行为的类，并为该类声明方法，但不能在该类中实现该类的情况。不能创建 abstract 类的实例。然而可以创建一个变量，其类型是一个抽象类，并让它指向具体子类的一个实例。不能有抽象构造函数或抽象静态方法。Abstract 类的子类为它们父类中的所有抽象方法提供实现，否则它们也是抽象类为。取而代之，在子类中实现该方法。知道其行为的其它类可以在类中实现这些方法。

接口（interface）是抽象类的变体。在接口中，所有方法都是抽象的。多继承性可通过实现这样的接口而获得。接口中的所有方法都是抽象的，没有一个有程序体。接口只可以定义 static final 成员变量。接口的实现与子类相似，除了该实现类不能从接口定义中继承行为。当类实现特殊

接口时，它定义（即将程序体给予）所有这种接口的方法。然后，它可以在实现了该接口的类的任何对象上调用接口的方法。由于有 抽象类，它允许使用接口名作为引用变量的类型。通常的动态联编将生效。引用可以转换到接口类型或从接口类型转换， instanceof 运算符可以用来 决定某对象的类是否实现了接口。

18、heap 和 stack 有什么区别。

栈是一种线形集合，其添加和删除元素的操作应在同一段完成。栈按照后进先出的方式进行处理。堆是栈的一个组成元素

19、forward 和 redirect 的区别

forward 是服务器请求资源，服务器直接访问目标地址的 URL，把那个 URL 的响应内容读取过来，然后把这些内容再发给浏览器，浏览器根本不知道服务器发送的内容是从哪儿来的，所以它的地址栏中还是原来的地址。

redirect 就是服务端根据逻辑,发送一个状态码,告诉浏览器重新去请求那个地址，一般来说浏览器会用刚才请求的所有参数重新请求，所以 session,request 参数都可以获取。

20、EJB 与 JAVA BEAN 的区别？

Java Bean 是可复用的组件，对 Java Bean 并没有严格的规范，理论上讲，任何一个 Java 类都可以是一个 Bean。但通常情况下，由于 Java Bean 是被容器所创建（如 Tomcat）的，所以 Java Bean 应具有一个无参的构造器，另外，通常 Java Bean 还要实现 Serializable 接口用于实现 Bean 的持久性。Java Bean 实际上相当于微软 COM 模型中的本地进程内 COM 组件，它是不能被跨进程访问的。Enterprise Java Bean 相当于 DCOM，即分布式组件。它是基于 Java 的远程方法调用（RMI）技术的，所以 EJB 可以被远程访问（跨进程、跨计算机）。但 EJB 必须被部署在诸如 Webspere、WebLogic 这样的容器中，EJB 客户从不直接访问真正的 EJB 组件，而是通过其容器访问。EJB 容器是 EJB 组件的代理，EJB 组件由容器所创建和管理。客户通过容器来访问真正的 EJB 组件。

21、Static Nested Class 和 Inner Class 的不同。

Static Nested Class 是被声明为静态（static）的内部类，它可以不依赖于外部类实例被实例化。而通常的内部类需要在外部类实例化后才能实例化。

22、JSP 中动态 INCLUDE 与静态 INCLUDE 的区别？

动态 INCLUDE 用 jsp:include 动作实现 <jsp:include page="included.jsp" flush="true" />它总是会检查所含文件中的变化，适合用于包含动态页面，并且可以带参数。

静态 INCLUDE 用 include 伪码实现,定不会检查所含文件的变化，适用于包含静态页面 <%@ include file="included.htm" %>

23、什么时候用 assert。

assertion (断言)在软件开发中是一种常用的调试方式，很多开发语言中都支持这种机制。在实现中，assertion 就是在程序中的一条语句，它对一个 boolean 表达式进行检查，一个正确程序必须保证这个 boolean 表达式的值为 true；如果该值为 false，说明程序已经处于不正确的状态下，系统将给出警告或退出。一般来说，assertion 用于保证程序最基本、关键的正确性。assertion 检查通常在开发和测试时开启。为了提高性能，在软件发布后，assertion 检查通常是关闭的。

24、GC 是什么？为什么要有 GC？

GC 是垃圾收集的意思（Garbage Collection），内存处理是编程人员容易出现问题的地方，忘记或者错误的内存回收会导致程序或系统的不稳定甚至崩溃，Java 提供的 GC 功能可以自动监测对象是否超过作用域从而达到自动回收内存的目的，Java 语言没有提供释放已分配内存的显示操作方法。

25、short s1 = 1; s1 = s1 + 1;有什么错？ short s1 = 1; s1 += 1;有什么错？

short s1 = 1; s1 = s1 + 1;（s1+1 运算结果是 int 型，需要强制转换类型）

short s1 = 1; s1 += 1; (可以正确编译)

26、Math.round(11.5)等於多少? Math.round(-11.5)等於多少?

Math.round(11.5)=12

Math.round(-11.5)=-11

round 方法返回与参数最接近的长整数，参数加 1/2 后求其 floor.

27、String s = new String("xyz");创建了几个 String Object?

两个

28、设计 4 个线程，其中两个线程每次对 j 增加 1，另外两个线程对 j 每次减少 1。写出程序。

以下程序使用内部类实现线程，对 j 增减的时候没有考虑顺序问题。

```
public class ThreadTest1{
    private int j;
    public static void main(String args[]){
        ThreadTest1 tt=new ThreadTest1();
        Inc inc=tt.new Inc();
        Dec dec=tt.new Dec();
        for(int i=0;i<2;i++){
            Thread t=new Thread(inc);
            t.start();
            t=new Thread(dec);
            t.start();
        }
        private synchronized void inc(){
            j++;
            System.out.println(Thread.currentThread().getName()+"-inc:"+j);
        }
        private synchronized void dec(){
            j--;
            System.out.println(Thread.currentThread().getName()+"-dec:"+j);
        }
        class Inc implements Runnable{
            public void run(){
                for(int i=0;i<100;i++){
                    inc();
                }
            }
        }
        class Dec implements Runnable{
            public void run(){
                for(int i=0;i<100;i++){
                    dec();
                }
            }
        }
    }
}
```

}

29、Java 有没有 goto?

java 中的保留字，现在没有在 java 中使用。

30、启动一个线程是用 run()还是 start()?

启动一个线程是调用 start()方法，使线程所代表的虚拟处理机处于可运行状态，这意味着它可以由 JVM 调度并执行。这并不意味着线程就会立即运行。run()方法可以产生必须退出的标志来停止一个线程。

31、EJB 包括 (SessionBean,EntityBean) 说出他们的生命周期，及如何管理事务的?

SessionBean: Stateless Session Bean 的生命周期是由容器决定的，当客户机发出请求要建立一个 Bean 的实例时，EJB 容器不一定要创建一个新的 Bean 的实例供客户机调用，而是随便找一个现有的实例提供给客户机。当客户机第一次调用一个 Stateful Session Bean 时，容器必须立即在服务器中创建一个新的 Bean 实例，并关联到客户机上，以后此客户机调用 Stateful Session Bean 的方法时容器会把调用分派到与此客户机相关联的 Bean 实例。

EntityBean: Entity Beans 能存活相对较长的时间，并且状态是持续的。只要数据库中的数据存在，Entity beans 就一直存活。而不是按照应用程序或者服务进程来说的。即使 EJB 容器崩溃了，Entity beans 也是存活的。Entity Beans 生命周期能够被容器或者 Beans 自己管理。

EJB 通过以下技术管理事务：对象管理组织 (OMG) 的对象事务服务 (OTS)，Sun Microsystems 的 Transaction Service (JTS)、Java Transaction API (JTA)，开发组 (X/Open) 的 XA 接口。

32、应用服务器有那些?

BEA WebLogic Server, IBM WebSphere Application Server, Oracle9i Application Server, jBoss, Tomcat

33、给我一个你最常见到的 runtime exception。

ArithmeticException, ArrayStoreException, BufferOverflowException, BufferUnderflowException, CannotRedoException, CannotUndoException, ClassCastException, CMMException, ConcurrentModificationException, DOMException, EmptyStackException, IllegalArgumentException, IllegalMonitorStateException, IllegalPathStateException, IllegalStateException, ImagingOpException, IndexOutOfBoundsException, MissingResourceException, NegativeArraySizeException, NoSuchElementException, NullPointerException, ProfileDataException, ProviderException, RasterFormatException, SecurityException, SystemException, UndeclaredThrowableException, UnmodifiableSetException, UnsupportedOperationException

34、接口是否可继承接口？抽象类是否可实现 (implements)接口？抽象类是否可继承实体类 (concrete class)?

接口可以继承接口。抽象类可以实现 (implements)接口，抽象类是否可继承实体类，但前提是实体类必须有明确的构造函数。

35、List, Set, Map 是否继承自 Collection 接口?

List, Set 是, Map 不是

6.10

最全的 java 面试题（五）

Java 基础方面:

1、作用域 public,private,protected,以及不写时的区别

答：区别如下：

作用域 当前类 同一 package 子孙类 其他 package

public √ √ √ √

protected √ √ √ ×

friendly √ √ × ×

private √ × × ×

不写时默认为 friendly

```
package test;
import test.FatherClass;
public class ChildClass extends FatherClass
{
    public ChildClass()
    {
        System.out.println("ChildClass Create");
    }
    public static void main(String[] args)
    {
        FatherClass fc = new FatherClass();
        ChildClass cc = new ChildClass();
    }
}
```

输出结果:

```
C:>java test.ChildClass
```

```
FatherClass Create
```

```
FatherClass Create
```

```
ChildClass Create
```

6、内部类的实现方式?

答: 示例代码如下:

```
package test;
public class OuterClass
{
    private class InterClass
    {
        public InterClass()
        {
            System.out.println("InterClass Create");
        }
    }
    public OuterClass()
    {
        InterClass ic = new InterClass();
    }
}
```



```

System.out.println("OuterClass Create");
}
public static void main(String[] args)
{
OuterClass oc = new OuterClass();
}
}

```

输出结果:

```
C:>java test/OuterClass
```

```
InterClass Create
```

```
OuterClass Create
```

再一个例题:

```

public class OuterClass {
private double d1 = 1.0;
//insert code here
}

```

You need to insert an inner class declaration at line 3. Which two inner class declarations are

valid?(Choose two.)

- A. class InnerOne{
public static double methoda() {return d1;}
}
- B. public class InnerOne{
static double methoda() {return d1;}
}
- C. private class InnerOne{
double methoda() {return d1;}
}
- D. static class InnerOne{
protected double methoda() {return d1;}
}
- E. abstract class InnerOne{
public abstract double methoda();
}

说明如下:

- 一.静态内部类可以有静态成员，而非静态内部类则不能有静态成员。故 A、B 错
- 二.静态内部类的非静态成员可以访问外部类的静态变量，而不可访问外部类的非静态变量；
return d1 出错。

故 D 错

- 三.非静态内部类的非静态成员可以访问外部类的非静态变量。故 C 正确

四.答案为 C、E

7、垃圾回收机制,如何优化程序?

希望大家补上，谢谢

8、float 型 float f=3.4 是否正确？

答:不正确。精度不准确,应该用强制类型转换，如下所示：float f=(float)3.4

9、介绍 JAVA 中的 Collection Framework(包括如何写自己的数据结构)?

答：Collection Framework 如下：

Collection

- └List
 - └└LinkedList
 - └└ArrayList
 - └└Vector
 - └└Stack
- └Set

Map

- └Hashtable
- └HashMap
- └WeakHashMap

Collection 是最基本的集合接口，一个 Collection 代表一组 Object，即 Collection 的元素 (Elements)

Map 提供 key 到 value 的映射

10、Java 中异常处理机制，事件机制？

11、JAVA 中的多形与继承？

希望大家补上，谢谢

12、抽象类与接口？

答：抽象类与接口都用于抽象，但是抽象类(JAVA中)可以有自己的部分实现，而接口则完全是一个标识(同时有多重继承的功能)。

13、Java 的通信编程，编程题(或问答)，用 JAVA SOCKET 编程，读服务器几个字符，再写入本地显示？

答:Server 端程序：

```
package test;
import java.net.*;
import java.io.*;

public class Server
{
    private ServerSocket ss;
    private Socket socket;
    private BufferedReader in;
    private PrintWriter out;
    public Server()
```

```

{
try
{
ss=new ServerSocket(10000);
while(true)
{
socket = ss.accept();
String RemoteIP = socket.getInetAddress().getHostAddress();
String RemotePort = ":"+socket.getLocalPort();
System.out.println("A client come in!IP:"+RemoteIP+RemotePort);
in = new BufferedReader(new

InputStreamReader(socket.getInputStream()));
String line = in.readLine();
System.out.println("Cleint send is ." + line);
out = new PrintWriter(socket.getOutputStream(),true);
out.println("Your Message Received!");
out.close();
in.close();
socket.close();
}
}catch (IOException e)
{
out.println("wrong");
}
}
public static void main(String[] args)
{
new Server();
}
};

```

Client 端程序:

```

package test;
import java.io.*;
import java.net.*;

```

```

public class Client
{
Socket socket;
BufferedReader in;
PrintWriter out;
public Client()
{
try

```

```

{
System.out.println("Try to Connect to 127.0.0.1:10000");
socket = new Socket("127.0.0.1",10000);
System.out.println("The Server Connected!");
System.out.println("Please enter some Character:");
BufferedReader line = new BufferedReader(new

InputStreamReader(System.in));
out = new PrintWriter(socket.getOutputStream(),true);
out.println(line.readLine());
in = new BufferedReader(new InputStreamReader(socket.getInputStream()));
System.out.println(in.readLine());
out.close();
in.close();
socket.close();
}catch(IOException e)
{
out.println("Wrong");
}
}
public static void main(String[] args)
{
new Client();
}
};

```

14、用 JAVA 实现一种排序， JAVA 类实现序列化的方法(二种)? 如在 COLLECTION 框架中，实现比较要实现什么样的接口?

答:用插入法进行排序代码如下

```

package test;
import java.util.*;
class InsertSort
{
ArrayList al;
public InsertSort(int num,int mod)
{
al = new ArrayList(num);
Random rand = new Random();
System.out.println("The ArrayList Sort Before:");
for (int i=0;i<num ;i++ )
{
al.add(new Integer(Math.abs(rand.nextInt()) % mod + 1));
System.out.println("al["+i+"]="+al.get(i));
}
}
}

```

```

    }
    public void SortIt()
    {
        Integer tempInt;
        int MaxSize=1;
        for(int i=1;i<al.size();i++)
        {
            tempInt = (Integer)al.remove(i);
            if(tempInt.intValue()>= (Integer)al.get(MaxSize-1).intValue())
            {
                al.add(MaxSize,tempInt);
                MaxSize++;
                System.out.println(al.toString());
            } else {
                for (int j=0;j<MaxSize ;j++ )
                {
                    if

                    (((Integer)al.get(j)).intValue()>=tempInt.intValue())
                    {
                        al.add(j,tempInt);
                        MaxSize++;
                        System.out.println(al.toString());
                        break;
                    }
                }
            }
        }
        System.out.println("The ArrayList Sort After:");
        for(int i=0;i<al.size();i++)
        {
            System.out.println("al["+i+"]="+al.get(i));
        }
    }
    public static void main(String[] args)
    {
        InsertSort is = new InsertSort(10,100);
        is.SortIt();
    }
}

```

JAVA 类实现序列化方法是实现 `java.io.Serializable` 接口

Collection 框架中实现比较要实现 `Comparable` 接口和 `Comparator` 接口

15、编程：编写一个截取字符串的函数，输入为一个字符串和字节数，输出为按字节截取的字符

串。但是要保证汉字不被截半个，如"我 ABC"4，应该截为"我 AB"，输入"我 ABC 汉 DEF"，6，应该输出为"我 ABC"而不是"我 ABC+汉的半个"。

答：代码如下：

```
package test;

class SplitString
{
    String SplitStr;
    int SplitByte;
    public SplitString(String str,int bytes)
    {
        SplitStr=str;
        SplitByte=bytes;
        System.out.println("The String is:'"+SplitStr+"';SplitBytes="+SplitByte);
    }
    public void SplitIt()
    {
        int loopCount;

        loopCount=(SplitStr.length()%SplitByte==0)?(SplitStr.length()/SplitByte):(SplitStr.length()/Split
        Byte+1);
        System.out.println("Will Split into "+loopCount);
        for (int i=1;i<=loopCount ;i++ )
        {
            if (i==loopCount){

                System.out.println(SplitStr.substring((i-1)*SplitByte,SplitStr.length()));
            } else {

                System.out.println(SplitStr.substring((i-1)*SplitByte,(i*SplitByte)));
            }
        }
        }
    public static void main(String[] args)
    {
        SplitString ss = new SplitString("test 中 dd 文 dsaf 中男大 3443n 中国 43 中国人
        0ewldfls=103",4);
        ss.SplitIt();
    }
}
```

}

16、JAVA 多线程编程。用 JAVA 写一个多线程程序，如写四个线程，二个加 1，二个对一个变量减一，输出。

希望大家补上，谢谢

17、STRING 与 STRINGBUFFER 的区别。

答：STRING 的长度是不可变的，STRINGBUFFER 的长度是可变的。如果你对字符串中的内容经常进行操作，特别是内容要修改时，那么使用 StringBuffer，如果最后需要 String，那么使用 StringBuffer 的 toString()方法

6.11

最全的 java 面试题（六）

Jsp 方面

1、jsp 有哪些内置对象？作用分别是什么？

答：JSP 共有以下 9 种基本内置组件（可与 ASP 的 6 种内部组件相对应）：

request 用户端请求，此请求会包含来自 GET/POST 请求的参数

response 网页传回用户端的回应

pageContext 网页的属性是在这里管理

session 与请求有关的会话期

application servlet 正在执行的内容

out 用来传送回应的输出

config servlet 的构架部件

page JSP 网页本身

exception 针对错误网页，未捕捉的例外

2、jsp 有哪些动作？作用分别是什么？

答：JSP 共有以下 6 种基本动作

jsp:include：在页面被请求的时候引入一个文件。

jsp:useBean：寻找或者实例化一个 JavaBean。

jsp:setProperty：设置 JavaBean 的属性。

jsp:getProperty：输出某个 JavaBean 的属性。

jsp:forward：把请求转到一个新的页面。

jsp:plugin：根据浏览器类型为 Java 插件生成 OBJECT 或 EMBED 标记

3、JSP 中动态 INCLUDE 与静态 INCLUDE 的区别？

答：动态 INCLUDE 用 jsp:include 动作实现

<jsp:include page="included.jsp" flush="true" />它总是会检查所含文件中的变化，适用于包含动态页面，并且可以带参数

静态 INCLUDE 用 include 伪码实现，定不会检查所含文件的变化，适用于包含静态页面

<%@ include file="included.htm" %>

4、两种跳转方式分别是什么?有什么区别?

答: 有两种, 分别为:

```
<jsp:include page="included.jsp" flush="true">
```

```
<jsp:forward page="nextpage.jsp"/>
```

前者页面不会转向 include 所指的页面, 只是显示该页的结果, 主页面还是原来的页面。执行完后还会回来, 相当于函数调用。并且可以带参数。后者完全转向新页面, 不会再回来。相当于 go to 语句。

Servlet 方面

1、说一说 Servlet 的生命周期?

答: servlet 有良好的生存期的定义, 包括加载和实例化、初始化、处理请求以及服务结束。这个生存期由 javax.servlet.Servlet 接口的 init, service 和 destroy 方法表达。

2、Servlet 版本间(忘了问的是哪两个版本了)的不同?

希望大家补上, 谢谢

3、JAVA SERVLET API 中 forward() 与 redirect()的区别?

答: 前者仅是容器中控制权的转向, 在客户端浏览器地址栏中不会显示出转向后的地址; 后者则是完全的跳转, 浏览器将会得到跳转的地址, 并重新发送请求链接。这样, 从浏览器的地址栏中可以看到跳转后的链接地址。所以, 前者更加高效, 在前者可以满足需要时, 尽量使用 forward() 方法, 并且, 这样也有助于隐藏 实际的链接。在有些情况下, 比如, 需要跳转到一个其它服务器上的资源, 则必须使用 sendRedirect()方法。

4、Servlet 的基本架构

```
public class ServletName extends HttpServlet {  
    public void doPost(HttpServletRequest request, HttpServletResponse response) throws  
        ServletException, IOException {  
    }  
    public void doGet(HttpServletRequest request, HttpServletResponse response) throws  
        ServletException, IOException {  
    }  
}
```

Jdbc、Jdo 方面

1、可能会让你写一段 Jdbc 连 Oracle 的程序,并实现数据查询。

答:程序如下:

```
package hello.ant;  
import java.sql.*;  
public class jdbc  
{  
    String dbUrl="jdbc:oracle:thin:@127.0.0.1:1521:orcl";
```



```

String theUser="admin";
String thePw="manager";
Connection c=null;
Statement conn;
ResultSet rs=null;
public jdbc()
{
try {
Class.forName("oracle.jdbc.driver.OracleDriver").newInstance();
c = DriverManager.getConnection(dbUrl,theUser,thePw);
conn=c.createStatement();
} catch (Exception e) {
e.printStackTrace();
}
}
public boolean executeUpdate(String sql)
{
try
{
conn.executeUpdate(sql);
return true;
}
catch (SQLException e)
{
e.printStackTrace();
return false;
}
}
public ResultSet executeQuery(String sql)
{
rs=null;
try
{
rs=conn.executeQuery(sql);
}
catch (SQLException e)
{
e.printStackTrace();
}
return rs;
}
public void close()
{
try

```

```

{
conn.close();
c.close();
}
catch (Exception e)
{
e.printStackTrace();
}
}
public static void main(String[] args)
{
ResultSet rs;
jdbc conn = new jdbc();
rs=conn.executeQuery("select * from test");
try{
while (rs.next())
{
System.out.println(rs.getString("id"));
System.out.println(rs.getString("name"));
}
}catch(Exception e)
{
e.printStackTrace();
}
}
}

```

2、Class.forName 的作用?为什么要用?

答：调用该访问返回一个以字符串指定类名的类的对象。

3、Jdo 是什么?

答: JDO 是 Java 对象持久化的新的规范, 为java data object 的简称, 也是一个用于存取某种数据仓库中的对象的标准 API。JDO 提供了透明的对象存储, 因此对开发人员来说, 存储数据对象完全不需要额外的代码 (如 JDBC API 的使用)。这些繁琐的例行工作已经转移到 JDO 产品提供商身上, 使开发人员解脱出来, 从而集中时间和精力在业务逻辑上。另外, JDO 很灵活, 因为它可以在任何数据底层上运行。JDBC 只是面向关系数据库 (RDBMS) JDO 更通用, 提供到任何数据底层的存储功能, 比如关系数据库、文件、XML 以及对象数据库 (ODBMS) 等等, 使得应用可移植性更强。

4、在 ORACLE 大数据量下的分页解决方法。一般用截取 ID 方法, 还有是三层嵌套方法。

答: 一种分页方法

```

<%
int i=1;
int numPages=14;

```

```

String pages = request.getParameter("page");
int currentPage = 1;
currentPage=(pages==null)?(1): {Integer.parseInt(pages)}
sql = "select count(*) from tables";
ResultSet rs = DBLink.executeQuery(sql);
while(rs.next()) i = rs.getInt(1);
int intPageCount=1;
intPageCount=(i%numPages==0)?(i/numPages):(i/numPages+1);
int nextPage;
int upPage;
nextPage = currentPage+1;
if (nextPage>=intPageCount) nextPage=intPageCount;
upPage = currentPage-1;
if (upPage<=1) upPage=1;
rs.close();
sql="select * from tables";
rs=DBLink.executeQuery(sql);
i=0;
while((i<numPages*(currentPage-1))&&rs.next()){i++;}
%>
//输出内容
//输出翻页连接
合计:<%=currentPage%>/<%=intPageCount%><a href="List.jsp?page=1">第一页</a><a
href="List.jsp?page=<%=upPage%>">上一页</a>
<%
for(int j=1;j<=intPageCount;j++){
if(currentPage!=j){
%>
<a href="list.jsp?page=<%=j%>">[<%=j%>]</a>
<%
}else{
out.println(j);
}
}
%>
<a href="List.jsp?page=<%=nextPage%>">下一页</a><a href="List.jsp?page=<%=intPageCount%>"
>最后页

</a>

```

Xml 方面

1、xml 有哪些解析技术?区别是什么?

答:有 DOM,SAX,STAX 等

DOM: 处理大型文件时其性能下降的非常厉害。这个问题是由 DOM 的树结构所造成的, 这种结构占用的内存较多, 而且 DOM 必须在解析文件之前把整个文档装入内存, 适合对 XML 的随机访问
SAX: 不现于 DOM,SAX 是事件驱动型的 XML 解析方式。它顺序读取 XML 文件, 不需要一次全部装载整个文件。当遇到像文件开 头, 文档结束, 或者标签开头与标签结束时, 它会触发一个事件, 用户通过在其回调事件中写入处理代码来处理 XML 文件, 适合对 XML 的顺序访问

STAX:Streaming API for XML (StAX)

2、你在项目中用到了 xml 技术的哪些方面?如何实现的?

答: 用到了数据存贮, 信息配置两方面。在做数据交换平台时, 将不能数据源的数据组装成 XML 文件, 然后将 XML 文件压缩打包加密后通过网络传送给接收者, 接收 解密与解压缩后再同 XML 文件中还原相关信息进行处理。在做软件配置时, 利用 XML 可以很方便的进行, 软件的各种配置参数都存贮在 XML 文件中。

3、用 jdom 解析 xml 文件时如何解决中文问题?如何解析?

答:看如下代码,用编码方式加以解决

```
package test;
import java.io.*;
public class DOMTest
{
    private String inFile = "c:\\people.xml";
    private String outFile = "c:\\people.xml";
    public static void main(String args[])
    {
        new DOMTest();
    }
    public DOMTest()
    {
        try
        {
            javax.xml.parsers.DocumentBuilder builder =
                javax.xml.parsers.DocumentBuilderFactory.newInstance().newDocumentBuilder();
            org.w3c.dom.Document doc = builder.newDocument();
            org.w3c.dom.Element root = doc.createElement("老师");
            org.w3c.dom.Element wang = doc.createElement("王");
            org.w3c.dom.Element liu = doc.createElement("刘");
            wang.appendChild(doc.createTextNode("我是王老师"));
            root.appendChild(wang);
            doc.appendChild(root);
            javax.xml.transform.Transformer transformer =
```

```

javax.xml.transform.TransformerFactory.newInstance().newTransformer();
transformer.setOutputProperty(javax.xml.transform.OutputKeys.ENCODING, "gb2312");
transformer.setOutputProperty(javax.xml.transform.OutputKeys.INDENT, "yes");

```

```

transformer.transform(new javax.xml.transform.dom.DOMSource(doc),
new

```

```

javax.xml.transform.stream.StreamResult(outFile));
}
catch (Exception e)
{
System.out.println (e.getMessage());
}
}
}
}

```

4、编程用 JAVA 解析 XML 的方式.

答:用 SAX 方式解析 XML，XML 文件如下:

```

<?xml version="1.0" encoding="gb2312"?>
<person>
<name> 王小明 </name>
<college> 信息学院 </college>
<telephone>6258113</telephone>
<notes>男,1955 年生,博士, 95 年调入海南大学 </notes>
</person>

```

事件回调类 SAXHandler.java

```

import java.io.*;
import java.util.Hashtable;
import org.xml.sax.*;

public class SAXHandler extends HandlerBase
{
private Hashtable table = new Hashtable();
private String currentElement = null;
private String currentValue = null;
public void setTable(Hashtable table)
{
this.table = table;
}
public Hashtable getTable()
{
return table;
}
public void startElement(String tag, AttributeList attrs)

```

```

throws SAXException
{
currentElement = tag;
}
public void characters(char[] ch, int start, int length)
throws SAXException
{
currentValue = new String(ch, start, length);
}
public void endElement(String name) throws SAXException
{
if (currentElement.equals(name))
table.put(currentElement, currentValue);
}
}

```

JSP 内容显示源码,SaxXml.jsp:

```

<HTML>
<HEAD>
<TITLE>剖析 XML 文件 people.xml</TITLE>
</HEAD>
<BODY>
<%@ page errorPage="ErrPage.jsp"
contentType="text/html;charset=GB2312" %>
<%@ page import="java.io.*" %>
<%@ page import="java.util.Hashtable" %>
<%@ page import="org.w3c.dom.*" %>
<%@ page import="org.xml.sax.*" %>
<%@ page import="javax.xml.parsers.SAXParserFactory" %>
<%@ page import="javax.xml.parsers.SAXParser" %>
<%@ page import="SAXHandler" %>
<%
File file = new File("c:\people.xml");
FileReader reader = new FileReader(file);
Parser parser;
SAXParserFactory spf = SAXParserFactory.newInstance();
SAXParser sp = spf.newSAXParser();
SAXHandler handler = new SAXHandler();
sp.parse(new InputSource(reader), handler);
Hashtable hashTable = handler.getTable();
out.println("<TABLE BORDER=2<<CAPTION>教师信息表 </CAPTION>");
out.println("<TR><TD>姓名</TD>" + "<TD>" +
(String)hashTable.get(new String("name")) + "</TD></TR>");
out.println("<TR><TD>学院</TD>" + "<TD>" +
(String)hashTable.get(new String("college"))+"</TD></TR>");

```

```

out.println("<TR><TD> 电话</TD>" + "<TD>" +
(String)hashTable.get(new String("telephone")) + "</TD></TR>");
out.println("<TR><TD> 备注</TD>" + "<TD>" +
(String)hashTable.get(new String("notes")) + "</TD></TR>");
out.println("</TABLE>");
%>
</BODY>
</HTML>

```

EJB 方面

1、EJB2.0 有哪些内容?分别用在什么场合? EJB2.0 和 EJB1.1 的区别?

答: 规范内容包括 Bean 提供者, 应用程序装配者, EJB 容器, EJB 配置工具, EJB 服务提供者, 系统管理员。这里面, EJB 容器是 EJB 之所以能够运行的 核心。EJB 容器管理着 EJB 的创建, 撤消, 激活, 去活, 与数据库的连接等等重要的核心工作。JSP,Servlet,EJB,JNDI,JDBC, JMS.....

2、EJB 与 JAVA BEAN 的区别?

答:Java Bean 是可复用的组件, 对 Java Bean 并没有 严格的规范, 理论上讲, 任何一个 Java 类都可以是一个 Bean。但通常情况下, 由于 Java Bean 是被容器所创建 (如 Tomcat)的, 所以 Java Bean 应具有一个无参的构造器, 另外, 通常 Java Bean 还要实现 Serializable 接口用于实现 Bean 的持久性。Java Bean 实际上相当于微软 COM 模型中的本地进程内 COM 组件, 它是不能被跨进程访问的。Enterprise Java Bean 相当于 DCOM, 即分布式组件。它是基于 Java 的远程方法调用 (RMI) 技术的, 所以 EJB 可以被远程访问 (跨进程、跨计算机)。但 EJB 必须被布署在诸如 Webspere、WebLogic 这样的容器中, EJB 客户从不直接访问真正的 EJB 组件, 而是通过其容器访问。EJB 容器是 EJB 组件的代理, EJB 组件由容器所创建和管理。客户通过容器来访问真正的 EJB 组件。

3、EJB 的基本架构

答:一个 EJB 包括三个部分:

Remote Interface 接口的代码

```

package Beans;
import javax.ejb.EJBObject;
import java.rmi.RemoteException;
public interface Add extends EJBObject
{
//some method declare
}

```

Home Interface 接口的代码

```

package Beans;
import java.rmi.RemoteException;
import javax.ejb.CreateException;
import javax.ejb.EJBHome;
public interface AddHome extends EJBHome
{

```

```
//some method declare
}
EJB 类的代码
package Beans;
import java.rmi.RemoteException;
import javax.ejb.SessionBean;
import javax.ejb.SessionContext;
public class AddBean implements SessionBean
{
//some method declare
}
```

J2EE,MVC 方面

1、MVC 的各个部分都有那些技术来实现?如何实现?

答: MVC 是 Model—View—Controller 的简写。"Model" 代表的是应用的业务逻辑(通过 JavaBean, EJB 组件实现), "View" 是应用的表示面(由 JSP 页面产生), "Controller" 是提供应用的处理过程控制(一般是一个 Servlet), 通过这种设计 模型把应用逻辑, 处理过程和显示逻辑分成不同的组件实现。这些组件可以进行交互和重用。

2、应用服务器与 WEB SERVER 的区别?

希望大家补上, 谢谢

3、J2EE 是什么?

答: J2EE 是 Sun 公司提出的多层(multi-tiered), 分布式(distributed), 基于组件(component-based)的企业级应用模型(enterprise application model). 在这样的一个应用系统中, 可按照功能划分为不同的组件, 这些组件又可在不同计算机上, 并且处于相应的层次(tier)中。所属层次包括客户层(client tier)组件, web 层和组件, Business 层和组件, 企业信息系统 (EIS)层。

4、WEB SERVICE 名词解释。JAXP 开发包的介绍。JAXP、JAXM 的解释。SOAP、UDDI, WSDL 解释。

答: Web Service 描述语言 WSDL

SOAP 即简单对象访问协议(Simple Object Access Protocol), 它是用于交换 XML 编码信息的轻量级协议。

UDDI 的目的是为电子商务建立标准; UDDI 是一套基于 Web 的、分布式的、为 Web Service 提供的、信息注册中心的实现标准规范, 同时也包含一组使企业能将自身提供的 Web Service 注册, 以使别的企业能够发现的访问协议的实现标准。

5、BS 与 CS 的联系与区别。

希望大家补上, 谢谢

6、STRUTS 的应用(如 STRUTS 架构)

答：Struts 是采用 Java Servlet/JavaServer Pages 技术，开发 Web 应用程序的开放源码的 framework。采用 Struts 能开发出基于 MVC(Model-View-Controller)设计模式的应用构架。Struts 有如下的主要功能：

- 一.包含一个 controller servlet，能将用户的请求发送到相应的 Action 对象。
- 二.JSP 自由 tag 库，并且在 controller servlet 中提供关联支持，帮助开发人员创建交互式表单应用。
- 三.提供了一系列实用对象：XML 处理、通过 Java reflection APIs 自动处理 JavaBeans 属性、国际化的提示和消息。

设计模式方面

1、开发中都用到了那些设计模式?用在什么场合？

答：每个模式都描述了一个在我们的环境中不断出现的问题，然后描述了该问题的解决方案的核心。通过这种方式，你可以无数次地使用那些已有的解决方案，无需在重复相同的工作。主要用到了 MVC 的设计模式。用来开发 JSP/Servlet 或者 J2EE 的相关应用。简单工厂模式等。

2、UML 方面

答：标准建模语言 UML。用例图,静态图(包括类图、对象图和包图),行为图,交互图(顺序图,合作图),实现图,

JavaScript 方面

1、如何校验数字型？

```
var re=/^d{1,8}$|.d{1,2}$\s/;
var str=document.form1.all(i).value;
var r=str.match(re);
if (r==null)
{
sign=-4;
break;
}
else{
document.form1.all(i).value=parseFloat(str);
}
```

CORBA 方面

1、CORBA 是什么?用途是什么？

答：CORBA 标准是公共对象请求代理结构(Common Object Request Broker Architecture)，由对象管理组织 (Object Management Group，缩写为 OMG)标准化。它的组成是接口定义语言 (IDL)，语言绑定(binding;也译为联编)和允许应用程序间互操作的协议。其目的为：

用不同的程序设计语言书写

在不同的进程中运行

为不同的操作系统开发

LINUX 方面

1、LINUX 下线程，GDI 类的解释。

答：LINUX 实现的就是基于核心轻量级进程的"一对一"线程模型，一个线程实体对应一个核心轻量级进程，而线程之间的管理在核外函数库中实现。

GDI 类为图像设备编程接口类库

最全的 java 面试题（七）

一些著名的大公司面试题目往往很基础。

一、Java 基础知识

1. Java 有那些基本数据类型，String 是不是基本数据类型，他们有何区别。

Integer literals, Floating-point literals, character literals, Boolean literal, String literal.

String 不是基本数据类型

2. 字符串的操作：

写一个方法，实现字符串的反转，如：输入 abc，输出 cba

```
public static String reverse(String s){
    int length=s.length();
    StringBuffer result=new StringBuffer(length);
    for(int i=length-1;i>=0;i--)
        result.append(s.charAt(i));
    return result.toString();
}
```

写一个方法，实现字符串的替换，如：输入 bbbwlirbbb，输出 bbbhhtccc。

3. 数据类型之间的转换

如何将数值型字符转换为数字（Integer，Double）

如何将数字转换为字符

如何去小数点前两位，并四舍五入。

4. 日期和时间

如何取得年月日，小时分秒

```
Date dat=new Date();
```

```
dat.getYear();dat.getMonth();dat.getDay();dat.getHours();...
```

如何取得从 1970 年到现在的毫秒数

```
long now=dat.getTime();
```

如何获取某个日期是当月的最后一天

如何格式化日期

```
DateFormate df=DateFormate.getInstance();
```

```
df.Format(dat);
```

5. 数组和集合

6. 文件和目录（I/O）操作

如何列出某个目录下的所有文件

如何列出某个目录下的所有子目录

判断一个文件或目录是否存在

如何读写文件

7. Java 多态的实现（继承、重载、覆盖）

8. 编码转换，怎样实现将 GB2312 编码的字符串转换为 ISO-8859-1 编码的字符串。

9. Java 中访问数据库的步骤，Statement 和 PreparedStatement 之间的区别。

10. 找出下列代码可能存在的错误，并说明原因：

二、JSP&Servlet 技术

1. 描述 JSP 和 Servlet 的区别、共同点、各自应用的范围

2. 在 Web 开发中需要处理 HTML 标记时，应做什么样的处理，要筛选那些字符（<> & ""）

3. 在 JSP 中如何读取客户端的请求，如何访问 CGI 变量，如何确定某个 Jsp 文件的真实路径。

4. 描述 Cookie 和 Session 的作用，区别和各自的应用范围，Session 工作原理。

5. 列出 Jsp 中包含外部文件的方式，两者有何区别。

6. 说明 Jsp 中 errorPage 的作用，应用范围。

7. 介绍在 Jsp 中如何使用 JavaBeans。

8. 简单介绍 JSP 的标记库

9. Jsp 和 Servlet 中的请求转发分别如何实现。

三、J2EE 相关知识

1. 介绍 J2EE、J2SE、J2SE 的区别。

2. J2EE 是一种技术还是一种平台，他提供了那些技术。

3. 什么是 Application Server，它有什么功能和优点。

4. 简单介绍连接池的优点和原理。

5. Web.xml 的作用

四、其他

1. Web 安全性的考虑（表单验证、浏览器 Basic 方式的验证，应用程序的安全性，SSL，代码考虑）

2. 简单介绍您所了解的 MVC。

3. 简单介绍所了解的 XML。

4. 文档和编码规范

5. Java 中的分页、效率考虑。

6. 简单介绍您所了解的 struts。

找出以下程序错误。

```
Class Test{  
    private String par1;  
    private String par2;  
    Test(){  
    }  
  
    public static void main(String[] arg){
```

```

    int a ;
    if(a){
        System.out.println("par1="+par1);
    }else{
        System.out.println("par2=" + par2);
    }
}
}

```

=====

```

public class Test
{
    public static int a = 5;
    public static void main(String[] args)
    {
        Test test = new Test();
        test = null;
        System.out.println(test.a);
    }
}

```

答案是 5，a 为静态变量；不依赖对象，即使为空。

6.12

最全的 java 面试题（八）

J2EE 初学者需要理解的问题

一、J2EE 提出的背景

1、企业级应用框架的需求

在许多企业级应用中，例如数据库连接、邮件服务、事务处理等都是些通用企业需求模块，这些模块如果每次再开发中都由开发人员来完成的话，将会造成开发周期长和代码可*性差等问题。于是许多大公司开发了自己的通用模块服务。这些服务性的软件系统统称为中间件。

2、为了通用必须要提出规范，不然无法达到通用

在上面的需求基础之上，许多公司都开发了自己的中间件，但其与用户的沟通都各有不同，从而导致用户无法将各个公司不同的中间件组装在一块为自己服务。从而产生瓶颈。于是提出标准的概念。其实 J2EE 就是基于 JAVA 技术的一系列标准。

注：中间件的解释中间件处在操作系统和更高级应用程序之间。他充当的功能是：将应用程序运行环境与操作系统隔离，从而实现应用程序开发者不必为更多系统问题忧虑，而直接关注该应用程序在解决问题上的能力。我们后面说到的容器的概念就是中间件的一种。

二、相关名词解释

容器：充当中间件的角色

WEB 容器：给处于其中的应用程序组件（JSP，SERVLET）提供一个环境，使 JSP,SERVLET 直接更容器中的环境变量接口交互，不必关注其它系统问题。主要有 WEB 服务器来实现。例如：TOMCAT,WEBLOGIC,WEBSPPHERE 等。该容器提供的接口严格遵守 J2EE 规范中的 WEB APPLICATION 标准。我们把遵守以上标准的 WEB 服务器就叫做 J2EE 中的 WEB 容器。

EJB 容器：Enterprise java bean 容器。更具有行业领域特色。他提供给运行在其中的组件 EJB 各种管理功能。只要满足 J2EE 规范的 EJB 放入 该容器，马上就会被容器进行高效率的管理。并且可以通过现成的接口来获得系统级别的服务。例如邮件服务、事务管理。

WEB 容器和 EJB 容器在原理上是大体相同的，更多的区别是被隔离的外界环境。WEB 容器更多的是跟基于 HTTP 的请求打交道。而 EJB 容器不是。它是更多的跟数据库、其它服务打交道。但他们都是把与外界的交互实现从而减轻应用程序的负担。例如 SERVLET 不用关心 HTTP 的细节，直接引用环境变量 session,request,response 就行、EJB 不用关心数据库连接速度、各种事务控制，直接由容器来完成。

RMI/IIOP:远程方法调用/internet 对象请求中介协议，他们主要用于通过远程调用服务。例如，远程有一台计算机上运行一个程序，它提供股票分析服务，我们可以在本地计算机上实现对其直接调用。当然这是要通过一定的规范才能在异构的系统之间进行通信。RMI 是 JAVA 特有的。

JNDI:JAVA 命名目录服务。主要提供的功能是：提供一个目录系统，让其它各地的应用程序在其上面留下自己的索引，从而满足快速查找和定位分布式应用程序的功能。

JMS:JAVA 消息服务。主要实现各个应用程序之间的通讯。包括点对点和广播。

JAVAMAIL:JAVA 邮件服务。提供邮件的存储、传输功能。他是 JAVA 编程中实现邮件功能的核心。相当 MS 中的 EXCHANGE 开发包。

JTA：JAVA 事务服务。提供各种分布式事务服务。应用程序只需调用其提供的接口即可。

JAF:JAVA 安全认证框架。提供一些安全控制方面的框架。让开发者通过各种部署和自定义实现自己的个性安全控制策略。

EAI:企业应用集成。是一种概念，从而牵涉到好多技术。J2EE 技术是一种很好的集成实现。

三、J2EE 的优越性

1、基于 JAVA 技术，平台无关性表现突出

2、开放的标准，许多大型公司已经实现了对该规范支持的应用服务器。如 BEA ,IBM,ORACLE 等。

3、提供相当专业的通用软件服务。

4、提供了一个优秀的企业级应用程序框架，对快速高质量开发打下基础

四、现状

J2EE 是由 SUN 公司开发的一套企业级应用规范。现在最高版本是 1.4。支持 J2EE 的应用服务器有 IBM WEBSPPHERE APPLICATION SERVER, BEA WEBLOGIC SERVER, JBOSS, ORACLE APPLICATION SERVER, SUN ONE APPLICATION SERVER 等。

ff2004 发表评论于 2004-12-14 22:07:37 | 引用

学习 Java 的 30 个基本概念

Java 概述:

目前 Java 主要应用于中间件的开发(middleware)---处理客户机于服务器之间的通信技术,早期的实践证明,Java 不适合 pc 应用程序的开发,其发展逐渐变成在开发手持设备,互联网信息站,及车载计算机的开发. Java 于其他语言所不同的是程序运行时提供了平台的独立性,称许可以在 windows,solaris,linux 其他操作系统上使用完全相同的代码. Java 的语法与 C++语法类似,C++/C 程序员很容易掌握,而且 Java 是完全的彻底的面向对象的,其中提出了很好的 GC(Garbage Collector)垃圾处理机制,防止内存溢出.

Java 的白皮书为我们提出了 Java 语言的 11 个关键特性.

(1)Easy:Java 的语法比 C++的相对简单,另一个方面就是 Java 能使软件在很小的机器上运行,基础解释其和类库的支持的大小约为 40kb,增加基本的标准库和线程支持的内存需要增加 125kb.

(2)分布式:Java 带有很强大的 TCP/IP 协议族的例程库,Java 应用程序能够通过 URL 来穿过网络来访问远程对象,由于 servlet 机制的出现,使 Java 编程非常的高效,现在许多的大的 web server 都支持 servlet.

(3)OO:面向对象设计是把重点放在对象及对象的接口上的一个编程技术.其面向对象和 C++有很多不同,在与多重继承的处理及 Java 的原类模型.

(4)健壮特性:Java 采取了一个安全指针模型,能减小重写内存和数据崩溃的可能性。

(5)安全:Java 用来设计网路和分布系统,这带来了新的安全问题,Java 可以用来构建防病毒和防攻击的 System. 事实证明 Java 在防毒这一方面做的比较好.

(6)中立体系结构:Java 编译其生成体系结构中立的目标文件格式可以在很多处理器上执行,编译器产生的指令字节码(Javabyte code)实现此特性,此字节码可以在任何机器上解释执行.

(7)可移植性:Java 中对基本数据结构类型的大小和算法都有严格的规定所以可移植性很好.

(8)多线程:Java 处理多线程的过程很简单,Java 把多线程实现交给底下操作系统或线程程序完成. 所以多线程是 Java 作为服务器端开发语言的流行原因之一

(9)Applet 和 servlet:能够在网页上执行的程序叫 Applet,需要支持 Java 的浏览器很多,而 applet 支持动态的网页,这是很多其他语言所不能做到的.

基本概念:

1.OOP 中唯一关系的是对象的接口是什么,就像计算机的销售商她不管电源内部结构是怎样的,他只关系能否给你提供电就行了,也就是只要知道 **can or not** 而 不是 **how and why**.所有的程序是由一定的属性和行为对象组成的,不同的对象的访问通过函数调用来完成,对象间所有的交流都是通过方法调用,通过 对封装对象数据,很大程度上提高复用率.

2.OOP 中最重要的思想是类,类是模板是蓝图,从类中构造一个对象,即创建了这个类的一个实例(instance)

3.封装:就是把数据和行为结合起在一个包中)并对对象使用者隐藏数据的实现过程,一个对象中的数据叫他的实例字段(instance field)

4.通过扩展一个类来获得一个新类叫继承(inheritance),而所有的类都是由 Object 根超类扩展而得,根超类下文会做介绍.

5.对象的 3 个主要特性

behavior--- 说明这个对象能做什么.

state---当对象施加方法时对象的反映.

identity---与其他相似行为对象的区分标志.

每个对象有唯一的 identity 而这 3 者之间相互影响.

6.类之间的关系:

use-a :依赖关系

has-a :聚合关系

is-a :继承关系 --例:A 类继承了 B 类,此时 A 类不仅有了 B 类的方法,还有其自己的方法.(个性存在于共性中)

7.构造对象使用构造器:构造器的提出,构造器是一种特殊的方法,构造对象并对其初始化.

例:Data 类的构造器叫 Data

new Data()---构造一个新对象,且初始化当前时间.

Data happyday=new

Data()---把一个对象赋值给一个变量 happyday,从而使该对象能够多次使用,此处要声明的使变量与对象变量二者是不同的.new 返回的值是一个引用.

构造器特点:构造器可以有 0 个,一个或多个参数

构造器和类有相同的名字

一个类可以有多个构造器

构造器没有返回值

构造器总是和 new 运算符一起使用.

8.重载:当多个方法具有相同的名字而含有不同的参数时,便发生重载.编译器必须挑选出调用哪个方法.

9.包(package)Java 允许把一个或多个类收集在一起成为一组,称作包,以便于组织任务,标准 Java 库

分为许多包 java.lang java.util java.net 等,包是分层次的所有 java 包都在 java 和 javax 包层次内.

10. 继承思想:允许在已经存在的类的基础上构建新的类,当你继承一个已经存在的类时,那么你就复用了这个类的方法和字段,同时你可以在新类中添加新的方法和字段.

11. 扩展类:扩展类充分体现了 is-a 的继承关系. 形式为: class (子类) extends (基类).

12. 多态:在 java 中,对象变量是多态的.而 java 中不支持多重继承.

13. 动态绑定:调用对象方法的机制.

(1) 编译器检查对象声明的类型和方法名.

(2) 编译器检查方法调用的参数类型.

(3) 静态绑定:若方法类型为 private static final 编译器会准确知道该调用哪个方法.

(4) 当程序运行并且使用动态绑定来调用一个方法时,那么虚拟机必须调用 x 所指向的对象的实际类型相匹配的方法版本.

(5) 动态绑定:是很重要的特性,它能使程序变得可扩展而不需要重编译已存代码.

14. final 类:为防止他人从你的类上派生新类,此类是不可扩展的.

15. 动态调用比静态调用花费的时间要长,

16. 抽象类:规定一个或多个抽象方法的类本身必须定义为 abstract 例: public abstract String getDescription

17. Java 中的每一个类都是从 Object 类扩展而来的.

18. Object 类中的 equals 和 toString 方法. equals 用于测试一个对象是否同另一个对象相等. toString 返回一个代表该对象的字符串,几乎每一个类都会重载该方法,以便返回当前状态的正确表示.(toString 方法是一个很重要的方法)

19. 通用编程:任何类类型的所有值都可以同 `object` 类性的变量来代替.

20. 数组列表: `ArrayList` 动态数组列表,是一个类库,定义在 `java.util` 包中,可自动调节数组的大小.

21. `Class` 类 `Object` 类中的 `getClass` 方法返回 `Class` 类型的一个实例,程序启动时包含在 `main` 方法的类会被加载,虚拟机要加载他需要的所有类,每一个加载的类都要加载它需要的类.

22. `Class` 类为编写可动态操纵 `java` 代码的程序提供了强大的功能反射,这项功能为 `JavaBeans` 特别有用,使用反射 `Java` 能支持 `VB` 程序员习惯使用的工具. 能够分析类能力的程序叫反射器,`Java` 中提供此功能的包叫 `java.lang.reflect` 反射机制十分强大.

1. 在运行时分析类的能力.

2. 在运行时探索类的对象.

3. 实现通用数组操纵代码.

4. 提供方法对象.

而此机制主要针对是工具者而不是应用及程序.

反射机制中的最重要的部分是允许你检查类的结构.用到的 API 有:

`java.lang.reflect.Field` 返回字段.

`java.reflect.Method` 返回方法.

`java.lang.reflect.Constructor` 返回参数.

方法指针 `java` 没有方法指针,把一个方法的地址传给另一个方法,可以在后面调用它,而接口是更好的解决方案.

23. 接口(`Interface`)说明类该做什么而不指定如何去做,一个类可以实现一个或多个 `interface`.

24. 接口不是一个类,而是对符合接口要求的类的一套规范.若实现一个接口需要 2 个步骤:

1. 声明类需要实现的指定接口.

2.提供接口中的所有方法的定义.

声明一个类实现一个接口需要使用 `implements` 关键字 `class actionB implements Comparable` 其 `actionb` 需要提供 `CompareTo` 方法,接口不是类,不能用 `new` 实例化一个接口.

25.一个类只有一个超类,但一个类能实现多个接口.Java 中的一个重要接口 `Cloneable`

26.接口和回调.编程一个常用的模式是回调模式,在这种模式中你可以指定当一个特定时间发生时回调对象上的方法.例:`ActionListener` 接口监听.

类似的 API 有:`java.swing.JOptionPane`

`java.swing.Timer`

`java.awt.Toolkit`

27.对象 clone:clone 方法是 `object` 一个保护方法,这意味着你的代码不能简单的调用它.

28.内部类:一个内部类的定义是定义在另一个内部的类

原因是:1.一个内部类的对象能够访问创建它的对象的实现,包括私有数据

2.对于同一个包中的其他类来说,内部类能够隐藏起来.

3.匿名内部类可以很方便的定义回调.

4.使用内部类可以非常方便的编写事件驱动程序.

29.代理类(proxy):1.指定接口要求所有代码 2.`object` 类定义的所有的的方法(`toString equals`)

30.数据类型:Java 是强调类型的语言,每个变量都必须先申明它都类型,java 中总共有 8 个基本类型.4 种是整型,2 种是浮点型,一种是字符型,被用于 `Unicode` 编码中的字符,布尔型.

6.13

最全的 java 面试题（九）

CORBA 方面

1、CORBA 是什么?用途是什么?

答: CORBA 标准是公共对象请求代理结构(Common Object Request Broker Architecture), 由对象管理组织 (Object Management Group, 缩写为 OMG)标准化。它的组成是接口定义语言 (IDL), 语言绑定(binding:也译为联 编)和允许应用程序间互操作的协议。其目的为:

用不同的程序设计语言书写

在不同的进程中运行

为不同的操作系统开发

LINUX 方面

1、LINUX 下线程, GDI 类的解释。

答: LINUX 实现的就是基于核心轻量级进程的 "一对一"线程模型, 一个线程实体对应一个核心轻量级进程, 而线程之间的管理在核外函数库中实现。

GDI 类为图像设备编程接口类库。

JAVA 华为面试题

JAVA 方面

1 面向对象的特征有哪些方面

2 String 是最基本的数据类型吗?

3 int 和 Integer 有什么区别

4 String 和 StringBuffer 的区别

5 运行时异常与一般异常有何异同?

异常表示程序运行过程中可能出现的非正常状态, 运行时异常表示虚拟机的通常操作中可能遇到的异常, 是一种常见运行错误。java 编译器要求方法必须声明抛出可能发生的非运行时异常, 但是并不要求必须声明抛出未被捕获的运行时异常。

6 说出一些常用的类, 包,接口, 请各举 5 个

7 说出 ArrayList,Vector,LinkedList 的存储性能和特性

ArrayList 和 Vector 都是使用数组方式存储数据，此数组元素数大于实际存储的数据以便增加和插入元素，它们都允许直接按序号索引元素，但是插入元素要涉及数组元素移动等内存操作，所以索引数据快而插入数据慢，Vector 由于使用了 synchronized 方法（线程安全），通常性能上较 ArrayList 差，而 LinkedList 使用双向链表实现存储，按序号索引数据需要进行前向或后向遍历，但是插入数据时只需要记录本项的前后项即可，所以插入速度较快。

8 设计 4 个线程，其中两个线程每次对 j 增加 1，另外两个线程对 j 每次减少 1。写出程序。

以下程序使用内部类实现线程，对 j 增减的时候没有考虑顺序问题。

```
public class ThreadTest1{

    private int j;

    public static void main(String args[]){

        ThreadTest1 tt=new ThreadTest1();

        Inc inc=tt.new Inc();

        Dec dec=tt.new Dec();

        for(int i=0;i<2;i++){

            Thread t=new Thread(inc);

            t.start();

            t=new Thread(dec);

            t.start();

        }

    }

    private synchronized void inc(){

        j++;

        System.out.println(Thread.currentThread().getName()+"-inc:"+j);

    }
```

```
private synchronized void dec(){

j--;

System.out.println(Thread.currentThread().getName()+"-dec:"+j);

}
```

```
class Inc implements Runnable{

public void run(){

for(int i=0;i<100;i++){

inc();

}

}

}
```

```
class Dec implements Runnable{

public void run(){

for(int i=0;i<100;i++){

dec();

}

}

}

}
```

9. JSP 的内置对象及方法。

request request 表示 HttpServletRequest 对象。它包含了有关浏览器请求的信息，并且提供了几个

用于获取 cookie, header, 和 session 数据的有用的方法。

`response` `response` 表示 `HttpServletResponse` 对象，并提供了几个用于设置送回浏览器的响应的方法（如 cookies, 头信息等）

`out` `out` 对象是 `javax.jsp.JspWriter` 的一个实例，并提供了几个方法使你能用于向浏览器回送输出结果。

`pageContext` `pageContext` 表示一个 `javax.servlet.jsp.PageContext` 对象。它是用于方便存取各种范围的名字空间、servlet 相关的对象的 API，并且包装了通用的 servlet 相关功能的方法。

`session` `session` 表示一个请求的 `javax.servlet.http.HttpSession` 对象。Session 可以存贮用户的状态信息

`application` `application` 表示一个 `javax.servelet.ServletContext` 对象。这有助于查找有关 servlet 引擎和 servlet 环境的信息

`config` `config` 表示一个 `javax.servlet.ServletConfig` 对象。该对象用于存取 servlet 实例的初始化参数。

`page` `page` 表示从该页面产生的一个 servlet 实例

10. 用 socket 通讯写出客户端和服务端端的通讯，要求客户发送数据后能够回显相同的数据。

参见课程中 socket 通讯例子。

11 说出 Servlet 的生命周期，并说出 Servlet 和 CGI 的区别。

Servlet 被服务器实例化后，容器运行其 `init` 方法，请求到达时运行其 `service` 方法，`service` 方法自动派遣运行与请求对应的 `doXXX` 方法（`doGet`, `doPost`）等，当服务器决定将实例销毁的时候调用其 `destroy` 方法。

与 `cgi` 的区别在于 `servlet` 处于服务器进程中，它通过多线程方式运行其 `service` 方法，一个实例可以服务于多个请求，并且其实例一般不会销毁，而 `CGI` 对每个请求都产生新的进程，服务完成后就销毁，所以效率上低于 `servlet`。

12. EJB 是基于哪些技术实现的？并说出 `SessionBean` 和 `EntityBean` 的区别，`StatefulBean` 和 `StatelessBean` 的区别。

13. EJB 包括（`SessionBean`, `EntityBean`）说出他们的生命周期，及如何管理事务的？

14. 说出数据连接池的工作机制是什么？

15 同步和异步有和异同，在什么情况下分别使用他们？举例说明。

16 应用服务器有那些？

17 你所知道的集合类都有哪些？主要方法？

18 给你一个:驱动程序 A,数据源名称为 B,用户名称为 C,密码为 D,数据库表为 T，请用 JDBC 检索出表 T 的所有数据。

19. 说出在 JSP 页面里是怎么分页的？

页面需要保存以下参数：

总行数：根据 sql 语句得到总行数

每页显示行数：设定值

当前页数：请求参数

页面根据当前页数和每页行数计算出当前页第一行行数，定位结果集到此行，对结果集取出每页显示行数的行即可。

数据库方面：

1. 存储过程和函数的区别

存储过程是用户定义的一系列 sql 语句的集合，涉及特定表或其它对象的任务，用户可以调用存储过程，而函数通常是数据库已定义的方法，它接收参数并返回某种类型的值并且不涉及特定用户表。

2. 事务是什么？

事务是作为一个逻辑单元执行的一系列操作，一个逻辑工作单元必须有四个属性，称为 ACID（原子性、一致性、隔离性和持久性）属性，只有这样才能成为一个事务：

原子性

事务必须是原子工作单元；对于其数据修改，要么全都执行，要么全都不执行。

一致性

事务在完成时，必须使所有的数据都保持一致状态。在相关数据库中，所有规则都必须应用于事务的修改，以保持所有数据的完整性。事务结束时，所有的内部数据结构（如 B 树索引或双向链表）都必须是正确的。

隔离性

由 并发事务所作的修改必须与任何其它并发事务所作的修改隔离。事务查看数据时数据所处的状态，要么是另一并发事务修改它之前的状态，要么是另一事务修改它之后的状态，事务不会查看中间状态的数据。这称为可串行性，因为它能够重新装载起始数据，并且重播一系列事务，以使数据结束时的状态与原始事务执行的状态相同。

持久性

事务完成之后，它对于系统的影响是永久性的。该修改即使出现系统故障也将一直保持。

3. 游标的作用？如何知道游标已经到了最后？

游标用于定位结果集的行，通过判断全局变量 @@FETCH_STATUS 可以判断是否到了最后，通常此变量不等于 0 表示出错或到了最后。

4. 触发器分为事前触发和事后触发，这两种触发有和区别。语句级触发和行级触发有何区别。

事前触发器运行于触发事件发生之前，而事后触发器运行于触发事件发生之后。通常事前触发器可以获取事件之前和新的字段值。

语句级触发器可以在语句执行前或后执行，而行级触发在触发器所影响的每一行触发一次。

最全的java 面试题（十）

中远面试题

- 1、面向对象的三个基本特征
- 2、方法重载和方法重写的概念和区别
- 3、接口和内部类、抽象类的特性
- 4、文件读写的基本类
- **5、串行化的注意事项以及如何实现串行化
- 6、线程的基本概念、线程的基本状态以及状态之间的关系
- 7、线程的同步、如何实现线程的同步
- 8、几种常用的数据结构及内部实现原理。
- 9、Socket 通信(TCP、UDP 区别及 Java 实现方式)
- **10、Java 的事件委托机制和垃圾回收机制
- 11、JDBC 调用数据库的基本步骤
- **12、解析 XML 文件的几种方式和区别
- 13、Java 四种基本权限的定义
- 14、Java 的国际化

二、JSP

- 1、至少要能说出 7 个隐含对象以及他们的区别
- ** 2、forward 和 redirect 的区别
- 3、JSP 的常用指令

三、servlet

- 1、什么情况下调用 doGet()和 doPost()?

2、servlet 的 init()方法和 service()方法的区别

3、servlet 的生命周期

4、如何现实 servlet 的单线程模式

5、servlet 的配置

6、四种会话跟踪技术

四、EJB

**1、EJB 容器提供的服务

主要提供声明周期管理、代码产生、持续性管理、安全、事务管理、锁和并发行管理等服务。

2、EJB 的角色和三个对象

EJB 角色主要包括 Bean 开发者应用组装者 部署者系统管理员 EJB 容器提供者 EJB 服务器提供者

三个对象是 Remote (Local) 接口、Home (LocalHome) 接口，Bean 类

2、EJB 的几种类型

会话 (Session) Bean，实体 (Entity) Bean 消息驱动的 (Message Driven) Bean

会话 Bean 又可分为有状态 (Stateful) 和无状态 (Stateless) 两种

实体 Bean 可分为 Bean 管理的持续性 (BMP) 和容器管理的持续性 (CMP) 两种

3、bean 实例的生命周期

对于 Stateless Session Bean、Entity Bean、Message Driven Bean 一般存在缓冲池管理，而对于 Entity Bean 和 Statefull Session Bean 存在 Cache 管理，通常包含创建实例，设置上下文、创建 EJB Object (create)、业务方法调用、remove 等过程，对于存在缓冲池管理的 Bean，在 create 之后实例并不从内存清除，而是采用缓冲池调度机制不断重用实例，而对于存在 Cache 管理的 Bean 则通过激活和去激活机制保持 Bean 的状态并限制内存中实例数量。

4、激活机制

以 Statefull Session Bean 为例：其 Cache 大小决定了内存中可以同时存在的 Bean 实例的数量，根据 MRU 或 NRU 算法，实例在激活和去激活状态之间迁移，激活机制是当客户端调用某个 EJB 实例业务方法时，如果对应 EJB Object 发现自己没有绑定对应的 Bean 实例则从其去激活 Bean 存储中(通过序列化机制存储实例)回复(激活)此实例。状态变迁前会调用对应的 ejbActive

和 `ejbPassivate` 方法。

5、remote 接口和 home 接口主要作用

`remote` 接口定义了业务方法，用于 EJB 客户端调用业务方法

`home` 接口是 EJB 工厂用于创建和移除查找 EJB 实例

6、客户端调用 EJB 对象的几个基本步骤

一、设置 JNDI 服务工厂以及 JNDI 服务地址系统属性

二、查找 Home 接口

三、从 Home 接口调用 Create 方法创建 Remote 接口

四、通过 Remote 接口调用其业务方法

五、数据库

1、存储过程的编写

2、基本的 SQL 语句

六、weblogic

1、如何给 weblogic 指定大小的内存？

在启动 Weblogic 的脚本中（位于所在 Domain 对应服务器目录下的 `startServerName`），增加 `set MEM_ARGS=-Xms32m -Xmx200m`，可以调整最小内存为 32M，最大 200M

2、如何设定的 weblogic 的热启动模式(开发模式)与产品发布模式？

可以在管理控制台中修改对应服务器的启动模式为开发或产品模式之一。或者修改服务的启动文件或者 `commenv` 文件，增加 `set PRODUCTION_MODE=true`。

3、如何启动时不需输入用户名与密码？

修改服务启动文件，增加 `WLS_USER` 和 `WLS_PW` 项。也可以在 `boot.properties` 文件中增加加密过的用户名和密码。

4、在 weblogic 管理台中对一个应用域(或者说是一个网站,Domain)进行 jms 及 ejb 或连接池等相关信息进行配置后,实际保存在什么文件中？

保存在此 Domain 的 `config.xml` 文件中，它是服务器的核心配置文件。

5、说说 weblogic 中一个 Domain 的缺省目录结构?比如要将一个简单的 helloWorld.jsp 放入何目录下,然的在浏览器上就可打入 <http://主机:端口号/helloworld.jsp> 就可以看到运行结果了?又比如这其中用到了一个自己写的 javaBean 该如何办?

Domain 目录\服务器目录\applications, 将应用目录放在此目录下将可以作为应用访问, 如果是 Web 应用, 应用目录需要满足 Web 应用目录要求, jsp 文件可以直接放在应用目录中, Javabeen 需要放在应用目录的 WEB-INF 目录的 classes 目录中, 设置服务器的缺省应用将可以实现在浏览器上无需输入应用名。

6、如何查看在 weblogic 中已经发布的 EJB?

可以使用管理控制台, 在它的 Deployment 中可以查看所有已发布的 EJB

7、如何在 weblogic 中进行 ssl 配置与客户端的认证配置或说说 j2ee(标准)进行 ssl 的配置

缺省安装中使用 DemoIdentity.jks 和 DemoTrust.jks KeyStore 实现 SSL, 需要配置服务器使用 Enable SSL, 配置其端口, 在产品模式下需要从 CA 获取私有密钥和数字证书, 创建 identity 和 trust keystore, 装载获得的密钥和数字证书。可以配置此 SSL 连接是单向还是双向的。

8、在 weblogic 中发布 ejb 需涉及到哪些配置文件

不同类型的 EJB 涉及的配置文件不同, 都涉及到的配置文件包括 ejb-jar.xml, weblogic-ejb-jar.xml CMP 实体 Bean 一般还需要 weblogic-cmp-rdbms-jar.xml

9、EJB 需直接实现它的业务接口或 Home 接口吗, 请简述理由。

远程接口和 Home 接口不需要直接实现, 他们的实现代码是由服务器产生的, 程序运行中对应实现类会作为对应接口类型的实例被使用。

10、说说在 weblogic 中开发消息 Bean 时的 persistent 与 non-persistent 的差别

persistent 方式的 MDB 可以保证消息传递的可靠性, 也就是如果 EJB 容器出现问题而 JMS 服务器依然会将消息在此 MDB 可用的时候发送过来, 而 non-persistent 方式的消息将被丢弃。

11、说说你所熟悉或听说过的 j2ee 中的几种常用模式?及对设计模式的一些看法

Session Facade Pattern: 使用 SessionBean 访问 EntityBean

Message Facade Pattern: 实现异步调用

EJB Command Pattern: 使用 Command JavaBeans 取代 SessionBean, 实现轻量级访问

Data Transfer Object Factory: 通过 DTO Factory 简化 EntityBean 数据提供特性

Generic Attribute Access: 通过 AttributeAccess 接口简化 EntityBean 数据提供特性

Business Interface: 通过远程（本地）接口和 Bean 类实现相同接口规范业务逻辑一致性

E J B 架构的设计好坏将直接影响系统的性能、可扩展性、可维护性、组件可重用性及开发效率。项目越复杂，项目队伍越庞大则越能体现良好设计的重要性

6.15

最全的java 面试题（十一）

基础知识:

1.C++或 Java 中的异常处理机制的简单原理和应用。

当 JAVA 程序违反了 JAVA 的语义规则时，JAVA 虚拟机就会将发生的错误表示为一个异常。违反语义规则包括 2 种情况。一种是 JAVA 类库内置的语义检查。例如数组下标越界,会引发 `IndexOutOfBoundsException`;访问 null 的对象时会引发 `NullPointerException`。另一种情况就是 JAVA 允许程序员扩展这种语义检查，程序员可以创建自己的异常，并自由选择何时用 `throw` 关键字引发异常。所有的异常都是 `java.lang.Throwable` 的子类。

2. Java 的接口和 C++的虚类的相同和不同处。

由于 Java 不支持多继承，而有可能某个类或对象要使用分别在几个类或对象里面的方法或属性，现有的单继承机制就不能满足要求。与继承相比，接口有更高的灵活性，因为接口中没有任何现代码。当一个类实现了接口以后，该类要实现接口里面所有的方法和属性，并且接口里面的属性在默认状态下面都是 `public static`,所有方法默认情况下是 `public`。一个类可以实现多个接口。

3. 垃圾回收的优点和原理。并考虑 2 种回收机制。

Java 语言中一个显著的特点就是引入了垃圾回收机制，使 c++程序员最头疼的内存管理的问题迎刃而解，它使得 Java 程序员在编写程序的时候不再需要考虑内存管理。由于有个垃圾回收机制，Java 中的对象不再有“作用域”的概念，只有对象的引用才有“作用域”。垃圾回收可以有效的防止内存泄露，有效的使用可以使用的内存。垃圾回收器通常是作为一个单独的低级别的线程运行，不可预知的情况下对内存堆中已经死亡的或者长时间没有使用的对象进行清楚和回收，程序员不能实时的调用垃圾回收器对某个对象或所有对象进行垃圾回收。回收机制有分代复制垃圾回收和标记垃圾回收，增量垃圾回收。

4. 请说出你所知道的线程同步的方法。

`wait()`:使一个线程处于等待状态，并且释放所持有的对象的 `lock`。

`sleep()`:使一个正在运行的线程处于睡眠状态，是一个静态方法，调用此方法要捕捉 `InterruptedException` 异常。

`notify()`:唤醒一个处于等待状态的线程，注意的是在调用此方法的时候，并不能确切的唤醒某一个等待状态的线程，而是由 JVM 确定唤醒哪个线程，而且不是按优先级。

`Allnotity()`:唤醒所有处于等待状态的线程，注意并不是给所有唤醒线程一个对象的锁，而是让它们竞争。

5. 请讲一讲析构函数和虚函数的用法和作用。

6. Error 与 Exception 有什么区别？

Error 表示系统级的错误和程序不必处理的异常，

Exception 表示需要捕捉或者需要程序进行处理的异常。

7. 在 java 中一个类被声明为 `final` 类型，表示什么意思？

表示该类不能被继承，是顶级类。

8. 描述一下你最常用的编程风格。

9. heap 和 stack 有什么区别。

栈是一种线性集合，其添加和删除元素的操作应在同一段完成。栈按照后进先出的方式进行处理。堆是栈的一个组成元素

10. 如果系统要使用超大整数（超过 long 长度范围），请你设计一个数据结构来存储这种超大型数字以及设计一种算法来实现超大整数加法运算）。

```
public class BigInt()
{
    int[] ArrOne = new ArrOne[1000];
    String intString="";
    public int[] Arr(String s)
    {
        intString = s;
        for(int i=0;i<ArrOne.length;i++)
        {
```

11. 如果要设计一个图形系统，请你设计基本的图形元件(Point,Line,Rectangle,Triangle)的简单实现

12. 谈谈 final, finally, finalize 的区别。

final—修饰符（关键字）如果一个类被声明为 **final**，意味着它不能再派生出新的子类，不能作为父类被继承。因此一个类不能既被声明为 **abstract** 的，又被声明为 **final** 的。将变量或方法声明为 **final**，可以保证它们在使用中不被改变。被声明为 **final** 的变量必须在声明时给一定初值，而在以后的引用中只能读取，不可修改。被声明为 **final** 的方法也同样只能使用，不能重载。

finally—再异常处理时提供 **finally** 块来执行任何清除操作。如果抛出一个异常，那么相匹配的 **catch** 子句就会执行，然后控制就会进入 **finally** 块（如果有的话）。

finalize—方法名。Java 技术允许使用 **finalize()** 方法在垃圾收集器将对象从内存中清除出去之前做必要的清理工作。这个方法是由垃圾收集器在确定这个对象没有被引用时对这个对象调用的。它是在 **Object** 类中定义的，因此所有的类都继承了它。子类覆盖 **finalize()** 方法以整理系统资源或者执行其他清理工作。**finalize()** 方法是在垃圾收集器删除对象之前对这个对象调用的。

13. Anonymous Inner Class (匿名内部类) 是否可以 extends(继承) 其它类，是否可以 implements(实现) interface(接口)?

匿名的内部类是没有名字的内部类。不能 extends(继承) 其它类，但一个内部类可以作为一个接口，由另一个内部类实现。

14. Static Nested Class 和 Inner Class 的不同，说得越多越好 (面试题有的很笼统)。

Nested Class（一般是 C++ 的说法），**Inner Class**（一般是 JAVA 的说法）。Java 内部类与 C++ 嵌套类最大的不同就在于是否有指向外部的引用上。具体可见 <http://www.frontfree.net/articles/services/view.asp?id=704&page=1>

注：静态内部类（**Inner Class**）意味着 1 创建一个 **static** 内部类的对象，不需要一个外部类对象，2 不能从一个 **static** 内部类的一个对象访问一个外部类对象

第四，&和&&的区别。

&是位运算符。&&是布尔逻辑运算符。

15. HashMap 和 Hashtable 的区别。

都属于 Map 接口的类，实现了将唯一键映射到特定的值上。

HashMap 类没有分类或者排序。它允许一个 null 键和多个 null 值。

Hashtable 类似于 HashMap，但是不允许 null 键和 null 值。它也比 HashMap 慢，因为它是同步的。

16, Collection 和 Collections 的区别。

Collections 是个 java.util 下的类，它包含有各种有关集合操作的静态方法。

Collection 是个 java.util 下的接口，它是各种集合结构的父接口。

17, 什么时候用 assert。

断言是一个包含布尔表达式的语句，在执行这个语句时假定该表达式为 true。如果表达式计算为 false，那么系统会报告一个 AssertionError。它用于调试目的：

```
assert(a > 0); // throws an AssertionError if a <= 0
```

断言可以有两种形式：

```
assert Expression1 ;
```

```
assert Expression1 : Expression2 ;
```

Expression1 应该总是产生一个布尔值。

Expression2 可以是得出一个值的任意表达式。这个值用于生成显示更多调试信息的 String 消息。

断言在默认情况下是禁用的。要在编译时启用断言，需要使用 source 1.4 标记：

```
javac -source 1.4 Test.java
```

要在运行时启用断言，可使用 -enableassertions 或者 -ea 标记。

要在运行时选择禁用断言，可使用 -da 或者 -disableassertions 标记。

要系统类中启用断言，可使用 -esa 或者 -dsa 标记。还可以在包的基础上启用或者禁用断言。

可以在预计正常情况下不会到达的任何位置上放置断言。断言可以用于验证传递给私有方法的参数。不过，断言不应该用于验证传递给公有方法的参数，因为不管是否启用了断言，公有方法都必须检查其参数。不过，既可以在公有方法中，也可以在非公有方法中利用断言测试后置条件。另外，断言不应该以任何方式改变程序的状态。

18, GC 是什么？为什么要有 GC？(基础)。

GC 是垃圾收集器。Java 程序员不用担心内存管理，因为垃圾收集器会自动进行管理。要请求垃圾收集，可以调用下面的方法之一：

```
System.gc()
```

```
Runtime.getRuntime().gc()
```

19, String s = new String("xyz"); 创建了几个 String Object?

两个对象，一个是 "xyz"，一个是指向 "xyz" 的引用对象 s。

20, Math.round(11.5) 等于多少？Math.round(-11.5) 等于多少？

Math.round(11.5) 返回 (long) 12，Math.round(-11.5) 返回 (long) -11；

21, short s1 = 1; s1 = s1 + 1; 有什么错？short s1 = 1; s1 += 1; 有什么错？

short s1 = 1; s1 = s1 + 1; 有错，s1 是 short 型，s1+1 是 int 型，不能显式转化为 short 型。可修改为 s1 = (short)(s1 + 1)。short s1 = 1; s1 += 1 正确。

22, sleep() 和 wait() 有什么区别？搞线程的最爱

sleep() 方法是使线程停止一段时间的方法。在 sleep 时间间隔期满后，线程不一定立即恢复执行。这是因为在那个时刻，其它线程可能正在运行而且没有被调度为放弃执行，除非 (a) "醒来" 的线程具有更高的优先级 (b) 正在运行的线程因为其它原因而阻塞。

wait() 是线程交互时，如果线程对一个同步对象 x 发出一个 wait() 调用，该线程会暂停执行，被调对象进入等待状态，直到被唤醒或等待时间到。

23, Java 有没有 goto?

Goto—java 中的保留字, 现在没有在 java 中使用。

24, 数组有没有 length()这个方法? String 有没有 length()这个方法?

数组没有 length()这个方法, 有 length 的属性。

String 有 length()这个方法。

25, Overload 和 Override 的区别。Overloaded 的方法是否可以改变返回值的类型?

方法的重写 Overriding 和重载 Overloading 是 Java 多态性的不同表现。重写 Overriding 是父类与子类之间多态性的一种表现, 重载 Overloading 是一个类中多态性的一种表现。如果在子类中定义某方法与其父类有相同的名称和参数, 我们说该方法被重写 (Overriding)。子类的对象使用这个方法时, 将调用子类中的定义, 对它而言, 父类中的定义如同被"屏蔽"了。如果在一个类中定义了多个同名的方法, 它们或有不同的参数个数或有不同的参数类型, 则称为方法的重载(Overloading)。Overloaded 的方法是可以改变返回值的类型。

26, Set 里的元素是不能重复的, 那么用什么方法来区分重复与否呢? 是用 == 还是 equals()? 它们有何区别?

Set 里的元素是不能重复的, 那么用 iterator()方法来区分重复与否。equals()是判断两个 Set 是否相等。

equals()和 == 方法决定引用值是否指向同一对象 equals()在类中被覆盖, 为的是当两个分离的对象的内容和类型相配的话, 返回真值。

27, 给我一个你最常见到的 runtime exception。

ArithmeticException, ArrayStoreException, BufferOverflowException, BufferUnderflowException, CannotRedoException, CannotUndoException, ClassCastException, CMMException, ConcurrentModificationException, DOMException, EmptyStackException, IllegalArgumentException, IllegalMonitorStateException, IllegalPathStateException, IllegalStateException, ImagingOpException, IndexOutOfBoundsException, MissingResourceException, NegativeArraySizeException, NoSuchElementException, NullPointerException, ProfileDataException, ProviderException, RasterFormatException, SecurityException, SystemException, UndeclaredThrowableException, UnmodifiableSetException, UnsupportedOperationException

28, error 和 exception 有什么区别?

error 表示恢复不是不可能但很困难的情况下的一种严重问题。比如说内存溢出。不可能指望程序能处理这样的情况。

exception 表示一种设计或实现问题。也就是说, 它表示如果程序运行正常, 从不会发生的情况。

29, List, Set, Map 是否继承自 Collection 接口?

List, Set 是

Map 不是

30, abstract class 和 interface 有什么区别?

声明方法的存在而不去实现它的类被叫做抽象类 (abstract class), 它用于要创建一个体现某些基本行为的类, 并为该类声明方法, 但不能在该类中实现该方法的情况。不能创建 abstract 类的实例。然而可以创建一个变量, 其类型是一个抽象类, 并让它指向具体子类的一个实例。不能有抽象构造函数或抽象静态方法。Abstract 类的子类为它们父类中的所有抽象方法提供实现, 否则它们也是抽象类。取而代之, 在子类中实现该方法。知道其行为的其它类可以在类中实现这些方法。

接口 (interface) 是抽象类的变体。在接口中, 所有方法都是抽象的。多继承性可通过实现

这样的接口而获得。接口中的所有方法都是抽象的，没有一个有程序体。接口只可以定义 `static final` 成员变量。接口的实现与子类相似，除了该实现类不能从接口定义中继承行为。当类实现特殊接口时，它定义（即将程序体给予）所有这种接口的方法。然后，它可以在实现了该接口的类的任何对象上调用接口的方法。由于有抽象类，它允许使用接口名作为引用变量的类型。通常的动态联编将生效。引用可以转换到接口类型或从接口类型转换，`instanceof` 运算符可以用来决定某对象的类是否实现了接口。

31, `abstract` 的 `method` 是否可同时是 `static`, 是否可同时是 `native`, 是否可同时是 `synchronized`?
都不能

32, 接口是否可继承接口? 抽象类是否可实现 (`implements`) 接口? 抽象类是否可继承实体类 (`concrete class`)?

接口可以继承接口。抽象类可以实现 (`implements`) 接口，抽象类是否可继承实体类，但前提是实体类必须有明确的构造函数。

33, 启动一个线程是用 `run()` 还是 `start()`?

启动一个线程是调用 `start()` 方法，使线程所代表的虚拟处理机处于可运行状态，这意味着它可以由 JVM 调度并执行。这并不意味着线程就会立即运行。`run()` 方法可以产生必须退出的标志来停止一个线程。

34, 构造器 `Constructor` 是否可被 `override`?

构造器 `Constructor` 不能被继承，因此不能重写 `Overriding`，但可以被重载 `Overloading`。

35, 是否可以继承 `String` 类?

`String` 类是 `final` 类故不可以继承。

36, 当一个线程进入一个对象的一个 `synchronized` 方法后，其它线程是否可进入此对象的其它方法?

不能，一个对象的一个 `synchronized` 方法只能由一个线程访问。

37, `try {}` 里有一个 `return` 语句，那么紧跟在这个 `try` 后的 `finally {}` 里的 `code` 会不会被执行，什么时候被执行，在 `return` 前还是后?

会执行，在 `return` 前执行。

38, 编程题：用最有效率的方法算出 2 乘以 8 等于几?

有 C 背景的程序员特别喜欢问这种问题。

`2 << 3`

39, 两个对象值相同 (`x.equals(y) == true`)，但却可有不同的 `hash code`，这句话对不对?

不对，有相同的 `hash code`。

40, 当一个对象被当作参数传递到一个方法后，此方法可改变这个对象的属性，并可返回变化后的结果，那么这里到底是值传递还是引用传递?

是值传递。Java 编程语言只由值传递参数。当一个对象实例作为一个参数被传递到方法中时，参数的值就是对该对象的引用。对象的内容可以在被调用的方法中改变，但对象的引用是永远不会改变的。

41, `switch` 是否能作用在 `byte` 上，是否能作用在 `long` 上，是否能作用在 `String` 上?

`switch (expr1)` 中，`expr1` 是一个整数表达式。因此传递给 `switch` 和 `case` 语句的参数应该是 `int`、`short`、`char` 或者 `byte`。`long`、`string` 都不能作用于 `switch`。

42, 编程题：写一个 `Singleton` 出来。

`Singleton` 模式主要作用是保证在 Java 应用程序中，一个类 `Class` 只有一个实例存在。

一般 `Singleton` 模式通常有几种形式：

第一种形式：定义一个类，它的构造函数为 `private` 的，它有一个 `static` 的 `private` 的该类变量，在类初始化时实例化，通过一个 `public` 的 `getInstance` 方法获取对它的引用,继而调用其中的方法。

```
public class Singleton {
    private Singleton(){}
    //在自己内部定义自己一个实例，是不是很奇怪？
    //注意这是 private 只供内部调用
    private static Singleton instance = new Singleton();
    //这里提供了一个供外部访问本 class 的静态方法，可以直接访问
    public static Singleton getInstance() {
        return instance;
    }
}
```

第二种形式：

```
public class Singleton {
    private static Singleton instance = null;
    public static synchronized Singleton getInstance() {
        //这个方法比上面有所改进，不用每次都进行生成对象，只是第一次
        //使用时生成实例，提高了效率！
        if (instance==null)
            instance=new Singleton();
        return instance;
    }
}
```

其他形式：

定义一个类，它的构造函数为 `private` 的，所有方法为 `static` 的。

一般认为第一种形式要更加安全些

Hashtable 和 HashMap

Hashtable 继承自 Dictionary 类，而 HashMap 是 Java1.2 引进的 Map interface 的一个实现

HashMap 允许将 null 作为一个 entry 的 key 或者 value，而 Hashtable 不允许

还有就是，HashMap 把 Hashtable 的 contains 方法去掉了，改成 containsvalue 和 containsKey。

因为 contains 方法容易让人引起误解。

最大的不同是，Hashtable 的方法是 Synchronize 的，而 HashMap 不是，在多个线程访问 Hashtable 时，不需要自己为它的方法实现同步，而 HashMap 就必须为之提供外同步。

Hashtable 和 HashMap 采用的 hash/rehash 算法都大概一样，所以性能不会有很大的差异。

43.描述一下 JVM 加载 class 文件的原理机制？

44.试举例说明一个典型的垃圾回收算法？

45.请用 java 写二叉树算法，实现添加数据形成二叉树功能，并以先序的方式打印出来。

46.请写一个 java 程序实现线程连接池功能？

47.给定一个 C 语言函数，要求实现在 java 类中进行调用。

48、编一段代码，实现在控制台输入一组数字后，排序后在控制台输出；

49、列出某文件夹下的所有文件；

50、调用系统命令实现删除文件的操作；

51、实现从文件中一次读出一个字符的操作;

52、列出一些控制流程的方法;

53、多线程有哪些状态?

54、编写了一个服务器端的程序实现在客户端输入字符然后在控制台上显示,直到输入"END"为止,让你写出客户端的程序;

55、作用域 public,private,protected,以及不写时的区别

答: 区别如下:

作用域 当前类 同一 package 子孙类 其他 package

public √ √ √ √

protected √ √ √ ×

friendly √ √ × ×

private √ × × ×

不写时默认为 friendly

56、ArrayList 和 Vector 的区别,HashMap 和 Hashtable 的区别

答: 就 ArrayList 与 Vector 主要从二方面来说.

一.同步性:Vector 是线程安全的,也就是说是同步的,而 ArrayList 是线程程序不安全的,不是同步的

二.数据增长:当需要增长时,Vector 默认增长为原来一倍,而 ArrayList 却是原来的一半

就 HashMap 与 Hashtable 主要从三方面来说。

一.历史原因:Hashtable 是基于陈旧的 Dictionary 类的,HashMap 是 Java 1.2 引进的 Map 接口的一个实现

二.同步性:Hashtable 是线程安全的,也就是说是同步的,而 HashMap 是线程程序不安全的,不是同步的

三.值: 只有 HashMap 可以让你将空值作为一个表的条目的 key 或 value

57、char 型变量中能不能存贮一个中文汉字?为什么?

答: 是能够定义成为一个中文的,因为 java 中以 unicode 编码,一个 char 占 16 个字节,所以放一个中文是没问题的

58、多线程有几种实现方法,都是什么?同步有几种实现方法,都是什么?

答: 多线程有两种实现方法,分别是继承 Thread 类与实现 Runnable 接口
同步的实现方面有两种,分别是 synchronized,wait 与 notify

59、垃圾回收机制,如何优化程序?

希望大家补上, 谢谢

60、float 型 float f=3.4 是否正确?

答:不正确。精度不准确,应该用强制类型转换, 如下所示: float f=(float)3.4

61、介绍 JAVA 中的 Collection Framework(包括如何写自己的数据结构)?

答: Collection Framework 如下:

Collection

├List

│├LinkedList

│├ArrayList

│└Vector

│└Stack

└Set

Map

└─Hashtable
└─HashMap
└─WeakHashMap

Collection 是最基本的集合接口，一个 Collection 代表一组 Object，即 Collection 的元素 (Elements)

Map 提供 key 到 value 的映射

62、Java 中异常处理机制，事件机制？

11、JAVA 中的多形与继承？

希望大家补上，谢谢

63、抽象类与接口？

答：抽象类与接口都用于抽象，但是抽象类 (JAVA 中) 可以有自己的部分实现，而接口则完全是一个标识 (同时有多重继承的功能)。

6.16

最全的 java 面试题（十二）

编程题：

1. 现在输入 n 个数字，以逗号，分开；

然后可选择升或者降序排序；

按提交键就在另一页面显示

按什么 排序，结果为， ，

提供 reset

答案（1）

```
public static String[] splitStringByComma(String source){  
    if(source==null||source.trim().equals(""))  
        return null;  
    StringTokenizer commaToker = new StringTokenizer(source,",");  
    String[] result = new String[commaToker.countTokens()];  
    int i=0;  
    while(commaToker.hasMoreTokens()){  
        result[i] = commaToker.nextToken();  
        i++;  
    }  
    return result;  
}
```

循环遍历 String 数组

Integer.parseInt(String s)变成 int 类型

组成 int 数组

Arrays.sort(int[] a),

a 数组升序

降序可以从尾部开始输出

2. 金额转换，阿拉伯数字的金额转换成中国传统的形式如：

（¥ 1011）—>（一千零一拾一元整）输出。

3、继承时候类的执行顺序问题，一般都是选择题，问你将会打印出什么？

答：父类：

```
package test;
public class FatherClass
{
public FatherClass()
{
System.out.println("FatherClass Create");
}
}
```

子类:

```
package test;
import test.FatherClass;
public class ChildClass extends FatherClass
{
public ChildClass()
{
System.out.println("ChildClass Create");
}
public static void main(String[] args)
{
FatherClass fc = new FatherClass();
ChildClass cc = new ChildClass();
}
}
```

输出结果:

```
C:>java test.ChildClass
FatherClass Create
FatherClass Create
ChildClass Create
```

4、内部类的实现方式?

答: 示例代码如下:

```
package test;
public class OuterClass
{
private class InterClass
{
public InterClass()
{
System.out.println("InterClass Create");
}
}
public OuterClass()
{
InterClass ic = new InterClass();
```

```

System.out.println("OuterClass Create");
}
public static void main(String[] args)
{
OuterClass oc = new OuterClass();
}
}

```

输出结果:

```

C:>java test/OuterClass
InterClass Create
OuterClass Create

```

再一个例题:

```

public class OuterClass {
private double d1 = 1.0;
//insert code here
}

```

You need to insert an inner class declaration at line 3. Which two inner class declarations are

valid?(Choose two.)

- A. class InnerOne{
public static double methoda() {return d1;}
}
- B. public class InnerOne{
static double methoda() {return d1;}
}
- C. private class InnerOne{
double methoda() {return d1;}
}
- D. static class InnerOne{
protected double methoda() {return d1;}
}
- E. abstract class InnerOne{
public abstract double methoda();
}

说明如下:

- 一.静态内部类可以有静态成员，而非静态内部类则不能有静态成员。故 A、B 错
- 二.静态内部类的非静态成员可以访问外部类的静态变量，而不可访问外部类的非静态变量；
return d1 出错。

故 D 错

- 三.非静态内部类的非静态成员可以访问外部类的非静态变量。故 C 正确

四.答案为 C、E

5、Java 的通信编程，编程题 (或问答)，用 JAVA SOCKET 编程，读服务器几个字符，再写入本地显示？

答:Server 端程序:

```
package test;
import java.net.*;
import java.io.*;
public class Server
{
    private ServerSocket ss;
    private Socket socket;
    private BufferedReader in;
    private PrintWriter out;
    public Server()
    {
        try
        {
            ss=new ServerSocket(10000);
            while(true)
            {
                socket = ss.accept();
                String RemoteIP = socket.getInetAddress().getHostAddress();
                String RemotePort = ":"+socket.getLocalPort();
                System.out.println("A client come in!IP:"+RemoteIP+RemotePort);
                in = new BufferedReader(new
                    InputStreamReader(socket.getInputStream()));
                String line = in.readLine();
                System.out.println("Cleint send is :"+ line);
                out = new PrintWriter(socket.getOutputStream(),true);
                out.println("Your Message Received!");
                out.close();
                in.close();
                socket.close();
            }
        }catch (IOException e)
        {
            out.println("wrong");
        }
    }
    public static void main(String[] args)
    {
        new Server();
    }
};
```


Client 端程序:

```
package test;
import java.io.*;
import java.net.*;

public class Client
{
    Socket socket;
    BufferedReader in;
    PrintWriter out;
    public Client()
    {
        try
        {
            System.out.println("Try to Connect to 127.0.0.1:10000");
            socket = new Socket("127.0.0.1", 10000);
            System.out.println("The Server Connected!");
            System.out.println("Please enter some Character:");
            BufferedReader line = new BufferedReader(new
                InputStreamReader(System.in));
            out = new PrintWriter(socket.getOutputStream(), true);
            out.println(line.readLine());
            in = new BufferedReader(new InputStreamReader(socket.getInputStream()));
            System.out.println(in.readLine());
            out.close();
            in.close();
            socket.close();
        } catch (IOException e)
        {
            out.println("Wrong");
        }
    }
    public static void main(String[] args)
    {
        new Client();
    }
};
```

6、用 JAVA 实现一种排序， JAVA 类实现序列化的方法(二种)? 如在 COLLECTION 框架中，实现比较要实现什么样的接口?

答:用插入法进行排序代码如下

```
package test;
import java.util.*;
class InsertSort
```

```

{
ArrayList al;
public InsertSort(int num,int mod)
{
al = new ArrayList(num);
Random rand = new Random();
System.out.println("The ArrayList Sort Before:");
for (int i=0;i<num ;i++ )
{
al.add(new Integer(Math.abs(rand.nextInt()) % mod + 1));
System.out.println("al["+i+"]="+al.get(i));
}
}
public void SortIt()
{
Integer tempInt;
int MaxSize=1;
for(int i=1;i<al.size();i++)
{
tempInt = (Integer)al.remove(i);
if(tempInt.intValue()>= (Integer)al.get(MaxSize-1)).intValue())
{
al.add(MaxSize,tempInt);
MaxSize++;
System.out.println(al.toString());
} else {
for (int j=0;j<MaxSize ;j++ )
{
if
(((Integer)al.get(j)).intValue()>=tempInt.intValue())
{
al.add(j,tempInt);
MaxSize++;
System.out.println(al.toString());
break;
}
}
}
}
System.out.println("The ArrayList Sort After:");
for(int i=0;i<al.size();i++)
{
System.out.println("al["+i+"]="+al.get(i));
}
}

```

```

}
}
public static void main(String[] args)
{
    InsertSort is = new InsertSort(10,100);
    is.SortIt();
}
}

```

JAVA 类实现序列化化的方法是实现 `java.io.Serializable` 接口

Collection 框架中实现比较要实现 `Comparable` 接口和 `Comparator` 接口

7、编程：编写一个截取字符串的函数，输入为一个字符串和字节数，输出为按字节截取的字符串。但是要保证汉字不被截半个，如"我 ABC"4，应该截为"我 AB"，输入"我 ABC 汉 DEF"，6，应该输出为"我 ABC"而不是"我 ABC+ 汉的半个"。

答：代码如下：

```

package test;

class SplitString
{
    String SplitStr;
    int SplitByte;
    public SplitString(String str,int bytes)
    {
        SplitStr=str;
        SplitByte=bytes;
        System.out.println("The String is:'"+SplitStr+"';Split Bytes='"+SplitByte);
    }
    public void SplitIt()
    {
        int loopCount;
        loopCount=(SplitStr.length()%SplitByte==0)?(SplitStr.length()/SplitByte):(SplitStr.length()/SplitByte+1);
        System.out.println("Will Split into "+loopCount);
        for (int i=1;i<=loopCount ;i++ )
        {
            if (i==loopCount){
                System.out.println(SplitStr.substring((i-1)*SplitByte,SplitStr.length()));
            } else {
                System.out.println(SplitStr.substring((i-1)*SplitByte,(i*SplitByte)));
            }
        }
    }
    public static void main(String[] args)
    {
        SplitString ss = new SplitString("test 中 dd 文 dsaf 中男大 3443n 中国 43 中国人

```

```
0ewldfls=103",4);  
ss.SplitIt();  
}  
}
```

8、JAVA 多线程编程。用 JAVA 写一个多线程程序，如写四个线程，二个加 1，二个对一个变量减一，输出。

希望大家补上，谢谢

9、STRING 与 STRINGBUFFER 的区别。

答：STRING 的长度是不可变的，STRINGBUFFER 的长度是可变的。如果你对字符串中的内容经常进行操作，特别是内容要修改时，那么使用 StringBuffer，如果最后需要 String，那么使用 StringBuffer 的 toString()方法

6.17

46 家中外知名企业面试题目-1

微软

智力题

1. 烧一根不均匀的绳子，从头烧到尾总共需要 1 个小时，问如何用烧绳子的方法来确

定半小时的时间呢？

2. 10 个海盗抢到了 100 颗宝石，每一颗都一样大小且价值连城。他们决定这么分：

(1) 抽签决定自己的号码（1~10）；

(2) 首先，由 1 号提出分配方案，然后大家表决，当且仅当超过半数的人同意时，按

照他的方案进行分配，否则将被扔进大海喂鲨鱼；

(3) 如果 1 号死后，再由 2 号提出分配方案，然后剩下的 4 个人进行表决，当且仅当超

过半数的人同意时，按照他的方案进行分配，否则将被扔入大海喂鲨鱼；

(4) 依此类推.....

条件：每个海盗都是很聪明的人，都能很理智地做出判断，从而做出选择。

问题：第一个海盗提出怎样的分配方案才能使自己的收益最大化？

3. 为什么下水道的盖子是圆的？

4. 中国有多少辆汽车？

5. 你让工人为你工作 7 天，回报是一根金条，这根金条平分成相连的 7 段，你必须在每

天结束的时候给他们一段金条。如果只允许你两次把金条弄断，你如何给你的工人付费？

6. 有一辆火车以每小时 15 公里的速度离开北京直奔广州，同时另一辆火车以每小时 2

0 公里的速度从广州开往北京。如果有一只鸟，以 30 公里每小时的速度和两辆火车同时启动，从北京出发，碰到另一辆车后就向相反的方向返回去飞，就这样依次在两辆火车之间来回地飞，直到两辆火车相遇。请问，这只鸟共飞行了多长的距离？

7. 你有两个罐子以及 50 个红色弹球和 50 个蓝色弹球，随机选出一个罐子，随机选出一个弹球放入罐子，怎样给出红色弹球最大的选中机会？在你的计划里，得到红球的几率是多少？

8. 想像你站在镜子前，请问，为什么镜子中的影像可以左右颠倒，却不能上下颠倒呢？

9. 如果你有无穷多的水，一个 3 公升的提桶，一个 5 公升的提桶，两只提桶形状上下都不均匀，问你如何才能准确称出 4 公升的水？

10. 你有一桶果冻，其中有黄色、绿色、红色三种，闭上眼睛抓取同种颜色的两个。抓取多少次就可以确定你肯定有两个同一颜色的果冻？

11. 连续整数之和为 1000 的共有几组？

12. 从同一地点出发的相同型号的飞机，可是每架飞机装满油只能绕地球飞半周，飞机之间可以加油，加完油的飞机必须回到起点。问至少要多少架次，才能满足有一架绕地球一周。

参考答案：

1. 两边一起烧。

2. 96, 0, 1, 0, 1, 0, 1, 0, 1, 0。

3. 因为口是圆的。

4. 很多。

5. 分 1, 2, 4。

6. 6/7 北京到广州的距离。

7. 100%。

8. 平面镜成像原理（或者是“眼睛是左右长的”）。

9. 3 先装满，倒在 5 里，再把 3 装满，倒进 5 里。把 5 里的水倒掉，把 3 里剩下的水倒进 5 里，再把 3 装满，倒进 5 里，ok！

10. 一次。

11. 首先 1000 为一个解。连续数的平均值设为 x ，1000 必须是 x 的整数倍。假如连续数的个数为偶数个， x 就不是整数了。 x 的 2 倍只能是 5, 25, 125 才行。因为平均值为 12.5, 要 62.5 是可以的。即 62, 63, 61, 64, 等等。连续数的个数 = 连续 80 个达不到。125/2

$2 \times 2 \times 2 \times 5 \times 5 \times 5$; x 可为奇数时, 平均值为整数。1000 为平均值的奇数倍。1000

以为 2, 4, 8, 40, 200 排除后剩下 40 和 200 是可以的。所以答案为平均值为 62.5, 40, 200

, 1000 的 4 组整数。

12. 答案是 5 架次。一般的解法可以分为如下两个部分:

(1) 直线飞行

一架飞机载满油飞行距离为 1, n 架飞机最远能飞多远? 在不是兜圈没有迎头接应的情况, 这问题就是 n 架飞机能飞多远? 存在的极值问题是不要重复飞行, 比如两架飞机同时给一架飞机加油且同时飞回来即可认为是重复, 或者换句话说, 离出发点越远, 在飞的飞机就越少, 这个极值条件是显然的, 因为 n 架飞机带的油是一定的, 如重复, 则浪费的油就越多。比如最后肯定是只有一架飞机全程飞行, 注意“全程”这两个字, 也就是不要重复的极值条件。如果是两架飞机的话, 肯定是一架给另一架加满油, 并使剩下的油刚好能回去, 就说第二架飞机带的油耗在 3 倍于从出发到加油的路程上, 有三架飞机第三架带的油耗在 $1+1/3+\dots+1/5=5$ 倍于从出发到其加油的路程上, 所以 n 架飞机最远能飞行的距离为 s ($2n+1$) 这个级数是发散的, 所以理论上只要飞机足够多最终可以使一架飞机飞到无穷远 1 个飞机加油。-, 当然实际上不可能一架飞机在飞行 $1/(2n+1)$ 时间内同时给 n

(2) 可以迎头接应加油

一架飞机载满油飞行距离为 $1/2$, 最少几架飞机能飞行距离 1? 也是根据不要重复飞行的极值条件, 得出最远处肯定是只有一架飞机飞行, 这样得出由 $1/2$ 处对称两边 $1/4$ 肯定是一架飞机飞行, 用上面的公式即可知道一边至少需要两架飞机支持, $(1/3+1/5)/2 > 1/4$ (左边除以 2 是一架飞机飞行距离为 $1/2$), 但是有一点点剩余, 所以想像为一个滑轮 (中间一个飞机是个绳子, 两边两架飞机是个棒) 的话, 可以滑动一点距离, 就说加油地点可以在一定距离内变动 (很容易算出来每架飞机的加油地点和加油数量, 等等)

数学篇

1. 1000! 有几位数, 为什么?

2. $F(n) = 1, n > 8, n < 12$

$F(n) = 2, n < 2$

$F(n) = 3, n = 6$

$F(n) = 4, n \text{ other}$

使用 + * / 和 $\text{sign}(n)$ 函数组合出 $F(n)$ 函数

$\text{sign}(n) = 0, n = 0$

$\text{sign}(n) = 1, n < 0$

$\text{sign}(n) = 1, n > 0$

3. 编一个程序求质数的和，例如 $F(7) = 1+3+5+7+11+13+17 = 57$ 。

逻辑推理题

1. 此题源于 1981 年柏林的德国逻辑思考学院，98% 的测验者无法解答此题。

有五间房屋排成一列；所有房屋的外表颜色都不一样；所有的屋主来自不同的国家；

所有的屋主都养不同的宠物；喝不同的饮料；抽不同的香烟。

(1) 英国人住在红色房屋里；(2) 瑞典人养了一只狗；(3) 丹麦人喝茶；(4) 绿

色的房子在白色的房子的左边；(5) 绿色房屋的屋主喝咖啡；(6) 吸 Pall Mall 香烟的屋

主养鸟；(7) 黄色屋主吸 Dunhill 香烟；(8) 位于最中间的屋主喝牛奶；(9) 挪威人住

在第一间房屋里；(10) 吸 Blend 香烟的人住在养猫人家的隔壁；(11) 养马的屋主在吸 D

unhill 香烟的人家的隔壁；(12) 吸 Blue Master 香烟的屋主喝啤酒；(13) 德国人吸 Pri

nce 香烟；(14) 挪威人住在蓝色房子隔壁；(15) 只喝开水的人住在吸 Blend 香烟的人的

隔壁

问：谁养鱼？

提示：首先确定

房子颜色：红、黄、绿、白、蓝 Color 1 2 3 4 5

国籍：英、瑞、丹、挪、德 \Rightarrow Nationality 1 2 3 4 5

饮料：茶、咖、奶、酒、水 \Rightarrow Drink 1 2 3 4 5

烟：PM、DH、BM、PR、混 \Rightarrow Tobacco 1 2 3 4 5

宠物：狗、鸟、马、猫、鱼 \Rightarrow Pet 1 2 3 4 5

然后有：

(9) \Rightarrow N1=挪威

(14) \Rightarrow C2=蓝

(4) \Rightarrow 如 C3=绿，C4=白，则 (8) 和 (5) 矛盾，所以 C4=绿，C5=白
剩下红黄只能为 C1，C3

(1) \Rightarrow C3=红，N3=英国，C1=黄

(8) =>D3=牛奶

(5) =>D4=咖啡

(7) =>T1=DH

(11) =>P2=马

那么:

挪威? 英国??

黄 蓝 红 绿 白

?? 牛奶 咖啡?

DH????

? 马???

(12) =>啤酒只能为 D2 或 D5, BM 只能为 T2 或 T5=>D1=矿泉水

(3) =>茶只能为 D2 或 D5, 丹麦只能为 N2 或 N5

(15) =>T2=混合烟=>BM=T5,

所以剩下啤酒=D5, 茶=T2=>丹麦=D2

然后:

挪威 丹麦 英国??

黄 蓝 红 绿 白

矿泉水 茶 牛奶 咖啡 啤酒

DH 混合烟?? BM

? 马???

(13) =>德国=N4, PR=T4

所以, 瑞典=N5, PM=T3

(2) =>狗=P5

(6) =>鸟=P3

(10) =>猫=P1

得到:

挪威 丹麦 英国 德国 瑞典

黄 蓝 红 绿 白

矿泉水 茶 牛奶 咖啡 啤酒

DH 混合烟 PM PR BM

猫 马 鸟? 狗

所以, 最后剩下的鱼只能由德国人养了。

2.

...

...

...

请仅用一笔画四根直线, 将上图 9 个点全部连接。

3. 对一批编号为 1~100 全部开关朝上(开)的灯进行以下操作:

凡是 1 的倍数反方向拨一次开关; 2 的倍数反方向又拨一次开关; 3 的倍数反方向又拨一

次开关.....

问：最后为关灯状态的灯的编号。

微软招聘总经理助理的三道面试题

1. 某手机厂家由于设计失误，有可能造成电池寿命比原来设计的寿命短一半（不是冲放电时间），解决方案就是更换电池或给 50 元购买该厂家新手机的折换券。请给所有已购买的用户写信告诉解决方案。
2. 一高层领导在参观某博物馆时，向博物馆馆员小王要了一块明代的城砖作为纪念，按国家规定，任何人不得将博物馆收藏品变为私有。博物馆馆长需要如何写信给这位领导，将城砖取回？
3. 王小姐由于工作失误，将 2 万元的笔记本电脑以 1.2 万元错卖给李先生，王小姐的经理应该怎么写信给李先生将钱要回？

6.18

英文面试题目

1. Algorithms

- * What's the difference between a linked list and an array?
- * Implement an algorithm to sort a linked list. Why did you pick the method you did?
- * Implement an algorithm to sort an array. Why did you pick the method you did?
- * Implement strstr() (or some other string library function).
- * Reverse a string. Optimize for speed. Optimize for space.
- * Count the number of set bits in a number. Now optimize for speed. Now optimize for size.
- * How would you find a cycle in a linked list?
- * Give me an algorithm to shuffle a deck of cards, given that the cards are stored in an array of ints.
- * Write a function that takes in a string parameter and checks to see whether or not it is an integer, and if it is then return the integer value.
- * Write a function to print all of the permutations of a string.
- * Implement malloc.
- * Write a function to print the Fibonacci numbers.
- * Write a function to copy two strings, A and B. The last few bytes of str

ing A overlap the first few bytes of string B.

- * How would you print out the data in a binary tree, level by level, start

ing at the top?

2. Applications

- * How can computer technology be integrated in an elevator system for a hu

ndred story office building? How do you optimize for availability? How would v

ariation of traffic over a typical work week or floor or time of day affect th

is?

- * How would you redesign an ATM?
- * Suppose we wanted to run a microwave oven from the computer. What kind o

f software would you write to do this?

- * How would you design a coffee-machine for an automobile.

3. Thinkers

- * How are M&Ms made?
- * If you had to learn a new computer language, how would you go about doin

g it?

- * If MS told you we were willing to invest \$5 million in a start up of you

r choice, what business would you start? Why?

- * If you could gather all of the computer manufacturers in the world toget

her into one room and then tell them one thing that they would be compelled to

do,what would it be?

- * Expla in a scenario for testing a salt shaker.
- * If you are going to receive an award in 5 years, what is it for and who

is the audience?

- * How would you expla in how to use Microsoft Excel to your grandma?
- * Why is it that when you turn on the hot water in any hotel, for exa mple,

the hot water comes pouring out almost instantaneously?

微软亚洲技术支持中心面试题目录

1. 进程和线程的差别。
2. Heap 与 stack 的差别。
3. Windows 下的内存是如何管理的？
4. 介绍 .Net 和 .Net 的安全性。
5. 客户端如何访问 .Net 组件实现 Web Service ？

6. C/C++编译器中虚表是如何完成的？
7. 谈谈 COM 的线程模型。然后讨论进程内 / 外组件的差别。
8. 谈谈 IA32 下的分页机制。
9. 给两个变量，如何找出一个带环单链表中是什么地方出现环的？
10. 在 IA32 中一共有多少种办法从用户态跳到内核态？
11. 如果只想让程序有一个实例运行，不能运行两个。像 winamp 一样，只能开一个窗口，怎样实现？

12. 如何截取键盘的响应，让所有的 'a' 变成 'b'？
13. Apartment 在 COM 中有什么用？为什么要引入？
14. 存储过程是什么？有什么用？有什么优点？
15. Template 有什么特点？什么时候用？
16. 谈谈 Windows DNA 结构的特点和优点。

微软研究院笔试题目

```
1. #include <stdio.h>
#include <String.h>
class CBuffer
{
    char * m_pBuffer;
    int m_size;
public:
    CBuffer()
    {
        m_pBuffer=NULL;
    }
    ~CBuffer()
    {
        Free();
    }
    void Allocte(int size)
    {
        m_size=size;
        m_pBuffer= new char[size];
    }
private:
    void Free()
    {
        if(m_pBuffer!=NULL)
        {
            delete m_pBuffer;
            m_pBuffer=NULL;
        }
    }
}
```

```

    }
public:
    void SaveString(const char* pText) const
    {
        strcpy(m_pBuffer, pText);
    }
    char* GetBuffer() const
    {
        return m_pBuffer;
    }
};

void main (int argc, char* argv[])
{
    cBuffer buffer1;
    buffer1.SaveString("Microsoft");
    printf(buffer1.Get Buffer());
}
}

```

找出 Allocate, SaveString, main 的错误。

2. 打印“Welcome MSR Asia”

```

#include <stdio.h>
#include <string.h>
char * GetName (void)
{
    //To return “MSR Asia” String
    char name[]="MSR Asia";
    return name;
}

void main(int argc, char* argv[])
{
    char name[32];
    //Fill in zeros into name
    for(int i=0;i<=32;i++)
    {
        name[i]='\0';
    }
    //copy “Welcome” to name
    name="Welcome";
    //Append a blank char
    name[8]=" ";
    //Append string to name
    strcat(name,GetName());
    //print out
    printf(name);
}

```

找出程序中的错误。

```
3. #include <stdio.h>
class A
{
public:
void FuncA()
{
printf("FuncA called\n");
}
virtual void FuncB()
{
printf("FuncB called\n");
}
};
class B: public A
{
public:
void FuncA()
{
A::FuncA();
printf("FuncAB called\n");
}
virtual void FuncB()
{
printf("FuncBB called\n");
}
};
void main(void)
{
B b;
A *pa;
pa=&b;
A *pa2=new A;
b.FuncA();
b.FuncB();
pa->FuncA();
pa->FuncB();
pa2->FuncA();
pa2->FuncB();
delete pa2;
}
```

What is the output of the above program?

```
4. #include <stdio.h>
#include <string.h>
```

```

int FindSubString(char* pch)
{
    int count=0;
    char* p1=pch;
    while(*p1!='\0')
    {
        if(*p1==p1[1]-1)
        {
            p1++;
            count++;
        }
        else
        {
            break;
        }
    }
    int count2=count;
    while(*p1!='\0')
    {
        if(*p1==p1[1]+1)
        {
            p1++;
            count2--;
        }
        else
        {
            break;
        }
    }
    if(count2==0)
    return count;
    return 0;
}

void ModifyString(char* pText)
{
    char* p1=pText;
    char* p2=p1;
    while(*p1!='\0')
    {
        int count=FindSubString(p1);
        if(count>0)
        {
            *p2++=*p1;
            sprintf(p2, "%I", count);
            while(*p2!= '\0')

```

```

{
p2++;
}
p1+=count+count+1;
}
else
{
*p2++=*p1++;
}
}
}
void main(void)
{
char text[32]="XYBCDCBA BABA ";
ModifyString(text);
printf(text);
}

```

In the main() function, after ModifyString(text) is called, what's the va

lue of 'text'?

微创笔试题目（微创，微软在中国的合资公司）

1. 上海的苏州河由于遭受多年的工业污染，一直是条臭水沟。上海市政府下了很大决心清理苏州河，你觉得需要几年能让河水变清？你的依据是什么？
2. 找出字符串 A 中包含的字符可以进行的所有不同组合。例如：abccd 中，ab，ac，bc，cc，abd 等都是可能的组合。（请用 C/C++编程，不允许上机操作）
3. 请估算月球的体积。
4. 经常去的技术网站，请举例。
5. 对软件开发过程的理解。
6. 上海有多少外籍和港澳台人士？你的依据是什么？（不得引用政府和调研机构数据）
7. 字符串 A 是由 n 个小写英文字母（a~z）构成的，定义为 char A[n]。你能用更少的空间表示这个字符串吗？请写出从 char A[n]到你的新的储存格式的转换函数。（请用 C/C++编程，不允许上机操作）
8. 哈希表和数组的定义，区别，优缺点。
9. 用递归实现菲波列数列。
10. 用 dhtml 写页面。
11. 一楼到十楼的每层电梯门口都放着一颗钻石，钻石大小不一。你乘坐电梯从一楼

到十楼，每层楼电梯门都会打开一次，只能拿一次钻石，问怎样才能拿到最大的一颗？

（去年应聘到微创的 S 小姐面试遇到的就是这道智力题。她的回答是：选择前五层楼都不拿，观察各层钻石的大小，做到心中有数。后五层楼再选择，选择大小接近前五层楼出现过最大钻石大小的钻石。她至今也不知道这道题的准确答案，“也许就没有准确答案，就是考一下你的思路，”她如是说。）

12.U2 合唱团在 17 分钟内得赶到演唱会场，途中必需跨过一座桥，四个人从桥的同一端出发，你得帮助他们到达另一端，天色很暗，而他们只有一只手电筒。一次同时最多可以有两人一起过桥，而过桥的时候必须持有手电筒，所以就得有人把手电筒带来带去，来回桥两端。手电筒是不能用丢的方式来传递的。四个人的步行速度各不同，若两人同行则以较慢者的速度为准。Bono 需花 1 分钟过桥，Edge 需花 2 分钟过桥，Adam 需花 5 分钟过桥，Larry 需花 10 分钟过桥。他们要如何在 17 分钟内过桥呢？（有个同济的学生写文章说他当时在微软面试时就是碰到了这道题，最短只能做出在 19 分钟内过桥，微软的人对他讲这样的结果已经是不错的了！）

13.烧一根不均匀的绳要用一个小时，如何用它来判断半个小时？（参考答案：两边一起烧）

14.为什么下水道的盖子是圆的？（从复旦大学一位计算机系教授那里听来的答案：因为如果是方的、长方形的或椭圆的，那无聊之徒拎起来它就可以直接扔进地下道啦！但圆形的盖子嘛，就可以避免这种情况了）

15.有 7 克、2 克砝码各一个，天平一只，如何只用这些物品三次将 140 克的盐分成 50、90 克各一份？

6.19

46 家中外知名企业面试题目-2

Intel 笔试面试题目
智力题

1. 每天中午从法国塞纳河畔的勒阿佛有一艘轮船驶往美国纽约，在同一时刻纽约也有一艘轮船驶往勒阿佛。已知横渡一次的时间是 7 天 7 夜，轮船匀速航行，在同一航线，轮船近距离可见。

请问今天中午从勒阿佛开出的船会遇到几艘从纽约来的船？

2. 巴拿赫病故于 1945 年 8 月 31 日。他的出生年份恰好是他在世时某年年龄的平方，问：他是哪年出生的？

答案：

设他在世时某年年龄为 x ，则 x 的平方 < 1945 ，且 x 为自然数。其出生年份 x 的平方 f ；

1) 1945 的平方 $-x(x-1)$ ，他在世年龄 $1945-x(x=85;x$

1) $g=44$ 时， $x(x=44.1)$ ，则 x 应为 44 或略小于此的数。而 $x=$ 根

；
 $=53$ ；又 $x=1892-1892$ ，算得其在世年龄为 $1945=01;44 \times 43$

$180-1806$ ，得其在世年龄为 $1945=43 \times 42=1)-43$ 时， $x(x$

44，即他出生于 $18=139$ ；若 x 再取小，其在世年龄越大，显然不妥。故 $x=6$

92 年，终年 53 岁。

3.

(图形描述:一个各边相等的十字图案)

上图中各边相等，要求：用最少的分割，拼成一个正方形。

笔试题目

1. 设计一个重采样系统，说明如何 anti-alias。

$x(n/2)$ ，问： $=x(2n)$ ， $y_2(n) =$ 2. $y_1(n)$

如果 y_1 为周期函数，那么 x 是否为周期函数？

如果 x 为周期函数，那么 y_1 是否为周期函数？

如果 y_2 为周期函数，那么 x 是否为周期函数？

如果 x 为周期函数，那么 y_2 是否为周期函数？

3. 如果模拟信号的带宽为 5kHz，要用 8k 的采样率，怎么办。

4. 某个程序在一个嵌入式系统（200M 的 CPU，50M 的 SDRAM）中已经最优化了，换到另

一个系统（300M 的 CPU，50M 的 SDRAM）中运行，还需要优化吗？

5. $x^4+a \cdot x^3+x^2+c \cdot x+d$ 最少需要做几次乘法。

6. 三个 float: a, b, c

问值：

$(b+a)+c==$ $(a+b)+c$

$(a+c)+b==$ $(a+b)+c$

7. 把一个链表反向填空。

8. 下面哪种排序法对 12354 最快? 。

A. quick sort

B. bubble sort

C. merge sort

9. 哪种结构平均来讲获取一个值最快? 。

A. binary tree

B. hash table

C. stack

10.

```
#include "stdafx.h"
```

```
#include <iostream.h>
```

```
struct bit
```

```
{ int a:3;
```

```
  int b:2;
```

```
  int c:3;
```

```
};
```

```
int main(int argc, char* argv[])
```

```
{ bit s;
```

```
  char *c = (char*)&s;
```

```
  *c = 0x99;
```

```
  cout << s.a <<endl <<s.b<<endl<<s.c<<endl;
```

```
  return 0;
```

```
}
```

Output:?

11.

挑 bug, 在 linux 下运行:

```
#include <stdio.h>
```

```
char *reverse(char* str)
```

```
{
```

```
  int len=0, i=0;
```

```
  char *pstr=str, *ptemp,*pd;
```

```
  while(*++pstr)
```

```
    len++;
```

```
  pstr--;
```

```
  //ptemp=(char*)malloc(len+1);
```

```
  ptemp=(char*)malloc(len+1);
```

```
  pd=ptemp;
```

```
  while(len--){
```

```
    *ptemp=*pstr;
```

```
    ptemp++;
```

```
    pstr--;
```

```
    i++;
```

```
  }
```

```
  *ptemp=*pstr;
```

```

ptemp++;
*ptemp='\0';
return pd;
}
main()
{
char string[40]= "Hello World!";
char *pstr=string;
printf("%s", pstr);
printf("%s", reverse(pstr));
}

```

实验室笔试题

1. 写出下列信号的奈奎斯特频率

$$\sin(4000\pi t) + \cos(2000\pi t) + 1 = \quad (1) \quad f(t)$$

$$\sin(4000\pi t) / \pi t = \quad (2) \quad f(t)$$

$$(\sin(4000\pi t) \text{ 的平方}) / \pi t = \quad (3) \quad f(t)$$

2. 有两个线程

```

void producer()
{
while(1)
{
GeneratePacket();
PutPacketIntoBuffer();
Signal(customer);
}
}
void customer()
{
while(1)
{
WaitForSignal();
if(PacketInBuffer>10)
{
ReadAllPackets();
ProcessPackets();
}
}
}

```

(1) 有没有其他方法可以提高程序的性能

(2) 可不可以不使用信号之类的机制来实现上述的功能

3. 优化下面的程序

```
0=      (0) sum
1=      (1) I
4*I=    (2) T1
4-address(A)= (3) T2
T2[T1]=   (4) T3
4-address(B)= (5) T4
4*I=     (6) T5
T4[T5]=   (7) T6
T3*T5=    (8) T7
sum+T6=   (9) sum
I+1=     (10) I
        (11) IF I<20 GOTO (2)
6.20
```

面试题

1. 下面这段代码不符合 Pipeline 要求，请你改动一下

```
if(a>b)
0:=    i
else
1:=    i
```

2. 对于运行在 ring3 上的这个指令，请你指出 CPU 和操作系统分别参与了哪部分操作？

```
mov eax, [0x12345678]
```

3. 如果有一个芯片，只是裸机，如何写它的操作系统？最难的部分在哪儿？如何解决

？

4. 如何写一个主板的 BIOS？

5. 没有操作系统来给你完成它的一些初始化工作，如何写出 PCI 的 driver？

Intel 2004 北京笔试题

问答题

1. 你觉得 C 程序中为什么会有 main()，有没有想过 exit，return，或什么都不做也可

以让程序正常终止？

2. TOTAL 个人围一圈，从 1 开始数到 N，谁数到 N 出圈，下一个人继续从 1 开始数，返回

最后一个出局的人。

```
#define TOTAL 15;
int xxxx(int N)
{
    int ring[TOTAL] = {1,1,1,1,1,1,1,1,1,1,1,1,1,1, 1} //全是 1
    int nextstart = 0, counter=0;
```

```

for (i=1; i<TOTAL; i++){
counter = 0;
while(counter<N) {
if (_____)
(;)
else
(_____;)
}
ring[nextstart]=0
}
return nextstart +1;
}

```

3. 列举 Intel 四种架构

4. 概率题

$\int_0^1 \int_0^x dx \cdot \int_0^x dx =$ (1) x, y 为随机变量, 联合概率密度 $f(x,y)$

?? $E(xy) = k \cdot dy$, k 为常数, 求 k

注: $\int_a^b f(x)dx$ 为 a 到 b 的定积分。

(2) A, B 为随机事件, 以下哪个正确。

A. $P(A \cup B) \cdot P(AB) < P(A)P(B) =$

B. $P(A \cup B) \cdot P(AB) > P(A)P(B) =$

C. $P(A \cup B) \cdot P(AB) < P(A) + P(B) =$

D. $P(A \cup B) \cdot P(AB) > P(A) + P(B) =$

5. 信道带宽 200kHz, 信噪比 10dB, 求信道波特率?

6. 以下代码运行结果是。

```

int main()
{
int a,b,c,abc = 0;
a=b=c=40;
if(c)
{
int abc;
abc = a*b+c;
}
printf("%d,%d", abc, c);
return 0;
}

```

7. 给出了从纽约出发和到达洛杉矶的各种航班信息, 写出找到一条从纽约到洛杉矶的

最短距离的航班组合的代码。

8. 从计算机图形上截取某个物体边缘的若干个坐标, 求这个物体的面积, 并判断是方

形还是圆形, 说明原因。

9. 离散卷积与 DFT 的区别与关系。快速求出不满足 2^N 长度的离散傅立叶变换的方法有

哪些？如何用 fft 求 $N \times M$ 点的离散卷积？

10. 给出 fir 和 iir 的优缺点。

11. 如何计算线性标量量化器的量化噪声？需要哪些假设？

6.21

IBM

IBM 笔试题目

字母矩阵题目（15 分钟）

给你一个矩阵：

（一） （二） （三） （四） （五）

1 a b c d e

2 b c a e d

3 c b e a d

4 c e d b a

5 e d a c b

回答以下问题。

（1）将第一行和第四行交换后，第一行第四个字母下面的左边的下面的右边的字母是

。

①a ②b ③c ④d ⑤e

（2）将所有出现在 d 左边的字母从矩阵中删掉。将所有出现在 a 左边的 c 字母从矩阵中

删掉。如果矩阵中剩下的字母的种类的数目大于 3，答案为原矩阵中左上方至右下方对角线

上出现两次的字母。如果矩阵中剩下的字母的种类的数目小于或者等于 3，答案为原矩阵中

右上至左下对角线上出现 4 次的字母是 。

①a ②b ③c ④d ⑤e

（3）将所有的 a 用 4 替换，所有的 d 用 2 替换，哪一列的总和 最大

①第 1 列 ②第 2 列 ③第 3 列 ④第 4 列 ⑤第五列

（4）从左上角的字母开始，顺时针沿矩阵外围，第 4 次出现的字母是。

①a ②b ③c ④d ⑤e

（5）沿第 5 列从上到下，接着沿第 3 列从下到上，接着沿第 4 列从上到下，接着沿第 1 列

从下到上，接着沿第 2 列从上到下，第 1 个出现 5 次的字母是 。

①a ②b ③c ④d ⑤e

（6）从左上角的字母开始，顺时针沿矩阵外围，第 4 次出现的字母是以下哪个。

①a ②b ③c ④d ⑤e

智力题

1. 有 50 家人家，每家一条狗。有一天警察通知，50 条狗当中有病狗，行为和正常狗不

一样。每人只能通过观察别人家的狗来判断自己家的狗是否生病，而不能看自己家的狗，

如果判断出自己家的狗病了，就必须当天一枪打死自己家的狗。结果，第一天没有枪声，

第二天没有枪声，第三天开始一阵枪响，问：一共死了几条狗？

2. 已知两个数字为 1~30 之间的数字，甲知道两数之和，乙知道两数之积，甲问乙：“

你知道是哪两个数吗？”乙说：“不知道”。乙问甲：“你知道是哪两个数吗？”甲说：

“也不知道”。于是，乙说：“那我知道了”，随后甲也说：“那我也知道了”，这两个

数是什么？

3. 一个经理有三个女儿，三个女儿的年龄加起来等于 13，三个女儿的年龄乘起来等于

经理自己的年龄。有一个下属已知道经理的年龄，但仍不能确定经理的三个女儿的年龄，

这时经理说只有一个女儿的头发是黑的，然后这个下属就知道了经理的三个女儿的年龄。

请问三个女儿的年龄分别是多少？为什么？

答案：

1. 死了 3 条（第几天枪响就有几条）。

简单分析：从有一条不正常的狗开始，显然第一天将会听到一声枪响。这里的要点是

你只需站在那条不正常狗的主人的角度考虑。

有两条的话思路继续，只考虑有两条不正常狗的人，其余人无需考虑。通过第一天他

们了解了对方的信息。第二天杀死自己的狗。换句话说每个人需要一天的时间证明自己的

狗是正常的。有三条的话，同样只考虑那三个人，其中每一个人需要两天的时间证明自己的

的狗是正常的狗。

2. 1 和 4，或者 4 和 7。

3. 分别是 2，2，9。

简单分析：

1 1 11 11 伪穷举，呵呵

1 2 10 20

1 3 9 27

1 4 8 32

1 5 7 35

1 6 6 36 在所有的可能性中，只有这两个相同，如果经理的年龄为其他，则他下属就

可以确定三个人分别为几岁了

2 2 9 36 所以只有两种可能：1，6，6 或者 2，2，9。如果是 1，6，6 的话，那么两个

同样大的 6 岁的孩子应该都是黑头发

2 3 8 40 所以只有 2, 2, 9 比较合理, 大的那个是黑头发, 另外两个是黄毛丫头

2 4 7 56

2 5 6 60

3 3 7 42

3 4 6 72

3 5 5 75

4 4 5 80

社会招聘笔试题

1. 一个粗细均匀的长直管子, 两端开口, 里面有 4 个白球和 4 个黑球, 球的直径、两端开口的直径等于管子的内径, 现在白球和黑球的排列是 `wwwbbb`, 要求不取出任何一个球, 使得排列变为 `bbwwwbb`。

2. 一只蜗牛从井底爬到井口, 每天白天蜗牛要睡觉, 晚上才出来活动, 一个晚上蜗牛可以向上爬 3 尺, 但是白天睡觉的时候会往下滑 2 尺, 井深 10 尺, 问蜗牛几天可以爬出来?

3. 在一个平面上画 1999 条直线最多能将这一平面划分成多少个部分?

4. 在太平洋的一个小岛上生活着土人, 他们不愿意被外人打扰, 一天, 一个探险家到了岛上, 被土人抓住, 土人的祭司告诉他, 你临死前还可以有一个机会留下一句话, 如果这句话是真的, 你将被烧死, 是假的, 你将被五马分尸, 可怜的探险家如何才能活下来?

5. 怎样种四棵树使得任意两棵树的距离相等。

6. 27 个小运动员在参加完比赛后, 口渴难耐, 去小店买饮料, 饮料店搞促销, 凭三个空瓶可以再换一瓶, 他们最少买多少瓶饮料才能保证一人一瓶?

7. 有一座山, 山上有座庙, 只有一条路可以从山上的庙到山脚, 每周一早上 8 点, 有一个聪明的小和尚去山下化缘, 周二早上 8 点从山脚回山上的庙里, 小和尚的上下山的速度是任意的, 在每个往返中, 他总是能在周一和周二的同一钟点到达山路上的同一点。例如

, 有一次他发现星期一的 8 点 30 和星期二的 8 点 30 他都到了山路靠山脚的 $\frac{3}{4}$ 的地方, 问这是

为什么？

8. 有两根不均匀分布的香，每根香烧完的时间是一个小时，你能用什么方法来确定一

段 15 分钟的时间？

英文面试题目

1. Describe your greatest achievement in the past 4-5 years?

2. What are your short & long term career objectives? What do you think is

the most ideal job for you?

3. Why do you want to join IBM? What do you think you can contribute to IB

M?

6.22

宝洁公司（P&G）面试题目

宝洁公司招聘题号称由高级人力资源专家设计，无论您如实或编造回答，都能反应您

某一方面的能力。核心部分的题目如下：

Please provide concise examples that will help us better understand your c

apabilities.

1. Describe an instance where you set your sights on a high/demanding goal

and saw it through completion.

2. Summarize a situation where you took the initiative to get others going

on an important task or issue, and played a leading role to achieve the resul

ts you wanted.

3. Describe a situation where you had to seek out relevant information, de

fine key issues, and decide on which steps to take to get the desired results.

4. Describe an instance where you made effective use of facts to secure th

e agreement of others.

5. Give an examples of how you worked effectively with people to accomplis

h an important result.

6. Describe a creative/innovative idea that you produced which led to a si

significant contribution to the success of an activity or project.

7. Provide an example of how you assessed a situation and achieved good re-

sults by focusing on the most important priorities.

8. Provide an example of how you acquired technical skills and converted

them to practical application

解答范例:

1. Demnding Goal: To design a musical and dramatic show to celebrate the c-

entennial Anniversary of Tianjin University.

The person who reach this goal: Chairman of Tianjin University Student Uni-

on What I learned from this observation: It is not necessary for a true leader

to be an expert in such or such field of his career. But he must possessthe c-

harismatic and the capacity to drive different people, who have diverging opin-

ions, or even conflicting interests, to proceed together to the sameorganizati-

onal goal.

2. The activity I initiated: To organize a group to sing English anthems o-

n Charistmas Eve, visit all domitories in university and send christmas gifts

on behalf of our English Association The desired result: To broaden the studen-

ts' horizons about Western culture.

My leading role: Combine the representatitives' suggestions with my idea

and draw the decision on:

* What songs to play?

* Who could attend the choir?

* Which spots we performed on?

The result: Many students said that they felt the warmness we sent to them

and they hoped we would hold such activities next Charistmas.

3. Background: I organized the first activity after the establishment of t-

he Management School English Association.

The desired result: To help the freshmen and the sophomores with their Eng-

lish while publicizing our group.

Key issue:

- * What aspect of the students' English abilities needed refining? Relevant

Information:

- * What kind of entertainment was popular among students and also offered chances

for them to learn English most effectively?

- * Which foreign teacher was suitable for this position?
- * When was our member free?
- * Which place was convenient for most attendances?
- * Other related factors, such as the availability of facilities and the layout of the spots.

4. Background: I advanced a plan to found an English Garden in collaboration

with fraternal association in neighboring university.

The disagreement:

- * The authority of our school may dissent.
- * The cost was expensive, and we had no enough human resource to carry on this project.

- * There were too many English corners. Another one was unnecessary. The facts I made use:

- * Our dean approved this proposal.
- * Our partner was willing to provide financial assistantship. And our members

volunteered to design the details of the plan and implement it.

- * The poll showed that the current English corners did not meet the students' requirement and lacked uniqueness. The result: Others were convinced and

we founded the English Garden successfully.

5. Background: In the military training, we hold a Military Songs Competition.

Working procedure:

Design: I cooperated with my colleagues to figure out the climax of the performance, the musical accompaniment of the songs and the whole arrangement of the narrative poem.

Rehearsal: I worked together with those who were in charge of the lights,

sounds and scenes to create the perfect artistic effect.

On stage: I reminded my fellows with gestures and eye expressions. Also, w

e coped with an emergency coherently.

The result: Our military team won the second prize in this competition.

6. Activity: To hold the Perspective Entrepreneur Contest.

The innovative idea I produced: To simulate a board meeting of a company, i

n which our candidates debated the feasibility of selling modern fitness equip

ment according to the market information they collected.

The result: The contest was held based on my proposal.

7. Background: On one morning when our promotion month first began, I found

d that the inventories in some department stores were not adequate.

My assessment of the situation: The four promoting stores were not very fa

r from each other, and the time they opened was not the same. It was possible

to fetch some stock from another store and make up for it later.

The priorities: To satisfy the stocking demand of the store which had the

largest number of customers.

8. Background: In the investigation of customers' opinions about the taste

of a new kind of beer, I found that the questionnaire form was out of date

and limited the freedom of the responders' choices.

Technical skills: The scientific arrangement of questionnaire form

The result: With the help of the converted form, our company obtained more ob

jective and effective information.

飞利浦笔试试题

1. 用逻辑门和 CMOS 电路实现 $ab+cd$ 。
2. 用一个二选一 mux 和一个 inv 实现异或。
3. 给了 reg 的 setup 和 hold 时间, 求中间组合逻辑的 delay 范围。
4. 如何解决亚稳态。
5. 用 Verilog/VHDL 写一个 fifo 控制器。
6. 用 Verilog/VDDL 检测 stream 中的特定字符串。

6.23

阿尔卡特(中国)的面试题目

全部用 C 语言完成:

1. 自己定义数据结构, 写出程序: 在一个单向链表中, 往 I 位置插入一个节点。
2. 自己定义数据结构, 写出程序: 二叉树的前序遍历。
3. 不允许使用系统时间, 写出一个随机数生成函数。

Google

这次是连环游戏, 每一题的答案将在下一题中用到。

1、{first 10-digit prime found in consecutive digits e}.com.

e 中出现的连续的第一个 10 个数字组成的质数。

2、7427466391.com

Congratulations, Youve made it to level 2. Go to www.Linux.org and enter Bo

bsyouruncle as the login and the answer to this equation as the password.

$f(1)=7182818284$

$f(2)=8182845904$

$f(3)=8747135266$

$f(4)=7427466391$

$f(5)=$ _____

update: 提示: $f(1)$ 到 $f(4)$ 是 e 中连续的 10 个数字满足总和等于 49 的前 4 个, $f(5)$ 当然

是让你搜索第 5 个咯, 编个小程序吧。

答案: 5966290435

3、www.Linux.org

4、www.google.com/labjobs/

via:

Google recruits eggheads with mystery billboard

Mysterious Billboard May Be Google Recruitment Ad

Myserious billboard

Google is behind mystery geek trap

写一句俳句来描述搜索流量季节性预测的可能方法。

用三种颜色为一个二十面体涂颜色, 每面都要覆盖, 你能够用多少种不同的涂法? 你

将选择哪三种颜色?

这是一个我们故意留给你你空白, 请填充一些你喜欢的东西。

戴尔

1.Choose one question and write down the trouble shooting steps in English

(4-5 steps)

1.Customer report his computer cannot start after sudden power lost. H

ow will you trouble shoot and find out the cause of the failure.

2. One computer was used normally last day, but today, the user cannot connect to internet via dial-up networking.

3. Customer complains the system send out great noise. What's the detail action to identify the faulty part?

4. My computer was suddenly disconnected from LAN, how to trouble shoot?

5. My computer was hung up. After reboot, only one cursor blinked on the upper-left corner. How to fix this problem.

II. Reading and translation:

Passage 1.

Customer called in and reported the battery weren't charging, System LED indicator was in yellow.

1. Suggested customer remove the battery from the battery bay by sliding the latch at the bottom of Notebook. Checked the battery power level by pressing the check button on the battery. It indicated no power.

2. Suggested customer remove the CD ROM off the Media bay, then plug the Battery to the media bay, it was still same result.

3. Suggested customer try with other battery, in the battery bay and media bay, it was ok. Battery could be charged.

4. Requested customer provide the DSN number from the fail battery. CN-05H980-69502-21U-01GB.

Informed customer that we would send a replaced battery to her.

Passage 2.

Customer bought a Notebook 20 day ago, then discovered that when she adjusted the Brightness of the LCD, the LCD would turn to very dim. She must reboot the Notebook few times before the LCD might display normally.

1. Suggested her try gently apply pressure above the keyboard near the LEDs and power button.

2. Suggested her swivel the LCD back and forth and lightly tap the plastic

tic back of LCD.

3.Suggested her gently apply pressure on the bezel surrounding the scr

een. the problem persisted, after these testing.

4.Requested customer to update BIOS and alter the LCD Brightness contr

ol setting in the BIOS. After checking BIOS the problem persisted.

I told customer we would have engineer onsite replace the LCD Panel.

Computer communications

Different kinds of computers use different methods, or protocols, to c

ommunicate with each other. Macintosh computers use the AppleTalk protocol. Ma

cintosh as well as PCs can use TCP/IP to share information on the Internet. So

me PCs require a Network Operating System (NOS) to communicate. For example, N

ovell Netware is a popular NOS with PC users.

6.24

46 家中外知名企业面试题目-3

意法半导体软件试题

A Test for The C Programming Language

I. History

1. C was originally designed for and implemented on the (what) operating s

ystem on the DEC PDP-11, by (who) .

2. The most recently approved ANSI/ISO C standard was issued in (when) , a

nd single line comments notation “//” is or isn’t a feature of C89.

II. Syntax and Semantics

1. In a runtime C program, auto variables are stored in , static variables

are stored in , and function parameters are stored in .

a. stack b. heap c. neither stack nor heap

2. The statement “extern int x;” is a , and the keyword extern is used d

uring .

a. variable declaration b. variable definition

c. compilation time d. runtime

3. There is a complicated declaration: void (* signal (int, void (*)(int)

)) (int);

If a statement “typedef void (*p) (int);” is given, please rewrite this

complicated declaration.

4. The following code is a segment of C program.

```
.....  
void func(int *p)  
{.....}  
.....  
main()  
{  
  int num=0;  
  .....  
  func(&num);  
  .....  
}  
.....
```

Here, the function argument “&num” is passed .

a. by value b. by reference

III. Practice

Create a tree, which has h ($h > 0$) layers, and its each node has w ($w > 0$) sub

-nodes.

Please complete the following incomplete solution.

```
#include <stdlib.h>  
#include <string.h>  
struct tree{  
  char info;  
  p_sub; //link to sub-nodes  
};  
//allocate memory and initiate  
void dnode ( struct tree* tmp )  
{  
  = malloc( sizeof (struct tree) );  
  = 0x41;  
  = NULL;  
}  
struct tree *dtree (struct tree* subtree, int height, int width)  
{  
  int i;  
  if ( !subtree ) //if necessary, allocate memory for subtree  
    denode(subtree);  
  if ( height == 1 )
```



```

return subtree;
else if ( height == 2 ) {
    struct tree *leaf = NULL;

    for ( i=0; i<width; i++ ) {
        denode ( );
        ;
        leaf = NULL;
    }

    return subtree;
}
else {

    for ( i=0; i<width; i++ ) {

    }

    return subtree;
}
}
main()
{
    .....
    struct tree *root = NULL;
    root = dtree (root, h, w) ; // h and w are integers get from input
    .....
}

```

Sony 笔试题

1. 完成下列程序

```

*
*.*
*..*..
*...*...
*...*...*...
*...*...*...*...
*...*...*...*...*...
*...*...*...*...*...*...
*...*...*...*...*...*...*...
*...*...*...*...*...*...*...*...
#include <stdio.h>
#define N 8
int main()

```

```

{
int i;
int j;
int k;
-----

||
||
||
-----

return 0;
}

```

2. 完成程序，实现对数组的降序排序

```

#include <stdio.h>
void sort( );
int main()
{
int array[]={45, 56, 76, 234, 1, 34, 23, 2, 3}; //数字任意给出
sort( );
return 0;
}
void sort( )
{
-----

||
||
|-----|
}

```

3. 费波那其数列，1，1，2，3，5.....编写程序求第十项。可以用递归，也可以用其

他方法，但要说明你选择的理由。

```

#include <stdio.h>
int Pheponatch(int);
int main()
{
printf("The 10th is %d",Pheponatch(10));
return 0;
}
int Pheponatch(int N)
{
-----

||
||
-----

}

```

4. 下列程序运行时会崩溃，请找出错误并改正，并且说明原因。

```
#include <stdio.h>
#include <malloc.h>
typedef struct{
    TNode* left;
    TNode* right;
    int value;
} TNode;
TNode* root=NULL;
void append(int N);
int main()
{
    append(63);
    append(45);
    append(32);
    append(77);
    append(96);
    append(21);
    append(17); // Again, 数字任意给出
}
void append(int N)
{
    TNode* NewNode=(TNode *)malloc(sizeof(TNode));
    NewNode->value=N;

    if(root==NULL)
    {
        root=NewNode;
        return;
    }
    else
    {
        TNode* temp;
        temp=root;
        while((N>=temp.value && temp.left!=NULL) || (N<temp.value && temp.right
!=NULL
        ))
        {
            while(N>=temp.value && temp.left!=NULL)
            temp=temp.left;
            while(N<temp.value && temp.right!=NULL)
            temp=temp.right;
        }
    }
}
```

```

if(N>=temp.value)
temp.left=NewNode;
else
temp.right=NewNode;
return;
}
}
6.25

```

华为笔试题

1. 请你分别画出 OSI 的七层网络结构图和 TCP/IP 的五层结构图。
2. 请你详细地解释一下 IP 协议的定义，在哪个层上面？主要有什么作用？TCP 与 UDP 呢

?

3. 请问交换机和路由器各自的实现原理是什么？分别在哪个层次上面实现的？
4. 请问 C++ 的类和 C 里面的 struct 有什么区别？
5. 请讲一讲析构函数和虚函数的用法和作用。
6. 全局变量和局部变量有什么区别？是怎么实现的？操作系统和编译器是怎么知道的

?

7. 8086 是多少位的系统？在数据总线上是怎么实现的？

联想笔试题

1. 设计函数 `int atoi(char *s)`。
2. `int i=(j=4,k=8,l=16,m=32); printf("%d", i);` 输出是多少？
3. 解释局部变量、全局变量和静态变量的含义。
4. 解释堆和栈的区别。
5. 论述含参数的宏与函数的优缺点。

普天 C++ 笔试题

1. 实现双向链表删除一个节点 P，在节点 P 后插入一个节点，写出这两个函数。
2. 写一个函数，将其中的 \t 都转换成 4 个空格。
3. Windows 程序的入口是哪里？写出 Windows 消息机制的流程。
4. 如何定义和实现一个类的成员函数为回调函数？
5. C++ 里面是不是所有的动作都是 `main()` 引起的？如果不是，请举例。
6. C++ 里面如何声明 `const void f(void)` 函数为 C 程序中的库函数？
7. 下列哪两个是等同的

`int b;`

A `const int* a = &b;`

B `const* int a = &b;`

C `const int* const a = &b;`

D `int const* const a = &b;`

8. 内联函数在编译时是否做参数类型检查？

```

void g(base & b){
    b.play;
}
void main(){
    son s;
    g(s);
    return;
}

```

大唐电信

DTT 笔试题

考试时间一小时，第一部分是填空和选择：

1. 数列 6, 10, 18, 32, “?”, 问“?”是几?

答案: 54 解释:

1. 我来解释 54: $10 = 6 * 1.5 + 1$, $18 = 10 * 1.5 + 3$, $32 = 18 * 1.5 + 5$, $54 = 32 * 1.5 + 7$ 再解释一个 57: $18 = 2 * 10 - 0.5 * (10/6 \text{ 的余数})$, 以此类推。

2.

$$6 = 2 * 3$$

$$10 = 2 * 5$$

$$18 = 2 * 9$$

$$32 = 2 * 16$$

观察数列 3, 5, 9, 16

$$5 - 3 = 2$$

$$9 - 5 = 4$$

$$16 - 9 = 7$$

再观察数列 2, 4, 7

$$4 - 2 = 2$$

$$7 - 4 = 3$$

由此我可以说 2, 4, 7 是差加 1 的数列, 该数列的下一项是 11

那么数列 3, 5, 9, 16 的下一项就是 27

所以原数列 6, 10, 18, 32, 的下一项就是 $27 * 2 = 54$

2. 某人出 70 买进一个 x, 80 卖出, 90 买回, 100 卖出, 这桩买卖怎么样?

分两次, 各挣 10

共挣 20,

不清楚可假设他一开始有 100 元钱。到最后有 120 元

3. 月球绕地球一圈, 至少要多少时间?

应该是一恒星月, 约 27.32166 日, 而不是一回归月 (约 27.32158 日)。

竟然还有两个人说是回归月, 看来不得不给这两位说一下回归月和恒星月的区别了。

回归月是月球连续两次经过零度黄经线的时间间隔, 而零度黄经线是在缓慢地向西移动的 (这种现象叫岁差), 所以这个周期其实把地轴的进动也算在内了; 恒星月是月球相对于恒星背景运动的周期, 排除了地球的公转和岁差, 这才是真正的月球的公转周期。正如恒星年才是地球的公转周期, 而回归年不是一样。

4. 7 个人用 7 小时挖了 7 米的沟, 以同样的速度在 50 小时挖 50 米的沟要多少人?

7 个人 1 小时 1 米
1 个人 1 小时 $1/7$ 米
1 个人 50 小时 $50/7$ 米
50 米为 $350/7$ 米
350 除与 50 等于 7
所以 7 个人 50 小时 50 米

或:

7 个人 1 小时 1 米
7 个人 50 小时 50 米

5. 鱼头长 9, 鱼尾等于鱼头加半个鱼身, 鱼身等于鱼头加鱼尾, 问鱼全长多少?

设鱼身长 X

鱼尾长 Y

$$Y = 9 + 1/2 X$$

$$X = 9 + Y$$

解得: $X=36$, $Y=27$

$$\text{全长} = 9 + 36 + 27 = 72$$

6. 一个小姐买了一块手表, 回家发现手表比她家的表慢了两分钟, 晚上看新闻的时候又发现她家的表比新闻里的时间慢了两分钟, 则。

A 手表和新闻里的时间一样

B 手表比新闻里的时间慢

C 手表比新闻里的时间快

?? C

电台一小时 60 分钟, 那么闹钟一小时就是 58 分. 将 58 分钟平均分 60 份, 也就是说闹钟的一分钟其实是 $58/60$ 分钟. 而手表就是比闹钟快标准的 $58/30$ 分钟, 小于实际的两分钟, 所以比标准慢.

7. 王先生看到一则招聘启事, 发现两个公司除了以下条件不同外, 其他条件都相同

A 半年年薪 50 万, 每半年涨 5 万

B 一年年薪 100 万, 每一年涨 20 万

王先生想去一家待遇比较优厚的公司, 他会去哪家?

第一年

$$a: 50 + 55 = 105$$

$$b: 100 = 100$$

第二年

$$a: 105 + 60 + 65 = 230$$

$$b: 100 + 120 = 220$$

第三年

a: $230+70+75=375$

b: $220+140=360$;

第四年

a: $375+80+85=540$

b: $360+160=520$

.....

实际每年的工资

a: (上半年的工资+10)*2+5

b: 上年度工资+20

所以

10. 问哪个袋子里有金子?

A 袋子上的标签是这样写的: B 袋子上的话是对的, 金子在 A 袋子。

B 袋子上的标签是这样写的: A 袋子上的话是错的, 金子在 A 袋子里。

金子在 A 袋。

在 A 里: B 对, 则 A 错, 则 B 错, 则 A 对, 从而可以退出 A、B 都对, 所以金子在 A 里

11. 3 个人住酒店 30 块钱, 经理找回 5 块钱, 服务生从中藏了 2 块钱, 找给每人 1 块钱,

1) $+2=29$, 问这是怎么回事? $-3 \times (10$

经理找回 5 块, 说明 3 人住宿要 $30-5=25$ 元. 然后加服务员藏的 2 元. 每人分的 1 元. 即 $25+2+1+1+1=30$

12. 三篇写作, 均为书信形式。

(1) 一片中文的祝贺信, 祝贺某男当了某公司 xx

(2) 两篇英文的, 一是说有事不能应邀, 派别人去; 另一篇是讨债的, 7 天不给钱就

原版加答疑

注意: 如果一答案不符合的情况, 请留言或发邮件 (zhuseahui@yahoo.com.cn) 告诉我, 谢谢合作!

走人 (主要考 business letter 格式)。

6.26

大唐面试题

1. 什么是中断? 中断发生时 CPU 做什么工作?
2. CPU 在上电后, 进入操作系统的 main() 之前必须做什么工作?
3. 简述 ISO OSI 的物理层 Layer1, 链路层 Layer2, 网络层 Layer3 的任务。
4. 有线电话和无线电话有何区别? 无线电话特别需要注意的是什么?
5. 软件开发五个主要 step 是什么?
6. 你在开发软件的时候, 这 5 个 step 分别占用的时间百分比是多少?
7. makefile 文件的作用是什么?
8. UNIX 显示文件夹中, 文件名的命令是什么? 能使文件内容显示在屏幕的命令是什么

?

9. (选做) 手机用户在从一个基站漫游到另一个基站的过程中, 都会发生什么?

网通笔试题

选择题 (每题 5 分, 只有一个正确答案)

1. 中国 1 号信令协议属于 的协议。

A ccs B cas C ip D atm

2. isdnpri 协议全称是 。

A 综合业务模拟网基速协议

B 综合业务模拟网模拟协议

C 综合业务数字网基率协议

D 综合业务数字网基次协议

3. 路由协议中, 协议是用距离作为向量的。

A ospf B bgp C is-is D rip

4. 中国智能网中, ssp 与 scp 间最上层的 ss7 协议是 。

A incs B is41b C is41c D inap

5. dtmf 全称是 。

A 双音多频 B 多音双频 C 多音三频 D 三音多频

6. 计算机的基本组成部分中, 不包含下面设备的是 。

A cpu B 输入设备 C 存储器 D 接口

7. 脉冲编码调制的简称是 。

A pcm B pam C (delta)M D atm

8. 普通电话线接口专业称呼是 。

A rj11 B rj45 C rs232 D bnc

9. 现有的公共数据网都采用 。

A 电路交换技术 B 报文交换技术

C 语音插空 D 分组交换

10. ss7 协议中的制止市忙消息简写为 。

A stb B slb C sub D spb

简答题 (每题 10 分)

1. 简述普通电话与 IP 电话的区别。

2. 简述随路信令与公路信令的根本区别。

3. 说明掩码的主要作用。

4. ss7 协议中, 有三大要素决定其具体定位, 哪三大要素?

5. 描述 ss7 的基本通话过程。

6. 简述通信网的组成结构。

7. 面向连接与面向非连接各有何利弊?

8. 写出爱尔兰的基本计算公式。

9. 数据网主要有哪些设备?

10. 中国一号协议是如何在被叫号码中插入主叫号码的?

东信笔试题目

笔试: 30 分钟。

1. 压控振荡器的英文缩写。

2. 动态随机存储器的英文缩写。

3. 选择电阻时要考虑什么？
4. 单片机上电后没有运转，首先要检查什么？
5. 计算机的基本组成部分及其各自的作用。
6. 怎样用 D 触发器、与或非门组成二分频电路？

中软融鑫笔试题

1. 关于工作

- (1) 你对未来的工作生活是怎样憧憬的？为何选择我公司作为求职公司？
- (2) 请用不超过 30 个字给出一个最能让我们录用你的理由。
- (3) 你认为比较理想的工作环境是怎样的？
- (4) 你个人的中长期的职业发展目标是怎样的？

2. 关于社会

- (1) 如果你是杨利伟，你在太空中向祖国人民说的第一句话是什么？
- (2) 宋美龄女士于 2003 年 10 月谢世，对这位著名人士在西安事变中的态度和作用，你

是如何看待的？（不超过 300 字）

- (3) 北京政府颁布的对拾金不昧者，失主奖励相当于财产 20% 奖金的公告，你是如

何看的？

- (4) 如果给你 50 万元人民币，你将会用这些钱做什么？
- (5) 在美国，男、女卫生间（厕所）的正确称呼为什么？请用英语写出答案。
- (6) 你认为麦当劳是世界最大的汉堡生产商吗？如果不是，请说出你的观点。

3. 教育背景

- (1) 你受过哪些正规的教育或培训？（自高中毕业起）
- (2) 在校期间进行过哪些社会活动？

6.27

Delphi 笔试题目

机械类笔试试题

1. Briefly describe what is blanking(cutting), forming, coining and emboss

ing in stamping process.

2. What is metal cladding?
3. What is the purpose of adding glass fiber to thermoplastic material?
4. In contrast with metal and thermoplastic material, which has a higher co

efficient of thermal expansion(CTE).

5. The most suitable material for a integral hinge design (typical plastic

thickness=0.25 to 0.5mm at hinge)

6. Can a bending load makes both compressive and tensile stress in a membe

r?

7. What is the design criteria used in plastics catch/snap?
8. What is FEA?
9. Why is natural frequency important in vibration analysis?
10. What is the deflection equation of a cantilever beam fixed at one edge

?

EE 笔试试题

1. Name 3 Vehicle Buses.
2. Name 2 possible sources of Electromagnetic interference on Electronics

Circuit ASM.

3. Wavelength for 12MHz frequency signal is _____
4. Name 2 important considerations for car radio performance related to

audio signal processing under multipath condition?

5. What is the typical FM receiver RF signal strength to achieve 30dB S/N

for car radio?

6. When a radio is tuned to 98.1 MHz & with a LO of 108.8 MHz, what is the

image frequency?

7. For a system with a matched impedance, what is the Reflection Coefficient

and SWR?

8. Which property of the output capacitor is the primary cause of Low Drop

Out(LDO) regulator loop instability?

- (1) Equivalent series resistance(ESR)
- (2) Effective series inductance(ESL)
- (3) Capacitance value
- (4) Dielectric material

9. The switching regulator is capable of:

- (1) Higher power conversion efficiency
- (2) Providing an output voltage that is higher than the input
- (3) Generating an output voltage opposite in polarity to the input
- (4) All of the above

10. A linear regulator op $V_{in(max)} = 10V$, $V_{out(min)} = 4.8V$, $I_{out(max)} = 2.$

5mA, $I_q(max) = 2.5mA$, $T_a(max) = 85^\circ C$, The regulator is available in 3 pac

kages. Each package has the following thermal characteristics:

Package R_{ja} ($^\circ C/W$) R_{jc} ($^\circ C/W$)

SO14 125 30

Choose the most suitable package to handle the power dissipation requirements without a heat sink and why.

软件笔试题

1. How do you code an infinite loop in C?

2. Volatile:

(1) What does the keyword volatile mean? Give an example

(2) Can a parameter be both const and volatile? Give an example

(3) Can a pointer be volatile? Give an example

3. What are the values of a, b, and c after the following instructions:

```
int a=5, b=7, c;
```

```
c = a+++b;
```

4. What do the following declarations mean?

(1) const int a;

(2) int const a;

(3) const int *a;

(4) int * const a;

(5) int const * a const;

5. Which of the following statements describe the use of the keyword static?

c?

(1) Within the body of a function: A static variable maintains its value

between function revocations

(2) Within a module: A static variable is accessible by all functions within that module

hin that module

(3) Within a module: A static function can only be called by other functions within that module

ons within that module

6. Embedded systems always require the user to manipulate bits in registers or variables. Given an integer variable a, write two code fragments.

The first should set bit 5 of a. The second should clear bit 5 of a. In both cases, the remaining bits should be unmodified.

7. What does the following function return?

```
char foo(void)
```

```
{
```

```
    unsigned int a = 6;
```

```
    int b = -20;
```

```
    char c;
```

```
    (a+b > 6) ? (c=1) : (c=0);
```

```
return c;
}
```

8. What will be the output of the following C code?

```
main()
{
    int k, num= 30;
    k =(num > 5 ? (num <=10 ? 100:200): 500);
    printf(“%d”, k);
}
```

9. What will the following C code do?

```
int *ptr;
ptr =(int *)0x67a9;
*ptr = 0xaa55;
```

10. What will be the output of the follow C code?

```
#define product(x) (x*x)
main()
{
    int i = 3, j, k;
    j = product(i++);
    k = product(++i);
    printf(“%d %d”, j,k);
}
```

11. Simplify the following Boolean expression

$\neg((i = 12) \parallel (j > 15))$

12. How many flip-flop circuits are needed to divide by 16?

13. Provides 3 properties that make an OS, a RTOS?

14. What is pre-emption?

15. Assume the BC register value is 8538H, and the DE register value is 62

A5H. Find the value of register BC after the following assembly operations:

```
MOV A,C
SUB E
MOV C,A
MOV A,B
SBB D
MOV B,A
```

16. In the Assembly code shown below

```
LOOP: MVI C,78H
      DCR C
      JNZ LOOP
      HLT
```

How many times is the DCR C Operation executed?

17. Describe the most efficient way (in term of execution time and code si

æ) to divide a number by 4 in assembly language

18. what value is stored in m in the following assembly language code frag

ment if n=7?

```
LDAA #n
```

```
LABEL1: CMPA #5
```

```
BHI L3
```

```
BEQ L2
```

```
DECA
```

```
BRA L1
```

```
LABEL2: CLRA
```

```
LABEL3: STAA #m
```

19. What is the state of a process if a resource is not available?

```
#define a 365*24*60*60
```

20. Using the #define statement, how would you declare a manifest constant

that returns the number of seconds in a year? Disregard leap years in your an

swer.

21. Interrupts are an important part of embedded systems. Consequently, ma

ny compiler vendors offer an extension to standard C to support interrupts. Ty

pically, the keyword is __interrupt. The following routine (ISR). Point out pr

blems in the code.

```
__interrupt double compute_area (double radius)
{
    double area = PI * radius * radius;
    printf("\nArea = %f", area);
    return area;
}
```

Hongkong Bank 笔试题

1. Please state why you chose to follow these activities and how they have

contributed to your personal development. You may wish to give details of you

r role whether anyone else was involved and any difficulties you encountered.

2. Please state how you have benefited from your work experience.

3. How much is your present monthly salary including allowances.
4. Do you need to compensate your present employer if you resign? If so, please give details.

lease give details.

5. Other than academic success, what has been your greatest achievement to date? What do you see as your personal strength, why?
6. Please state why the position you have applied for is appropriate for you; Why you have selected HongKong Bank and what your career objectives are.

A.T. Kearney 笔试题

1. Describe your greatest achievement in the past 4-5 years?
2. What are your short-term and long-term career objectives? What do you think is the most ideal job for you?
3. Why do you want to join A.T. Kearney? What do you think you can contribute to A.T. Kearney?
4. Why are you applying for a position at Arthur Anderson?
5. What are your expectations of our firm.
6. Describe your hobbies and interests.

Shell company 笔试题

1. How would your colleagues/classmates describe you in five words? On what evidence would they base this assessment.
2. If you are asked to recruit the best graduates for Shell, what would you do to attract them? What would you do to select them?
3. Please describe a new activity that you have initiated and implemented.

Please highlight your role out.

4. Please describe your outstanding non-academic achievements.
5. Please describe any other significant activities you have been involved in including organizing people.
6. Imagine that Shell has found oil in an inland province of China, near a large river. You are responsible for planning how to transport the oil to the coast thousands of miles away. What are the main issue you would consider, and

d what would you do?

6.28

46 家中外知名企业面试题目-4

KPMG 笔试题

“The big economic difference between nuclear and fossil-fuelled power stations is that nuclear reactors are more expensive to build and decommission, but much cheaper to run. So disputes over the relative efficiency of the two systems revolve not just around prices of coal and uranium today and tomorrow, but also around the way in which future income should be compared with current income.”

1. The main difference between nuclear and fossil-fuelled power stations is an economic one.

TRUE

UNTRUE

CANNOT SAY

2. The price of coal is not relevant to discussions about the relative efficiency of nuclear reactors.

TRUE

UNTRUE

CANNOT SAY

3. If nuclear reactors were cheaper to build and decommission than fossil-fuelled power stations, they would definitely have the economic advantage.

TRUE

UNTRUE

CANNOT SAY

“At any given moment we are being bombarded by physical and psychological stimuli competing for our attention. Although our eyes are capable of handling more than 5 million bits of data per second, our brain are capable of interpreting only about 500 bits per second. With similar disparities between each o

f the other senses and the brain, it is easy to see that we must select the visual, auditory, or tactile stimuli that we wish to compute at any specific time.”

4. Physical stimuli usually win in the competition for our attention.

TRUE

UNTRUE

CANNOT SAY

5. The capacity of the human brain is sufficient to interpret nearly all the stimuli the senses can register under optimum conditions.

TRUE

UNTRUE

CANNOT SAY

6. Eyes are able to cope with a greater input of information than ears.

TRUE

UNTRUE

CANNOT SAY

VERBAL ANSWER :

(1) C CANNOT SAY

(2) B UNTRUE

(3) A TRUE

(4) C CANNOT SAY

(5) B UNTRUE

(6) C CANNOT SAY

PartII NUMERICAL TEST

1. Which country had the highest number of people aged 60 or over at the start of 1985?

A. UK

B. France

C. Italy

D. W. Germany

E. Spain

2. What percentage of the total 15mm button production was classed as sub-

standard in September?

AA 10.5% BB 13% CC 15% DD 17.5% EE 20% AB 23.5% AC 25%

AD 27.5% AE 28% BC 30.5%

3. How many live births occurred in 1985 in Spain and Italy together (to the

nearest 1000)?

A. 104 000

B. 840 000

C. 1 044 000

D. 8 400 000

E. 10 440 000

4. What was the net effect on the UK population of the live birth and death rates in 1985?

A. Decrease of 66 700

B. Increase of 752 780

C. Increase of 84 900

D. Cannot Say

E. Increase of 85 270

5. By how much did the total sales value of November's button production vary from October's?

A. 8.50 (Decrease)

B. 42.50 (Decrease)

C. 85.00 (Increase)

D. 27.50 (Decrease)

E. No change

6. What was the loss in potential sales revenue attributable to the production of sub-standard (as opposed to standard) buttons over the 6 month period?

A. 13.75

B. 27.50

C. 137.50

D. 280.00

E. 275.00

香港电信笔试题

1. Based on your understanding of the following java related technologies:

servlets, JavaServerPage, JavaBeans, Enterprise JavaBeans, how do you think these technologies work together or are applied in the development of an internet-based application (25marks).

2. In your opinion, what do you think are the advantages or benefits of using an object-oriented approach to software development? how do you think those benefits can be achieved or realized? (15marks).

3. In designing your classes, given the choice between inheritance and aggregation which do you choose (15marks).

4. How would you work around the lack of multiple inheritance feature in Java (15marks).
5. What would you consider to be the hardest part of OO analysis and design and why (10marks).
6. How do you keep yourself up to date with the latest in software technology, especially in the field of software development (10marks).
7. What is your career aspiration? Why do you think this E-Commerce Development Center can help you in achieving your career goals (10marks) (1hr, answer in English).

OR 慈 L 的笔试题

1. Would you please describe yourself in 3-4 lines? (limited in 500 words)
2. Could you tell us why we should choose you as a Loreal Person, and what makes you unique? (limited in 500 words)
3. What is your short-term and long-term career plan? (limited in 500 words)
4. What kind of group activities are you interested in and what type of role do you often play? (limited in 500 words)
5. Please use one sentence to give a definition of 'Beauty', and describe the most beautiful thing in your life. (limited in 500 words)

维尔 VERITAS 软件笔试题

1. A class B network on the internet has a subnet mask of 255.255.240.0, what is the maximum number of hosts per subnet .
a. 240 b. 255 c. 4094 d. 65534
2. What is the difference: between $O(\log n)$ and $O(\log n^2)$, where both logarithms have base 2 .
a. $O(\log n^2)$ is bigger b. $O(\log n)$ is bigger
c. no difference
3. For a class what would happen if we call a class's constructor from within the class?

th the same class's constructor .

- a. compilation error b. linking error
 - c. stack overflow d. none of the above
4. "new" in c++ is a: .
- a. library function like malloc in c
 - b. key word c. operator
 - d. none of the above
5. Which of the following information is not contained in an inode .
- a. file owner b. file size
 - c. file name d. disk address
6. What's the number of comparisons in the worst case to merge two sorted

lists containing n elements each .

- a. $2n$ b. $2n-1$ c. $2n+1$ d. $2n-2$
7. Time complexity of n algorithm $T(n)$, where n is the input size ,is $T(n)$

$=T(n-1)+1/n$ if $n>1$ otherwise 1 the order of this algorithm is .

- a. $\log(n)$ b. n c. n^2 d. n^n
8. The number of 1's in the binary representation of $3*4096+ 15*256+5*16+$

3 are .

- a. 8 b. 9 c. 10 d. 12

百威啤酒(武汉公司)

- 1,为什么申请来百威?
- 2,将来有什么打算?
- 3,有没有社会活动经历?
- 4,有没有当众演讲的经历?
- 5,经常使用那些软件?
- 6,喜欢哪些课程?
- 7,你认为工作中的什么因素对你来说最重要?
- 8,什么时候可以来上班?可以在这里工作多久?
- 9,八点上班,要加班和出差,能不能做到?

星巴克

- 1、 您是一家咖啡店的店经理，你发现店内同时出现下列状况：
 - 1) 许多张桌子桌面上有客人离去后留下的空杯未清理，桌面不干净待整理。
 - 2) 有客人正在询问店内卖哪些品种，他不知如何点咖啡菜单。
 - 3) 已有客人点完成咖啡，正在收银机旁等待结帐。
 - 4) 有厂商正准备要进货，需要店经理签收。请问，针对上述同时发生的情况，你要如何排定处理之先后顺序，为什么
- 2、 有一位甲员工脾气不好以致在前三家店因为与店内其他同事相处不佳而屡屡调动

，现在甲被调到你的店里面来，请问身为店经理的你，将如何应对??

- 3、 你是店经理，本周五结帐后，发现门市总销售额较上周五减少 30%，请问可能原因

会是哪几种，各原因如何应对？

凹凸电子软件笔试题

1. Select ONE of the following projects to discuss:

a. Signal Filtering: You are given a sampled realtime waveform consisting

of a sensor reading mixed with highly periodic impulses and high frequency noise.

The desired output is the realtime filtered sensor signal with the impulses

and noise removed, and a readout of the impulse period. The FFT may not be used.

ed.

b. Interrupt Processing. A hardware register consisting of eight independent

edge triggered latches is used to record external asynchronous interrupt requests.

When any of the request bits are latched, a software interrupt is generated.

The software may read the latch to see which interrupt(s) occurred. Writing

a one to any latch bit will clear the latch. How does that software assure

that no interrupt request is ever missed?

c. User Interface: a prototype MP3 player interface consisting of a playlist

display and a few control buttons is given to you. How would you make the interface

“skinnable”, with user selected graphics, options, and control button

placement?

Each project description is incomplete. What questions would you ask to completely

specify the project? What development tools would you prefer to use?

What algorithm /data structures/design would you use?

2. What program(s) have you coded for your own enjoyment (not part of a school

ool project, not for pay). What type of software project would you most enjoy w

orking on?

3. Have you participated in a team programming project? What is the hardes

t part of programming as a team, as opposed to programming alone?

友立资讯笔试题目

1. 一堆鸡蛋，3 个 3 个数剩余 2 个，5 个 5 个数剩余 1 个，7 个 7 个数剩余 3 个，问这堆鸡蛋

最少有多少个？并给出通解。

2. 列举五岳，及其所在省份。
3. 何为四书。
4. 按顺序默写 24 节气。
5. 默写于谦的《吟石灰》。
6. 英语翻译约 300 字。
7. 作文一篇：求职有感。

普华永道 PWC 笔试题目（作文）

1. 最近 10 年来中国媒体的变化。
2. 你认为发展汽车产业和公共交通哪个更重要？
3. 如何理解风险投资？
4. 如何理解广告的消极作用和积极作用？

Avant! 微电子 EE 笔试题

1. 名词解释：VLSI，CMOS，EDA，VHDL，Verilog，HDL，ROM，RAM，DRC，LVS。
2. 简述 CMOS 工艺流程。
3. 画出 CMOS 与非门的电路，并画出波形图简述其功能。
4. 画出 N 沟道增强型 MOSFET 的剖面图。
5. 简述 ESD 和 latch-up 的含义。
6. 简述三极管与 MOS 管的区别。
7. 简述 MOORE 模型和 MEALY 模型。
8. 简述堆栈与队列的区别。

奇码数字信息有限公司笔试题

1. 画出 NMOS 的特性曲线（指明饱和区，截至区，线性区，击穿区和 C-V 曲线）
2. 2.2um 工艺下， $K_n = 3K_p$ ，设计一个反相器，说出器件尺寸。
3. 说出制作 N-well 的工艺流程。
4. 雪崩击穿和齐纳击穿的机理和区别。
5. 用 CMOS 画一个 D 触发器（clk，d，q，q-）。

德勤笔试题

五个人来自不同地方，住不同房子，养不同动物，吸不同牌子香烟，喝不同饮料，喜

欢不同食物。根据以下线索确定谁是养猫的人。

- (1) 红房子在蓝房子的右边，白房子的左边（不一定紧邻）
- (2) 黄房子的主人来自香港，而且他的房子不在最左边。
- (3) 爱吃比萨饼的人住在爱喝矿泉水的人的隔壁。
- (4) 来自北京的人爱喝茅台，住在来自上海的人的隔壁。
- (5) 吸希尔顿香烟的人住在养马的人右边隔壁。
- (6) 爱喝啤酒的人也爱吃鸡。
- (7) 绿房子的人养狗。
- (8) 爱吃面条的人住在养蛇的人的隔壁。
- (9) 来自天津的人的邻居（紧邻）一个爱吃牛肉，另一个来自成都。
- (10) 养鱼的人住在最右边的房子里。
- (11) 吸万宝路香烟的人住在吸希尔顿香烟的人和吸“555”香烟的人的中间（紧邻）
- (12) 红房子的人爱喝茶。
- (13) 爱喝葡萄酒的人住在爱吃豆腐的人的右边隔壁。
- (14) 吸红塔山香烟的人既不住在吸健牌香烟的人的隔壁，也不与来自上海的人相邻。
- (15) 来自上海的人住在左数第二间房子里。
- (16) 爱喝矿泉水的人住在最中间的房子里。
- (17) 爱吃面条的人也爱喝葡萄酒。
- (18) 吸“555”香烟的人比吸希尔顿香烟的人住的靠右。

扬智（科技）笔试题目

软件题目

1. Queue is a useful structure

* What is a queue?

* Write 5 operations or functions, without details, that can be done on a

queue.

2. Insert a sequence of keys(24,49,13,20,59,23,90,35) into a data structure,

which has no keys initially. Depict the data structure after these insertions, if it is:

* a heap tree

* an AVL tree

3. * What is a synchronous I/O bus?
- * What is an asynchronous I/O bus?
- * Compare the advantages and disadvantages of synchronous and asynchronous I/O bus.

s I/O bus.

4. Explain the following terminology:
 - * Baud rate
 - * Handshaking
 - * Memory mapped I/O
5. Explain the key issues in supporting a real-time operation system for embedded system.

embedded system.

6. Explain the mapping of virtual addresses to real addresses under paging by

- * direct mapping
- * associative mapping
- * combined direct/associative mapping

7. Please explain what is “write-back” and “write-through”, and discuss

the advantage and disadvantage about these two methods.

8. Explain the concept and benefit of threads
9. What is hardware interrupt? What is software interrupt? What is exception?

on? Please tell me all you know about interrupt.

10. Write a recursive function that tests whether a string is a palindrome.

A palindrome is a string such as “abcba” or “otto” that reads the same in

both directions. If you can write this function recursively, you can write an iterative version of this function instead.

11. 什么是进程（Process）和线程（Thread）？有何区别？
12. MFC 和 SDK 有何区别？
13. IRP 是什么？有何作用？
14. Windows 2000 操作系统下用户模式和内核模式下编程有何区别？
15. 驱动程序的 BUFFER 能 swap 到磁盘上去吗？为什么？
16. 试编写 3 个函数实现
 - (1) 建立一个双向链表
 - (2) 插入一个节点
 - (3) 删除一个节点
17. 简述 Hardware interrupt 和 software 中断的区别，简述其应用。
18. 试编写一个函数，计算一个字符串中 A 的个数。
19. 画出其相应流程图并编写一个函数实现一个整数到二进制数的转换，如输入 6，输出 110。

出 110。

20.

- (1) 编写一个递归函数，删除一个目录。
- (2) 编写一个非递归函数，删除一个目录。

并比较其性能。

21. 附加题：简单叙述编程经历

硬件题目

1. 用 mos 管搭出一个二输入与非门。
2. 集成电路前段设计流程，写出相关的工具。
3. 解释名词 IRQ, BIOS, USB, VHDL, SDR。
4. 简述如下 Unix 命令 cp -r, rm, uname。
5. 用波形表示 D 触发器的功能。
6. 写异步 D 触发器的 verilog module。
7. What is PC Chipset?
8. 用传输门和倒向器搭一个边沿触发器。
9. 画状态机，接受 1, 2, 5 分钱的卖报机，每份报纸 5 分钱。

DSP 题目

1) $y(n) = b \cdot \delta(n) - a \cdot h(n-1)$ 1. $H(n)$

- (1) 求 $h(n)$ 的 z 变换
 - (2) 该系统是否为稳定系统
 - (3) 写出 FIR 数字滤波器的差分方程
2. 写出下面模拟信号所需的最小采样带宽
- (1) 模拟信号的频率范围是 0~4kHz
 - (2) 模拟信号的频率范围是 2~4kHz
3. 名词解释
- (1) 量化误差
 - (2) 直方图
 - (3) 白平衡
 - (4) MMX
4. 写出下面几种格式中用到的压缩技术
- (1) JPEG
 - (2) MPEG2
 - (3) MP3

6.29

高通笔试题

1. Can you describe the trend of wireless mobile communication industry? (

2000 letters)

2. Compare the major third generation technologies.(2000 letters)
3. Describe the characteristics of Walsh function. Explain how to generate

Walsh Function. (2000 letters)

4. List factors that will affect the capacity of forward and reverse links

of a CDMA system. (2000 letters)

5. What are the differences between IS-95 A/B and cdma2000 1X? (2000 letters)

rs)

威盛笔试题

2002 年软件笔试题

1. 三组程序，找出你认为的错误。

(1) a.c long temp[255];

b.c extern *temp;

(2) a.c long temp[255];

b.c extern temp[256];

(3) a.c long temp[255];

b.c extern temp[];

2. 在第一个声明处编译出了奇怪的错误，为什么？

```
#include <stdio.h>
```

```
#include "myfun1.h"
```

```
#include "myfun2.h"
```

```
int myInt1;
```

```
int myInt2;
```

3. printf("0x%x", (&1)); 请问打印了什么？ -0[

4. 汇编，用 ax,bx,cx,dx，求 $1000 \times 1000 / 30$ （四舍五入），结果放在 ax 中。

5. 编最优化 Bubble(int *pIntArray,int L)，要求:交换元素不能用临时变量，如果

有序需要最优。

6. 用任意一种编程语言写 n!的算法。

2003 Asic 部分

1. 一个四级的 Mux，其中第二级信号为关键信号，如何改善 timing？

2. 一个状态机的题目用 Verilog 实现。

3. Asic 中的 design flow 的实现。

4. 用逻辑门画出 D 触发器。

— 5. 给出某个一般时序电路的图，有 Tsetup, Tdelay, Tck>q 还有 clock 的 del

ay, 写出决定最大时钟的因素，同时给出表达式。

6. 用 C 语言实现统计某个 cell 在某.v 文件调用的次数。

7. Cache 的主要部分。

2003 EE 笔试题目

1. 写出电流公式。

2. 写出平板电容公式。
3. 电阻 R 和电容 C 串联，输入电压为 R 和 C 之间的电压，输出电压分别为 C 上电压和 R 上电压，要求绘制这两种电路输入电压的频谱，判断这两种电路何为高通滤波器，何为低通滤波器。当 $RC \ll T$ 时，给出输入电压波形图，绘制两种电路的输出波形图。
4. 给出时域信号，求其直流分量。
5. 给出一时域信号，要求写出频率分量，并写出其傅立叶变换级数。当波形经过低通滤波器滤掉高次谐波而只保留一次谐波时，画出滤波后的输出波形。
6. 有一时域信号 $S = V_0 \sin(2\pi f_0 t) + V_1 \cos(2\pi f_1 t) + V_2 \sin(2\pi f_3 t + 90)$ ，写出当其通过低通、带通、高通滤波器后的信号表示方式。
7. 给出一差分电路，告诉其输出电压 Y_+ 和 Y_- ，求共模分量和差模分量。
8. 一电源和一段传输线相连（长度为 L ，传输时间为 T ），画出终端处波形，考虑传输线无损耗。给出电源电压波形图，要求绘制终端波形图。
9. 求锁相环的输出频率，给了一个锁相环的结构图。
10. 给出一个堆栈的结构，求中断后显示结果，主要是考堆栈压入返回地址存放在低端地址还是高端。

2003 Graphic 笔试题目

1. 问答题
 - (1) texture mapping 是什么？为什么要用 filter？
 - (2) 用 float 和 int 表示一个数，比如 2，说明优点和缺点。
 - (3) 在 MPEG 哪部分可以加速硬件？
 - (4) 解释 cubic 和 B-spline 的差别，写出各自函数。
 - (5) 写出几个 WinAPI 中的 OpenGL 函数。
 - (6) 说出固定小数表示和浮点小数表示的优缺点。
 - (7) 说出显卡可以优化哪些 MPEG 中的计算？
 - (8) 说出 Bezier 和 B-Spline 曲线的区别。
2. 用最简单的方法判断一个数是否是 2 的指数次幂。
3. S23E8 和 S10E5 两种浮点数表示方法分析，表示 0.25 写一个类 S10E5，实现从 S23E8 转换。
4. 用模版的方式实现三个量取最大值。
5. 题目告诉你 IEEE 16 和 32 浮点数表示的规范，要求将 0.25 分别用 IEEE 16 和 IEEE 32 表示并写一个 C++ 函数将输入的 IEEE 16 表示转化为 IEEE 32 的表示。

$x * 0.5$ 要求只能用整数操作。 = 6. 用 C 语言写一个函数 $f(x)$

2003 Software Engineer 笔试题

1. Describe x86 PC's architecture in a diagram cpu, core chipset, Cache, DR

AM, IO-subsystem, IO-Bus

2. SWI instruction is often called a “supervisor call”, describe the act

ions in detail

- * Save the address of the instruction after the SWI in `rl4_svc`.
 - * Save the CPSR in `SPSR_svc`.
 - * Enter supervisor mode and disable IRQs.
 - * Set the PC to 08 and begin executing the instruction there.
- 3.
- * What is PIO operation? advantage and disadvantage?
 - * DMA operation? advantage and disadvantage?
 - * Scatter/Gather DMA engine? how does it operate?
4. MP3 decoder related. (a flow chart of decoding is presented)
- * advantages of Huffman encoding?
 - * why the aliasing reduction is necessary?
 - * analytical expression in mathematics of the IMDCT?
 - * which block in the flow chart is suitable for the software implementatio

n and which for the hardware? why?

5. Assembly codes -> C language (about 15 lines).

6. Graduation thesis description.

6.30

46 家中外知名企业面试题目-5

汉王笔试题

高级研究人员（模式识别、图像处理类）招聘试题

说明：

可能您的专业并不完全符合本试题所涉及的领域。因此，并非所有的问题都需要回答

，您可以只回答您所熟悉和能够回答的问题。允许参考任意的资料，但请独立完成此试题

，我们更欣赏您独立的思考和创新的精神。本试题并非我们录用或者不录用您的惟一依据

。应聘高级研究人员者请回答这部分问题。

1. 人工智能与模式识别的研究已有多多年，但似乎公认的观点认为它仍然非常困难。试

对你所熟悉的任一方向（如指纹识别、人像识别、语音识别、字符识别、自然语言理解等

）的发展状况进行描述。并设想如果你将从事该方向的研究，你打算如何着手，以建立有

效的识别理论和方法；或者你认为现在的理论和方法有何缺陷，有什么办法来进行改进？

(500 字以内即可，不要太长)

2. 简述下面任一主题的主要理论框架或主要观点 (500 字以内即可，不要太长)

- (1) David Marr 的视觉计算理论框架
- (2) 格式塔 (Gestalt) 心理学派的主要观点
- (3) Bayes 决策理论
- (4) 人工神经网络中的 BP 网络、自组织网络和联想记忆网络的主要内容
- (5) 基因算法
- (6) 小波分析
- (7) 目前流行的有损静态图像压缩方法

3. 设想你要设计一个算法，检测给定的图像中是否有矩形结构。所要检测的矩形可能

有多种形态，试提出你的算法框架。要求你的算法至少能检测出样本中的矩形，而拒绝其

他的任意非矩形结构。矩形的大小、位置和方向未知，要求你的算法能确定这些参数。

如果你认为这个问题太难而不能解决，请说明理由。

高级软件开发人员招聘试题

说明：

可能您的专业并不完全符合本试题所涉及的领域。因此，并非所有的问题都需要回答

，您可以只回答您所熟悉和能够回答的问题。允许参考任意的资料，但请独立完成此试题

，我们更欣赏您独立的思考和创新的精神。本试题并非我们录用或者不录用您的惟一依据

。

应聘高级软件开发人员者请回答这部分问题。

1. 数据的逻辑存储结构 (如数组，队列，树等) 对于软件开发具有十分重要的影响，

试对你所了解的各种存储结构从运行速度、存储效率和适用场合等方面进行简要地分析。

2. 数据库技术是计算机系统中一个非常重要的领域，几乎所有的计算机应用中都或多或少地用到了数据库。试简要地谈谈数据库设计中应当注意哪些问题，以及如何解决？给

出两种你所熟悉的 DBMS，要求一种适用于小型应用，另一种适用于大型应用，给出你做出选择的理由。

3. 某公司的主要业务是提供 WWW 和 E-mail 服务，出于安全考虑，该公司要求我公司提供一套网络指纹登录系统，该系统要求能够利用指纹替代 E-mail 中常用的密码，并对所提供的部分网页通过指纹认证后才能访问，请利用你所学过的知识对该系统进行分析设计，

你可以指定网络的配置（包括协议），但必须保证邮件用户既可通过网页（http 方式）收取信件，也可通过 Outlook 收取信件。请分析该系统的可行性，可行时给出系统结构和主要的存储结构，指出系统中的难点和解决方法。（假设指纹识别的问题已经解决）

高级硬件开发人员招聘试题

说明：

可能您的专业并不完全符合本试题所涉及的领域。因此，并非所有的问题都需要回答，您可以只回答您所熟悉和能够回答的问题。允许参考任意的资料，但请独立完成此试题，我们更欣赏您独立的思考和创新的精神。本试题并非我们录用或者不录用您的惟一依据。

应聘高级硬件开发人员者请回答这部分问题。

1. 下面是一些基本的数字电路知识问题，请简要回答：

- (1) 什么是 Setup 和 Holdup 时间？
- (2) 什么是竞争与冒险现象？怎样判断？如何消除？
- (3) 请画出用 D 触发器实现 2 倍分频的逻辑电路。
- (4) 什么是“线与”逻辑？要实现它，在硬件特性上有什么具体要求？
- (5) 什么是同步逻辑和异步逻辑？
- (6) 请画出微机接口电路中，典型的输入设备与微机接口逻辑示意图（数据接口、控制接口、寄存器 / 缓冲器）。

(7) 你知道哪些常用的逻辑电平？TTL 与 COMS 电平可以直接互连吗？

2. 可编程逻辑器件在现代电子设计中越来越重要，请问：

- (1) 你所知道的可编程逻辑器件有哪些？
- (2) 试用 VHDL 或 Verilog, ABLE 描述 8 位 D 触发器逻辑

3. 设想你将设计完成一个电子电路方案。请简述用 EDA 软件（如 PROTEL）进行设计（

包括原理图和 PCB 图）到调试出样机的整个过程。在各个环节应注意哪些问题？

北京信威通信技术股份有限公司面试题

1. DSP 和通用处理器在结构上有什么不同？请简要画出你熟悉的一种 DSP 结构图。
2. 说说定点 DSP 和浮点 DSP 的定义（或者说出他们的区别）。
3. 说说你对循环寻址和位反序寻址的理解。

8, 7】的二进制补码和二进制偏置码。用 Q15 表示出 0.5 和 f- 4. 请写出【

；

85;0.5。

中国国际金融有限公司 CICC 笔试题

1. Please tell us about an achievement that you are especially proud of because it was difficult or demanding.
 - (1) What the objective was?
 - (2) Why it is important to you?
 - (3) How you achieved it and the obstacles that you had to overcome in order to do so?
2. What is your career plan? Three years after graduation, and five years after graduation?
3. Why are you interested in investment bank? What other industries do you also have interests?
4. Why do you think you can be a qualified investment banker? How can you contribute in this industry?

国泰君安笔试题

一列火车上有三个工人，史密斯、琼斯和罗伯特，三人工作为消防员、司闸员和机械师，有三个乘客与这三人的名字相同。罗伯特住在底特律；司闸员住在芝加哥和底特律中间的地方；琼斯一年赚 2 万美金；有一个乘客和司闸员住在一个地方，每年的薪水是司闸员的 3 倍整；史密斯台球打得比消防员好；和司闸员同名的乘客住在芝加哥。请问谁是机械师？

Briny 笔试题

1. 说出 RC 振荡器的构成和工作原理。
2. 什么是 SDH?
3. 什么是共模、差模?画出差分电路的结构。
4. $a=5$; $b=6$; $a+=b++$; 执行结果是什么?
5. 什么是 TDM? 什么是 CDMA?
6. 什么是采样定理?
7. 什么是香农定理?
8. 计算机的中断有哪几类?

7.1

广东北电面试题目

英文笔试题

1. Translation (Mandatory)

CDMA vendors have worked hard to give CDMA roaming capabilities via the de

velopment of RUIM-essentially, a SIM card for CDMA handsets currently being deployed in China for new CDMA operator China Unicom. Korean cellco KTF demonstrated earlier this year the ability to roam between GSM and CDMA using such cards. However, only the card containing the user's service data can roam-not the CDMA handset or the user's number (except via call forwarding).

2. Programming (Mandatory)

Linked list

- a. Implement a linked list for integers, which supports the insert after (insert a node after a specified node) and remove after (remove the node after a specified node) methods;
 - b. Implement a method to sort the linked list to descending order.
3. Debugging (Mandatory)
 - a. For each of the following recursive methods, enter Y in the answer box if the method terminates (assume $i=5$), Otherwise enter N.

```
static int f(int i){
    return f(i-1)*f(i-1);
}
```

Answer:

```
static int f(int i){
    if(i==0){return 1;}
    else {return f(i-1)*f(i-1);}
}
```

Answer:

```
static int f(int i){
    if(i==0){return 1;}
    else {return f(i-1)*f(i-2);}
}
```

Answer:

- b. There are two errors in the following JAVA program:

```
static void g(int i){
    if(i==1){return;}
    if(i%2==0){g(i/2);return;}
    else {g(3*i);return;}
}
```

please correct them to make sure we can get the printed-out result as below

w:

3 10 5 16 8 4 2 1

中文笔试题

1. 汉译英

北电网络的开发者计划使来自于不同组织的开发者，能够在北电网络的平台上开发圆满的补充业务。北电网络符合工业标准的开放接口，为补充业务的开展引入了无数商机，开发者计划为不同层面的开发者提供不同等级的资格，资格的划分还考虑到以下因素：补充业务与北电网络平台的集合程度，开发者团体与北电网络的合作关系，等等。

2. 编程

将整数转换成字符串：`void itoa(int, char);`

例如 `itoa(-123, s[])` 则 `s="-123"`;

U2 合唱团在 17 分钟内得赶到演唱会场，途中必需跨过一座桥，四个人从桥的同一端出发，你得帮助他们到达另一端，天色很暗，而他们只有一只手电筒。一次同时最多可以有两人一起过桥，而过桥的时候必须持有手电筒，所以就得有人把手电筒带来带去，来回桥两端。手电筒是不能用丢的方式来传递的。四个人的步行速度各不同，若两人同行则以较慢者的速度为准。Bono 需花 1 分钟过桥，Edge 需花 2 分钟过桥，Adam 需花 5 分钟过桥，Larry 需花 10 分钟过桥。他们要如何在 17 分钟内过桥呢？（有个同济的学生写文章说他当时在微软面试时就是碰到了这道题，最短只能做出在 19 分钟内过桥，微软的人对他讲这样的结果已经是不错的了！）

A 点到 B 点

1 和 2 过去 2 分钟 2

2 过来 4 分钟 $2+2=4$

10 和 5 过去 14 分钟 $4+10=14$

1 过来 15 分钟 $14+1=15$

1 和 2 过去 17 分钟 $15+2=17$

19 分钟还很不错？？？

广州本田笔试题

1. 排序 s-m-t-w-t-f-?
2. 如果六千，六百，六表示成 6606，那么十一千，十一百，十一表示成什么？
3. grass 后面加一个词， agent 前面加一个单词，组成两个新词，这个词是什么？
4. 农场不知道有多少鸡，现有一批饲料，如果卖掉 75 只鸡饲料够 20 天用，买进 100 只

鸡饲料够用 15 天，问原来有多少只鸡？

5. 6 个桶，装着两种液体，一种液体的价格是另外一种的 double，桶容量为 8，13，1

5，17，19，31，有一个美国人，各用了 14 美元买两种液体，剩下一个桶。问剩下哪个？

6. 篮球场，还剩 6 秒，差对手 4 分，没可能追得上，现在有一个暂停，你会怎么指导球

员去做？

明基面试问题

1. 自我介绍（2 分钟）。
2. 你大学期间最辉煌的一件事是什么？
3. 如果你明天去火星呆上 300 年，今天晚上你最想做的一件事是什么？

网易

- 1、10 个人分成 4 组 有几种分法？
- 2、如图：

7 8 9 10

6 1 2 11

5 4 3 12

16 15 14 13

设“1”的坐标为（0，0）“7”的坐标为（-1，-1）编写一个小程序，使程

序做到输入坐标（X,Y）之后显示出相应的数字。

- 3、#include<stdio.h>

```
//example input and output
//in 1 2 3 out 1 3 1
//in 123456789 2 100 out 123456789 100 21
long mex(long a,long b,long c)
{ long d;
  if(b==0) return 0;
  if(b==1) return a%c;
  d=mex(a,b/2,c); d*=d;这里忘了;d*=mex(a,b%2,c);d%=c;
  return d;
}
int main(void)
{ long x,y,z;
```

```

while(1)
{ if(scanf("%d %d %d",&x,&y,&z)>3) return 0;
  if(x<0) { printf("too small\n");continue;}
  if(y<0) { printf("too small\n");continue;}
  if(z<1) { printf("too small\n");continue;}
  if(y>z) { printf("too big\n");continue;}
  if(z>1000000010) {printf("too big\n");continue}
  printf("%d %d %d",x,z,mex(x,y,z);
}
}

```

根据这个程序，当已知一个输入，算出输出，如：输入 1 3 1 则输出 1 2 3 输

入 123456789 100 21 输出 123456789 2 100

广州日报

- 1、填空部分是一些时事题，如：我国有多少网民，三个代表、北京申奥什么的，及记者的一些常识性的问题：如我国第一个以写新闻通讯出名的记者是谁？蔡元培曾经夸奖过的记者是谁？
- 2、选择题范围与填空基本一样，包括时政和新闻知识：如深度采访的实质，记者的职业道德等。
- 3、简答题就比较专业：一道是你参加一条高速公路的开通典礼，如何在记者会上发的新闻通稿之外写出会上没说的内容。一道是你去一个单位采访，但没有任何该单位的证件、邀请函之类东西，你如何骗过门卫混进去。第三道是有几家香水公司都想让你说好话，就是做软新闻了，你该如何处理。第四道是 A 明星与 B 明星不和，你如何报道 A 骂 B 的话而又不能让 B 告你诽谤。
- 4、写作题是以“今年冬天不太冷”为题任意想象，加叙加议。
- 5、五道智力测验：如何喝道啤酒杯底部的啤酒、汽车过隧道但高 2 厘米该怎么办、你吃苹果时吃到几条虫最恶心之类，10 只点燃的蜡烛，让风吹灭了 2 只，后来在关窗户前又吹灭 1 只。问最后还有几支。

7.2

JAVA 程序员面试题收集

一、Java 基础知识

1. Java 有那些基本数据类型，String 是不是基本数据类型，他们有何区别。
2. 字符串的操作：
写一个方法，实现字符串的反转，如：输入 abc，输出 cba
写一个方法，实现字符串的替换，如：输入 bbbwlirbbb，输出 bbbhhtccc。
3. 数据类型之间的转换
如何将数值型字符转换为数字（Integer，Double）
如何将数字转换为字符
如何取小数点前两位，并四舍五入。
4. 日期和时间
如何取得年月日，小时分秒
如何取得从 1970 年到现在的毫秒数
如何获取某个日期是当月的最后一天
如何格式化日期
5. 数组和集合
6. 文件和目录（I/O）操作
如何列出某个目录下的所有文件
如何列出某个目录下的所有子目录
判断一个文件或目录是否存在
如何读写文件
7. Java 多态的实现（继承、重载、覆盖）
8. 编码转换，怎样实现将 GB2312 编码的字符串转换为 ISO-8859-1 编码的字符串。
9. Java 中访问数据库的步骤，Statement 和 PreparedStatement 之间的区别。
10. 找出下列代码可能存在的错误，并说明原因：

二、JSP&Servlet 技术

1. 描述 JSP 和 Servlet 的区别、共同点、各自应用的范围
2. 在 Web 开发中需要处理 HTML 标记时，应做什么样的处理，要筛选那些字符（<> & “”）
3. 在 JSP 中如何读取客户端的请求，如何访问 CGI 变量，如何确定某个 Jsp 文件的真实路径。
4. 描述 Cookie 和 Session 的作用，区别和各自的应用范围，Session 工作原理。
5. 列出 Jsp 中包含外部文件的方式，两者有何区别。
6. 说明 Jsp 中 errorPage 的作用，应用范围。
7. 介绍在 Jsp 中如何使用 JavaBeans。
8. 简单介绍 JSP 的标记库
9. Jsp 和 Servlet 中的请求转发分别如何实现。

三、J2EE 相关知识

1. 介绍 J2EE、J2SE、J2SE 的区别。
2. J2EE 是一种技术还是一种平台，他提供了那些技术。
3. 什么是 Application Server，它有什么功能和优点。
4. 简单介绍连接池的优点和原理。
5. Web.xml 的作用

四、其他

1.Web 安全性的考虑（表单验证、浏览器 Basic 方式的验证，应用程序的安全性，SSL，代码考虑）

2.简单介绍您所了解的 MVC。

3.简单介绍您所了解的 XML。

4.文档和编码规范

5.Java 中的分页、效率考虑。

6.简单介绍您所了解的 struts。

收藏：

1.xml 在项目中的作用

2.s-EJB 与 e-EJB 的区别

3.会话面的作用

4.cmp 与 bmp 的优缺点

5.j2me 程序的必需的几个部分

6.c/s 与 b/s 的区别

7.构建一个 connect pool,然后再调用它，

8.j2ee 平台与 dotnet 平台的区别

9.ejb 的 life cycle

10.session bean 和 entity bean 的区别

11.ejb 中的 transaction 机制

12.synchronized (生产者和消费)

13.String 和 StringBuffer

14.Serializable

15.MVC（Struts 的工作流程）

16.什么是 MDA

17.tcp 与 udp 的区别

18.链表与散列表和数组的区别

19.堆和栈的区别

20.ejb 的分类及区别

21.你对现在软件业以及国内软件业的看法

22.谈谈 java 多线程

23.谈谈文件加密技术

24.软件开发生命周期

25.路由协议种类及特点

26.java 的 awt 和 swing 组件的 GUI 设计的关键

27.对于 java 流的认识

28.简单描述一下 awt 与 swing 区别。

29.简述 java 编程中事件处理模式。

30.你编写过 applet 吗？applet 的安全权限如何？试列举 java application 或者 applet 中与 servlet/jsp 通信可以采用的方式。

31. 简述逻辑操作 (如&,|)与条件操作 (如&&,||)的区别。
32. 简述 Java Server Page 和 Servlet 的联系和区别。
33. 简述 synchronized 和 java.util.concurrent.locks.Lock 的异同?
34. EJB 规范规定 EJB 中禁止的操作有哪些?

贝尔笔试面试及答案

一、请填写 BOOL, float, 指针变量与“零值”比较的 if 语句。(10 分)

请写出 BOOL flag 与“零值”比较的 if 语句。(3 分)

标准答案:

if (flag)

if (!flag) 如下写法均属不良风格, 不得分。

if (flag == TRUE)

if (flag == 1)

if (flag == FALSE)

if (flag == 0)

请写出 float x 与“零值”比较的 if 语句。(4 分)

标准答案示例:

```
const float EPSINON = 0.00001;
```

```
if ((x >= - EPSINON) && (x <= EPSINON))
```

不可将浮点变量用“=”或“!=”与数字比较, 应该设法转化成“>=”或“<=”此类形式。

如下是错误的写法, 不得分。

```
if (x == 0.0)
```

```
if (x != 0.0)
```

请写出 char *p 与“零值”比较的 if 语句。(3 分)

标准答案:

if (p == NULL)

if (p != NULL) 如下写法均属不良风格，不得分。

if (p == 0)

if (p != 0)

if (p)

if (!)

二、以下为 Windows NT 下的 32 位 C++ 程序，请计算 sizeof 的值(10 分)

```
char str[] = "Hello";
```

```
char *p = str;
```

```
int n = 10;
```

请计算

sizeof (str) = 6 (2 分)

sizeof (p) = 4 (2 分)

sizeof (n) = 4 (2 分) void Func (char str[100])

```
{
```

请计算

sizeof(str) = 4 (2 分)

```
}
```

```
void *p = malloc( 100 );
```

请计算

sizeof (p) = 4 (2 分)

三、简答题(25 分)

1、头文件中的 `ifndef/define/endif` 干什么用? (5 分)

答: 防止该头文件被重复引用。

2、`#include?` 和 `#include "filename.h"` 有什么区别? (5 分)

答: 对于 `#include?`, 编译器从标准库路径开始搜索 `filename.h`

对于 `#include "filename.h"`, 编译器从用户的工作路径开始搜索 `filename.h`

3、`const` 有什么用途? (请至少说明两种)(5 分)

答: (1)可以定义 `const` 常量

(2)`const` 可以修饰函数的参数、返回值, 甚至函数的定义体。被 `const` 修饰的东西都

受到

强制保护, 可以预防意外的变动, 能提高程序的健壮性。

4、在 C++ 程序中调用被 C 编译器编译后的函数, 为什么要加 `extern "C"`? (5 分)

)

答: C++ 语言支持函数重载, C 语言不支持函数重载。函数被 C++ 编译后在库中的名字与

C 语言

的不同。假设某个函数的原型为: `void foo(int x, int y);`

该函数被 C 编译器编译后在库中的名字为 `_foo`, 而 C++ 编译器则会产生像 `_foo_int_int`

之类的

名字。

C++ 提供了 C 连接交换指定符号 `extern "C"` 来解决名字匹配问题。

5、请简述以下两个 `for` 循环的优缺点 (5 分)

`for (i=0; i<N; i++)`

```

{

if (condition)

DoSomething();

else

DoOtherthing();

}

if (condition)

{

for (i=0; i<N; i++)

DoSomething();

}

else

{

for (i=0; i<N; i++)

DoOtherthing();

}

```

优点：程序简洁

缺点：多执行了 N-1 次逻辑判断，并且打断了循环“流水线”作业，使得编译器不能

对循环进行优化处理，降低了效率。 优点：循环的效率高

缺点：程序不简洁

四、有关内存的思考题(每小题 5 分，共 20 分)

```
void GetMemory(char *p)
```



```

{

p = (char *)malloc(100);

}

void Test(void)

{

char *str = NULL;

GetMemory(str);

strcpy(str, "hello world");

printf(str);

}

```

请问运行 Test 函数会有什么样的结果？

答：程序崩溃。

因为 GetMemory 并不能传递动态内存，

Test 函数中的 str 一直都是 NULL。

strcpy(str, "hello world");将使程序崩溃。

```

char *GetMemory(void)

{

char p[] = "hello world";

return p;

}

void Test(void)

{

```

```
char *str = NULL;

str = GetMemory();

printf(str);

}
```

请问运行 Test 函数会有什么样的结果？

答：可能是乱码。

因为 GetMemory 返回的是指向“栈内存”的指针，该指针的地址不是 NULL，但其原现
的

内容已经被清除，新内容不可知。

```
void GetMemory2(char **p, int num)

{

    *p = (char *)malloc(num);

}

void Test(void)

{

    char *str = NULL;

    GetMemory(&str, 100);

    strcpy(str, "hello");

    printf(str);

}
```

请问运行 Test 函数会有什么样的结果？

答：

(1)能够输出 hello

(2)内存泄漏

```
void Test(void)
```

```
{
```

```
char *str = (char *) malloc(100);
```

```
strcpy(str, "hello");
```

```
free(str);
```

```
if(str != NULL)
```

```
{
```

```
strcpy(str, "world");
```

```
printf(str);
```

```
}
```

```
}
```

请问运行 Test 函数会有什么样的结果？

答：篡改动态内存区的内容，后果难以预料，非常危险。

因为 free(str);之后，str 成为野指针，

if(str != NULL)语句不起作用。

五、编写 strcpy 函数(10 分)

已知 strcpy 函数的原型是

```
char *strcpy(char *strDest, const char *strSrc);
```

其中 strDest 是目的字符串，strSrc 是源字符串。

(1)不调用 C++/C 的字符串库函数，请编写函数 strcpy

```

char *strcpy(char *strDest, const char *strSrc);

{

assert((strDest!=NULL) && (strSrc !=NULL)); // 2 分

char *address = strDest; // 2 分

while( (*strDest++ = * strSrc++) != '\0' ) // 2 分

NULL ;

return address ; // 2 分

}

```

(2)strcpy 能把 strSrc 的内容复制到 strDest，为什么还要 char * 类型的返回值？

答：为了实现链式表达式。 // 2 分

例如 int length = strlen(strcpy(strDest, “hello world”));

六、编写类 String 的构造函数、析构函数和赋值函数 (25 分)

已知类 String 的原型为：

```

class String

{

public:

String(const char *str = NULL); // 普通构造函数

String(const String &other); // 拷贝构造函数

~String(void); // 析构函数

String & operate =(const String &other); // 赋值函数

private:

char *m_data; // 用于保存字符串

```

```
};
```

请编写 String 的上述 4 个函数。

标准答案：

// String 的析构函数

```
String::~String(void) // 3 分
```

```
{
```

```
delete [] m_data;
```

```
// 由于 m_data 是内部数据类型，也可以写成 delete m_data;
```

```
}
```

// String 的普通构造函数

```
String::String(const char *str) // 6 分
```

```
{
```

```
if(str==NULL)
```

```
{
```

```
m_data = new char[1]; // 若能加 NULL 判断则更好
```

```
*m_data = '\0';
```

```
}
```

```
else
```

```
{
```

```
int length = strlen(str);
```

```
m_data = new char[length+1]; // 若能加 NULL 判断则更好
```

```
strcpy(m_data, str);
```

```
}
```

```
}
```

```
// 拷贝构造函数
```

```
String::String(const String &other) // 3 分
```

```
{
```

```
int length = strlen(other.m_data);
```

```
m_data = new char[length+1]; // 若能加 NULL 判断则更好
```

```
strcpy(m_data, other.m_data);
```

```
}
```

```
// 赋值函数
```

```
String &String::operator =(const String &other) // 13 分
```

```
{
```

```
//(1) 检查自赋值 // 4 分
```

```
if(this == &other)
```

```
return *this;
```

```
//(2) 释放原有的内存资源 // 3 分
```

```
delete [] m_data;
```

```
//(3)分配新的内存资源，并复制内容 // 3 分
```

```
int length = strlen(other.m_data);
```

```
m_data = new char[length+1]; // 若能加 NULL 判断则更好
```

```
strcpy(m_data, other.m_data);
```

```
//(4)返回本对象的引用 // 3 分
```

华为面试记

[size=10]六点钟起床，刷牙洗脸整理发型，穿上自己唯一一套比较正式的装扮（被芽芽和我称为面试套装），带上简历匆匆赶到上地大厦准备接受华为 8 点开始的面试。

7: 40 分，存好破烂的自行车，信步来到前台大厅却只见一片冷清。和前台服务员招呼过后才知道，一般的面试将近九点才会开始，没搞明白负责通知面试的那位为什么要我八点到。。。乘电梯到达 9 层华为面试大厅。空无一人。如果不是报着见识见识华为是如何面试新人的态度，真想一走了之了。幸好时间不大就来了一些应试的朋友，大家互相攀谈一阵，原来他们的约定时间都是 8: 30 分。

8: 45 分，人差不多到齐了，华为的人力资源部干事开始发题目和答题纸。按照电话通知的预约，我得到了一份 C 语言的笔试题。

9 点 10 分左右，华为的考官们稀稀拉拉的到场了。

不久便陆续有人答题完毕，应试者提交题目和答题纸之后，就等候在自己的座位上听候需要自己的部门面试。

看得出来一共有四轮面试，真是层层选拔一点儿也不含糊。

因为四轮比较费时间，所以有几位应聘人员等候近 1 小时后见没有轮到自己，就离去了。

交卷之后两个小时，到我上场了。

面试官看过简历后发现我是 JAVA 程序员，也没有 C/C++ 任何相关经验，就问我愿意从事 JAVA 还是 C，我当然选择老本行。

于是又得到一套 JAVA 试卷。

这套卷子相当侧重 JAVA 的多线程编程，写游戏和游戏服务器当然离不开多线程的程序设计，再加上俺深厚的基础功底，十来分钟就结束了这份答卷。

主考官见我做得比较快，又是二次答卷就比较照顾的先面我了。意外的是他看了我的简历后发现我是自考生，而且刚进入毕设阶段，就问我什么时候可以拿到学位证书，我如实回答 12 月拿到毕业证书，明年初拿到学位证。

面试官面露难色，又详细的看了一遍我的简历，跟我说华为招的都是本科以上的应聘者，不知道我这种情况是否可以过资格审查，说过后他跑到人力那里去询问具体情况，结果沮丧而归。其实面试通知人员在电话预约我的时候就提到过这个问题，对方说是没有证书不是很重要，华为还是很务实的，主要看能力等等。。。

只可惜当前这位面试官把我的简历看了再三，跑了两次前台，还是无功而返，最后要了我记下的那位通知我来面试的工作人员手机号码，搞笑的是整个华为招聘现场竟没有人知道那是谁的手机。

出于对工作失误的歉意，面试官还是一脸尴尬的简单和我聊了几句。当然很快的，我就离开了华为招聘会。

size]

IT 公司面试笔试秘籍大公开

爱立信：

爱立信的特色是楼里有超多老外。笔试题出得不错，给一张考卷一本技术手册。技术手册提供了一套汇编指令集，有每种指令基本使用方法和例程，现场学会这套汇编语言之后开始做考卷，考卷基本上是些程序填空之类的题，期间肯定要不停地翻那本技术手册，感觉题都不难，但要

求 45 分钟内做完就非常有难度了。其实汇编和单片机学得稍微好点就不会有什么技术上得障碍了，感觉很大程度上是在考快速阅读英文技术资料的能力，外恶的英语啊……面完出来就感觉不好，时间太紧了好多题都没做完，结果果然就被鄙视了～

IBM China globe delivery:

IBM 这个部门名字起得真是鬼斧神工，收到笔试信的时候一直弄不明白这个部门应该是干什么的，按字面翻译大概应该是 IBM 全球快递吧……

笔试很常规，考些所谓的逻辑题智力题，都不难，但对速度有要求。分三个部分，第一个是矩阵转换，就是把一个矩阵颠来倒去地进行变换和运算最后问某一行某一列 是个什么数，我就弄不明白为什么这种问题就偏要被写成长达 5, 6 行的定语从句，然后每道题我都吃不准自己理解的是否正确……再次鄙视一下自己的英语。第二部分 是填数列，这个做的还不错，他说停的时候正好做完而且觉得都对的。第三部分考初中应用题，就是解二元一次方程，强一点的小学生都会做，但我最后还是差 一个空没填完。

面试形式很怪，3 对 3 坐着，任何一个考官都能提问，有时候会问诸如“你对刚才那个同学的观点有什么看法”一类的问题，感觉 我们三个可怜的学生在不停地被挑拨离间……考官问我有没有学过日语我说没，他又问为什么我说学校教，他居然说学校不教你就不自己学啦，你是不是对日本有什么偏见？或者你觉得自己没有语言天赋还是对学语言完全不感兴趣？

oracle:

本来投得是 SE 职位，结果居然通知我去笔他们的 trainee，想想轮岗两年后还有次机会再选择自己的发展方向也不错就去了

笔试地点在香格里拉，一个超级大的礼堂，笔试面试都和技术没关系，所以来的人什么专业都有，复旦交大同济都有专车过去，还蛮方便的。笔试也是考些逻辑题之类的，也是不比难度比速度的（怎么 IT 类企业都好这口），量比 IBM CGD 的更大，基本都不怎么做得完。居然还有填反义词，感觉好难啊我又没背过 GRE 怎么尽弄些那么牛 b 的单词呢……

笔完了觉得没什么希望了，感觉早上没怎么睡醒做题速度比较慢但没想到最后还是接到面试通知，而且就是 final 了。据说是 3 对 1 的形式但轮到我的时候大概一个 HR 去吃午饭了所以就变成了 2 对 1……一男一女，男的非常 tough 女的一直面带微笑但很少提问。唯一提到和技术稍微有点关系的一个问题是“请举个例子，说明自己怎么向非专业技术人员解释技术问题”，其他的都是些比较常规的问题了，面试前最好准备下 P&G 经典题什么的。比较有特色的一个环节是给 6 张卡片让你抽一张然后根据卡片上写的单词说一段 1 分钟的 presentation，我抽到的是 customer concentric。整个流程大约半小时，有时候会说中文，不过我觉得香港人的英语发音和普通话发音一样难听。

IBM China Software Development Lab (CDL):

笔试题太恶心了，3 个半小时，啥都考，我选得是 J2EE 卷，然后一直不停地在做 c,java,jsp,xml,uml,DB,linux,struts……最后很多都不会，出来什么感觉都没有了，都不知道自己做

得好还是坏

面试的是两个 GG，一个 PM 一个 HR，还是问项目经历为主，还问了很多 Tomcat 配置的问题，我觉得很奇怪就问难道 IBM 自己不是用 Websphere 的么？他说很多项目都不用这些重量级东西的，觉得把 struts,spring,hibernate 之类的流行技术弄熟，然后英语能交流点技术问题应该就能过了，觉得面得还行可最后还是被鄙视了。

EMC:

EMC 是一直很想去的地方，但它出得笔试题实在是烂，以至于对它的 好感骤减。最恶心的就是那些考 c 语言的题目，老是弄些边角料的花边技巧甚至还考运算优先级之类的.....ft. 还有些就是写数列的题目，咋一看毫无规律然后换算成 16 进制数就很容易了，蛮有意思的。有到题印象蛮深的，说一个碗里有 n 根面条，每次从碗里拿出面条的两个端点（不一定是同一根面条的两个端点），把端点粘起来，直到没有端点了，问最后碗里环的个数的期望值（反正是概率里的，我忘了是不是这个专有名词了）是多少。我现在还不确定当时给的答案对不对，有兴趣的可以算一下。

第一轮面试的 GG 英语非常非常地流利，先问我项目经历，然后问细节，说这个项目不小你负责的是哪块呢，我说是权限管理，他说那就谈谈碰到过的具体问题，我就说了那个比较经典的 1248 算法，然后我看他表情好像很好奇就觉得他应该没做过类似的东西看来可以随便忽悠了，就告诉他其实 windows 下的权限用的也是这个算法，只不过到了 struts 里面每个权限就变成 action 了，然后就 blablabla不过他领悟能力超强，一下子就看到问题所在，开始问我在数据库里的实现方法，然后继续 blablabla。然后问了道智力题，说有 9 个人排成一排，每个人头上一顶帽子，或红或黑，但是每个人只能看到排在自己前面的人的帽子的颜色而且看不见自己的。从最后一个人开始猜自己头上的帽子是什么颜色的，猜错就拖出去打死，问找一种方案让幸存者人数最大化，我觉得貌似是道小学奥数题但自己还真没做过，就说是不是就凭概率啊，他说当然不是，然后我就汗了，当时真的很紧张，要看着他的脸当场想而且他还会不停的施加压力说这道题其实很简单啊很简单，然后我就说了种比较傻的方案，能确保 4 个人活下来其他人听天由命，自己都 觉得不可能只有那么少但决定就这样蒙了。后来出来等车的时候又想了想，发现应该是能确保 6 个人活下来的，而且这个数字应该就是最佳答案了，然后郁闷。

第二轮面试考基础题，上来先问我这学期学什么，我说编译和网络，他说那好，我就问你这两个方面的知识吧，你现学现卖，然后我超郁闷，答得非常非常之差，最后他说你觉得这个样子你考试能过么我只好如实说其实每个学期得课程也就最后考试前看看，而现在还没开始复习所以书基本都是新的，其实你问我大三大二学过得课程说不定我还能答的好点呢。然后他就笑了，开始问我些操作系统数据结构得东西，这次觉得问题还蛮对胃口的，对自己基本满意。然后他说了个想法，说要做个网络爬虫的服务，然后我说爬虫是什么，他就描述了一遍，然后说要我提出意见，说一下这个网络服务的前景和可行性，然后口头做一个策划方案包括开发推广和 实施，说一下要注意到的问题。

第三轮来了个 JJ，这个 JJ 自以为是得要命觉得自己英语很好说得暴快其实就是发音含混一点都听不清在说什么，然后我就不停地 pardon.....最后说到一个词她自己想不起来怎么说了，对着天花板想了半天蹦出来个中文词我汗然后下面就都变成了中文面。这次基本就考编程能力，让我站在黑板前写程序，都是些链表题，什么用链表实现个 LRU 算法啊什么找大链表中是否有

环啊之类的，考反应和编程速度，然后我就太注重速度了，很多边界值都没考虑，然后就不停地被她抓住把柄，说 c 语言很危险地啊你这样不考虑细节会造成很严重的内存泄漏的啊，我只好说这种问题我一般都是在 调试阶段修改的。

intel 通讯部门：

先是打电话来说是 intel，但今年 intel 全球大裁员所以本部门可能要被卖给 其他公司了，不过肯定也是大公司，问还有兴趣么，我说有。她就介绍了下自己部门的情况，我忘了具体名字，只知道是搞通讯和语音设备的，然后她说来一趟紫竹 也很远我们就电面吧，然后开始问一些关于 mp3 编码的问题，大约一共 15 分钟，貌似互相感觉都不怎么样

微软全球技术支持中心 GTSC：

笔试的东西都是微软技术，.net 的运行机制，C#的函数用法，win32 api，windows 的一些运行状态等等，做了 15 分钟就交卷了，不会做。我的简历上没有涉及任何有关微软技术的东西，不知道怎么就会被叫过来笔试的

intel 主板部门：

每 次接到 intel 都会有点有意思的事情这次她说他们那有一份半年前我投的简历，问我是否还有兴趣，办事效率真不是一般的低额~~~我说那时候我应该是 在应聘实习职位吧，难道现在去还是实习？她说是的，我想想反正下半学期应该比较空，就说好。然后就跋山涉水来到紫竹。

一面是个 GG，问了几道智力题。

二 面是个 JJ，考编程，发现好几个 intel 的女冷 coding 都超强啊，题目比较简单，做的也比较快，但做完她问我为什么很多边界值不考虑，我说一般在调 试时候加上去，比较习惯在第一遍写的时候少考虑些东西，她说我们这是测试部门，考虑的就是边界问题，你应该养成良好习惯。然后问了黑盒白盒测试的问题，还 问有没有学过软件工程，说有。

三面又是个 GG，上来还问智力题，然后给了个程序，问如果让我测试会给些什么样的数据，然后我就列了些数据。

四 面是 HRJJ，上来先介绍下这个部门的情况，原来他们是做主板 BIOS 的，要找实习生做测试，然后说我对你印象很不错，但不知道你对这份工作有没有兴趣， 我说我想通过实习留下来，不知道可能性多大。她说裁员厉害，留实习生可能要看 07 年公司的计划，不过按现在的发展态势希望不大。我说哦，然后她问还有什么 问题，我说怎么回家比较方便。

7.6

面试题汇集（一）

、面向对象的特征有哪些方面

1.抽象:

抽象就是忽略一个主题中与当前目标无关的那些方面，以便更充分地注意与当前目标有关的方面。抽象并不打算了解全部问题，而只是选择其中的一部分，暂时不用部分细节。抽象包括两个方面，一是过程抽象，二是数据抽象。

2.继承:

继承是一种联结类的层次模型，并且允许和鼓励类的重用，它提供了一种明确表述共性的方法。对象的一个新类可以从现有的类中派生，这个过程称为类继承。新类继承了原始类的特性，

新类称为原始类的派生类（子类），而原始类称为新类的基类（父类）。派生类可以从它的基类那里继承方法和实例变量，并且类可以修改或增加新的方法使之更适合特殊的需要。

3.封装:

封装是把过程和数据包围起来，对数据的访问只能通过已定义的界面。面向对象计算始于这个基本概念，即现实世界可以被描绘成一系列完全自治、封装的对象，这些对象通过一个受保护的接口访问其他对象。

4.多态性:

多态性是指允许不同类的对象对同一消息作出响应。多态性包括参数化多态性和包含多态性。多态性语言具有灵活、抽象、行为共享、代码共享的优势，很好的解决了应用程序函数同名问题。

2、String 是最基本的数据类型吗？

基本数据类型包括 byte、int、char、long、float、double、boolean 和 short。

java.lang.String 类是 final 类型的，因此不可以继承这个类、不能修改这个类。为了提高效率节省空间，我们应该用 StringBuffer 类

3、int 和 Integer 有什么区别

Java 提供两种不同的类型：引用类型和原始类型（或内置类型）。Int 是 java 的原始数据类型，Integer 是 java 为 int 提供的封装类。Java 为每个原始类型提供了封装类。原始类型封装类
booleanBoolean charCharacter byteByte shortShort intInteger longLong floatFloat
doubleDouble

引用类型和原始类型的行为完全不同，并且它们具有不同的语义。引用类型和原始类型具有不同的特征和用法，它们包括：大小和速度问题，这种类型以哪种类型的数据结构存储，当引用类型和原始类型用作某个类的实例数据时所指定的缺省值。对象引用实例变量的缺省值为 null，而原始类型实例变量的缺省值与它们的类型有关。

4、String 和 StringBuffer 的区别

JAVA 平台提供了两个类：String 和 StringBuffer，它们可以储存和操作字符串，即包含多个字符的字符数据。这个 String 类提供了数值不可改变的字符串。而这个 StringBuffer 类提供

的字符串进行修改。当你知道字符数据要改变的时候你就可以使用 `StringBuffer`。典型地，你可以使用 `StringBuffers` 来动态构造字符数据。

5、运行时异常与一般异常有何异同？

异常表示程序运行过程中可能出现的非正常状态，运行时异常表示虚拟机的通常操作中可能遇到的异常，是一种常见运行错误。`java` 编译器要求方法必须声明抛出可能发生的非运行时异常，但是并不要求必须声明抛出未被捕获的运行时异常。

6、说出 `Servlet` 的生命周期，并说出 `Servlet` 和 `CGI` 的区别。

`Servlet` 被服务器实例化后，容器运行其 `init` 方法，请求到达时运行其 `service` 方法，`service` 方法自动派遣运行与请求对应的 `doXXX` 方法（`doGet`，`doPost`）等，当服务器决定将实例销毁的时候调用其 `destroy` 方法。

与 `cgi` 的区别在于 `servlet` 处于服务器进程中，它通过多线程方式运行其 `service` 方法，一个实例可以服务于多个请求，并且其实例一般不会销毁，而 `CGI` 对每个请求都产生新的进程，服务完成后就销毁，所以效率上低于 `servlet`。

7、说出 `ArrayList`、`Vector`，

`ArrayList` 和 `Vector` 都是使用数组方式存储数据，此数组元素数大于实际存储的数据以便增加和插入元素，

它们都允许直接按序号索引元素，但是插入元素要涉及数组元素移动等内存操作，所以索引数据快而插入数据慢，

`Vector` 由于使用了 `synchronized` 方法（线程安全），通常性能上较 `ArrayList` 差，而 `LinkedList` 使用双向链表实现存储，

按序号索引数据需要进行前向或后向遍历，但是插入数据时只需要记录本项的前后项即可，所以插入速度较快

8、EJB 是基于哪些技术实现的？并说出 `SessionBean` 和 `EntityBean` 的区别，`StatefulBean` 和 `StatelessBean` 的区别。

EJB 包括 `Session Bean`、`Entity Bean`、`Message Driven Bean`，基于 `JNDI`、`RMI`、`JAT` 等技术实现。`SessionBean` 在 `J2EE` 应用程序中被用来完成一些服务器端的业务操作，例如访问数据库、调用其他 EJB 组件。

`EntityBean` 被用来代表应用系统中用到的数据。对于客户机，`SessionBean` 是一种非持久性对象，它实现某些在服务器上运行的业务逻辑。对于客户机，`EntityBean` 是一种持久性对象，它代表一个存储在持久性存储器中的实体的对象视图，或是一个由现有企业应用程序实现的实体。

`Session Bean` 还可以再细分为 `Stateful Session Bean` 与 `Stateless Session Bean`，

这两种的 `Session Bean` 都可以将系统逻辑放在 `method` 之中执行，不同的是 `Stateful Session Bean` 可以记录呼叫者的状态，因此通常来说，一个使用者会有一个相对应的 `Stateful Session Bean` 的实体。

`Stateless Session Bean` 虽然也是逻辑组件，但是他却不负责记录使用者状态，也就是说当使用者呼叫

`Stateless Session Bean` 的时候，`EJB Container` 并不会找寻特定的 `Stateless Session Bean`

的实体来执行这个 `method`。换言之，很可能数个使用者在执行某个 `Stateless Session Bean` 的 `methods` 时，

会是同一个 Bean 的 Instance 在执行。从内存方面来看，Stateful Session Bean 与 Stateless Session Bean 比较，

Stateful Session Bean 会消耗 J2EE Server 较多的内存，然而 Stateful Session Bean 的优势却在于他可以维持使用者的状态。

9、Collection 和 Collections 的区别。

Collection 是集合类的上级接口，继承与他的接口主要有 Set 和 List.

Collections 是针对集合类的一个帮助类，他提供一系列静态方法实现对各种集合的搜索、排序、线程安全化等操作。

10、&和&&的区别。

&是位运算符，表示按位与运算，&&是逻辑运算符，表示逻辑与（and）。

11、HashMap 和 Hashtable 的区别。

HashMap 是 Hashtable 的轻量级实现（非线程安全的实现），他们都完成了 Map 接口，主要区别在于 HashMap 允许空（null）键值（key），

由于非线程安全，效率上可能高于 Hashtable。HashMap 允许将 null 作为一个 entry 的 key 或者 value，而 Hashtable 不允许。

HashMap 把 Hashtable 的 contains 方法去掉了，改成 containsvalue 和 containsKey。因为 contains 方法容易让人引起误解。

Hashtable 继承自 Dictionary 类，而 HashMap 是 Java1.2 引进的 Map interface 的一个实现。最大的不同是，Hashtable 的方法是

Synchronized 的，而 HashMap 不是，在多个线程访问 Hashtable 时，不需要自己为它的方法实现同步，

而 HashMap 就必须为之提供外同步。Hashtable 和 HashMap 采用的 hash/rehash 算法都大概一样，所以性能不会有很大的差异。

12、final, finally, finalize 的区别。 final

用于声明属性，方法和类，分别表示属性不可变，方法不可覆盖，类不可继承。

finally 是异常处理语句结构的一部分，表示总是执行。

finalize 是 Object 类的一个方法，在垃圾收集器执行的时候会调用被回收对象的此方法，可以覆盖此方法提供垃圾收集时的其他资源回收，例如关闭文件等。

13、sleep() 和 wait() 有什么区别？

sleep 是线程类（Thread）的方法，导致此线程暂停执行指定时间，给执行机会给其他线程，但是监控状态依然保持，

到时后会自动恢复。调用 sleep 不会释放对象锁。wait 是 Object 类的方法，对此对象调用 wait 方法导致本线程放弃对象锁，

进入等待此对象的等待锁定池，只有针对此对象发出 notify 方法（或 notifyAll）后本线程才进入对象锁定池准备获得对象

锁进入运行状态。

14、Overload 和 Override 的区别。Overloaded 的方法是否可以改变返回值的类型？

方法的重写 Overriding 和重载 Overloading 是 Java 多态性的不同表现。重写 Overriding 是父类与

子类之间多态性的一种表现，

重载 **Overloading** 是一个类中多态性的一种表现。如果在子类中定义某方法与其父类有相同的名称和参数，

我们说该方法被重写 (**Overriding**)。子类的对象使用这个方法时，将调用子类中的定义，对它而言，

父类中的定义如同被"屏蔽"了。如果在一个类中定义了多个同名的方法，它们或有不同的参数个数或有不同的参数类型，

则称为方法的重载 (**Overloading**)。Overloaded 的方法是可以改变返回值的类型。

15、error 和 exception 有什么区别? error

表示恢复不是不可能但很困难的情况下的一种严重问题。比如说内存溢出。不可能指望程序能处理这样的情况。 **exception** 表示一种设计或实现问题。也就是说，它表示如果程序运行正常，从不会发生的情况。

16、同步和异步有何异同，在什么情况下分别使用他们？举例说明。

如果数据将在线程间共享。例如正在写的的数据以后可能被另一个线程读到，或者正在读的数据可能已经被另一个线程写过了，那么这些数据就是共享数据，必须进行同步存取。

当应用程序在对象上调用了一个需要花费很长时间来执行的方法，并且不希望让程序等待方法的返回时，就应该使用异步编程，在很多情况下采用异步途径往往更有效率。

17、abstract class 和 interface 有什么区别？

声明方法的存在而不去实现它的类被叫做抽象类 (**abstract class**)，它用于要创建一个体现某些基本行为的类，

并为该类声明方法，但不能在该类中实现该方法的情况。不能创建 **abstract** 类的实例。然而可以创建一个变量，

其类型是一个抽象类，并让它指向具体子类的一个实例。不能有抽象构造函数或抽象静态方法。

Abstract 类的子类为它们父类中的所有抽象方法提供实现，否则它们也是抽象类为。取而代之，在子类中实现该方法。知道其行为的其它类可以在类中实现这些方法。接口 (**interface**) 是抽象类的变体。

在接口中，所有方法都是抽象的。多继承性可通过实现这样的接口而获得。接口中的所有方法都是抽象的，

没有一个有程序体。接口只可以定义 **static final** 成员变量。接口的实现与子类相似，除了该实现类不能从接口定义中继承行为。

当类实现特殊接口时，它定义（即将程序体给予）所有这种接口的方法。然后，它可以在实现了该接口的类的任何对象上调用接口

的方法。由于有抽象类，它允许使用接口名作为引用变量的类型。通常的动态联编将生效。引用可以转换到接口类型或从接口

类型转换， **instanceof** 运算符可以用来决定某对象的类是否实现了接口

18、heap 和 stack 有什么区别。

栈是一种线形集合，其添加和删除元素的操作应在同一段完成。栈按照后进先出的方式进行处理。堆是栈的一个组成元素

19、forward 和 redirect 的区别

forward 是服务器请求资源，服务器直接访问目标地址的 URL，把那个 URL 的响应内容读取过来，然后把这些内容再发给浏览器，
浏览器根本不知道服务器发送的内容是从哪儿来的，所以它的地址栏中还是原来的地址。redirect 就是服务端根据逻辑，
发送一个状态码，告诉浏览器重新去请求那个地址，一般来说浏览器会用刚才请求的所有参数重新请求，
所以 session,request 参数都可以获取。

20、EJB 与 JAVA BEAN 的区别？ Java Bean 是可复用的组件，对 Java

Bean 并没有严格的规范，理论上讲，任何一个 Java 类都可以是一个 Bean。但通常情况下，由于 Java Bean 是被容器所创建（如 Tomcat）的，所以 Java Bean 应具有一个无参的构造器，另外，通常 Java Bean 还要实现 Serializable 接口用于实现 Bean 的持久性。Java Bean 实际上相当于微软 COM 模型中的本地进程内 COM 组件，它是不能被跨进程访问的。Enterprise Java Bean 相当于 DCOM，即分布式组件。它是基于 Java 的远程方法调用（RMI）技术的，所以 EJB 可以被
远程访问（跨进程、跨计算机）。但 EJB 必须被布署在诸如 Websphere、WebLogic 这样的容器？
校斑 JB 客户从不直接访问真正的 EJB 组件，而是通过其容器访问。EJB 容器是 EJB 组件的代理？
斑 JB 组件由容器所创建和管理。客户通过容器来访问真正的 EJB 组件。

21、Static Nested

Class 和 Inner Class 的不同。Static Nested

Static Nested Class 是被声明为静态（static）的内部类，
它可以不依赖于外部类实例被实例化。而通常的内部类需要在外部类实例化后才能实例化。

22、JSP 中动态 INCLUDE 与静态 INCLUDE 的区别？

动态 INCLUDE 用 jsp:include 动作实现 <jsp:include page="included.jsp" flush="true" />它总是会检查所含文件中的变化，
适用于包含动态页面，并且可以带参数。
静态 INCLUDE 用 include 伪码实现，定不会检查所含文件的变化，适用于包含静态页面 <%@ include file="included.htm" %>

23、什么时候用 assert。

assertion(断言)在软件开发中是一种常用的调试方式，很多开发语言中都支持这种机制。在实现中，
assertion 就是在程序中的一条语句，它对一个 boolean 表达式进行检查，一个正确程序必须保证这个 boolean 表达式的值为 true；
如果该值为 false，说明程序已经处于不正确的状态下，系统将给出警告或退出。一般来说，
assertion 用于保证程序最基本、关键的正确性。assertion 检查通常在开发和测试时开启。为了提高性能，在软件发布后，
assertion 检查通常是关闭的。

24、GC 是什么? 为什么要有 GC?

GC 是垃圾收集的意思 (Garbage Collection), 内存处理是编程人员容易出现问题的地方, 忘记或者错误的内存回收会导致程序或系统的不稳定甚至崩溃, Java 提供的 GC 功能可以自动监测对象是否超过作用域从而达到自动回收内存的目的, Java 语言没有提供释放已分配内存的显示操作方法。

25、short s1 = 1; s1 = s1 + 1; 有什么错? short s1 = 1; s1 += 1; 有什么错?

short s1 = 1; s1 = s1 + 1; (s1+1 运算结果是 int 型, 需要强制转换类型)

short s1 = 1; s1 += 1; (可以正确编译)

26、Math.round(11.5) 等於多少? Math.round(-11.5) 等於多少?

Math.round(11.5) == 12

Math.round(-11.5) == -11

round 方法返回与参数最接近的长整数, 参数加 1/2 后求其 floor.

27、String s = new String("xyz"); 创建了几个 String Object?

两个

28、设计 4 个线程, 其中两个线程每次对 j 增加 1, 另外两个线程对 j 每次减少 1。写出程序。

以下程序使用内部类实现线程, 对 j 增减的时候没有考虑顺序问题。

```
public class ThreadTest1 {
    private int j;
    public static void main(String args[]) {
        ThreadTest1 tt = new ThreadTest1();
        Inc inc = tt.new Inc();
        Dec dec = tt.new Dec();
        for (int i = 0; i < 2; i++) {
            Thread t = new Thread(inc);
            t.start();
            t = new Thread(dec);
            t.start();
        }
    }
    private synchronized void inc() {
        j++;
        System.out.println(Thread.currentThread().getName() + "-inc:" + j);
    }
    private synchronized void dec() {
        j--;
        System.out.println(Thread.currentThread().getName() + "-dec:" + j);
    }
    class Inc implements Runnable {
        public void run() {
```



```

for(int i=0;i<100;i++){
inc();
}
}
}
class Dec implements Runnable{
public void run(){
for(int i=0;i<100;i++){
dec();
}
}
}
}
}

```

29、Java 有没有 goto?

java 中的保留字，现在没有在 java 中使用。

30、启动一个线程是用 run()还是 start()?

启动一个线程是调用 start()方法，使线程所代表的虚拟处理机处于可运行状态，这意味着它可以由 JVM 调度并执行。

这并不意味着线程就会立即运行。run()方法可以产生必须退出的标志来停止一个线程。

31、EJB 包括（SessionBean,EntityBean）说出他们的生命周期，及如何管理事务的？

SessionBean: Stateless Session Bean 的生命周期是由容器决定的，当客户机发出请求要建立一个 Bean 的实例时，

EJB 容器不一定要创建一个新的 Bean 的实例供客户机调用，而是随便找一个现有的实例提供给客户机。

当客户机第一次调用一个 Stateful Session Bean 时，容器必须立即在服务器中创建一个新的 Bean 实例，并关联到客户机上，

以后此客户机调用 Stateful Session Bean 的方法时容器会把调用分派到与此客户机相关联的 Bean 实例。

EntityBean: Entity Beans 能存活相对较长的时间，并且状态是持续的。只要数据库中的数据存在，Entity beans 就一直存活。

而不是按照应用程序或者服务进程来说的。即使 EJB 容器崩溃了，Entity beans 也是存活的。

Entity Beans 生命周期能够被容器或者 Beans 自己管理。EJB 通过以下技术管理实务：对象管理组织（OMG）的对象实务服务（OTS），

Sun Microsystems 的 Transaction Service（JTS）、Java Transaction API（JTA），开发组（X/Open）的 XA 接口。

32、应用服务器有那些？

BEA WebLogic Server，IBM WebSphere Application Server，Oracle9i Application Server，jBoss，Tomcat

33、给我一个你最常见到的 runtime exception。

ArithmeticException, ArrayStoreException, BufferOverflowException, BufferUnderflowException, CannotRedoException,

CannotUndoException, ClassCastException, CMMException, ConcurrentModificationException, DOMException,

EmptyStackException, IllegalArgumentException, IllegalMonitorStateException, IllegalPathStateException,

IllegalStateException, ImagingOpException, IndexOutOfBoundsException, MissingResourceException,

NegativeArraySizeException, NoSuchElementException, NullPointerException, ProfileDataException, ProviderException, RasterFormatException, SecurityException, SystemException, UndeclaredThrowableException,

UnmodifiableSetException, UnsupportedOperationException

34、接口是否可继承接口？抽象类是否可实现 (implements) 接口？抽象类是否可继承实体类 (concrete class)？

接口可以继承接口。抽象类可以实现 (implements) 接口，抽象类是否可继承实体类，但前提是实体类必须有明确的构造函数。

35、List, Set, Map 是否继承自 Collection 接口？

List, Set 是，Map 不是

36、说出数据连接池的工作机制是什么？

J2EE 服务器启动时会建立一定数量的池连接，并一直维持不少于此数目的池连接。客户端程序需要连接时，

池驱动程序会返回一个未使用的池连接并将其标记为忙。如果当前没有空闲连接，池驱动程序就新建一定数量的连接，

新建连接的数量有配置参数决定。当使用的池连接调用完成后，池驱动程序将此连接标记为空闲，其他调用就可以使用这个连接。

37、abstract 的 method 是否可同时是 static, 是否可同时是 native, 是否可同时是 synchronized? 都不能

38、数组有没有 length() 这个方法？String 有没有 length() 这个方法？

数组没有 length() 这个方法，有 length 的属性。String 有 length() 这个方法。

39、Set 里的元素是不能重复的，那么用什么方法来区分重复与否呢？是用 == 还是 equals()？它们有何区别？

Set 里的元素是不能重复的，那么用 iterator() 方法来区分重复与否。equals() 是判断两个 Set 是否相等。

equals() 和 == 方法决定引用值是否指向同一对象 equals() 在类中被覆盖，为的是当两个分离的对象的内容和类型相配的话，返回真值。

40、构造器 Constructor 是否可被 override？

构造器 Constructor 不能被继承，因此不能重写 Overriding，但可以被重载 Overloading。

41、是否可以继承 String 类?

String 类是 final 类故不可以继承。

42、switch 是否能作用在 byte 上，是否能作用在 long 上，是否能作用在 String 上?

switch (expr1) 中，expr1 是一个整数表达式。因此传递给 switch 和 case 语句的参数应该是 int、short、char

或者 byte。long,string 都不能作用于 switch。

43、try {} 里有一个 return 语句，那么紧跟在这个 try 后的 finally {} 里的 code 会不会被执行，什么时候被执行，在 return 前还是后?

会执行，在 return 前执行。

44、编程题：用最有效率的方法算出 2 乘以 8 等於几?

$2 \ll 3$

45、两个对象值相同(x.equals(y) == true)，但却可有不同的 hash code，这句话对不对?

不对，有相同的 hash code。

46、当一个对象被当作参数传递到一个方法后，此方法可改变这个对象的属性，并可返回变化后的结果，

那么这里到底是值传递还是引用传递?

是值传递。Java 编程语言只有值传递参数。当一个对象实例作为一个参数被传递到方法中时，参数的值就是对该对象的引用。对象的内容可以在被调用的方法中改变，但对象的引用是永远不会改变的。

47、当一个线程进入一个对象的一个 synchronized 方法后，其它线程是否可进入此对象的其它方法?

不能，一个对象的一个 synchronized 方法只能由一个线程访问。

48、编程题：写一个 Singleton 出来。

Singleton 模式主要作用是保证在 Java 应用程序中，一个类 Class 只有一个实例存在。

一般 Singleton 模式通常有几种形式：

第一种形式：定义一个类，它的构造函数为 private 的，它有一个 static 的 private 的该类变量，在类初始化时实例化，通过一个 public 的 getInstance 方法获取对它的引用，继而调用其中的方法。

```
public class Singleton {
```

```
private Singleton(){}
```

```
    //在自己内部定义自己一个实例，是不是很奇怪?
```

```
    //注意这是 private 只供内部调用
```

```
    private static Singleton instance = new Singleton();
```

```
    //这里提供了一个供外部访问本 class 的静态方法，可以直接访问
```

```
    public static Singleton getInstance() {
```

```
        return instance;
```

```
    }
```

```
}
```

第二种形式:

```
public class Singleton {  
    private static Singleton instance = null;  
    public static synchronized Singleton getInstance() {  
        //这个方法比上面有所改进,不用每次都进行生成对象,只是第一次  
        //使用时生成实例,提高了效率!  
        if (instance==null)  
            instance=new Singleton();  
        return instance;    }  
}
```

其他形式:

定义一个类,它的构造函数为 `private` 的,所有方法为 `static` 的。

一般认为第一种形式要更加安全些

49、Java 的接口和 C++的虚类的相同和不同处。

由于 Java 不支持多继承,而有可能某个类或对象要使用分别在几个类或对象里面的方法或属性,现有的单继承机制就不能满足要求。

与继承相比,接口有更高的灵活性,因为接口中没有任何实现代码。当一个类实现了接口以后,该类要实现接口里面所有的方法和

属性,并且接口里面的属性在默认状态下面都是 `public static`,所有方法默认情况下是 `public`.一个类可以实现多个接口。

50、Java 中的异常处理机制的简单原理和应用。

当 JAVA 程序违反了 JAVA 的语义规则时, JAVA 虚拟机就会将发生的错误表示为一个异常。违反语义规则包括 2 种情况。一种是 JAVA 类库

内置的语义检查。例如数组下标越界,会引发 `IndexOutOfBoundsException`;访问 `null` 的对象时会引发 `NullPointerException`。

另一种情况就是 JAVA 允许程序员扩展这种语义检查,程序员可以创建自己的异常,并自由选择何时用 `throw` 关键字引发异常。

所有的异常都是 `java.lang.Throwable` 的子类。

7.7

面试题汇集(二)

51、垃圾回收的优点和原理。并考虑 2 种回收机制。

Java 语言中一个显著的特点就是引入了垃圾回收机制,使 c++程序员最头疼的内存管理的问题迎刃而解,它使得 Java 程序员在编写

程序的时候不再需要考虑内存管理。由于有个垃圾回收机制,Java 中的对象不再有"作用域"的概念,只有对象的引用才有"作用域"。

垃圾回收可以有效的防止内存泄露,有效的使用可以使用的内存。垃圾回收器通常是作为一个单独的低级别的线程运行,

不可预知的情况下对内存堆中已经死亡的或者长时间没有使用的对象进行清楚和回收,程序员不能实时的调用垃圾回收器对某个对象

或所有对象进行垃圾回收。回收机制有分代复制垃圾回收和标记垃圾回收,增量垃圾回收。

52、请说出你所知道的线程同步的方法。

`wait()`:使一个线程处于等待状态,并且释放所持有的对象的 `lock`。

`sleep()`:使一个正在运行的线程处于睡眠状态,是一个静态方法,调用此方法要捕捉 `InterruptedException` 异常。

`notify()`:唤醒一个处于等待状态的线程,注意的是在调用此方法的时候,并不能确切的唤醒某一个等待状态的线程,而是由 JVM 确定

唤醒哪个线程,而且不是按优先级。`Allnotity()`:唤醒所有处于等待状态的线程,注意并不是给所有唤醒线程一个对象的锁,而是让它们竞争。

53、你所知道的集合类都有哪些?主要方法?

最常用的集合类是 `List` 和 `Map`。`List` 的具体实现包括 `ArrayList` 和 `Vector`,它们是可变大小的列表,比较适合构建、存储

和操作任何类型对象的元素列表。`List` 适用于按数值索引访问元素的情形。`Map` 提供了一个更通用的元素存储方法。

`Map` 集合类用于存储元素对(称作"键"和"值"),其中每个键映射到一个值。

54、描述一下 JVM 加载 class 文件的原理机制?

JVM 中类的装载是由 `ClassLoader` 和它的子类来实现的,`Java ClassLoader` 是一个重要的 Java 运行时系统组件。它负责在运行时查找和装入类文件的类。

55、`char` 型变量中能不能存贮一个中文汉字?为什么?

能够定义成为一个中文的,因为 java 中以 `unicode` 编码,一个 `char` 占 16 个字节,所以放一个中文是没问题的

56、多线程有几种实现方法,都是什么?同步有几种实现方法,都是什么?

多线程有两种实现方法,分别是继承 `Thread` 类与实现 `Runnable` 接口同步的实现方面有两种,分别是 `synchronized`,`wait` 与 `notify`

57、JSP 的内置对象及方法。

`request` 表示 `HttpServletRequest` 对象。它包含了有关浏览器请求的信息,并且提供了几个用于获取 `cookie`,`header`,和 `session` 数据的有用的方法。

`response` 表示 `HttpServletResponse` 对象,并提供了几个用于设置送回浏览器的响应的方法(如 `cookies`,头信息等)

`out` 对象是 `javax.jsp.JspWriter` 的一个实例,并提供了几个方法使你能用于向浏览器回送输出结果。

`pageContext` 表示一个 `javax.servlet.jsp.PageContext` 对象。它是用于方便存取各种范围的名字空间、`servlet` 相关的对象的 API,并且包装了通用的 `servlet` 相关功能的方法。

`session` 表示一个请求的 `javax.servlet.http.HttpSession` 对象。`Session` 可以存贮用户的状态信息

`applicaton` 表示一个 `javax.servle.ServletContext` 对象。这有助于查找有关 `servlet` 引擎和 `servlet` 环境的信息

`config` 表示一个 `javax.servlet.ServletConfig` 对象。该对象用于存取 `servlet` 实例的初始化参数。

`page` 表示从该页面产生的一个 `servlet` 实例

58、线程的基本概念、线程的基本状态以及状态之间的关系

线程指在程序执行过程中，能够执行程序代码的一个执行单位，每个程序至少都有一个线程，也就是程序本身。

Java 中的线程有四种状态分别是：运行、就绪、挂起、结束。

59、JSP 的常用指令

```
<%@page language="java" contentType="text/html; charset=gb2312" session="true" buffer="64kb" autoFlush="true" isThreadSafe="true" info="text" errorPage="error.jsp" isErrorPage="true" isELIgnored="true" pageEncoding="gb2312" import="java.sql.*"%>
```

isErrorPage(是否能使用 Exception 对象), isELIgnored(是否忽略表达式)

```
<%@include file="filename"%>
```

```
<%@taglib prefix="c" uri="http://....."%>
```

60、什么情况下调用 doGet() 和 doPost()?

Jsp 页面中的 form 标签里的 method 属性为 get 时调用 doGet(), 为 post 时调用 doPost()。

61、servlet 的生命周期

web 容器加载 servlet，生命周期开始。通过调用 servlet 的 init() 方法进行 servlet 的初始化。通过调用 service() 方法实现

，根据请求的不同调用不同的 do***() 方法。结束服务，web 容器调用 servlet 的 destroy() 方法。

62、如何现实 servlet 的单线程模式

```
<%@ page isThreadSafe="false"%>
```

63、页面间对象传递的方法

request, session, application, cookie 等

64、JSP 和 Servlet 有哪些相同点和不同点，他们之间的联系是什么？

JSP 是 Servlet 技术的扩展，本质上是 Servlet 的简易方式，更强调应用的外表表达。JSP 编译后是“类 servlet”。

Servlet 和 JSP 最主要的不同点在于，Servlet 的应用逻辑是在 Java 文件中，并且完全从表示层中的 HTML 里分离开来。

而 JSP 的情况是 Java 和 HTML 可以组合成一个扩展名为 .jsp 的文件。JSP 侧重于视图，Servlet 主要用于控制逻辑。

65、四种会话跟踪技术

会话作用域 Servlets JSP 页面描述

page 否是代表与一个页面相关的对象和属性。一个页面由一个编译好的 Java servlet 类（可以带有任何的 include 指令，但是没有 include 动作）表示。这既包括 servlet 又包括被编译成 servlet 的 JSP 页面

request 是代表与 Web 客户机发出的一个请求相关的对象和属性。一个请求可能跨越多个页面，涉及多个 Web 组件（由于 forward 指令和 include 动作的关系）

session 是代表与用于某个 Web 客户机的一个用户体验相关的对象和属性。一个 Web 会话可以

也经常会跨越多个客户机请求

application 是代表与整个 Web 应用程序相关的对象和属性。这实质上是跨越整个 Web 应用程序，包括多个页面、请求和会话的一个全局作用域

66、Request 对象的主要方法：

setAttribute(String name,Object)：设置名字为 name 的 request 的参数值

getAttribute(String name)：返回由 name 指定的属性值

getAttributeNames()：返回 request 对象所有属性的名字集合，结果是一个枚举的实例

getCookies()：返回客户端的所有 Cookie 对象，结果是一个 Cookie 数组

getCharacterEncoding()：返回请求中的字符编码方式

getContentLength()：返回请求的 Body 的长度

getHeader(String name)：获得 HTTP 协议定义的文件头信息

getHeaders(String name)：返回指定名字的 request Header 的所有值，结果是一个枚举的实例

getHeaderNames()：返回所以 request Header 的名字，结果是一个枚举的实例

getInputStream()：返回请求的输入流，用于获得请求中的数据

getMethod()：获得客户端向服务器端传送数据的方法

getParameter(String name)：获得客户端传送给服务器端的有 name 指定的参数值

getParameterNames()：获得客户端传送给服务器端的所有参数的名字，结果是一个枚举的实例

getParameterValues(String name)：获得有 name 指定的参数的所有值

getProtocol()：获取客户端向服务器端传送数据所依据的协议名称

getQueryString()：获得查询字符串

getRequestURI()：获取发出请求字符串的客户端地址

getRemoteAddr()：获取客户端的 IP 地址

getRemoteHost()：获取客户端的名字

getSession([Boolean create])：返回和请求相关 Session

getServerName()：获取服务器的名字

getServletPath()：获取客户端所请求的脚本文件的路径

getServerPort()：获取服务器的端口号

removeAttribute(String name)：删除请求中的一个属性

67、J2EE 是技术还是平台还是框架？

J2EE 本身是一个标准，一个为企业分布式应用的开发提供的标准平台。

J2EE 也是一个框架，包括 JDBC、JNDI、RMI、JMS、EJB、JTA 等技术。

68、我们在 web 应用开发过程中经常遇到输出某种编码的字符，如 iso8859-1 等，如何输出一个某种编码的字符串？

```
Public String translate(String str) {  
    String tempStr = "";  
    try {  
        tempStr = new String(str.getBytes("ISO-8859-1"), "GBK");  
        tempStr = tempStr.trim();  
    }  
    catch (Exception e) {  
        System.err.println(e.getMessage());  
    }  
}
```

```
}  
return tempStr;  
}
```

69、简述逻辑操作 (&,&^)与条件操作(&&,&||)的区别。

区别主要答两点： a.条件操作只能操作布尔型的,而逻辑操作不仅可以操作布尔型,而且可以操作数值型

b.逻辑操作不会产生短路

70、XML 文档定义有几种形式？它们之间有何本质区别？解析 XML 文档有哪几种方式？

a: 两种形式 dtd schema,

b: 本质区别: schema 本身是 xml 的, 可以被 XML 解析器解析(这也是从 DTD 上发展 schema 的根本目的),

c: 有 DOM,SAX,STAX 等

DOM: 处理大型文件时其性能下降的非常厉害。这个问题是由 DOM 的树结构所造成的, 这种结构占用的内存较多,

而且 DOM 必须在解析文件之前把整个文档装入内存, 适合对 XML 的随机访问 SAX: 不现于 DOM, SAX 是事件驱动型的 XML 解析方式。

它顺序读取 XML 文件, 不需要一次全部装载整个文件。当遇到像文件开头, 文档结束, 或者标签开头与标签结束时,

它会触发一个事件, 用户通过在其回调事件中写入处理代码来处理 XML 文件, 适合对 XML 的顺序访问

STAX: Streaming API for XML (StAX)

71、简述 synchronized 和 java.util.concurrent.locks.Lock 的异同？

主要相同点: Lock 能完成 synchronized 所实现的所有功能

主要不同点: Lock 有比 synchronized 更精确的线程语义和更好的性能。synchronized 会自动释放锁,

而 Lock 一定要求程序员手工释放, 并且必须在 finally 从句中释放。

72、EJB 的角色和三个对象

一个完整的基于 EJB 的分布式计算结构由六个角色组成, 这六个角色可以由不同的开发商提供, 每个角色所作的工作必须遵循 Sun 公司提供的 EJB 规范, 以保证彼此之间的兼容性。这六个角色分别是 EJB 组件开发者 (Enterprise Bean Provider)、应用组合者 (Application Assembler)、部署者 (Deployer)、EJB 服务器提供者 (EJB Server Provider)、EJB 容器提供者 (EJB Container Provider)、系统管理员 (System Administrator)

三个对象是 Remote (Local) 接口、Home (LocalHome) 接口, Bean 类

73、EJB 容器提供的服务

主要提供声明周期管理、代码产生、持续性管理、安全、事务管理、锁和并发管理等服务。

74、EJB 规范规定 EJB 中禁止的操作有哪些？

1. 不能操作线程和线程 API (线程 API 指非线程对象的方法如 notify, wait 等),

2. 不能操作 awt,

- 3.不能实现服务器功能，
- 4.不能对静态属性存取，
- 5.不能使用 IO 操作直接存取文件系统，
- 6.不能加载本地库，
- 7.不能将 this 作为变量和返回，
- 8.不能循环调用。

75、remote 接口和 home 接口主要作用

remote 接口定义了业务方法，用于 EJB 客户端调用业务方法。

home 接口是 EJB 工厂用于创建和移除查找 EJB 实例

76、bean 实例的生命周期

对于 Stateless Session Bean、Entity Bean、Message Driven Bean 一般存在缓冲池管理，而对于 Entity Bean 和

Statefull Session Bean 存在 Cache 管理，通常包含创建实例，设置上下文、创建 EJB Object(create)、业务方法调用、

remove 等过程，对于存在缓冲池管理的 Bean，在 create 之后实例并不从内存清除，而是采用缓冲池调度机制不断重用实例，

而对于存在 Cache 管理的 Bean 则通过激活和去激活机制保持 Bean 的状态并限制内存中实例数量。

77、EJB 的激活机制

以 Stateful Session Bean 为例：其 Cache 大小决定了内存中可以同时存在的 Bean 实例的数量，根据 MRU 或 NRU 算法，

实例在激活和去激活状态之间迁移，激活机制是当客户端调用某个 EJB 实例业务方法时，如果对应 EJB Object 发现自己

没有绑定对应的 Bean 实例则从其去激活 Bean 存储中（通过序列化机制存储实例）回复（激活）此实例。状态变迁前会调用

对应的 ejbActive 和 ejbPassivate 方法。

78、EJB 的几种类型

会话（Session）Bean，实体（Entity）Bean 消息驱动的（Message Driven）Bean

会话 Bean 又可分为有状态（Stateful）和无状态（Stateless）两种

实体 Bean 可分为 Bean 管理的持续性（BMP）和容器管理的持续性（CMP）两种

79、客户端调用 EJB 对象的几个基本步骤

设置 JNDI 服务工厂以及 JNDI 服务地址系统属性，查找 Home 接口，从 Home 接口调用 Create 方法创建 Remote 接口，

通过 Remote 接口调用其业务方法。

80、如何给 weblogic 指定大小的内存？

在启动 Weblogic 的脚本中（位于所在 Domain 对应服务器目录下的 startServerName），增加 set MEM_ARGS=-Xms32m -Xmx200m，可以调整最小内存为 32M，最大 200M

81、如何设定的 weblogic 的热启动模式(开发模式)与产品发布模式？

可以在管理控制台中修改对应服务器的启动模式为开发或产品模式之一。或者修改服务的启动文件或者 `commenv` 文件，

增加 `set PRODUCTION_MODE=true`。

82、如何启动时不需输入用户名与密码？

修改服务启动文件，增加 `WLS_USER` 和 `WLS_PW` 项。也可以在 `boot.properties` 文件中增加加密过的用户名和密码。

83、在 weblogic 管理制台中对一个应用域(或者说是一个网站,Domain)进行 jms 及 ejb 或连接池等相关信息进行配置后，

实际保存在什么文件中？

保存在此 Domain 的 `config.xml` 文件中，它是服务器的核心配置文件。

84、说说 weblogic 中一个 Domain 的缺省目录结构？比如要将一个简单的 `helloWorld.jsp` 放入何目录下，

然的在浏览器上就可打入 <http://主机:端口号/helloworld.jsp> 就可以看到运行结果了？又比如这其中用到了一个自己写的 `javaBean` 该如何办？

Domain 目录服务器目录 `applications`，将应用目录放在此目录下将可以作为应用访问，如果是 Web 应用，

应用目录需要满足 Web 应用目录要求，`.jsp` 文件可以直接放在应用目录中，`JavaBean` 需要放在应用目录的 `WEB-INF`

目录的 `classes` 目录中，设置服务器的缺省应用将可以实现在浏览器上无需输入应用名。

85、在 weblogic 中发布 ejb 需涉及到哪些配置文件

不同类型的 EJB 涉及的配置文件不同，都涉及到的配置文件包括 `ejb-jar.xml`, `weblogic-ejb-jar.xml` `CMP` 实体 Bean

一般还需要 `weblogic-cmp-rdbms-jar.xml`

86、如何在 weblogic 中进行 ssl 配置与客户端的认证配置或说说 j2ee(标准)进行 ssl 的配置

缺省安装中使用 `DemoIdentity.jks` 和 `DemoTrust.jks` `KeyStore` 实现 SSL，需要配置服务器使用 `Enable SSL`，配置其端口，

在产品模式下需要从 CA 获取私有密钥和数字证书，创建 `identity` 和 `trust keystore`，装载获得的密钥和数字证书。

可以配置此 SSL 连接是单向还是双向的。

87、如何查看在 weblogic 中已经发布的 EJB？

可以使用管理控制台，在它的 `Deployment` 中可以查看所有已发布的 EJB

88、CORBA 是什么？用途是什么？

CORBA 标准是公共对象请求代理结构 (Common Object Request Broker Architecture)，

由对象管理组织 (Object Management Group，缩写为 `OMG`) 标准化。它的组成是接口定义语言 (`IDL`)，

语言绑定 (`binding`; 也译为联编) 和允许应用程序间互操作的协议。其目的为：用不同的程序设计语

言书写在不同的进程中运行，
为不同的操作系统开发。

89、说说你所熟悉或听说过的 j2ee 中的几种常用模式?及对设计模式的一些看法

Session Facade Pattern: 使用 SessionBean 访问 EntityBean

Message Facade Pattern: 实现异步调用

EJB Command Pattern: 使用 Command JavaBeans 取代 SessionBean, 实现轻量级访问

Data Transfer Object Factory: 通过 DTO Factory 简化 EntityBean 数据提供特性

Generic Attribute Access: 通过 AttributeAccess 接口简化 EntityBean 数据提供特性

Business Interface: 通过远程(本地)接口和 Bean 类实现相同接口规范业务逻辑一致性

EJB 架构的设计好坏将直接影响系统的性能、可扩展性、可维护性、组件可重用性及开发效率。

项目越复杂,

项目队伍越庞大则越能体现良好设计的重要性。

90、说说在 weblogic 中开发消息 Bean 时的 persistent 与 non-persistent 的差别

persistent 方式的 MDB 可以保证消息传递的可靠性,也就是如果 EJB 容器出现问题而 JMS 服务器依然会将消息在此 MDB 可用的时候

发送过来,而 non-persistent 方式的消息将被丢弃。

91、Servlet 执行时一般实现哪几个方法?

```
public void init(ServletConfig config)
```

```
public ServletConfig getServletConfig()
```

```
public String getServletInfo()
```

```
public void service(ServletRequest request,ServletResponse response)
```

```
public void destroy()
```

92、j2ee 常用的设计模式?说明工厂模式。

Java 中的 23 种设计模式:

Factory (工厂模式), Builder (建造模式), Factory Method (工厂方法模式),

Prototype (原始模型模式), Singleton (单例模式), Facade (门面模式),

Adapter (适配器模式), Bridge (桥梁模式), Composite (合成模式),

Decorator (装饰模式), Flyweight (享元模式), Proxy (代理模式),

Command (命令模式), Interpreter (解释器模式), Visitor (访问者模式),

Iterator (迭代子模式), Mediator (调停者模式), Memento (备忘录模式),

Observer (观察者模式), State (状态模式), Strategy (策略模式),

Template Method (模板方法模式), Chain Of Responsibility (责任链模式)

工厂模式: 工厂模式是一种经常被使用到的模式,根据工厂模式实现的类可以根据提供的数据生成一组类中某一个类的实例,

通常这一组类有一个公共的抽象父类并且实现了相同的方法,但是这些方法针对不同的数据进行了不同的操作。

首先需要定义一个基类,该类的子类通过不同的方法实现了基类中的方法。然后需要定义一个工厂类,

工厂类可以根据条件生成不同的子类实例。当得到子类的实例后,开发人员可以调用基类中的方法而不必考虑到底返回的是哪

一个子类的实例。

93、EJB 需直接实现它的业务接口或 Home 接口吗，请简述理由。

远程接口和 Home 接口不需要直接实现，他们的实现代码是由服务器产生的，程序运行中对应实现类会作为对应接口类型的实例被使用。

94、排序都有哪几种方法？请列举。用 JAVA 实现一个快速排序。

排序的方法有：插入排序（直接插入排序、希尔排序），交换排序（冒泡排序、快速排序），选择排序（直接选择排序、堆排序），归并排序，分配排序（箱排序、基数排序）

快速排序的伪代码。

//使用快速排序方法对 $a[0..n-1]$ 排序

从 $a[0..n-1]$ 中选择一个元素作为 $middle$ ，该元素为支点

把余下的元素分割为两段 $left$ 和 $right$ ，使得 $left$ 中的元素都小于等于支点，而 $right$ 中的元素都大于等于支点

递归地使用快速排序方法对 $left$ 进行排序

递归地使用快速排序方法对 $right$ 进行排序

所得结果为 $left+middle+right$

95、请对以下在 J2EE 中常用的名词进行解释(或简单描述)

web 容器：给处于其中的应用程序组件（JSP，SERVLET）提供一个环境，使 JSP,SERVLET 直接更容器中的环境变量接**互，不必关注其它系统问题。主要有 WEB 服务器来实现。例如：TOMCAT,WEBLOGIC,WEBSPPHERE 等。该容器提供的接口严格遵守 J2EE 规范中的 WEB APPLICATION 标准。我们把遵守以上标准的 WEB 服务器就叫做 J2EE 中的 WEB 容器。

EJB 容器：Enterprise java bean 容器。更具有行业领域特色。他提供给运行在其中的组件 EJB 各种管理功能。只要满足 J2EE 规范的 EJB 放入 该容器，马上就会被容器进行高效率的管理。并且可以通过现成的接口来获得系统级别的服务。例如邮件服务、事务管理。

JNDI：（Java Naming & Directory Interface）JAVA 命名目录服务。主要提供的功能是：提供一个目录系统，让其它各地的应用程序在其上面留下自己的索引，从而满足快速查找和定位分布式应用程序的功能。

JMS：（Java Message Service）JAVA 消息服务。主要实现各个应用程序之间的通讯。包括点对点和广播。

JTA：（Java Transaction API）JAVA 事务服务。提供各种分布式事务服务。应用程序只需调用其提供的接口即可。

JAF：（Java Action FrameWork）JAVA 安全认证框架。提供一些安全控制方面的框架。让开发者通过各种部署和自定义实现自己的个性安全控制策略。

RMI/IIOP：（Remote Method Invocation /internet 对象请求中介协议）他们主要用于通过远程调用服务。例如，远程有一台计算机上运行一个程序，它提供股票分析服务，我们可以在本地计算机上实现对其直接调用。当然这是要通过一定的规范才能在异构的系统之间进行通信。RMI 是 JAVA 特有的。

96、JAVA 语言如何进行异常处理，关键字：throws,throw,try,catch,finally 分别代表什么意义？在 try 块中可以抛出异常吗？

Java 通过面向对象的方法进行异常处理，把各种不同的异常进行分类，并提供了良好的接口。在 Java 中，每个异常都是一个对象，它是 `Throwable` 类或其它子类的实例。当一个方法出现异常后便抛出一个异常对象，该对象中包含有异常信息，调用这个方法可以捕获到这个异常并进行处理。Java 的异常处理是通过 5 个关键词来实现的：`try`、`catch`、`throw`、`throws` 和 `finally`。一般情况下是用 `try` 来执行一段程序，如果出现异常，系统会抛出一个（throws）异常，这时候你可以通过它的类型来捕捉（`catch`）它，或最后（`finally`）由缺省处理器来处理。

用 `try` 来指定一块预防所有“异常”的程序。紧跟在 `try` 程序后面，应包含一个 `catch` 子句来指定你想要捕捉的“异常”的类型。

`throw` 语句用来明确地抛出一个“异常”。

`throws` 用来标明一个成员函数可能抛出的各种“异常”。

`Finally` 为确保一段代码不管发生什么“异常”都被执行一段代码。

可以在一个成员函数调用的外面写一个 `try` 语句，在这个成员函数内部写另一个 `try` 语句保护其他代码。每当遇到一个 `try` 语句，“异常”的框架就放到堆栈上面，直到所有的 `try` 语句都完成。如果下一级的 `try` 语句没有对某种“异常”进行处理，堆栈就会展开，直到遇到有处理这种“异常”的 `try` 语句。

97、一个“.java”源文件中是否可以包括多个类（不是内部类）？有什么限制？

可以。必须只有一个类名与文件名相同。

98、MVC 的各个部分都有那些技术来实现？如何实现？

MVC 是 Model—View—Controller 的简写。“Model”代表的是应用的业务逻辑（通过 `JavaBean`、`EJB` 组件实现），“View”是应用的表示面（由 `JSP` 页面产生），“Controller”是提供应用的处理过程控制（一般是一个 `Servlet`），通过这种设计模型把应用逻辑，处理过程和显示逻辑分成不同的组件实现。这些组件可以进行交互和重用。

99、java 中有几种方法可以实现一个线程？用什么关键字修饰同步方法？`stop()`和 `suspend()`方法为何不推荐使用？

有两种实现方法，分别是继承 `Thread` 类与实现 `Runnable` 接口

用 `synchronized` 关键字修饰同步方法

反对使用 `stop()`，是因为它不安全。它会解除由线程获取的所有锁定，而且如果对象处于一种不连贯状态，那么其他线程能在那种状态下检查和修

改它们。结果很难检查出真正的问题所在。`suspend()`方法容易发生死锁。调用 `suspend()`的时候，目标线程会停下来，但却仍然持有在这之前获得的锁定。此时，其他任何线程都不能访问锁定的资源，除非被“挂起”的线程恢复运行。对任何线程来说，如果它们想恢复目标线程，同时又试图使用任何一个锁定的资源，就会造成死锁。所以不应该使用 `suspend()`，而应在自己的 `Thread` 类中置入一个标志，指出线程应该活动还是挂起。若标志指出线程应该挂起，便用 `wait()`命其进入等待状态。若标志指出线程应当恢复，则用一个 `notify()`重新启动线程。

100、java 中有几种类型的流？JDK 为每种类型的流提供了一些抽象类以供继承，请说出他们分别是哪些类？

字节流，字符流。字节流继承于 `InputStream` `OutputStream`，字符流继承于 `InputStreamReader` `OutputStreamWriter`。在 `java.io` 包中还有许多其他的流，主要是为了提高性能和使用方便。

面试题汇集（三）

面试题汇集（三）

101、java 中会存在内存泄漏吗，请简单描述。

会。如：`int i,i2; return (i-i2);` //when i 为足够大的正数,i2 为足够大的负数。结果会造成溢位，导致错误。

102、java 中实现多态的机制是什么？

方法的重写 **Overriding** 和重载 **Overloading** 是 Java 多态性的不同表现。重写 **Overriding** 是父类与子类之间多态性的一种表现，重载 **Overloading** 是一个类中多态性的一种表现。

103、垃圾回收器的基本原理是什么？垃圾回收器可以马上回收内存吗？有什么办法主动通知虚拟机进行垃圾回收？

对于 GC 来说，当程序员创建对象时，GC 就开始监控这个对象的地址、大小以及使用情况。通常，GC 采用有向图的方式记录和管理堆(heap)中的所有对象。通过这种方式确定哪些对象是"可达的"，哪些对象是"不可达的"。当 GC 确定一些对象为"不可达"时，GC 就有责任回收这些内存空间。可以。程序员可以手动执行 `System.gc()`，通知 GC 运行，但是 Java 语言规范并不保证 GC 一定会执行。

104、静态变量和实例变量的区别？

`static i = 10;` //常量

`class A a; a.i =10;` //可变

105、什么是 java 序列化，如何实现 java 序列化？

序列化就是一种用来处理对象流的机制，所谓对象流也就是将对象的内容进行流化。可以对流化后的对象进行读写操作，也可将流化后的对象传输于网络之间。序列化是为了解决在对对象流进行读写操作时所引发的问题。

序列化的实现：将需要被序列化的类实现 **Serializable** 接口，该接口没有需要实现的方法，`implements Serializable` 只是为了标注该对象是可被序列化的，然后使用一个输出流 (如：`FileOutputStream`)来构造一个 `ObjectOutputStream`(对象流)对象，接着，使用 `ObjectOutputStream` 对象的 `writeObject(Object obj)`方法就可以将参数为 `obj` 的对象写出(即保存其状态)，要恢复的话则用输入流。

106、是否可以从一个 `static` 方法内部发出对非 `static` 方法的调用？

不可以,如果其中包含对象的 `method()`；不能保证对象初始化。

107、写 `clone()`方法时，通常都有一行代码，是什么？

`Clone` 有缺省行为，`super.clone()`；他负责产生正确大小的空间，并逐位复制。

108、在 JAVA 中，如何跳出当前的多重嵌套循环？

用 `break`; `return` 方法。

109、`List`、`Map`、`Set` 三个接口，存取元素时，各有什么特点？

`List` 以特定次序来持有元素，可有重复元素。`Set` 无法拥有重复元素,内部排序。`Map` 保存 `key-value`

值，value 可多值。

110、J2EE 是什么？

J2EE 是 Sun 公司提出的多层 (multi-tiered), 分布式 (distributed), 基于组件 (component-base) 的企业级应用模型

(enterprise application model)。在这样的一个应用系统中，可按照功能划分为不同的组件，这些组件又可在不同计算机上

，并且处于相应的层次 (tier) 中。所属层次包括客户层 (client tier) 组件, web 层和组件, Business 层和组件, 企业信息系统 (EIS) 层。

111、UML 方面

标准建模语言 UML。用例图, 静态图 (包括类图、对象图和包图), 行为图, 交互图 (顺序图, 合作图), 实现图。

112、说出一些常用的类，包，接口，请各举 5 个

常用的类：BufferedReader BufferedWriter FileReader FileWriter String Integer

常用的包：java.lang java.awt java.io java.util java.sql

常用的接口：Remote List Map Document NodeList

113、开发中都用到了那些设计模式？用在什么场合？

每个模式都描述了一个在我们的环境中不断出现的问题，然后描述了该问题的解决方案的核心。通过这种方式，

你可以无数次地使用那些已有的解决方案，无需在重复相同的工作。主要用到了 MVC 的设计模式。用来开发 JSP/Servlet

或者 J2EE 的相关应用。简单工厂模式等。

114、jsp 有哪些动作？作用分别是什么？

JSP 共有以下 6 种基本动作 jsp:include：在页面被请求的时候引入一个文件。jsp:useBean：寻找或者实例化一个 JavaBean。

jsp:setProperty：设置 JavaBean 的属性。jsp:getProperty：输出某个 JavaBean 的属性。

jsp:forward：把请求转到一个新的页面。jsp:plugin：根据浏览器类型为 Java 插件生成 OBJECT 或 EMBED 标记。

115、Anonymous Inner Class (匿名内部类) 是否可以 extends (继承) 其它类，是否可以 implements (实现) interface (接口)？

可以继承其他类或完成其他接口，在 swing 编程中常用此方式。

116、应用服务器与 WEB SERVER 的区别？

应用服务器：Weblogic、Tomcat、Jboss

WEB SERVER：IIS、Apache

117、BS 与 CS 的联系与区别。

C/S 是 Client/Server 的缩写。服务器通常采用高性能的 PC、工作站或小型机，并采用大型数据库系统，

如 Oracle、Sybase、Informix 或 SQL Server。客户端需要安装专用的客户端软件。B/S 是 Browser/Server 的缩写,

客户机上只要安装一个浏览器 (Browser), 如 Netscape Navigator 或 Internet Explorer, 服务器安装 Oracle、Sybase、Informix 或 SQL Server 等数据库。在这种结构下, 用户界面完全通过 WWW 浏览器实现,

一部分事务逻辑在前端实现, 但是主要事务逻辑在服务器端实现。浏览器通过 Web Server 同数据库进行数据交互。

C/S 与 B/S 区别:

1. 硬件环境不同:

C/S 一般建立在专用的网络上, 小范围里的网络环境, 局域网之间再通过专门服务器提供连接和数据交换服务。

B/S 建立在广域网之上的, 不必是专门的网络硬件环境, 例与电话上网, 租用设备. 信息自己管理. 有比 C/S 更强的适应范围, 一般只要有操作系统和浏览器就行

2. 对安全要求不同

C/S 一般面向相对固定的用户群, 对信息安全的控制能力很强. 一般高度机密的信息系统采用 C/S 结构适宜. 可以通过 B/S 发布部分可公开信息。

B/S 建立在广域网之上, 对安全的控制能力相对弱, 可能面向不可知的用户。

3. 对程序架构不同

C/S 程序可以更加注重流程, 可以对权限多层次校验, 对系统运行速度可以较少考虑。

B/S 对安全以及访问速度的多重的考虑, 建立在需要更加优化的基础之上. 比 C/S 有更高的要求 B/S 结构的程序架构是发展的趋势, 从 MS 的 .Net 系列的 BizTalk 2000 Exchange 2000 等, 全面支持网络的构件搭建的系统. SUN 和 IBM 推的 JavaBean 构件 技术等, 使 B/S 更加成熟。

4. 软件重用不同

C/S 程序可以不可避免的整体性考虑, 构件的重用性不如在 B/S 要求下的构件的重用性好。

B/S 对的多重结构, 要求构件相对独立的功能. 能够相对较好的重用. 就买来的餐桌可以再利用, 而不是做在墙上的石头桌子

5. 系统维护不同

C/S 程序由于整体性, 必须整体考察, 处理出现的问题以及系统升级. 升级难. 可能是再做一个全新的系统

B/S 构件组成, 方面构件个别的更换, 实现系统的无缝升级. 系统维护开销减到最小. 用户从网上自己下载安装就可以实现升级。

6. 处理问题不同

C/S 程序可以处理用户面固定, 并且在相同区域, 安全要求高需求, 与操作系统相关. 应该都是相同的系统

B/S 建立在广域网上, 面向不同的用户群, 分散地域, 这是 C/S 无法作到的. 与操作系统平台关系最小。

7. 用户接口不同

C/S 多是建立的 Window 平台上, 表现方法有限, 对程序员普遍要求较高

B/S 建立在浏览器上, 有更加丰富和生动的表现方式与用户交流. 并且大部分难度减低, 减低开发成本。

8. 信息流不同

C/S 程序一般是典型的中央集权的机械式处理, 交互性相对低

B/S 信息流向可变化, B-B B-C B-G 等信息、流向的变化, 更像交易中心。

118、Linux 下线程，GDI 类的解释。

Linux 实现的就是基于核心轻量级进程的"一对一"线程模型，一个线程实体对应一个核心轻量级进程，而线程之间的管理在核外

函数库中实现。GDI 类为图像设备编程接口类库。

119、Struts 的应用(如 Struts 架构)

Struts 是采用 Java Servlet/JavaServer Pages 技术，开发 Web 应用程序的开放源码的 framework。

采用 Struts 能开发出基于

MVC(Model-View-Controller)设计模式的应用构架。Struts 有如下的主要功能：

一.包含一个 controller servlet，能将用户的请求发送到相应的 Action 对象。

二.JSP 自由 tag 库，并且在 controller servlet 中提供关联支持，帮助开发员创建交互式表单应用。

三.提供了一系列实用对象：XML 处理、通过 Java reflection APIs 自动处理 JavaBeans 属性、国际化的提示和消息。

120、JDO 是什么？

JDO 是 Java 对象持久化的新的规范，为 java data object 的简称,也是一个用于存取某种数据仓库中的对象的标准化 API。

JDO 提供了透明的对象存储，因此对开发人员来说，存储数据对象完全不需要额外的代码（如 JDBC API 的使用）。

这些繁琐的例行工作已经转移到 JDO 产品提供商身上，使开发人员解脱出来，从而集中时间和精力在业务逻辑上。

另外，JDO 很灵活，因为它可以在任何数据底层上运行。JDBC 只是面向关系数据库（RDBMS）JDO 更通用，

提供到任何数据底层的存储功能，比如关系数据库、文件、XML 以及对象数据库（ODBMS）等等，使得应用可移植性更强。

121、内部类可以引用他包含类的成员吗？有没有什么限制？

一个内部类对象可以访问创建它的外部类对象的内容

122、Web Service 名词解释。JSWDL 开发包的介绍。JAXP、JAXM 的解释。SOAP、UDDI、WSDL 解释。

Web Service Web Service 是基于网络的、分布式的模块化组件，它执行特定的任务，遵守具体的技术规范，这些规范使得 Web Service 能与其他兼容的组件进行互操作。

JAXP(Java API for XML Parsing)定义了使用 DOM, SAX, XSLT 的通用的接口。这样在你的程序中你只要使用这些通用的接口，当你需要改变具体的实现时候也不需要修改代码。

JAXM(Java API for XML Messaging)是为 SOAP 通信提供访问方法和传输机制的 API。

WSDL 是一种 XML 格式，用于将网络服务描述为一组端点，这些端点对包含面向文档信息或面向过程信息的信息进行操作。

这种格式首先对操作和信息进行抽象描述，然后将其绑定到具体的网络协议和信息格式上以定义端点。

相关的具体端点即组合成为抽象端点（服务）。

SOAP 即简单对象访问协议(Simple Object Access Protocol)，它是用于交换 XML 编码信息的轻量级协议。

UDDI 的目的是为电子商务建立标准；UDDI 是一套基于 Web 的、分布式的、为 Web Service 提

供的、信息注册中心的实现标准规范，同时也包含一组使企业能将自身提供的 Web Service 注册，以使别的企业能够发现的访问协议的实现标准。

7.9

JAVA 代码查错

1.

```
abstract class Name {  
    private String name;  
    public abstract boolean isStupidName(String name) {}  
}
```

大侠们，这有何错误？

答案: 错。abstract method 必须以分号结尾，且不帶花括号。

2.

```
public class Something {  
    void doSomething () {  
        private String s = "";  
        int l = s.length();  
    }  
}
```

有错吗？

答案: 错。局部变量前不能放置任何访问修饰符 (private, public, 和 protected)。final 可以用来修饰局部变量

(final 如同 abstract 和 strictfp, 都是非访问修饰符, strictfp 只能修饰 class 和 method 而非 variable)。

3.

```
abstract class Something {  
    private abstract String doSomething ();  
}
```

这好像没什么错吧？

答案: 错。abstract 的 methods 不能以 private 修饰。abstract 的 methods 就是让子类 implement(实现)具体细节的，

怎么可以用 private 把 abstract method 封锁起来呢？(同理，abstract method 前不能加 final)。

4.

```
public class Something {  
    public int addOne(final int x) {  
        return ++x;  
    }  
}
```

这个比较明显。

答案: 错。int x 被修饰成 final，意味着 x 不能在 addOne method 中被修改。

5.

```
public class Something {
    public static void main(String[] args) {
        Other o = new Other();
        new Something().addOne(o);
    }
    public void addOne(final Other o) {
        o.i++;
    }
}
class Other {
    public int i;
}
```

和上面的很相似，都是关于 `final` 的问题，这有错吗？

答案: 正确。在 `addOne` method 中，参数 `o` 被修饰成 `final`。如果在 `addOne` method 里我们修改了 `o` 的 `reference`

(比如: `o = new Other()`)，那么如同上例这题也是错的。但这里修改的是 `o` 的 `member variable` (成员变量)，而 `o` 的 `reference` 并没有改变。

6.

```
class Something {
    int i;
    public void doSomething() {
        System.out.println("i = " + i);
    }
}
```

有什么错呢？看不出来啊。

答案: 正确。输出的是 `"i = 0"`。`int i` 属于 `instant variable` (实例变量，或叫成员变量)。`instant variable` 有 `default value`。`int` 的 `default value` 是 `0`。

7.

```
class Something {
    final int i;
    public void doSomething() {
        System.out.println("i = " + i);
    }
}
```

和上面一题只有一个地方不同，就是多了一个 `final`。这难道就错了吗？

答案: 错。`final int i` 是个 `final` 的 `instant variable` (实例变量，或叫成员变量)。`final` 的 `instant variable` 没有 `default value`，

必须在 `constructor` (构造器)结束之前被赋予一个明确的值。可以修改为 `"final int i = 0;"`。

8.

```
public class Something {
```

```

public static void main(String[] args) {
    Something s = new Something();
    System.out.println("s.doSomething() returns " + doSomething());
}
public String doSomething() {
    return "Do something ...";
}
}

```

看上去很完美。

答案: 错。看上去在 `main` 里 `call doSomething` 没有什么问题, 毕竟两个 `methods` 都在同一个 `class` 里。但仔细看, `main` 是 `static` 的。

`static method` 不能直接 `call non-static methods`。可改成 `"System.out.println("s.doSomething() returns " + s.doSomething());"`。同理, `static method` 不能访问 `non- static instant variable`。

9.

此处, `Something` 类的文件名叫 `OtherThing.java`

```

class Something {
    private static void main(String[] something_to_do) {
        System.out.println("Do something ...");
    }
}

```

这个好像很明显。

答案: 正确。从来没有人说过 `Java` 的 `Class` 名字必须和其文件名相同。但 `public class` 的名字必须和文件名相同。

这个是错的, 主方法必须是 `public` 的

10.

```

interface A {
    int x = 0;
}
class B {
    int x = 1;
}
class C extends B implements A {
    public void pX() {
        System.out.println(x);
    }
    public static void main(String[] args) {
        new C().pX();
    }
}

```

答案: 错误。在编译时会发生错误(错误描述不同的 `JVM` 有不同的信息, 意思就是未明确的 `x` 调用,

两个 x 都匹配（就象在同时 import java.util 和 java.sql 两个包时直接声明 Date 一样）。对于父类的变量,可以用 super.x 来明确,而接口的属性默认隐含为 public static final.所以可以通过 A.x 来明确。

11.

```
interface Playable {
    void play();
}
interface Bounceable {
    void play();
}
interface Rollable extends Playable, Bounceable {
    Ball ball = new Ball("PingPang");
}
class Ball implements Rollable {
    private String name;
    public String getName() {
        return name;
    }
    public Ball(String name) {
        this.name = name;
    }
    public void play() {
        ball = new Ball("Football");
        System.out.println(ball.getName());
    }
}
```

这个错误不容易发现。

答案：错。"interface Rollable extends Playable, Bounceable"没有问题。interface 可继承多个 interfaces,

所以这里没错。问题出在 interface Rollable 里的"Ball ball = new Ball("PingPang");"。

任何在 interface 里声明的 interface variable (接口变量, 也可称成员变量), 默认为 public static final. 也就是说 "Ball ball = new Ball("PingPang");"实际上是 "public static final Ball ball = new Ball("PingPang");"。

在 Ball 类的 Play()方法中, "ball = new Ball("Football");"改变了 ball 的 reference,

而这里的 ball 来自 Rollable interface, Rollable interface 里的 ball 是 public static final 的,

final 的 object 是不能被改变 reference 的。因此编译器将在 "ball = new Ball("Football");"这里显示有错。

7.10

Java 基础方面（1）

一、Java 基础方面

1、作用域 public,private,protected,以及不写时的区别

答：区别如下：

作用域	当前类	同一 package	子孙类	其他 package
public	√	√	√	√
protected	√	√	√	×
friendly	√	√	×	×
private	√	×	×	×

不写时默认为 friendly

2、Anonymous Inner Class (匿名内部类) 是否可以 extends(继承)其它类，是否可以 implements(实现)interface(接口)

答：匿名的内部类是没有名字的内部类。可以继承抽象(SDK1.5 没有限制)，但一个内部类可以作为一个接口，由另一个内部类实现

3、Static Nested Class 和 Inner Class 的不同

答：Nested Class（一般是 C++ 的说法），Inner Class（一般是 JAVA 的说法）。Java 内部类与 C++ 嵌套类最大的不同 就在于是否有指向外部的引用上。注：静态内部类（Inner Class）意味着 1 创建一个 static 内部类的对象，不需要一个外部类对象，2 不能从一个 static 内部类的一个对象访问一个外部类对象

4、&和&&的区别

答：&是位运算符，表示按位与运算，&&是逻辑运算符，表示逻辑与（and）

5、Collection 和 Collections 的区别

答：Collection 是集合类的上级接口，继承与他的接口主要有 Set 和 List。Collections 是针对集合类的一个帮助类，他提供一系列静态方法实现对各种集合的搜索、排序、线程安全化等操作

6、什么时候用 assert

答：assertion(断言)在软件开发中是一种常用的调试方式，很多开发语言中都支持这种机制。在实现中，assertion 就是在程序中的一条语句，它对一个 boolean 表达式进行检查，一个正确程序必须保证这个 boolean 表达式的值为 true；如果该值为 false，说明程序已经处于不正确的状态下，系统将给出警告或退出。一般来说，assertion 用于保证程序最基本、关键的正确性。assertion 检查通常在开发和测试时开启。为了提高性能，在软件发布后，assertion 检查通常是关闭的

7、String s = new String("xyz");创建了几个 String Object

答: "xyz" 本身作为字符常量, 在汇编语言中应该作为常量放在数据段, Java 有一个类似数据段的 constant pool 保存这个常量, 在 classloader 加载这个类的时候就把 "xyz" 和这个类的其他一些信息放在 constant pool new String ("xyz") 根据常量 "xyz" 在 heap 上创建 String 对象所以, 一共两个对象

String(String original) Initializes a newly created String object so that it represents the same sequence of characters as the argument; in other words, the newly created string is a copy of the argument string.

8、Math.round(11.5)等於多少? Math.round(-11.5)等於多少

答: Math.round(11.5)=12;Math.round(-11.5)=-11;round 方法返回与参数最接近的长整数, 参数加 1/2 后求其 floor

9、short s1 = 1; s1 = s1 + 1;有什么错? short s1 = 1; s1 += 1;有什么错

答: short s1 = 1; s1 = s1 + 1; (s1+1 运算结果是 int 型, 需要强制转换类型) short s1 = 1; s1 += 1; (可以正确编译)

10、Java 有没有 goto

答: java 中的保留字, 现在没有在 java 中使用

11、数组有没有 length()这个方法? String 有没有 length()这个方法

答: 数组没有 length()这个方法, 有 length 的属性。String 有 length()这个方法

12、Overload 和 Override 的区别。Overloaded 的方法是否可以改变返回值的类型

答: 方法的重写 Overriding 和重载 Overloading 是 Java 多态性的不同表现。重写 Overriding 是父类与子类之间多态性的一种表现, 重载 Overloading 是一个类中多态性的一种表现。如果在子类中定义某方法与其父类有相同的名称和参数, 我们说该方法被重写 (Overriding)。子类的对象使用这个方法时, 将调用子类中的定义, 对它而言, 父类中的定义如同被"屏蔽"了。如果在一个类中定义了多个同名的方法, 它们或有不同的参数个数或有不同的参数类型, 则称为方法的重载(Overloading)。Overloaded 的方法是可以改变返回值的类型

13、Set 里的元素是不能重复的, 那么用什么方法来区分重复与否呢? 是用 == 还是 equals()? 它们有何区别

答: Set 里的元素是不能重复的, 那么用 iterator()方法来区分重复与否。equals()是判断两个 Set 是否相等, equals()和 == 方法决定引用值是否指向同一对象 equals()在类中被覆盖, 为的是当两个分离的对象的内容和类型相配的话, 返回真值

14、给我一个你最常见到的 runtime exception

答：常见的运行时异常有如下这些 ArithmeticException, ArrayStoreException, BufferOverflowException, BufferUnderflowException, CannotRedoException, CannotUndoException, ClassCastException, CMMException, ConcurrentModificationException, DOMException, EmptyStackException, IllegalArgumentException, IllegalMonitorStateException, IllegalPathStateException, IllegalStateException, ImagingOpException, IndexOutOfBoundsException, MissingResourceException, NegativeArraySizeException, NoSuchElementException, NullPointerException, ProfileDataException, ProviderException, RasterFormatException, SecurityException, SystemException, UndeclaredThrowableException, UnmodifiableSetException, UnsupportedOperationException

15、error 和 exception 有什么区别

答：error 表示恢复不是不可能但很困难的情况下的一种严重问题，比如说内存溢出，不可能指望程序能处理这样的情况；exception 表示一种设计或实现问题。也就是说，它表示如果程序运行正常，从不会发生的情况

16、List, Set, Map 是否继承自 Collection 接口

答：List, Set 是，Map 不是

17、abstract class 和 interface 有什么区别

答：声明方法的存在而不去实现它的类被叫做抽象类（abstract class），它用于要创建一个体现某些基本行为的类，并为该类声明方法，但不能在该类中实现该方法的情况。不能创建 abstract 类的实例。然而可以创建一个变量，其类型是一个抽象类，并让它指向具体子类的一个实例。不能有抽象构造函数或抽象静态方法。Abstract 类的子类为它们父类中的所有抽象方法提供实现，否则它们也是抽象类。取而代之，在子类中实现该方法。知道其行为的其它类可以在类中实现这些方法

接口（interface）是抽象类的变体。在接口中，所有方法都是抽象的。多继承性可通过实现这样的接口而获得。接口中的所有方法都是抽象的，没有一个有程序体。接口只可以定义 static final 成员变量。接口的实现与子类相似，除了该实现类不能从接口定义中继承行为。当类实现特殊接口时，它定义（即将程序体给予）所有这种接口的方法。然后，它可以在实现了该接口的类的任何对象上调用接口的方法。由于有抽象类，它允许使用接口名作为引用变量的类型。通常的动态联编将生效。引用可以转换到接口类型或从接口类型转换，instanceof 运算符可以用来决定某对象的类是否实现了接口

18、abstract 的 method 是否可同时是 static, 是否可同时是 native, 是否可同时是 synchronized

答：都不能

19、接口是否可继承接口？抽象类是否可实现 (implements) 接口？抽象类是否可继承实体类

(concrete class)

答：接口可以继承接口。抽象类可以实现(implements)接口，抽象类是否可继承实体类，但前提是实体类必须有明确的构造函数

20、构造器 Constructor 是否可被 override

答：构造器 Constructor 不能被继承，因此不能重写 Overriding，但可以被重载 Overloading

21、是否可以继承 String 类

答：String 类是 final 类故不可以继承

22、try {} 里有一个 return 语句，那么紧跟在这个 try 后的 finally {} 里的 code 会不会被执行，什么时候被执行，在 return 前还是后

答：会执行，在 return 前执行

23、用最有效率的方法算出 2 乘以 8 等於几

答：2 << 3

24、两个对象值相同(x.equals(y) == true)，但却可有不同的 hash code，这句话对不对

答：不对，有相同的 hash code

25、当一个对象被当作参数传递到一个方法后，此方法可改变这个对象的属性，并可返回变化后的结果，那么这里到底是值传递还是引用传递

答：是值传递。Java 编程语言只有值传递参数。当一个对象实例作为一个参数被传递到方法中时，参数的值就是对该对象的引用。对象的内容可以在被调用的方法中改变，但对象的引用是永远不会改变的

26、switch 是否能作用在 byte 上，是否能作用在 long 上，是否能作用在 String 上

答：switch (expr1) 中，expr1 是一个整数表达式。因此传递给 switch 和 case 语句的参数应该是 int、short、char 或者 byte。long、string 都不能作用于 switch

27、ArrayList 和 Vector 的区别,HashMap 和 Hashtable 的区别

答：就 ArrayList 与 Vector 主要从二方面来说：

1、同步性:Vector 是线程安全的，也就是说同步的，而 ArrayList 是线程不安全的，不是同步的

2、数据增长:当需要增长时,Vector 默认增长为原来一倍, 而 ArrayList 却是原来的一半

就 HashMap 与 Hashtable 主要从三方面来说:

1、历史原因:Hashtable 是基于陈旧的 Dictionary 类的, HashMap 是 Java 1.2 引进的 Map 接口的一个实现

2、同步性:Hashtable 是线程安全的, 也就是说是同步的, 而 HashMap 是线程程序不安全的, 不是同步的

3、值: 只有 HashMap 可以让你将空值作为一个表的条目的 key 或 value

28、char 型变量中能不能存贮一个中文汉字?为什么?

答: 是能够定义成为一个中文的, 因为 java 中以 unicode 编码, 一个 char 占 16 个字节, 所以放一个中文是没问题的

如果用 gbk 的本地编码的话可以.如果用 utf-8 的话, 可能不行。

29、GC 是什么? 为什么要有 GC

答: GC 是垃圾收集的意思 (Garbage Collection), 内存处理是编程人员容易出现问题的地方, 忘记或者错误的内存回收会导致程序或系统的不稳定甚至崩溃, Java 提供的 GC 功能可以自动监测对象是否超过作用域从而达到自动回收内存的目的, Java 语言没有提供释放已分配内存的显示操作方 法。

30、float 型 float f=3.4 是否正确?

答:不正确。精度不准确,应该用强制类型转换, 如下所示: float f=(float)3.4; float f = 3.4f;

31、介绍 JAVA 中的 Collection Framework(包括如何写自己的数据结构)?

答: Collection Framework 如下:

```
Collection
├── List
│   ├── LinkedList
│   ├── ArrayList
│   └── Vector
│       └── Stack
└── Set
Map
├── Hashtable
└── HashMap
```

↳WeakHashMap

Collection 是最基本的集合接口，一个 Collection 代表一组 Object，即 Collection 的元素 (Elements)

Map 提供 key 到 value 的映射

32、抽象类与接口？

答：抽象类与接口都用于抽象，但是抽象类(JAVA 中)可以有自己的部分实现，而接口则完全是一个标识(同时有多实现的功能)。

JAVA 类实现序列化的方法是实现 java.io.Serializable 接口

Collection 框架中实现比较要实现 Comparable 接口和 Comparator 接口

33、STRING 与 STRINGBUFFER 的区别。

答：STRING 的长度是不可变的，STRINGBUFFER 的长度是可变的。如果你对字符串中的内容经常进行操作，特别是内容要修改时，那么使用 StringBuffer，如果最后需要 String，那么使用 StringBuffer 的 toString()方法

34、谈谈 final, finally, finalize 的区别

答：final—修饰符（关键字）如果一个类被声明为 final，意味着它不能再派生出新的子类，不能作为父类被继承。因此一个类不能既被声明为 abstract 的，又被声明为 final 的。将变量或方法声明为 final，可以保证它们在使用中不被改变。被声明为 final 的变量必须在声明时给一定初值，而在以后的引用中只能读取，不可修改。被声明为 final 的方法也同样只能使用，不能重载

finally—再异常处理时提供 finally 块来执行任何清除操作。如果抛出一个异常，那么相匹配的 catch 子句就会执行，然后控制就会进入 finally 块（如果有的话）

finalize—方法名。Java 技术允许使用 finalize() 方法在垃圾收集器将对象从内存中清除出去之前做必要的清理工作。这个方法是由垃圾收集器在确定这个对象没有被引用时对这个对象调用的。它是在 Object 类中定义的，因此所有的类都继承了它。子类覆盖 finalize() 方法以整理系统资源或者执行其他清理工作。finalize() 方法是在垃圾收集器删除对象之前对这个对象调用的

35、面向对象的特征有哪些方面

答：主要有以下四方面：

1.抽象：抽象就是忽略一个主题中与当前目标无关的那些方面，以便更充分地注意与当前目标有关的方面。抽象并不打算了解全部问题，而只是选择其中的一部分，暂时不用部分细节。抽象包括两个方面，一是过程抽象，二是数据抽象。

2.继承：继承是一种联结类的层次模型，并且允许和鼓励类的重用，它提供了一种明确表述共性的方法。对象的一个新类可以从现有的类中派生，这个过程称为 类继承。新类继承了原始类的特性，新类称为原始类的派生类（子类），而原始类称为新类的基类（父类）。派生类可以从它的基类那里继承方法和实例变量，并且 类可以修改或增加新的方法使之更适合特殊的需要。

3.封装：封装是把过程和数据包围起来，对数据的访问只能通过已定义的界面。面向对象计算始于这个基本概念，即现实世界可以被描绘成一系列完全自治、封装的对象，这些对象通过一个受保护的接口访问其他对象。

4.多态性：多态性是指允许不同类的对象对同一消息作出响应。多态性包括参数化多态性和包含多态性。多态性语言具有灵活、抽象、行为共享、代码共享的优势，很好的解决了应用程序函数同名问题。

36、String 是最基本的数据类型吗

答：不是，基本数据类型包括 byte、int、char、long、float、double、boolean 和 short。
java.lang.String 类是 final 类型的，因此不可以继承这个类、不能修改这个类。为了提高效率节省空间，我们应该用 StringBuffer 类

37、int 和 Integer 有什么区别

答：Java 提供两种不同的类型：引用类型和原始类型（或内置类型）。

Int 是 java 的原始数据类型，Integer 是 java 为 int 提供的 wrapper。Java 为每个原始类型提供了封装类。

原始类型封装类：

boolean : Boolean

char : Character

byte : Byte

short : Short

int : Integer

long : Long

float : Float

double : Double

引用类型和原始类型的行为完全不同，并且它们具有不同的语义。引用类型和原始类型具有不同的特征和用法，它们包括：大小和速度问题，这种类型以哪种类型的数据结构存储，当引用类型和原始类型用作某个类的实例数据时所指定的缺省值。对象引用实例变量的缺省值为 null，而原始类型实例变量的缺省值与它们的类型有关

38、运行时异常与一般异常有何异同

答：异常表示程序运行过程中可能出现的非正常状态，运行时异常表示虚拟机的通常操作中

可能遇到的异常，是一种常见运行错误。java 编译器要求方法必须声明抛出可能发生的非运行时异常，但是并不要求必须声明抛出未被捕获的运行时异常。

39、说出 ArrayList,Vector,LinkedList 的存储性能和特性

答：ArrayList 和 Vector 都是使用数组方式存储数据，此数组元素数大于实际存储的数据以便增加和插入元素，它们都允许直接按序号索引元素，但是插入元素要涉及数组元素移动等内存操作，所以索引数据快而插入数据慢，Vector 由于使用了 synchronized 方法（线程安全），通常性能上较 ArrayList 差，而 LinkedList 使用双向链表实现存储，按序号索引数据需要进行前向或后向遍历，但是插入数据时只需要记录本项的前后项即可，所以插入速度较快。

40、HashMap 和 Hashtable 的区别

答：HashMap 是 Hashtable 的轻量级实现（非线程安全的实现），他们都完成了 Map 接口，主要区别在于 HashMap 允许空（null）键值（key），由于非线程安全，效率上可能高于 Hashtable。

HashMap 允许将 null 作为一个 entry 的 key 或者 value，而 Hashtable 不允许。

HashMap 把 Hashtable 的 contains 方法去掉了，改成 containsvalue 和 containsKey。因为 contains 方法容易让人引起误解。

Hashtable 继承自 Dictionary 类，而 HashMap 是 Java1.2 引进的 Map interface 的一个实现。

最大的不同是，Hashtable 的方法是 Synchronize 的，而 HashMap 不是，在多个线程访问 Hashtable 时，不需要自己为它的方法实现同步，而 HashMap 就必须为之提供外同步。

Hashtable 和 HashMap 采用的 hash/rehash 算法都大概一样，所以性能不会有很大的差异。

41、heap 和 stack 有什么区别

答：栈是一种线形集合，其添加和删除元素的操作应在同一段完成。栈按照后进先出的方式进行处理。堆是栈的一个组成元素

42、Java 的接口和 C++的虚类的相同和不同处

答：由于 Java 不支持多继承，而有可能某个类或对象要使用分别在几个类或对象里面的方法或属性，现有的单继承机制就不能满足要求。与继承相比，接口有更高的灵活性，因为接口中没有任何实现代码。当一个类实现了接口以后，该类要实现接口里面所有的方法和属性，并且接口里面的属性在默认状态下面都是 public static,所有方法默认情况下是 public.一个类可以实现多个接口。

43、Java 中的异常处理机制的简单原理和应用

答：当 JAVA 程序违反了 JAVA 的语义规则时，JAVA 虚拟机就会将发生的错误表示为一个

异常。违反语义规则包括 2 种情况。一种是 JAVA 类库内置的 语义检查。例如数组下标越界,会引发 `IndexOutOfBoundsException`;访问 `null` 的对象时会引发 `NullPointerException`。另一种情况就是 JAVA 允许程序员扩展这种语义检查,程序员可以创建自己的异常,并自由选择何时用 `throw` 关键字引发异常。所有的异常都是 `java.lang.Throwable` 的子类。

43、垃圾回收的优点和原理。并考虑 2 种回收机制

答: Java 语言中一个显著的特点就是引入了垃圾回收机制,使 c++程序员最头疼的内存管理的问题迎刃而解,它使得 Java 程序员在编写程序的时候不再 需要考虑内存管理。垃圾回收可以有效地防止内存泄露,有效的使用可以使用的内存。垃圾回收器通常是作为一个单独的低级别的线程运行,不可预知的情况下对内存堆中已经死亡的或者长时间没有使用的对象进行清楚和回收,程序员不能实时的调用垃圾回收器对某个对象或所有对象进行垃圾回收。回收机制有分代复制垃圾回收和标记垃圾回收,增量垃圾回收。

44、你所知道的集合类都有哪些? 主要方法?

答: 最常用的集合类是 `List` 和 `Map`。`List` 的具体实现包括 `ArrayList` 和 `Vector`, 它们是可变大小的列表,比较适合构建、存储和操作任何类型对象的元素列表。`List` 适用于按数值索引访问元素的情形。

`Map` 提供了一个更通用的元素存储方法。`Map` 集合类用于存储元素对 (称作 "键"和"值"), 其中每个键映射到一个值。

45、描述一下 JVM 加载 class 文件的原理机制?

答: JVM 中类的装载是由 `ClassLoader` 和它的子类来实现的,Java `ClassLoader` 是一个重要的 Java 运行时系统组件。它负责在运行时查找和装入类文件的类。

46、排序都有哪几种方法? 请列举

答: 排序的方法有: 插入排序 (直接插入排序、希尔排序), 交换排序 (冒泡排序、快速排序), 选择排序 (直接选择排序、堆排序), 归并排序, 分配排序 (箱排序、基数排序) 快速排序的伪代码。

使用快速排序方法对 `a[0 : n-1]` 排序:

从 `a[0 : n-1]` 中选择一个元素作为 `middle`, 该元素为支点, 把余下的元素分割为两段 `left` 和 `right`, 使得 `left` 中的元素都小于等于支点, 而 `right` 中的元素都大于等于支点。

递归地使用快速排序方法对 `left` 进行排序

递归地使用快速排序方法对 `right` 进行排序

所得结果为 `left + middle + right`

47、JAVA 语言如何进行异常处理，关键字：throws,throw,try,catch,finally 分别代表什么意义？在 try 块中可以抛出异常吗？

答：Java 通过面向对象的方法进行异常处理，把各种不同的异常进行分类，并提供了良好的接口。在 Java 中，每个异常都是一个对象，它是 Throwable 类或其它子类的实例。当一个方法出现异常后便抛出一个异常对象，该对象中包含有异常信息，调用这个方法可以捕获到这个异常并进行处理。Java 的异常处理是通过 5 个关键词来实现的：try、catch、throw、throws 和 finally。一般情况下是用 try 来执行一段程序，如果出现异常，系统会抛出（throws）一个异常，这时候你可以通过它的类型来捕捉（catch）它，或最后（finally）由缺省处理器来处理。

用 try 来指定一块预防所有"异常"的程序。紧跟在 try 程序后面，应包含一个 catch 子句来指定你想要捕捉的"异常"的类型。

throw 语句用来明确地抛出一个"异常"。

throws 用来标明一个成员函数可能抛出的各种"异常"。

Finally 为确保一段代码不管发生什么"异常"都被执行一段代码。

可以在一个成员函数调用的外面写一个 try 语句，在这个成员函数内部写另一个 try 语句保护其他代码。每当遇到一个 try 语句，"异常"的框架就放到堆栈上面，直到所有的 try 语句都完成。如果下一级的 try 语句没有对某种"异常"进行处理，堆栈就会展开，直到遇到有处理这种"异常"的 try 语句。

48、一个".java"源文件中是否可以包括多个类（不是内部类）？有什么限制？

答：可以。必须只有一个类名与文件名相同。

49、java 中有几种类型的流？JDK 为每种类型的流提供了一些抽象类以供继承，请说出他们分别是哪些类？

答：字节流，字符流。字节流继承于 InputStream OutputStream，字符流继承于 InputStreamReader OutputStreamWriter。在 java.io 包中还有许多其他的流，主要是为了提高性能和使用方便。

50、java 中会存在内存泄漏吗，请简单描述。

答：会。自己实现堆载的数据结构时有可能出现内存泄露，可参看 effective java.

7.11

Java 基础方面（2）

51、java 中实现多态的机制是什么？

答：方法的重写 **Overriding** 和重载 **Overloading** 是 Java 多态性的不同表现。重写 **Overriding** 是父类与子类之间多态性的一种表现，重载 **Overloading** 是一个类中多态性的一种表现。

52、垃圾回收器的基本原理是什么？垃圾回收器可以马上回收内存吗？有什么办法主动通知虚拟机进行垃圾回收

答：对于 GC 来说，当程序员创建对象时，GC 就开始监控这个对象的地址、大小以及使用情况。通常，GC 采用有向图的方式记录和管理堆(heap)中的所有对象。通过这种方式确定哪些对象是"可达的"，哪些对象是"不可达的"。当 GC 确定一些对象为"不可达"时，GC 就有责任回收这些内存空间。可以。程序员可以手动执行 **System.gc()**，通知 GC 运行，但是 Java 语言规范并不保证 GC 一定会执行。

53、静态变量和实例变量的区别？

54、什么是 java 序列化，如何实现 java 序列化？

答：序列化就是一种用来处理对象流的机制，所谓对象流也就是将对象的内容进行流化。可以对流化后的对象进行读写操作，也可将流化后的对象传输于网络之间。序列化是为了解决在对对象流进行读写操作时所引发的问题。

序列化的实现：将需要被序列化的类实现 **Serializable** 接口，该接口没有需要实现的方法，**implements Serializable** 只是 为了标注该对象是可被序列化的，然后使用一个输出流(如：**FileOutputStream**)来构造一个 **ObjectOutputStream**(对象流) 对象，接着，使用 **ObjectOutputStream** 对象的 **writeObject(Object obj)**方法就可以将参数为 **obj** 的对象写出(即保存其状态)，要恢复的话则用输入流。

55、是否可以从一个 **static** 方法内部发出对非 **static** 方法的调用？

答：不可以,如果其中包含对象的 **method()**；不能保证对象初始化。

56、写 **clone()**方法时，通常都有一行代码，是什么？

答：**Clone** 有缺省行为，**super.clone()**；他负责产生正确大小的空间，并逐位复制。

57、在 JAVA 中，如何跳出当前的多重嵌套循环？

答：用 **break**; **return** 方法。

58、**List**、**Map**、**Set** 三个接口，存取元素时，各有什么特点？

答：**List** 以特定次序来持有元素，可有重复元素。**Set** 无法拥有重复元素,内部排序。**Map** 保存 **key-value** 值，**value** 可多值。

59、说出一些常用的类，包，接口，请各举 5 个

答：常用的类：BufferedReader BufferedWriter FileReader FileWriter String Integer

常用的包：java.lang java.awt java.io java.util java.sql

常用的接口：Remote List Map Runnable Serializable ActionListener

二、线程编程方面

60、java 中有几种方法可以实现一个线程？用什么关键字修饰同步方法？stop()和 suspend()方法为何不推荐使用？

答：有两种实现方法，分别是继承 Thread 类与实现 Runnable 接口

用 synchronized 关键字修饰同步方法

反对使用 stop()，是因为它不安全。它会解除由线程获取的所有锁定，而且如果对象处于一种不连贯状态，那么其他线程能在那种状态下检查和修改它们。结果很难检查出真正的问题所在。suspend()方法容易发生死锁。调用 suspend()的时候，目标线程会停下来，但却仍然持有在这之前获得的锁定。此时，其他任何线程都不能访问锁定的资源，除非被"挂起"的线程恢复运行。对任何线程来说，如果它们想恢复目标线程，同时又试图使用任何一个锁定的资源，就会造成死锁。所以不应该使用 suspend()，而应在自己的 Thread 类中置入一个标志，指出线程应该活动还是挂起。若标志指出线程应该挂起，便用 wait()命其进入等待状态。若标志指出线程应当恢复，则用一个 notify()重新启动线程。

61、sleep() 和 wait() 有什么区别？

答：sleep 是线程类（Thread）的方法，导致此线程暂停执行指定时间，给执行机会给其他线程，但是监控状态依然保持，到时后会自动恢复。调用 sleep 不会释放对象锁。

wait 是 Object 类的方法，对此对象调用 wait 方法导致本线程放弃对象锁，进入等待此对象的等待锁定池，只有针对此对象发出 notify 方法（或 notifyAll）后本线程才进入对象锁定池准备获得对象锁进入运行状态。

62、同步和异步有何异同，在什么情况下分别使用他们？举例说明。

答：如果数据将在线程间共享。例如正在写的数据以后可能被另一个线程读到，或者正在读的数据可能已经被另一个线程写过了，那么这些数据就是共享数据，必须进行同步存取。

当应用程序在对象上调用了需要花费很长时间来执行的方法，并且不希望让程序等待方法的返回时，就应该使用异步编程，在很多情况下采用异步途径往往更有效率。

63、启动一个线程是用 `run()` 还是 `start()`?

答：启动一个线程是调用 `start()` 方法，使线程所代表的虚拟处理机处于可运行状态，这意味着它可以由 JVM 调度并执行。这并不意味着线程就会立即运行。`run()` 方法可以产生必须退出的标志来停止一个线程。

64、当一个线程进入一个对象的一个 `synchronized` 方法后，其它线程是否可进入此对象的其它方法?

答：不能，一个对象的一个 `synchronized` 方法只能由一个线程访问。

65、请说出你所知道的线程同步的方法。

答：`wait()`: 使一个线程处于等待状态，并且释放所持有的对象的 `lock`。

`sleep()`: 使一个正在运行的线程处于睡眠状态，是一个静态方法，调用此方法要捕捉 `InterruptedException` 异常。

`notify()`: 唤醒一个处于等待状态的线程，注意的是在调用此方法的时候，并不能确切的唤醒某一个等待状态的线程，而是由 JVM 确定唤醒哪个线程，而且不是按优先级。

`Allnotify()`: 唤醒所有处于等待状态的线程，注意并不是给所有唤醒线程一个对象的锁，而是让它们竞争。

66、多线程有几种实现方法,都是什么?同步有几种实现方法,都是什么?

答：多线程有两种实现方法，分别是继承 `Thread` 类与实现 `Runnable` 接口
同步的实现方面有两种，分别是 `synchronized`, `wait` 与 `notify`

67、线程的基本概念、线程的基本状态以及状态之间的关系

答：线程指在程序执行过程中，能够执行程序代码的一个执行单位，每个程序至少都有一个线程，也就是程序本身。

Java 中的线程有四种状态分别是：运行、就绪、挂起、结束

68、简述 `synchronized` 和 `java.util.concurrent.locks.Lock` 的异同?

答：主要相同点：`Lock` 能完成 `synchronized` 所实现的所有功能

主要不同点：`Lock` 有比 `synchronized` 更精确的线程语义和更好的性能。`synchronized` 会自动释放锁，而 `Lock` 一定要求程序员手工释放，并且必须在 `finally` 从句中释放。

三、Jsp 方面

69、forward 和 redirect 的区别

答: forward 是服务器请求资源, 服务器直接访问目标地址的 URL, 把那个 URL 的响应内容读取过来, 然后把这些内容再发给浏览器, 浏览器根本不知道服务器发送的内容是从哪儿来的, 所以它的地址栏中还是原来的地址。

redirect 就是服务端根据逻辑, 发送一个状态码, 告诉浏览器重新去请求那个地址, 一般来说浏览器会用刚才请求的所有参数重新请求, 所以 session, request 参数都可以获取。

70、jsp 有哪些内置对象? 作用分别是什么?

答: JSP 共有以下 9 种基本内置组件 (可与 ASP 的 6 种内部组件相对应):

request 用户端请求, 此请求会包含来自 GET/POST 请求的参数

response 网页传回用户端的回应

pageContext 网页的属性是在这里管理

session 与请求有关的会话期

application servlet 正在执行的内容

out 用来传送回应的输出

config servlet 的构架部件

page JSP 网页本身

exception 针对错误网页, 未捕捉的例外

71、jsp 有哪些动作? 作用分别是什么?

答: JSP 共有以下 6 种基本动作:

jsp:include: 在页面被请求的时候引入一个文件。

jsp:useBean: 寻找或者实例化一个 JavaBean。

jsp:setProperty: 设置 JavaBean 的属性。

jsp:getProperty: 输出某个 JavaBean 的属性。

jsp:forward: 把请求转到一个新的页面。

jsp:plugin: 根据浏览器类型为 Java 插件生成 OBJECT 或 EMBED 标记

72、JSP 中动态 INCLUDE 与静态 INCLUDE 的区别?

答: 动态 INCLUDE 用 jsp:include 动作实现。<jsp:include page="included.jsp" flush="true" /> 它总是会检查所含文件中的变化, 适合用于包含动态页面, 并且可以带参数

静态 INCLUDE 用 include 伪码实现, 定不会检查所含文件的变化, 适用于包含静态页面: <%@include file="included.htm" %>

73、两种跳转方式分别是什么? 有什么区别?

答：有两种，分别为：

```
<jsp:include page="included.jsp" flush="true">
<jsp:forward page="nextpage.jsp"/>
```

前者页面不会转向 `include` 所指的页面，只是显示该页的结果，主页面还是原来的页面。执行完后还会回来，相当于函数调用。并且可以带参数。后者完全转向新页面，不会再回来。相当于 `goto` 语句。

74、JSP 的内置对象及方法。

答：`request` 表示 `HttpServletRequest` 对象。它包含了有关浏览器请求的信息，并且提供了几个用于获取 `cookie`, `header`, 和 `session` 数据的有用的方法。

`response` 表示 `HttpServletResponse` 对象，并提供了几个用于设置送回浏览器的响应的方法（如 `cookies`, 头信息等）

`out` 对象是 `javax.jsp.JspWriter` 的一个实例，并提供了几个方法使你能用于向浏览器回送输出结果。

`pageContext` 表示一个 `javax.servlet.jsp.PageContext` 对象。它是用于方便存取各种范围的名字空间、`servlet` 相关的对象的 API，并且包装了通用的 `servlet` 相关功能的方法。

`session` 表示一个请求的 `javax.servlet.http.HttpSession` 对象。`Session` 可以存贮用户的状态信息

`applicaton` 表示一个 `javax.servle.ServletContext` 对象。这有助于查找有关 `servlet` 引擎和 `servlet` 环境的信息

`config` 表示一个 `javax.servlet.ServletConfig` 对象。该对象用于存取 `servlet` 实例的初始化参数。

`page` 表示从该页面产生的一个 `servlet` 实例

四、Servlet 方面

75、说一说 Servlet 的生命周期？

答：`servlet` 有良好的生存期的定义，包括加载和实例化、初始化、处理请求以及服务结束。这个生存期由 `javax.servlet.Servlet` 接口的 `init`, `service` 和 `destroy` 方法表达。`Servlet` 被服务器实例化后，容器运行其 `init` 方法，请求到达时运行其 `service` 方法，`service` 方法自动派遣运行与请求对应的 `doXXX` 方法（`doGet`, `doPost`）等，当服务器决定将实例销毁的时候调用其 `destroy` 方法。

与 `cgi` 的区别在于 `servlet` 处于服务器进程中，它通过多线程方式运行其 `service` 方法，一个实例可以服务于多个请求，并且其实例一般不会销毁，而 `CGI` 对每个请求都产生新的进程，服

务完成后就销毁，所以效率上低于 servlet。

76、JAVA SERVLET API 中 forward() 与 redirect()的区别？

答:前者仅是容器中控制权的转向，在客户端浏览器地址栏中不会显示出转向后的地址；后者则是完全的跳转，浏览器将会得到跳转的地址，并重新发送请求链接。这样，从浏览器的地址栏中可以看到跳转后的链接地址。所以，前者更加高效，在前者可以满足需要时，尽量使用 forward()方法，并且，这样也有助于隐藏实际的链接。在有些情况下，比如，需要跳转到一个其它服务器上的资源，则必须使用 sendRedirect()方法。

77、Servlet 的基本架构

答：

```
public class ServletName extends HttpServlet {
    public void doPost(HttpServletRequest request, HttpServletResponse response) throws
        ServletException, IOException {
    }
    public void doGet(HttpServletRequest request, HttpServletResponse response) throws
        ServletException, IOException {
    }
}
```

78、什么情况下调用 doGet() 和 doPost()？

答：Jsp 页面中的 form 标签里的 method 属性为 get 时调用 doGet()，为 post 时调用 doPost()。

79、servlet 的生命周期

答：web 容器加载 servlet，生命周期开始。通过调用 servlet 的 init()方法进行 servlet 的初始化。通过调用 service() 方法实现，根据请求的不同调用不同的 do***()方法。结束服务，web 容器调用 servlet 的 destroy()方法。

80、如何实现 servlet 的单线程模式

答：<%@ page isThreadSafe="false"%>

81、页面间对象传递的方法

答：request, session, application, cookie 等

82、JSP 和 Servlet 有哪些相同点和不同点，他们之间的联系是什么？

答：JSP 是 Servlet 技术的扩展，本质上是 Servlet 的简易方式，更强调应用的外表表达。JSP

编译后是"类 servlet"。Servlet 和 JSP 最主要的不同点在于,Servlet 的应用逻辑是在 Java 文件中,并且完全从表示层中的 HTML 里分离开来。而 JSP 的情况是 Java 和 HTML 可以组合成一个扩展名为.jsp 的文件。JSP 侧重于视图,Servlet 主要用于控制逻辑。

83、四种会话跟踪技术

答: 会话作用域 ServletsJSP 页面描述

page 否是代表与一个页面相关的对象和属性。一个页面由一个编译好的 Java servlet 类(可以带有任何的 include 指令,但是没有 include 动作)表示。这既包括 servlet 又包括被编译成 servlet 的 JSP 页面

request 是代表与 Web 客户机发出的一个请求相关的对象和属性。一个请求可能跨越多个页面,涉及多个 Web 组件(由于 forward 指令和 include 动作的关系)

session 是代表与用于某个 Web 客户机的一个用户体验相关的对象和属性。一个 Web 会话可以也经常跨越多个客户机请求

application 是代表与整个 Web 应用程序相关的对象和属性。这实质上是跨越整个 Web 应用程序,包括多个页面、请求和会话的一个全局作用域

84、Request 对象的主要方法

答:

setAttribute(String name,Object): 设置名字为 name 的 request 的参数值

getAttribute(String name): 返回由 name 指定的属性值

getAttributeNames(): 返回 request 对象所有属性的名字集合,结果是一个枚举的实例

getCookies(): 返回客户端的所有 Cookie 对象,结果是一个 Cookie 数组

getCharacterEncoding(): 返回请求中的字符编码方式

getContentTypeLength(): 返回请求的 Body 的长度

getHeader(String name): 获得 HTTP 协议定义的文件头信息

getHeaders(String name): 返回指定名字的 request Header 的所有值,结果是一个枚举的实例

getHeaderNames(): 返回所以 request Header 的名字,结果是一个枚举的实例

getInputStream(): 返回请求的输入流,用于获得请求中的数据

getMethod(): 获得客户端向服务器端传送数据的方法

getParameter(String name): 获得客户端传送给服务器端的有 name 指定的参数值

getParameterNames(): 获得客户端传送给服务器端的所有参数的名字,结果是一个枚举的实例

例

getParameterValues(String name): 获得有 name 指定的参数的所有值

getProtocol(): 获取客户端向服务器端传送数据所依据的协议名称

getQueryString(): 获得查询字符串

getRequestURI(): 获取发出请求字符串的客户端地址

getRemoteAddr(): 获取客户端的 IP 地址

getRemoteHost(): 获取客户端的名字
getSession([Boolean create]): 返回和请求相关 Session
getServerName(): 获取服务器的名字
getServletPath(): 获取客户端所请求的脚本文件的路径
getServerPort(): 获取服务器的端口号
removeAttribute(String name): 删除请求中的一个属性

85、我们在 web 应用开发过程中经常遇到输出某种编码的字符，如 iso8859-1 等，如何输出一个某种编码的字符串？

答：

```
Public String translate (String str) {  
    String tempStr = "";  
    try {  
        tempStr = new String(str.getBytes("ISO-8859-1"), "GBK");  
        tempStr = tempStr.trim();  
    }  
    catch (Exception e) {  
        System.err.println(e.getMessage());  
    }  
    return tempStr;  
}
```

86、Servlet 执行时一般实现哪几个方法？

答：

```
public void init(ServletConfig config)  
public ServletConfig getServletConfig()  
public String getServletInfo()  
public void service(ServletRequest request,ServletResponse response)  
public void destroy()
```

五、Jdbc、Jdo 方面

87、Class.forName 的作用?为什么要用？

答：调用该访问返回一个以字符串指定类名的类的对象。

88、Jdo 是什么？

答：JDO 是 Java 对象持久化的新的规范，为 java data object 的简称,也是一个用于存取某种数据仓库中的对象的标准 API。JDO 提供了透明的对象存储，因此对开发人员来说，存储数

据对象完全不需要额外的代码（如 JDBC API 的使用）。这些繁琐的例行工作已经转移到 JDO 产品提供商身上，使开发人员解脱出来，从而集中时间和精力在业务逻辑上。另外，JDO 很灵活，因为它可以在任何数据底层上运行。JDBC 只是面向关系数据库（RDBMS）JDO 更通用，提供到任何数据底层的存储功能，比如关系数据库、文件、XML 以及对象数据库（ODBMS）等等，使得应用可移植性更强。

89、说出数据连接池的工作机制是什么？

答：J2EE 服务器启动时会建立一定数量的池连接，并一直维持不少于此数目的池连接。客户端程序需要连接时，池驱动程序会返回一个未使用的池连接并将其标记为忙。如果当前没有空闲连接，池驱动程序就新建一定数量的连接，新建连接的数量有配置参数决定。当使用的池连接调用完成后，池驱动程序将此连接标记为空闲，其他调用就可以使用这个连接。

90、Jdo 是什么？

答：JDO 是 Java 对象持久化的新的规范，为 java data object 的简称,也是一个用于存取某种数据仓库中的对象的标准 API。JDO 提供了透明的对象存储，因此对开发人员来说，存储数据对象完全不需要额外的代码（如 JDBC API 的使用）。这些繁琐的例行工作已经转移到 JDO 产品提供商身上，使开发人员解脱出来，从而集中时间和精力在业务逻辑上。另外，JDO 很灵活，因为它可以在任何数据底层上运行。JDBC 只是面向关系数据库（RDBMS）JDO 更通用，提供到任何数据底层的存储功能，比如关系数据库、文件、XML 以及对象数据库（ODBMS）等等，使得应用可移植性更强。

六、XML 方面

91、xml 有哪些解析技术？区别是什么？

答：有 DOM,SAX,STAX 等

DOM:处理大型文件时其性能下降的非常厉害。这个问题是由 DOM 的树结构所造成的，这种结构占用的内存较多，而且 DOM 必须在解析文件之前把整个文档装入内存,适合对 XML 的随机访问 SAX:不现于 DOM,SAX 是事件驱动型的 XML 解析方式。它顺序读取 XML 文件，不需要一次全部装载整个文件。当遇到像文件开头，文档结束，或者标签开头与标签结束时，它会触发一个事件，用户通过在其回调事件中写入处理代码来处理 XML 文件，适合对 XML 的顺序访问

STAX:Streaming API for XML (StAX)

92、你在项目中用到了 xml 技术的哪些方面？如何实现的？

答：用到了数据存贮，信息配置两方面。在做数据交换平台时，将不同数据源的数据组装成 XML 文件，然后将 XML 文件压缩打包加密后通过网络传送给接收者，接收解密与解压缩后再同 XML 文件中还原相关信息进行处理。在做软件配置时，利用 XML 可以很方便的进行，软件的各种配置参数都存贮在 XML 文件中。

93、XML 文档定义有几种形式？它们之间有何本质区别？解析 XML 文档有哪几种方式？

答：XML 文档定义有两种形式：dtd, schema。

本质区别:schema 本身是 xml 的, 可以被 XML 解析器解析(这也是从 DTD 上发展 schema 的根本目的)。

解析 XML 文档有 DOM,SAX,STAX 等方式:

DOM:处理大型文件时其性能下降的非常厉害。这个问题是由 DOM 的树结构所造成的, 这种结构占用的内存较多, 而且 DOM 必须在解析文件之前把整个文档装入内存,适合对 XML 的随机访问

SAX:不同于 DOM,SAX 是事件驱动型的 XML 解析方式。它顺序读取 XML 文件, 不需要一次全部装载整个文件。当遇到像文件开头, 文档结束, 或者标签开头与标签结束时, 它会触发一个事件, 用户通过在其回调事件中写入处理代码来处理 XML 文件, 适合对 XML 的顺序访问

STAX:Streaming API for XML (StAX)

七、EJB 方面

94、EJB2.0 有哪些内容?分别用在什么场合? EJB2.0 和 EJB1.1 的区别?

答: 规范内容包括 Bean 提供者, 应用程序装配者, EJB 容器, EJB 配置工具, EJB 服务提供者, 系统管理员。这里面, EJB 容器是 EJB 之所以能够运行的核心。EJB 容器管理着 EJB 的创建, 撤消, 激活, 去活, 与数据库的连接等等重要的核心工作。JSP,Servlet,EJB,JNDI,JDBC,JMS.....

95、EJB 与 JAVA BEAN 的区别?

答: Java Bean 是可复用的组件, 对 Java Bean 并没有严格的规范, 理论上讲, 任何一个 Java 类都可以是一个 Bean。但通常情况下, 由于 Java Bean 是被容器所创建(如 Tomcat)的, 所以 Java Bean 应具有一个无参的构造器, 另外, 通常 Java Bean 还要实现 Serializable 接口用于实现 Bean 的持久性。Java Bean 实际上相当于微软 COM 模型中的本地进程内 COM 组件, 它是不能被跨进程访问的。Enterprise Java Bean 相当于 DCOM, 即分布式组件。它是基于 Java 的远程方法调用(RMI)技术的, 所以 EJB 可以被远程访问(跨进程、跨计算机)。但 EJB 必须被布署在诸如 Webspere、WebLogic 这样的容器中, EJB 客户从不直接访问真正的 EJB 组件, 而是通过其容器访问。EJB 容器是 EJB 组件的代理, EJB 组件由容器所创建和管理。客户通过容器来访问真正的 EJB 组件。

96、EJB 是基于哪些技术实现的? 并说出 SessionBean 和 EntityBean 的区别, StatefulBean 和 StatelessBean 的区别。

答：EJB 包括 Session Bean、Entity Bean、Message Driven Bean，基于 JNDI、RMI、JAT 等技术实现。

SessionBean 在 J2EE 应用程序中被用来完成一些服务器端的业务操作，例如访问数据库、调用其他 EJB 组件。EntityBean 被用来代表应用系统中用到的数据。

对于客户机，SessionBean 是一种非持久性对象，它实现某些在服务器上运行的业务逻辑。

对于客户机，EntityBean 是一种持久性对象，它代表一个存储在持久性存储器中的实体的对象视图，或是一个由现有企业应用程序实现的实体。

Session Bean 还可以再细分为 Stateful Session Bean 与 Stateless Session Bean，这两种的 Session Bean 都可以将系统逻辑放在 method 之中执行，不同的是 Stateful Session Bean 可以记录呼叫者的状态，因此通常来说，一个使用者会有一个相对应的 Stateful Session Bean 的实体。Stateless Session Bean 虽然也是逻辑组件，但是他却不负责记录使用者状态，也就是说当使用者呼叫 Stateless Session Bean 的时候，EJB Container 并不会找寻特定的 Stateless Session Bean 的实体来执行这个 method。换言之，很可能数个使用者在执行某个 Stateless Session Bean 的 methods 时，会是同一个 Bean 的 Instance 在执行。从内存方面来看，Stateful Session Bean 与 Stateless Session Bean 比较，Stateful Session Bean 会消耗 J2EE Server 较多的内存，然而 Stateful Session Bean 的优势却在于他可以维持使用者的状态。

97、EJB 与 JAVA BEAN 的区别？

答：Java Bean 是可复用的组件，对 Java Bean 并没有严格的规范，理论上讲，任何一个 Java 类都可以是一个 Bean。但通常情况下，由于 Java Bean 是被容器所创建（如 Tomcat）的，所以 Java Bean 应具有一个无参的构造器，另外，通常 Java Bean 还要实现 Serializable 接口用于实现 Bean 的持久性。Java Bean 实际上相当于微软 COM 模型中的本地进程内 COM 组件，它是不能被跨进程访问的。Enterprise Java Bean 相当于 DCOM，即分布式组件。它是基于 Java 的远程方法调用（RMI）技术的，所以 EJB 可以被远程访问（跨进程、跨计算机）。但 EJB 必须被布署在诸如 Webspere、WebLogic 这样的容器中，EJB 客户从不直接访问真正的 EJB 组件，而是通过其容器访问。EJB 容器是 EJB 组件的代理，EJB 组件由容器所创建和管理。客户通过容器来访问真正的 EJB 组件。

EJB 包括（SessionBean,EntityBean）说出他们的生命周期，及如何管理事务的？

SessionBean: Stateless Session Bean 的生命周期是由容器决定的，当客户机发出请求要建立一个 Bean 的实例时，EJB 容器不一定要创建一个新的 Bean 的实例供客户机调用，而是随便找一个现有的实例提供给客户机。当客户机第一次调用一个 Stateful Session Bean 时，容器必须立即在服务器中创建一个新的 Bean 实例，并关联到客户机上，以后此客户机调用 Stateful Session Bean 的方法时容器会把调用分派到与此客户机相关联的 Bean 实例。

EntityBean: Entity Beans 能存活相对较长的时间，并且状态是持续的。只要数据库中的数据存在，Entity beans 就一直存活。而不是按照应用程序或者服务进程来说的。即使 EJB 容器崩溃了，Entity beans 也是存活的。Entity Beans 生命周期能够被容器或者 Beans 自己管理。

EJB 通过以下技术管理事务：对象管理组织(OMG)的对象实务服务(OTS), Sun Microsystems 的 Transaction Service (JTS)、Java Transaction API (JTA), 开发组 (X/Open) 的 XA 接口。

98、EJB 的角色和三个对象

答：一个完整的基于 EJB 的分布式计算结构由六个角色组成，这六个角色可以由不同的开发商提供，每个角色所作的工作必须遵循 Sun 公司提供的 EJB 规范，以保证彼此之间的兼容性。这六个角色分别是 EJB 组件开发者 (Enterprise Bean Provider)、应用组合者 (Application Assembler)、部署者 (Deployer)、EJB 服务器提供者 (EJB Server Provider)、EJB 容器提供者 (EJB Container Provider)、系统管理员 (System Administrator)

三个对象是 Remote (Local) 接口、Home (LocalHome) 接口，Bean 类

99、EJB 容器提供的服务

答：主要提供声明周期管理、代码产生、持续性管理、安全、事务管理、锁和并发管理等服务。

100、EJB 规范规定 EJB 中禁止的操作有哪些？

答：1.不能操作线程和线程 API(线程 API 指非线程对象的方法如 notify,wait等), 2.不能操作 awt, 3.不能实现服务器功能, 4.不能对静态属性存取, 5.不能使用 IO 操作直接存取文件系统, 6.不能加载本地库, 7.不能将 this 作为变量和返回, 8.不能循环调用。

101、remote 接口和 home 接口主要作用

答：remote 接口定义了业务方法，用于 EJB 客户端调用业务方法。

home 接口是 EJB 工厂用于创建和移除查找 EJB 实例

home 接口是 EJB 工厂用于创建和移除查找 EJB 实例

7.12

Java 基础方面 (3)

101、remote 接口和 home 接口主要作用

答：remote 接口定义了业务方法，用于 EJB 客户端调用业务方法。

home 接口是 EJB 工厂用于创建和移除查找 EJB 实例

102、bean 实例的生命周期

答：对于 Stateless Session Bean、Entity Bean、Message Driven Bean 一般存在缓冲池管理，

而对于 Entity Bean 和 Statefull Session Bean 存在 Cache 管理，通常包含创建实例，设置上下文、创建 EJB Object (create)、业务方法调用、remove 等过程，对于存在缓冲池管理的 Bean，在 create 之后实例并不从内存清除，而是采用缓冲池调度机制不断重用实例，而对于存在 Cache 管理的 Bean 则通过激活和去激活机制保持 Bean 的状态并限制内存中实例数量。

103、EJB 的激活机制

答：以 Stateful Session Bean 为例：其 Cache 大小决定了内存中可以同时存在的 Bean 实例的数量，根据 MRU 或 NRU 算法，实例在激活和去激活状态之间迁移，激活机制是当客户端调用某个 EJB 实例业务方法时，如果对应 EJB Object 发现自己没有绑定对应的 Bean 实例则从其去激活 Bean 存储中（通过序列化机制存储实例）回复（激活）此实例。状态变迁前会调用对应的 ejbActive 和 ejbPassivate 方法。

104、EJB 的几种类型

答：会话（Session）Bean，实体（Entity）Bean 消息驱动的（Message Driven）Bean
会话 Bean 又可分为有状态（Stateful）和无状态（Stateless）两种
实体 Bean 可分为 Bean 管理的持续性（BMP）和容器管理的持续性（CMP）两种

105、客户端调用 EJB 对象的几个基本步骤

答：设置 JNDI 服务工厂以及 JNDI 服务地址系统属性，查找 Home 接口，从 Home 接口调用 Create 方法创建 Remote 接口，通过 Remote 接口调用其业务方法。

八、应用服务器方面

106、如何给 weblogic 指定大小的内存？

答：在启动 Weblogic 的脚本中（位于所在 Domian 对应服务器目录下的 startServerName），增加 set MEM_ARGS=-Xms32m -Xmx200m，可以调整最小内存为 32M，最大 200M

EJB 需直接实现它的业务接口或 Home 接口吗，请简述理由。

远程接口和 Home 接口不需要直接实现，他们的实现代码是由服务器产生的，程序运行中对实现类会作为对应接口类型的实例被使用。

107、应用服务器有那些？

答：BEA WebLogic Server，IBM WebSphere Application Server，Oracle9i Application Server，jBoss，Tomcat

108、如何设定的 weblogic 的热启动模式(开发模式)与产品发布模式？

答：可以在管理控制台中修改对应服务器的启动模式为开发或产品模式之一。或者修改服务

的启动文件或者 `commenv` 文件, 增加 `set PRODUCTION_MODE=true`。

109、如何启动时不需输入用户名与密码?

答: 修改服务启动文件, 增加 `WLS_USER` 和 `WLS_PW` 项。也可以在 `boot.properties` 文件中增加加密过的用户名和密码。

110、在 weblogic 管理制台中对一个应用域(或者说是一个网站,Domain)进行 jms 及 ejb 或连接池等相关信息进行配置后,实际保存在什么文件中?

答: 保存在此 Domain 的 `config.xml` 文件中, 它是服务器的核心配置文件。

111、说说 weblogic 中一个 Domain 的缺省目录结构?比如要将一个简单的 `helloWorld.jsp` 放入何目录下,然的在浏览器上就可打入 <http://主机:端口号/helloworld.jsp> 就可以看到运行结果了? 又比如这其中用到了一个自己写的 `javaBean` 该如何办?

答: Domain 目录服务器目录 `applications`, 将应用目录放在此目录下将可以作为应用访问, 如果是 Web 应用, 应用目录需要满足 Web 应用 目录要求, `.jsp` 文件可以直接放在应用目录中, `JavaBean` 需要放在应用目录的 `WEB-INF` 目录的 `classes` 目录中, 设置服务器的缺省应用将可以实现在浏览器上无需输入应用名。

112、在 weblogic 中发布 ejb 需涉及到哪些配置文件

答: 不同类型的 EJB 涉及的配置文件不同, 都涉及到的配置文件包括 `ejb-jar.xml`, `weblogic-ejb-jar.xml`, `CMP` 实体 Bean 一般还需要 `weblogic-cmp-rdbms-jar.xml`

86、如何在 weblogic 中进行 ssl 配置与客户端的认证配置或说说 j2ee(标准)进行 ssl 的配置

答: 缺省安装中使用 `DemoIdentity.jks` 和 `DemoTrust.jks` `KeyStore` 实现 SSL, 需要配置服务器使用 `Enable SSL`, 配置其端口, 在产品模式下需要从 CA 获取私有密钥和数字证书, 创建 `identity` 和 `trust keystore`, 装载获得的密 钥和数字证书。可以配置此 SSL 连接是单向还是双向的。

113、如何查看在 weblogic 中已经发布的 EJB?

答: 可以使用管理控制台, 在它的 `Deployment` 中可以查看所有已发布的 EJB

说说在 weblogic 中开发消息 Bean 时的 `persistent` 与 `non-persistent` 的差别

`persistent` 方式的 `MDB` 可以保证消息传递的可靠性, 也就是如果 EJB 容器出现问题而 `JMS` 服务器依然会将消息在此 `MDB` 可用的时候发送过来, 而 `non-persistent` 方式的消息将被丢弃。

九、J2EE 及 MVC 方面

114、MVC 的各个部分都有那些技术来实现?如何实现?

答：MVC 是 Model—View—Controller 的简写。"Model" 代表的是应用的业务逻辑（通过 JavaBean, EJB 组件实现），"View" 是应用的表示面（由 JSP 页面产生），"Controller" 是提供应用的处理过程控制（一般是一个 Servlet），通过这种设计 模型把应用逻辑，处理过程和显示逻辑分成不同的组件实现。这些组件可以进行交互和重用。

115、J2EE 是什么？

答：Je22 是 Sun 公司提出的多层 (multi-tiered), 分布式 (distributed), 基于组件 (component-base) 的企业 级应用模型 (enterprise application model). 在这样的一个应用系统中，可按照功能划分为不同的组件，这些组件又可在不同 计算机上，并且处于相应的层次 (tier) 中。所属层次包括客户层 (client tier) 组件, web 层和组件, Business 层和组件, 企业信息 系统 (EIS) 层。

116、WEB SERVICE 名词解释。JSDDL 开发包的介绍。JAXP、JAXM 的解释。SOAP、UDDI、WSDL 解释。

答：Web Service 描述语言 WSDL

SOAP 即简单对象访问协议 (Simple Object Access Protocol)，它是用于交换 XML 编码信息的轻量级协议。

UDDI 的目的是为电子商务建立标准；UDDI 是一套基于 Web 的、分布式的、为 Web Service 提供的、信息注册中心的实现标准规范，同时也包含一组使企业能将自身提供的 Web Service 注册，以使别的企业能够发现的访问协议的实现标准。

117、J2EE 是技术还是平台还是框架？

答：J2EE 本身是一个标准，一个为企业分布式应用的开发提供的标准平台。

J2EE 也是一个框架，包括 JDBC、JNDI、RMI、JMS、EJB、JTA 等技术。

118、STRUTS 的应用 (如 STRUTS 架构)

答：Struts 是采用 Java Servlet/JavaServer Pages 技术，开发 Web 应用程序的开放源码的 framework。采用 Struts 能开发出基于 MVC (Model-View-Controller) 设计模式的应用构架。Struts 有如下的主要功能：

1. 包含一个 controller servlet，能将用户的请求发送到相应的 Action 对象。
2. JSP 自由 tag 库，并且在 controller servlet 中提供关联支持，帮助开发人员创建交互式表单应用。
3. 提供了一系列实用对象：XML 处理、通过 Java reflection APIs 自动处理 JavaBeans 属性、国际化的提示和消息。

119、WEB SERVICE 名词解释。JSDDL 开发包的介绍。JAXP、JAXM 的解释。SOAP、UDDI、WSDL 解释。

答：Web Service 是基于网络的、分布式的模块化组件，它执行特定的任务，遵守具体的技术规范，这些规范使得 Web Service 能与其他兼容的组件进行互操作。

JAXP(Java API for XML Parsing) 定义了使用 DOM, SAX, XSLT 的通用的接口。这样在你的程序中你只要使用这些通用的接口，当你需要改变具体的实现时候也不需要修改代码。

JAXM(Java API for XML Messaging) 是为 SOAP 通信提供访问方法和传输机制的 API。

WSDL 是一种 XML 格式，用于将网络服务描述为一组端点，这些端点对包含面向文档信息或面向过程信息的信息进行操作。这种格式首先对操作和信息进行抽象描述，然后将其绑定到具体的网络协议和信息格式上以定义端点。相关的具体端点即组合成为抽象端点（服务）。

SOAP 即简单对象访问协议 (Simple Object Access Protocol)，它是用于交换 XML 编码信息的轻量级协议。

UDDI 的目的是为电子商务建立标准；UDDI 是一套基于 Web 的、分布式的、为 Web Service 提供的、信息注册中心的实现标准规范，同时也包含一组使企业能将自身提供的 Web Service 注册，以使别的企业能够发现的访问协议的实现标准。

120、C/S 与 B/S 区别：

答：有如下八个方面的不同：

1. 硬件环境不同：

C/S 一般建立在专用的网络上，小范围里的网络环境，局域网之间再通过专门服务器提供连接和数据交换服务。

B/S 建立在广域网之上的，不必是专门的网络硬件环境，例与电话上网，租用设备。信息自己管理。有比 C/S 更强的适应范围，一般只要有操作系统和浏览器就行。

2. 对安全要求不同

C/S 一般面向相对固定的用户群，对信息安全的控制能力很强。一般高度机密的信息系统采用 C/S 结构适宜。可以通过 B/S 发布部分可公开信息。

B/S 建立在广域网之上，对安全的控制能力相对弱，可能面向不可知的用户。

3. 对程序架构不同

C/S 程序可以更加注重流程,可以对权限多层次校验,对系统运行速度可以较少考虑.

B/S 对安全以及访问速度的多重的考虑,建立在需要更加优化的基础之上.比 C/S 有更高的要求 B/S 结构的程序架构是发展的趋势,从 MS 的. Net 系列的 BizTalk 2000 Exchange 2000 等,全面支持网络的构件搭建的系统. SUN 和 IBM 推的 JavaBean 构件 技术等,使 B/S 更加成熟.

4. 软件重用不同

C/S 程序可以不可避免的整体性考虑,构件的重用性不如在 B/S 要求下的构件的重用性好.

B/S 对的多重结构,要求构件相对独立的功能.能够相对较好的重用.就买来的餐桌可以再利用,而不是做在墙上的石头桌子

5. 系统维护不同

C/S 程序由于整体性,必须整体考察,处理出现的问题以及系统升级.升级难.可能是再做一个全新的系统

B/S 构件组成,方面构件个别的更换,实现系统的无缝升级.系统维护开销减到最小.用户从网上自己下载安装就可以实现升级.

6. 处理问题不同

C/S 程序可以处理用户面固定,并且在相同区域,安全要求高需求,与操作系统相关.应该都是相同的系统

B/S 建立在广域网上,面向不同的用户群,分散地域,这是 C/S 无法作到的.与操作系统平台关系最小.

7. 用户接口不同

C/S 多是建立的 Window 平台上,表现方法有限,对程序员普遍要求较高

B/S 建立在浏览器上,有更加丰富和生动的表现方式与用户交流.并且大部分难度减低,减低开发成本.

8. 信息流不同

C/S 程序一般是典型的中央集权的机械式处理,交互性相对低

B/S 信息流向可变化, B-B B-C B-G 等信息、流向的变化,更像交易中心。

答：给处于其中的应用程序组件（JSP，SERVLET）提供一个环境，使 JSP,SERVLET 直接跟容器中的环境变量交互，不必关注其它系统问题。主要有 WEB 服务器来实现。例如：TOMCAT,WEBLOGIC,WEBSPPHERE 等。该容器提供的接口严格遵守 J2EE 规范中的 WEB APPLICATION 标准。我们把遵守以上标准的 WEB 服务器就叫做 J2EE 中的 WEB 容器。

122、什么是 EJB 容器

答：Enterprise java bean 容器。更具有行业领域特色。他提供给运行在其中的组件 EJB 各种管理功能。只要满足 J2EE 规范的 EJB 放入该容器，马上就会被容器进行高效率的管理。并且可以通过现成的接口来获得系统级别的服务。例如邮件服务、事务管理

123、什么是 JNDI

答：（Java Naming & Directory Interface）JAVA 命名目录服务。主要提供的功能是：提供一个目录系统，让其它各地的应用程序在其上面留下自己的索引，从而满足快速查找和定位分布式应用程序的功能。

124、什么是 JMS

答：（Java Message Service）JAVA 消息服务。主要实现各个应用程序之间的通讯。包括点对点和广播。

125、什么是 JTA

答：（Java Transaction API）JAVA 事务服务。提供各种分布式事务服务。应用程序只需调用其提供的接口即可。

126、什么是 JAF

答：（Java Action FrameWork）JAVA 安全认证框架。提供一些安全控制方面的框架。让开发者通过各种部署和自定义实现自己的个性安全控制策略。

RMI/IIOP:（Remote Method Invocation /internet 对象请求中介协议）他们主要用于通过远程调用服务。例如， 远程有一台计算机上运行一个程序，它提供股票分析服务，我们可以在本地计算机上实现对其直接调用。当然这是要通过一定的规范才能在异构的系统之间进行通信。RMI 是 JAVA 特有的。

127、MVC 的各个部分都有那些技术来实现?如何实现?

答：MVC 是 Model— View—Controller 的简写。"Model" 代表的是应用的业务逻辑（通过 JavaBean, EJB 组件实现），"View" 是应用的表示面（由 JSP 页面产生），"Controller" 是提供应用的处理过程控制（一般是一个 Servlet），通过这种设计模型把应用逻辑，处理过程和显示逻辑分成不同的组件实现。这些组件可以进行交互和重用。

十、设计模式方面

128、开发中都用到了那些设计模式?用在什么场合?

答：每个模式都描述了一个在我们的环境中不断出现的问题，然后描述了该问题的解决方案的核心。通过这种方式，你可以无数次地使用那些已有的解决方案，无需在重复相同的工作。主要用到了 MVC 的设计模式。用来开发 JSP/Servlet 或者 J2EE 的相关应用。简单工厂模式等。

129、说说你所熟悉或听说过的 j2ee 中的几种常用模式？及对设计模式的一些看法

答：Session Facade Pattern：使用 SessionBean 访问 EntityBean

Message Facade Pattern：实现异步调用

EJB Command Pattern：使用 Command JavaBeans 取代 SessionBean，实现轻量级访问

Data Transfer Object Factory：通过 DTO Factory 简化 EntityBean 数据提供特性

Generic Attribute Access：通过 AttributeAccess 接口简化 EntityBean 数据提供特性

Business Interface：通过远程（本地）接口和 Bean 类实现相同接口规范业务逻辑一致性

EJB 架构的设计好坏将直接影响系统的性能、可扩展性、可维护性、组件可重用性及开发效率。项目越复杂，项目队伍越庞大则越能体现良好设计的重要性。

130、j2ee 常用的设计模式？说明工厂模式。

答：Java 中的 23 种设计模式：

Factory（工厂模式）， Builder（建造模式）， Factory Method（工厂方法模式），
Prototype（原始模型模式）， Singleton（单例模式）， Facade（门面模式），
Adapter（适配器模式）， Bridge（桥梁模式）， Composite（合成模式），
Decorator（装饰模式）， Flyweight（享元模式）， Proxy（代理模式），
Command（命令模式）， Interpreter（解释器模式）， Visitor（访问者模式），
Iterator（迭代子模式）， Mediator（调停者模式）， Memento（备忘录模式），
Observer（观察者模式）， State（状态模式）， Strategy（策略模式），
Template Method（模板方法模式）， Chain Of Responsibility（责任链模式）

工厂模式：工厂模式是一种经常被使用到的模式，根据工厂模式实现的类可以根据提供的
数据生成一组类中某一个类的实例，通常这一组类有一个公共的抽象父类 并且实现了相同的方法，但是这些方法针对不同的数据进行了不同的操作。首先需要定义一个基类，该类的子类通过不同的方法实现了基类中的方法。然后需要定义 一个工厂类，工厂类可以根据条件生成不同的子类实例。当得到子类的实例后，开发人员可以调用基类中的方法而不必考虑到底返回的是哪一个子类的实例。

131、UML 方面

答：标准建模语言 UML。用例图,静态图(包括类图、对象图和包图),行为图,交互图(顺序图,合作图),实现图,
UML 方面

十一、CORBA 方面

132、CORBA 是什么?用途是什么?

答：CORBA 标准是公共对象请求代理结构(Common Object Request Broker Architecture),由对象管理组织(Object Management Group, 缩写为 OMG)标准化。它的组成是接口定义语言(IDL), 语言绑定(binding;也译 为联编)和允许应用程序间互操作的协议。其目的为:

用不同的程序设计语言书写
在不同的进程中运行
为不同的操作系统开发

十二、LINUX 方面

133、LINUX 下线程, GDI 类的解释。

答：LINUX 实现的就是基于核心轻量级进程的"一对一"线程模型, 一个线程实体对应一个核心轻量级进程, 而线程之间的管理在核外函数库中实现。

GDI 类为图像设备编程接口类库。

Java 面试笔试题大汇总（1）

java 面试笔试题大汇总

第一，谈谈 final, finally, finalize 的区别。

最常被问到。

第二， Anonymous Inner Class (匿名内部类) 是否可以 extends(继承) 其它类，是否可以 implements(实现) interface(接口)?

第三， Static Nested Class 和 Inner Class 的不同，说得越多越好 (面试题有的很笼统)。

第四， &和&&的区别。

这个问得很少。

第五， HashMap 和 Hashtable 的区别。

常问。

第六， Collection 和 Collections 的区别。

你千万别说一个是单数一个是复数。

第七，什么时候用 assert。

API 级的技术人员有可能会问这个。

第八， GC 是什么? 为什么要有 GC?

基础。

第九， String s = new String("xyz"); 创建了几个 String Object?

第十， Math.round(11.5) 等於多少? Math.round(-11.5) 等於多少?

第十一， short s1 = 1; s1 = s1 + 1; 有什么错? short s1 = 1; s1 += 1; 有什么错?

面试题都是很变态的，要做好受虐的准备。

第十二， sleep() 和 wait() 有什么区别?

搞线程的最爱。

第十三， Java 有没有 goto?

JAVA 相关基础知识

1、面向对象的特征有哪些方面

1.抽象:

抽象就是忽略一个主题中与当前目标无关的那些方面，以便更充分地注意与当前目标有关的方面。抽象并不打算了解全部问题，而只是选择其中的一部分，暂时不用部分细节。抽象包括两个方面，一是过程抽象，二是数据抽象。

2.继承:

继承是一种联结类的层次模型，并且允许和鼓励类的重用，它提供了一种明确表述共性的方法。对象的一个新类可以从现有的类中派生，这个过程称为类继承。新类继承了原始类的特性，新类称为原始类的派生类（子类），而原始类称为新类的基类（父类）。派生类可以从它的基类那里继承方法和实例变量，并且类可以修改或增加新的方法使之更适合特殊的需要。

3.封装:

封装是把过程和数据包围起来，对数据的访问只能通过已定义的界面。面向对象计算始于这个基本概念，即现实世界可以被描绘成一系列完全自治、封装的对象，这些对象通过一个受保护的接口访问其他对象。

4.多态性:

多态性是指允许不同类的对象对同一消息作出响应。多态性包括参数化多态性和包含多态性。多态性语言具有灵活、抽象、行为共享、代码共享的优势，很好的解决了应用程序函数同名问题。

2、String 是最基本的数据类型吗？

基本数据类型包括 byte、int、char、long、float、double、boolean 和 short。

java.lang.String 类是 final 类型的，因此不可以继承这个类、不能修改这个类。为了提高效率节省空间，我们应该用 StringBuffer 类

3、int 和 Integer 有什么区别

Java 提供两种不同的类型：引用类型和原始类型（或内置类型）。Int 是 java 的原始数据类型，Integer 是 java 为 int 提供的封装类。Java 为每个原始类型提供了封装类。

原始类型封装类

booleanBoolean

charCharacter

byteByte

shortShort

intInteger

longLong

floatFloat

doubleDouble

引用类型和原始类型的行为完全不同，并且它们具有不同的语义。引用类型和原始类型具有不同的特征和用法，它们包括：大小和速度问题，这种类型以哪种类型的数据结构存储，当引用类型和原始类型用作某个类的实例数据时所指定的缺省值。对象引用实例变量的缺省值为 null，而原始类型实例变量的缺省值与它们的类 型有关。

4、String 和 StringBuffer 的区别

JAVA 平台提供了两个类：String 和 StringBuffer，它们 可以储存和操作字符串，即包含多个字符的字符数据。这个 String 类提供了数值不可改变的字符串。而这个 StringBuffer 类提供的字符串进行修改。当你知道字符数据要改变的时候你就可以使用 StringBuffer。典型地，你可以使用

StringBuffers 来动态构造字符数据。

5、运行时异常与一般异常有何异同？

异常表示程序运行过程中可能出现的非正常状态，运行时异常表示虚拟机的通常操作中可能遇到的异常，是一种常见运行错误。java 编译器要求方法必须声明抛出可能发生的非运行时异常，但是并不要求必须声明抛出未被捕获的运行时异常。

6、说出 Servlet 的生命周期，并说出 Servlet 和 CGI 的区别。

Servlet 被服务器实例化后，容器运行其 init 方法，请求到达时运行其 service 方法，service 方法自动派遣运行与请求对应的 doXXX 方法（doGet，doPost）等，当服务器决定将实例销毁的时候调用其 destroy 方法。

与 cgi 的区别在于 servlet 处于服务器进程中，它通过多线程方式运行其 service 方法，一个实例可以服务于多个请求，并且其实例一般不会销毁，而 CGI 对每个请求都产生新的进程，服务完成后就销毁，所以效率上低于 servlet。

7、说出 ArrayList,Vector,LinkedList 的存储性能和特性

ArrayList 和 Vector 都是使用数组方式存储数据，此数组元素数大于实际存储的数据以便增加和插入元素，它们都允许直接按序号索引元素，但是插入元素要涉及数组元素移动等内存操作，所以索引数据快而插入数据慢，Vector 由于使用了 synchronized 方法（线程安全），通常性能上较 ArrayList 差，而 LinkedList 使用双向链表实现存储，按序号索引数据需要进行前向或后向遍历，但是插入数据时只需要记录本项的前后项即可，所以插入速度较快。

8、EJB 是基于哪些技术实现的？并说出 SessionBean 和 EntityBean 的区别，StatefulBean 和 StatelessBean 的区别。

EJB 包括 Session Bean、Entity Bean、Message Driven Bean，基于 JNDI、RMI、JAT 等技术实现。SessionBean 在 J2EE 应用程序中被用来完成一些服务器端的业务操作，例如访问数据库、调用其他 EJB 组件。EntityBean 被用来代表应用系统中用到的数据。

对于客户机，SessionBean 是一种非持久性对象，它实现某些在服务器上运行的业务逻辑。

对于客户机，EntityBean 是一种持久性对象，它代表一个存储在持久性存储器中的实体的对象视图，或是一个由现有企业应用程序实现的实体。

Session Bean 还可以再细分为 Stateful Session Bean 与 Stateless Session Bean，这两种的 Session Bean 都可以将系统逻辑放在 method 之中执行，不同的是 Stateful Session Bean 可以记录呼叫者的状态，因此通常来说，一个使用者会有一个相对应的 Stateful Session Bean 的实体。Stateless Session Bean 虽然也是逻辑组件，但是他却不负责记录使用者状态，也就是说当使用者呼叫 Stateless Session Bean 的时候，EJB Container 并不会找寻特定的 Stateless Session Bean 的实体来执行这个 method。换言之，很可能数个使用者在执行某个 Stateless Session Bean 的 methods 时，会是同一个 Bean 的 Instance 在执行。从内存方面来看，Stateful Session Bean 与 Stateless Session Bean 比较，Stateful Session Bean 会消耗 J2EE Server 较多的内存，然而 Stateful Session Bean 的优势却在于他可以维持使用者的状态。

9、Collection 和 Collections 的区别。

Collection 是集合类的上级接口，继承与他的接口主要有 Set 和 List。

Collections 是针对集合类的一个帮助类，他提供一系列静态方法实现对各种集合的搜索、排序、线程安全化等操作。

10、&和&&的区别。

&是位运算符，表示按位与运算，&&是逻辑运算符，表示逻辑与（and）。

11、HashMap 和 Hashtable 的区别。

HashMap 是 Hashtable 的轻量级实现（非线程安全的实现），他们都完成了 Map 接口，主要区别在于 HashMap 允许空（null）键值（key），由于非线程安全，效率上可能高于 Hashtable。

HashMap 允许将 null 作为一个 entry 的 key 或者 value，而 Hashtable 不允许。

HashMap 把 Hashtable 的 contains 方法去掉了，改成 containsvalue 和 containsKey。因为 contains 方法容易让人引起误解。

Hashtable 继承自 Dictionary 类，而 HashMap 是 Java1.2 引进的 Map interface 的一个实现。

最大的不同是，Hashtable 的方法是 Synchronize 的，而 HashMap 不是，在多个线程访问 Hashtable 时，不需要自己为它的方法实现同步，而 HashMap 就必须为之提供外同步。

Hashtable 和 HashMap 采用的 hash/rehash 算法都大概一样，所以性能不会有很大的差异。

12、final, finally, finalize 的区别。

final 用于声明属性，方法和类，分别表示属性不可变，方法不可覆盖，类不可继承。

finally 是异常处理语句结构的一部分，表示总是执行。

finalize 是 Object 类的一个方法，在垃圾收集器执行的时候会调用被回收对象的此方法，可以覆盖此方法提供垃圾收集时的其他资源回收，例如关闭文件等。

13、sleep() 和 wait() 有什么区别？

sleep 是线程类（Thread）的方法，导致此线程暂停执行指定时间，给执行机会给其他线程，但是监控状态依然保持，到时后会自动恢复。调用 sleep 不会释放对象锁。

wait 是 Object 类的方法，对此对象调用 wait 方法导致本线程放弃对象锁，进入等待此对象的等待锁定池，只有针对此对象发出 notify 方法（或 notifyAll）后本线程才进入对象锁定池准备获得对象锁进入运行状态。

14、Overload 和 Override 的区别。Overloaded 的方法是否可以改变返回值的类型？

方法的重写 Overriding 和重载 Overloading 是 Java 多态性的不同表现。重写 Overriding 是父类与子类之间多态性的一种表现，重载 Overloading 是一个类中多态性的一种表现。如果在子类中定义某方法与其父类有相同的名称和参数，我们说该方法被重写（Overriding）。子类的对象使用这个方法时，将调用子类中的定义，对它而言，父类中的定义如同被“屏蔽”了。如果在一个类中定义了多个同名的方法，它们或有不同的参数个数或有不同的参数类型，则称为方法的重载（Overloading）。Overloaded 的方法是可以改变返回值的类型。

15、error 和 exception 有什么区别？

error 表示恢复不是不可能但很困难的情况下的一种严重问题。比如说内存溢出。不可能指望程序能处理这样的情况。

exception 表示一种设计或实现问题。也就是说，它表示如果程序运行正常，从不会发生的情况。

16、同步和异步有何异同，在什么情况下分别使用他们？举例说明。

如果数据将在线程间共享。例如正在写的的数据以后可能被另一个线程读到，或者正在读的数据可能已经被另一个线程写过了，那么这些数据就是共享数据，必须进行同步存取。

当应用程序在对象上调用了一个需要花费很长时间来执行的方法，并且不希望让程序等待方法的返回时，就应该使用异步编程，在很多情况下采用异步途径往往更有效率。

17、abstract class 和 interface 有什么区别？

声明方法的存在而不去实现它的类被叫做抽象类（abstract class），它用于要创建一个体现某些基本行为的类，并为该类声明方法，但不能在该类中实现该方法的情况。不能创建 abstract 类的实例。然而可以创建一个变量，其类型是一个抽象类，并让它指向具体子类的一个实例。不能有抽象构造函数或抽象静态方法。Abstract 类的子类为它们父类中的所有抽象方法提供实现，否则它们也是抽象类为。取而代之，在子类中实现该方法。知道其行为的其它类可以在类中实现这些方法。

接口（interface）是抽象类的变体。在接口中，所有方法都是抽象的。多继承性可通过实现这样的接口而获得。接口中的所有方法都是抽象的，没有一个有程序体。接口只可以定义 static final 成员变量。接口的实现与子类相似，除了该实现类不能从接口定义中继承行为。当类实现特殊

接口时，它定义（即将程序体给予）所有这种接口的方法。然后，它可以在实现了该接口的类的任何对象上调用接口的方法。由于有抽象类，它允许使用接口名作为引用变量的类型。通常的动态联编将生效。引用可以转换到接口类型或从接口类型转换，`instanceof`运算符可以用来决定某对象的类是否实现了接口。

18、heap 和 stack 有什么区别。

栈是一种线形集合，其添加和删除元素的操作应在同一段完成。栈按照后进先出的方式进行处理。堆是栈的一个组成元素

19、forward 和 redirect 的区别

forward 是服务器请求资源，服务器直接访问目标地址的 URL，把那个 URL 的响应内容读取过来，然后把这些内容再发给浏览器，浏览器根本不知道服务器发送的内容是从哪儿来的，所以它的地址栏中还是原来的地址。

redirect 就是服务端根据逻辑，发送一个状态码，告诉浏览器重新去请求那个地址，一般来说浏览器会用刚才请求的所有参数重新请求，所以 session,request 参数都可以获取。

20、EJB 与 JAVA BEAN 的区别？

Java Bean 是可复用的组件，对 Java Bean 并没有严格的规范，理论上讲，任何一个 Java 类都可以是一个 Bean。但通常情况下，由于 Java Bean 是被容器所创建（如 Tomcat）的，所以 Java Bean 应具有一个无参的构造器，另外，通常 Java Bean 还要实现 `Serializable` 接口用于实现 Bean 的持久性。Java Bean 实际上相当于微软 COM 模型中的本地进程内 COM 组件，它是不能被跨进程访问的。Enterprise Java Bean 相当于 DCOM，即分布式组件。它是基于 Java 的远程方法调用（RMI）技术的，所以 EJB 可以被远程访问（跨进程、跨计算机）。但 EJB 必须被布署在诸如 Webspere、WebLogic 这样的容器中，EJB 客户从不直接访问真正的 EJB 组件，而是通过其容器访问。EJB 容器是 EJB 组件的代理，EJB 组件由容器所创建和管理。客户通过容器来访问真正的 EJB 组件。

21、Static Nested Class 和 Inner Class 的不同。

Static Nested Class 是被声明为静态（static）的内部类，它可以不依赖于外部类实例被实例化。而通常的内部类需要在外部类实例化后才能实例化。

22、JSP 中动态 INCLUDE 与静态 INCLUDE 的区别？

动态 INCLUDE 用 `jsp:include` 动作实现它总是会检查所含文件中的变化，适合用于包含动态页面，并且可以带参数。

静态 INCLUDE 用 `include` 伪码实现，定不会检查所含文件的变化，适用于包含静态页面

23、什么时候用 assert。

assertion (断言)在软件开发中是一种常用的调试方式，很多开发语言中都支持这种机制。在实现中，assertion 就是在程序中的一条语句，它对一个 boolean 表达式进行检查，一个正确程序必须保证这个 boolean 表达式的值为 true；如果该值为 false，说明程序已经处于不正确的状态下，系统将给出警告或退出。一般来说，assertion 用于保证程序最基本、关键的正确性。assertion 检查通常在开发和测试时开启。为了提高性能，在软件发布后，assertion 检查通常是关闭的。

24、GC 是什么？为什么要有 GC？

GC 是垃圾收集的意思（Garbage Collection），内存处理是编程人员容易出现问题的地方，忘记或者错误的内存回收会导致程序或系统的不稳定甚至崩溃，Java 提供的 GC 功能可以自动监测对象是否超过作用域从而达到自动回收内存的目的，Java 语言没有提供释放已分配内存的显示操作方法。

25、`short s1 = 1; s1 = s1 + 1;`有什么错？`short s1 = 1; s1 += 1;`有什么错？

`short s1 = 1; s1 = s1 + 1;`（`s1+1` 运算结果是 int 型，需要强制转换类型）

`short s1 = 1; s1 += 1;`（可以正确编译）

26、Math.round(11.5)等於多少? Math.round(-11.5)等於多少?

Math.round(11.5)=12

Math.round(-11.5)=-11

round 方法返回与参数最接近的长整数，参数加 1/2 后求其 floor.

27、String s = new String("xyz");创建了几个 String Object?

两个

28、设计 4 个线程，其中两个线程每次对 j 增加 1，另外两个线程对 j 每次减少 1。写出程序。

以下程序使用内部类实现线程，对 j 增减的时候没有考虑顺序问题。

```
public class ThreadTest1 {
    private int j;
    public static void main(String args[]) {
        ThreadTest1 tt = new ThreadTest1();
        Inc inc = tt.new Inc();
        Dec dec = tt.new Dec();
        for (int i = 0; i < 2; i++) {
            Thread t = new Thread(inc);
            t.start();
            t = new Thread(dec);
            t.start();
        }
        private synchronized void inc() {
            j++;
            System.out.println(Thread.currentThread().getName() + "-inc:" + j);
        }
        private synchronized void dec() {
            j--;
            System.out.println(Thread.currentThread().getName() + "-dec:" + j);
        }
        class Inc implements Runnable {
            public void run() {
                for (int i = 0; i < 100; i++) {
                    inc();
                }
            }
        }
        class Dec implements Runnable {
            public void run() {
                for (int i = 0; i < 100; i++) {
                    dec();
                }
            }
        }
    }
}
```

29、Java 有没有 goto?

java 中的保留字，现在没有在 java 中使用。

30、启动一个线程是用 run()还是 start()?

启动一个线程是调用 start()方法，使线程所代表的虚拟处理机处于可运行状态，这意味着它可以由 JVM 调度并执行。这并不意味着线程就会立即运行。run()方法可以产生必须退出的标志来停止一个线程。

31、EJB 包括 (SessionBean,EntityBean) 说出他们的生命周期，及如何管理事务的?

SessionBean: Stateless Session Bean 的生命周期是由容器决定的，当客户机发出请求要建立一个 Bean 的实例时，EJB 容器不一定要创建一个新 的 Bean 的实例供客户机调用，而是随便找一个现有的实例提供给客户机。当客户机第一次调用一个 Stateful Session Bean 时，容器必须立即在服务器中创建一个新的 Bean 实例，并关联到客户机上，以后此客户机调用 Stateful Session Bean 的方法时容器会把调用分派 到与此客户机相关联的 Bean 实例。

EntityBean: Entity Beans 能存活相对较长的时间，并且状态是持续的。只要数据库中的数据存在，Entity beans 就一直存活。而不是按照应用程序或者服务进程来说的。即使 EJB 容器崩溃了，Entity beans 也是存活的。Entity Beans 生命周期能够被容器或者 Beans 自己管理。

EJB 通过以下技术管理实务：对象管理组织 (OMG) 的对象实务服务 (OTS)，Sun Microsystems 的 Transaction Service (JTS)、Java Transaction API (JTA)，开发组 (X/Open) 的 XA 接口。

32、应用服务器有那些?

BEA WebLogic Server, IBM WebSphere Application Server, Oracle9i Application Server, jBoss, Tomcat

33、给我一个你最常见到的 runtime exception。

ArithmeticException, ArrayStoreException, BufferOverflowException, BufferUnderflowException, CannotRedoException, CannotUndoException, ClassCastException, CMMException, ConcurrentModificationException, DOMException, EmptyStackException, IllegalArgumentException, IllegalMonitorStateException, IllegalPathStateException, IllegalStateException, ImagingOpException, IndexOutOfBoundsException, MissingResourceException, NegativeArraySizeException, NoSuchElementException, NullPointerException, ProfileDataException, ProviderException, RasterFormatException, SecurityException, SystemException, UndeclaredThrowableException, UnmodifiableSetException, UnsupportedOperationException

34、接口是否可继承接口? 抽象类是否可实现 (implements)接口? 抽象类是否可继承实体类 (concrete class)?

接口可以继承接口。抽象类可以实现 (implements)接口，抽象类是否可继承实体类，但前提是实体类必须有明确的构造函数。

35、List, Set, Map 是否继承自 Collection 接口?

List, Set 是，Map 不是

36、说出数据连接池的工作机制是什么?

J2EE 服务器启动时会建立一定数量的池连接，并一直维持不少于此数目的池连接。客户端程序需要连接时，池驱动程序会返回一个未使用的池连接并将其标记为忙。如果 当前没有空闲连接，池驱动程序就新建一定数量的连接，新建连接的数量有配置参数决定。当使用的池连接调用完成后，池驱动程序将此连接标记为空闲，其他调用 就可以使用这个连接。

37、abstract 的 method 是否可同时是 static,是否可同时是 native, 是否可同时是 synchronized?

都不能

38、数组有没有 length()这个方法? String 有没有 length()这个方法?

数组没有 `length()`这个方法，有 `length` 的属性。String 有 `length()`这个方法。

39、Set 里的元素是不能重复的，那么用什么方法来区分重复与否呢？是用 `==` 还是 `equals()`？它们有何区别？

Set 里的元素是不能重复的，那么用 `iterator()` 方法来区分重复与否。`equals()` 是判断两个 Set 是否相等。

`equals()` 和 `==` 方法决定引用值是否指向同一对象 `equals()` 在类中被覆盖，为的是当两个分离的对象的内容和类型相配的话，返回真值。

40、构造器 Constructor 是否可被 override？

构造器 Constructor 不能被继承，因此不能重写 Overriding，但可以被重载 Overloading。

41、是否可以继承 String 类？

String 类是 `final` 类故不可以继承。

42、switch 是否能作用在 byte 上，是否能作用在 long 上，是否能作用在 String 上？

`switch (expr1)` 中，`expr1` 是一个整数表达式。因此传递给 `switch` 和 `case` 语句的参数应该是 `int`、`short`、`char` 或者 `byte`。`long`、`string` 都不能作用于 `switch`。

43、try {} 里有一个 return 语句，那么紧跟在这个 try 后的 finally {} 里的 code 会不会被执行，什么时候被执行，在 return 前还是后？

会执行，在 return 前执行。

44、编程题：用最有效率的方法算出 2 乘以 8 等於几？

`2 << 3`

45、两个对象值相同 (`x.equals(y) == true`)，但却可有不同的 hash code，这句话对不对？

不对，有相同的 hash code。

46、当一个对象被当作参数传递到一个方法后，此方法可改变这个对象的属性，并可返回的引用。对象的内容可以在被调用的方法中改变，但对象的引用是永远不会改变的。

47、当一个线程进入一个对象的一个 `synchronized` 方法后，其它线程是否可进入此对象的其它方法？

不能，一个对象的一个 `synchronized` 方法只能由一个线程访问。

48、编程题：写一个 Singleton 出来。

Singleton 模式主要作用是保证在 Java 应用程序中，一个类 Class 只有一个实例存在。

一般 Singleton 模式通常有几种形式：

第一种形式：定义一个类，它的构造函数为 `private` 的，它有一个 `static` 的 `private` 的该类变量，在类初始化时实例化，通过一个 `public` 的 `getInstance` 方法获取对它的引用，继而调用其中的方法。

```
public class Singleton {
    private Singleton(){}
    //在自己内部定义自己一个实例，是不是很奇怪？
    //注意这是 private 只供内部调用
    private static Singleton instance = new Singleton();
    //这里提供了一个供外部访问本 class 的静态方法，可以直接访问
    public static Singleton getInstance() {
        return instance;
    }
}
```

第二种形式：

```
public class Singleton {
    private static Singleton instance = null;
```

```

public static synchronized Singleton getInstance() {
    //这个方法比上面有所改进，不用每次都进行生成对象，只是第一次
    //使用时生成实例，提高了效率！
    if (instance==null)
        instance=new Singleton();
    return instance;
}
}

```

其他形式：

定义一个类，它的构造函数为 `private` 的，所有方法为 `static` 的。

一般认为第一种形式要更加安全些

49、Java 的接口和 C++的虚类的相同和不同处。

由于 Java 不支持多继承，而有可能某个类或对象要使用分别在几个类或对象里面的方法或属性，现有的单继承机制就不能满足要求。与继承相比，接口有更高的灵活性，因为接口中没有任何实现代码。当一个类实现了接口以后，该类要实现接口里面所有的方法和属性，并且接口里面的属性在默认状态下面都是 `public static`，所有方法默认情况下是 `public`。一个类可以实现多个接口。

50、Java 中的异常处理机制的简单原理和应用。

当 JAVA 程序违反了 JAVA 的语义规则时，JAVA 虚拟机就会将发生的错误表示为一个异常。违反语义规则包括 2 种情况。一种是 JAVA 类库内置的语义检查。例如数组下标越界，会引发 `IndexOutOfBoundsException`；访问 `null` 的对象时会引发 `NullPointerException`。另一种情况就是 JAVA 允许程序员扩展这种语义检查，程序员可以创建自己的异常，并自由选择何时用 `throw` 关键字引发异常。所有的异常都是 `java.lang.Throwable` 的子类。

7.15

Java 面试笔试题大汇总（2）

51、垃圾回收的优点和原理。并考虑 2 种回收机制。

Java 语言中一个显著的特点就是引入了垃圾回收机制，使 c++程序员最头疼的内存管理的问题迎刃而解，它使得 Java 程序员在编写程序的时候不再需要考虑内存管理。由于有个垃圾回收机制，Java 中的对象不再有“作用域”的概念，只有对象的引用才有“作用域”。垃圾回收可以有效的防止内存泄露，有效的使用可以使用的内存。垃圾回收器通常是作为一个单独的较低级别的线程运行，不可预知的情况下对内存堆中已经死亡的或者长时间没有使用的对象进行清楚和回收，程序员不能实时的调用垃圾回收器对某个对象或所有对象进行垃圾回收。回收机制有分代复制垃圾回收和标记垃圾回收，增量垃圾回收。

52、请说出你所知道的线程同步的方法。

`wait()`:使一个线程处于等待状态，并且释放所持有的对象的 `lock`。

`sleep()`:使一个正在运行的线程处于睡眠状态，是一个静态方法，调用此方法要捕捉 `InterruptedException` 异常。

`notify()`:唤醒一个处于等待状态的线程，注意的是在调用此方法的时候，并不能确切的唤醒某一个等待状态的线程，而是由 JVM 确定唤醒哪个线程，而且不是按优先级。

`Allnotify()`:唤醒所有处于等待状态的线程，注意并不是给所有唤醒线程一个对象的锁，而是让它们竞争。

53、你所知道的集合类都有哪些？主要方法？

最常用的集合类是 `List` 和 `Map`。`List` 的具体实现包括 `ArrayList` 和 `Vector`，它们是可变大小的列表，比较适合构建、存储和操作任何类型对象的元素列表。`List` 适用于按数值索引访问元素的情

形。

Map 提供了一个更通用的元素存储方法。Map 集合类用于存储元素对（称作“键”和“值”），其中每个键映射到一个值。

54、描述一下 JVM 加载 class 文件的原理机制？

JVM 中类的装载是由 ClassLoader 和它的子类来实现的,Java ClassLoader 是一个重要的 Java 运行时系统组件。它负责在运行时查找和装入类文件的类。

55、char 型变量中能不能存贮一个中文汉字?为什么？

能够定义成为一个中文的，因为 java 中以 unicode 编码，一个 char 占 16 个字节，所以放一个中文是没问题的

56、多线程有几种实现方法,都是什么?同步有几种实现方法,都是什么?

多线程有两种实现方法，分别是继承 Thread 类与实现 Runnable 接口

同步的实现方面有两种，分别是 synchronized,wait 与 notify

57、JSP 的内置对象及方法。

request 表示 HttpServletRequest 对象。它包含了有关浏览器请求的信息，并且提供了几个用于获取 cookie, header, 和 session 数据的有用的方法。

response 表示 HttpServletResponse 对象，并提供了几个用于设置送回浏览器的响应的方法（如 cookies,头信息等）

out 对象是 javax.jsp.JspWriter 的一个实例，并提供了几个方法使你能用于向浏览器回送输出结果。

pageContext 表示一个 javax.servlet.jsp.PageContext 对象。它是用于方便存取各种范围的名字空间、servlet 相关的对象的 API，并且包装了通用的 servlet 相关功能的方法。

session 表示一个请求的 javax.servlet.http.HttpSession 对象。Session 可以存贮用户的状态信息

application 表示一个 javax.servelet.ServletContext 对象。这有助于查找有关 servlet 引擎和 servlet 环境的信息

config 表示一个 javax.servlet.ServletConfig 对象。该对象用于存取 servlet 实例的初始化参数。

page 表示从该页面产生的一个 servlet 实例

58、线程的基本概念、线程的基本状态以及状态之间的关系

线程指在程序执行过程中，能够执行程序代码的一个执行单位，每个程序至少都有一个线程，也就是程序本身。

Java 中的线程有四种状态分别是：运行、就绪、挂起、结束。

59、JSP 的常用指令

isErrorPage(是否能使用 Exception 对象)，isELIgnored(是否忽略表达式)

60、什么情况下调用 doGet() 和 doPost()？

Jsp 页面中的 form 标签里的 method 属性为 get 时调用 doGet()，为 post 时调用 doPost()。

61、servlet 的生命周期

web 容器加载 servlet，生命周期开始。通过调用 servlet 的 init()方法进行 servlet 的初始化。通过调用 service()方法实现，根据请求的不同调用不同的 do***()方法。结束服务，web 容器调用 servlet 的 destroy()方法。

62、如何现实 servlet 的单线程模式

63、页面间对象传递的方法

request, session, application, cookie 等

64、JSP 和 Servlet 有哪些相同点和不同点，他们之间的联系是什么？

JSP 是 Servlet 技术的扩展，本质上是 Servlet 的简易方式，更强调应用的外表表达。JSP 编译后是"类 servlet"。Servlet 和 JSP 最主要的不同点在于，Servlet 的应用逻辑是在 Java 文件中，并且完全从表示层中的 HTML 里分离开来。而 JSP 的情况是 Java 和 HTML 可以组合成一个扩展名为.jsp 的文件。JSP 侧重于视图，Servlet 主要用于控制逻辑。

65、四种会话跟踪技术

会话作用域 ServletsJSP 页面描述

page 否是代表与一个页面相关的对象和属性。一个页面由一个编译好的 Java servlet 类（可以带有任何的 include 指令，但是没有 include 动作）表示。这既包括 servlet 又包括被编译成 servlet 的 JSP 页面

request 是代表与 Web 客户机发出的一个请求相关的对象和属性。一个请求可能跨越多个页面，涉及多个 Web 组件（由于 forward 指令和 include 动作的关系）

session 是代表与用于某个 Web 客户机的一个用户体验相关的对象和属性。一个 Web 会话可以也经常跨越多个客户机请求

application 是代表与整个 Web 应用程序相关的对象和属性。这实质上是跨越整个 Web 应用程序，包括多个页面、请求和会话的一个全局作用域

66、Request 对象的主要方法：

setAttribute(String name,Object)：设置名字为 name 的 request 的参数值

getAttribute(String name)：返回由 name 指定的属性值

getAttributeNames()：返回 request 对象所有属性的名字集合，结果是一个枚举的实例

getCookies()：返回客户端的所有 Cookie 对象，结果是一个 Cookie 数组

getCharacterEncoding()：返回请求中的字符编码方式

getContentLength()：返回请求的 Body 的长度

getHeader(String name)：获得 HTTP 协议定义的文件头信息

getHeaders(String name)：返回指定名字的 request Header 的所有值，结果是一个枚举的实例

getHeaderNames()：返回所以 request Header 的名字，结果是一个枚举的实例

getInputStream()：返回请求的输入流，用于获得请求中的数据

getMethod()：获得客户端向服务器端传送数据的方法

getParameter(String name)：获得客户端传送给服务器端的有 name 指定的参数值

getParameterNames()：获得客户端传送给服务器端的所有参数的名字，结果是一个枚举的实例

getParameterValues(String name)：获得有 name 指定的参数的所有值

getProtocol()：获取客户端向服务器端传送数据所依据的协议名称

getQueryString()：获得查询字符串

getRequestURI()：获取发出请求字符串的客户端地址

getRemoteAddr()：获取客户端的 IP 地址

getRemoteHost()：获取客户端的名字

getSession([Boolean create])：返回和请求相关 Session

getServerName()：获取服务器的名字

getServletPath()：获取客户端所请求的脚本文件的路径

getServerPort()：获取服务器的端口号

removeAttribute(String name)：删除请求中的一个属性

67、J2EE 是技术还是平台还是框架？

J2EE 本身是一个标准，一个为企业分布式应用的开发提供的标准平台。

J2EE 也是一个框架，包括 JDBC、JNDI、RMI、JMS、EJB、JTA 等技术。

68、我们在 web 应用开发过程中经常遇到输出某种编码的字符，如 iso8859-1 等，如何输出一个某种编码的字符串？

```
Public String translate(String str) {  
String tempStr = "";  
try {  
tempStr = new String(str.getBytes("ISO-8859-1"), "GBK");  
tempStr = tempStr.trim();  
}  
catch (Exception e) {  
System.err.println(e.getMessage());  
}  
return tempStr;  
}
```

69、简述逻辑操作 (&,&|,^) 与条件操作 (&&,&&||) 的区别。

区别主要答两点：a. 条件操作只能操作布尔型的，而逻辑操作不仅可以操作布尔型，而且可以操作数值型

b. 逻辑操作不会产生短路

70、XML 文档定义有几种形式？它们之间有何本质区别？解析 XML 文档有哪几种方式？

a: 两种形式 dtd schema，b: 本质区别: schema 本身是 xml 的，可以被 XML 解析器解析(这也是从 DTD 上发展 schema 的根本目的)，c: 有 DOM, SAX, STAX 等

DOM: 处理大型文件时其性能下降的非常厉害。这个问题是由 DOM 的树结构所造成的，这种结构占用的内存较多，而且 DOM 必须在解析文件之前把整个文档装入内存，适合对 XML 的随机访问

SAX: 不现于 DOM, SAX 是事件驱动型的 XML 解析方式。它顺序读取 XML 文件，不需要一次全部装载整个文件。当遇到像文件开头，文档结束，或者标签开头与标签结束时，它会触发一个事件，用户通过在其回调事件中写入处理代码来处理 XML 文件，适合对 XML 的顺序访问

STAX: Streaming API for XML (StAX)

71、简述 synchronized 和 java.util.concurrent.locks.Lock 的异同？

主要相同点：Lock 能完成 synchronized 所实现的所有功能

主要不同点：Lock 有比 synchronized 更精确的线程语义和更好的性能。synchronized 会自动释放锁，而 Lock 一定要求程序员手工释放，并且必须在 finally 从句中释放。

72、EJB 的角色和三个对象

一个完整的基于 EJB 的分布式计算结构由六个角色组成，这六个角色可以由不同的开发商提供，每个角色所作的工作必须遵循 Sun 公司提供的 EJB 规范，以保证彼此之间的兼容性。这六个角色分别是 EJB 组件开发者 (Enterprise Bean Provider)、应用组合者 (Application Assembler)、部署者 (Deployer)、EJB 服务器提供者 (EJB Server Provider)、EJB 容器提供者 (EJB Container Provider)、系统管理员 (System Administrator)

三个对象是 Remote (Local) 接口、Home (LocalHome) 接口，Bean 类

73、EJB 容器提供的服务

主要提供声明周期管理、代码产生、持续性管理、安全、事务管理、锁和并发管理等服务。

74、EJB 规范规定 EJB 中禁止的操作有哪些？

1. 不能操作线程和线程 API (线程 API 指非线程对象的方法如 notify, wait 等)，2. 不能操作 awt，3. 不能实现服务器功能，4. 不能对静态属性生存取，5. 不能使用 IO 操作直接存取文件系统，6. 不能加载本地库，7. 不能将 this 作为变量和返回，8. 不能循环调用。

75、remote 接口和 home 接口主要作用

remote 接口定义了业务方法，用于 EJB 客户端调用业务方法。

home 接口是 EJB 工厂用于创建和移除查找 EJB 实例

76、bean 实例的生命周期

对于 Stateless Session Bean、Entity Bean、Message Driven Bean 一般存在缓冲池管理，而对于 Entity Bean 和 Statefull Session Bean 存在 Cache 管理，通常包含创建实例，设置上下文、创建 EJB Object（create）、业务方法调用、remove 等过程，对于存在缓冲池管理的 Bean，在 create 之后实例并不从内存清除，而是采用缓冲池调度机制不断重用实例，而对于存在 Cache 管理的 Bean 则通过激活和去激活机制保持 Bean 的状态并限制内存中实例数量。

77、EJB 的激活机制

以 Stateful Session Bean 为例：其 Cache 大小决定了内存中可以同时存在的 Bean 实例的数量，根据 MRU 或 NRU 算法，实例在激活和去激活状态之间迁移，激活机制是当客户端调用某个 EJB 实例业务方法时，如果对应 EJB Object 发现自己没有绑定对应的 Bean 实例则从其去激活 Bean 存储中（通过序列化机制存储实例）回复（激活）此实例。状态变迁前会调用对应的 ejbActive 和 ejbPassivate 方法。

78、EJB 的几种类型

会话（Session）Bean，实体（Entity）Bean 消息驱动的（Message Driven）Bean

会话 Bean 又可分为有状态（Stateful）和无状态（Stateless）两种

实体 Bean 可分为 Bean 管理的持续性（BMP）和容器管理的持续性（CMP）两种

79、客户端调用 EJB 对象的几个基本步骤

设置 JNDI 服务工厂以及 JNDI 服务地址系统属性，查找 Home 接口，从 Home 接口调用 Create 方法创建 Remote 接口，通过 Remote 接口调用其业务方法。

80、如何给 weblogic 指定大小的内存？

在启动 Weblogic 的脚本中（位于所在 Domain 对应服务器目录下的 startServerName），增加 set MEM_ARGS=-Xms32m -Xmx200m，可以调整最小内存为 32M，最大 200M

81、如何设定的 weblogic 的热启动模式(开发模式)与产品发布模式？

可以在管理控制台中修改对应服务器的启动模式为开发或产品模式之一。或者修改服务的启动文件或者 commenv 文件，增加 set PRODUCTION_MODE=true。

82、如何启动时不需输入用户名与密码？

修改服务启动文件，增加 WLS_USER 和 WLS_PW 项。也可以在 boot.properties 文件中增加加密过的用户名和密码。

83、在 weblogic 管理制台中对一个应用域(或者说是一个网站,Domain)进行 jms 及 ejb 或连接池等相关信息进行配置后,实际保存在什么文件中？

保存在此 Domain 的 config.xml 文件中，它是服务器的核心配置文件。

84、说说 weblogic 中一个 Domain 的缺省目录结构?比如要将一个简单的 helloWorld.jsp 放入何目录下,然的在浏览器上就可打入 <http://主机:端口号/helloworld.jsp> 就可以看到运行结果了?又比如这其中用到了一个自己写的 javaBean 该如何办？

Domain 目录服务器目录 applications，将应用目录放在此目录下将可以作为应用访问，如果是 Web 应用，应用目录需要满足 Web 应用目录要求，jsp 文件可以直接放在应用目录中，Javabeen 需要放在应用目录的 WEB-INF 目录的 classes 目录中，设置服务器的缺省应用将可以实现浏览器上无需输入应用名。

85、在 weblogic 中发布 ejb 需涉及到哪些配置文件

不同类型的 EJB 涉及的配置文件不同，都涉及到的配置文件包括 ejb-jar.xml,weblogic-ejb-jar.xmlCMP 实体 Bean 一般还需要 weblogic-cmp-rdbms-jar.xml

86、如何在 weblogic 中进行 ssl 配置与客户端的认证配置或说说 j2ee(标准)进行 ssl 的配置
缺省安装中使用 DemoIdentity.jks 和 DemoTrust.jks KeyStore 实现 SSL，需要配置服务器使用 Enable SSL，配置其端口，在产品模式下需要从 CA 获取私有密钥和数字证书，创建 identity 和 trust keystore，装载获得的密钥和数字证书。可以配置此 SSL 连接是单向还是双向的。

87、如何查看在 weblogic 中已经发布的 EJB?

可以使用管理控制台，在它的 Deployment 中可以查看所有已发布的 EJB

88、CORBA 是什么?用途是什么?

CORBA 标准是公共对象请求代理结构(Common Object Request Broker Architecture)，由对象管理组织 (Object Management Group，缩写为 OMG) 标准化。它的组成是接口定义语言(IDL)，语言绑定(binding;也译为联编)和允许应用程序间互操作的协议。其目的为：用不同的程序设计语言书写在不同的进程中运行，为不同的操作系统开发。

89、说说你所熟悉或听说过的 j2ee 中的几种常用模式?及对设计模式的一些看法

Session Facade Pattern：使用 SessionBean 访问 EntityBean

Message Facade Pattern：实现异步调用

EJB Command Pattern：使用 Command JavaBeans 取代 SessionBean，实现轻量级访问

Data Transfer Object Factory：通过 DTO Factory 简化 EntityBean 数据提供特性

Generic Attribute Access：通过 AttributeAccess 接口简化 EntityBean 数据提供特性

Business Interface：通过远程（本地）接口和 Bean 类实现相同接口规范业务逻辑一致性

EJB 架构的设计好坏将直接影响系统的性能、可扩展性、可维护性、组件可重用性及开发效率。项目越复杂，项目队伍越庞大则越能体现良好设计的重要性。

90、说说在 weblogic 中开发消息 Bean 时的 persistent 与 non-persistent 的差别

persistent 方式的 MDB 可以保证消息传递的可靠性,也就是如果 EJB 容器出现问题而 JMS 服务器依然会将消息在此 MDB 可用的时候发送过来，而 non-persistent 方式的消息将被丢弃。

91、Servlet 执行时一般实现哪几个方法?

```
public void init(ServletConfig config)
```

```
public ServletConfig getServletConfig()
```

```
public String getServletInfo()
```

```
public void service(ServletRequest request,ServletResponse response)
```

```
public void destroy()
```

92、j2ee 常用的设计模式? 说明工厂模式。

Java 中的 23 种设计模式:

Factory (工厂模式), Builder (建造模式), Factory Method (工厂方法模式),

Prototype (原始模型模式), Singleton (单例模式), Facade (门面模式),

Adapter (适配器模式), Bridge (桥梁模式), Composite (合成模式),

Decorator (装饰模式), Flyweight (享元模式), Proxy (代理模式),

Command (命令模式), Interpreter (解释器模式), Visitor (访问者模式),

Iterator (迭代子模式), Mediator (调停者模式), Memento (备忘录模式),

Observer (观察者模式), State (状态模式), Strategy (策略模式),

Template Method (模板方法模式), Chain Of Responsibility (责任链模式)

工厂模式: 工厂模式是一种经常被使用到的模式, 根据工厂模式实现的类可以根据提供的数据生成一组类中某一个类的实例, 通常这一组类有一个公共的抽象父类并且实现了相同的方法, 但是这些方法针对不同的数据进行了不同的操作。首先需要定义一个基类, 该类的子类通过不同的方法实现了基类中的方法。然后需要定义一个工厂类, 工厂类可以根据条件生成不同的子类实例。当得到子类的实例后, 开发人员可以调用基类中的方法而不必考虑到底返回的是哪一个子

类的实例。

93、EJB 需直接实现它的业务接口或 Home 接口吗，请简述理由。

远程接口和 Home 接口不需要直接实现，他们的实现代码是由服务器产生的，程序运行中对应实现类会作为对应接口类型的实例被使用。

94、排序都有哪几种方法？请列举。用 JAVA 实现一个快速排序。

排序的方法有：插入排序（直接插入排序、希尔排序），交换排序（冒泡排序、快速排序），选择排序（直接选择排序、堆排序），归并排序，分配排序（箱排序、基数排序）

快速排序的伪代码。

//使用快速排序方法对 a[0..n-1]排序

从 a[0..n-1]中选择一个元素作为 middle，该元素为支点

把余下的元素分割为两段 left 和 right，使得 left 中的元素都小于等于支点，而 right 中的元素都大于等于支点

递归地使用快速排序方法对 left 进行排序

递归地使用快速排序方法对 right 进行排序

所得结果为 left+middle+right

95、请对以下在 J2EE 中常用的名词进行解释(或简单描述)

web 容器：给处于其中的应用程序组件（JSP，SERVLET）提供一个环境，使 JSP,SERVLET 直接更容器中的环境变量接口交互，不必关注其它系统问题。主要有 WEB 服务器来实现。例如：TOMCAT,WEBLOGIC,WEBSPPHERE 等。该容器提供的接口严格遵守 J2EE 规范中的 WEB APPLICATION 标准。我们把遵守以上标准的 WEB 服务器就叫做 J2EE 中的 WEB 容器。

EJB 容器：Enterprise java bean 容器。更具有行业领域特色。他提供给运行在其中的组件 EJB 各种管理功能。只要满足 J2EE 规范的 EJB 放入 该容器，马上就会被容器进行高效率的管理。并且可以通过现成的接口来获得系统级别的服务。例如邮件服务、事务管理。

JNDI：（Java Naming & Directory Interface）JAVA 命名目录服务。主要提供的功能是：提供一个目录系统，让其它各地的应用程序在其上面留下自己的索引，从而满足快速查找和定位分布式应用程序的功能。

JMS：（Java Message Service）JAVA 消息服务。主要实现各个应用程序之间的通讯。包括点对点和广播。

JTA：（Java Transaction API）JAVA 事务服务。提供各种分布式事务服务。应用程序只需调用其提供的接口即可。

JAF：（Java Action FrameWork）JAVA 安全认证框架。提供一些安全控制方面的框架。让开发者通过各种部署和自定义实现自己的个性安全控制策略。

RMI/IIOP：（Remote Method Invocation /internet 对象请求中介协议）他们主要用于通过远程调用服务。例如，远程有一台计算机上运行一个程序，它提供股票分析服务，我们可以在本地计算机上实现对其直接调用。当然这是要通过一定的规范才能在异构的系统之间进行通信。RMI 是 JAVA 特有的。

96、JAVA 语言如何进行异常处理，关键字：throws,throw,try,catch,finally 分别代表什么意义？在 try 块中可以抛出异常吗？

Java 通过面向对象的方法进行异常处理，把各种不同的异常进行分类，并提供了良好的接口。在 Java 中，每个异常都是一个对象，它是 Throwable 类或其它子类的实例。当一个方法出现异常后便抛出一个异常对象，该对象中包含有异常信息，调用这个方法可以捕获到这个异常并进行处理。Java 的异常处理是通过 5 个关键词来实现的：try、catch、throw、throws 和 finally。一般情况下是用 try 来执行一段程序，如果出现异常，系统会抛 出（throws）一个异常，这时候你可以通过它的类型来捕捉（catch）它，或最后（finally）由缺省处理器来处理。

用 try 来指定一块预防所有"异常"的程序。紧跟在 try 程序后面,应包含一个 catch 子句来指定你想要捕捉的"异常"的类型。

throw 语句用来明确地抛出一个"异常"。

throws 用来标明一个成员函数可能抛出的各种"异常"。

Finally 为确保一段代码不管发生什么"异常"都被执行一段代码。

可以在一个成员函数调用的外面写一个 try 语句,在这个成员函数内部写另一个 try 语句保护其他代码。每当遇到一个 try 语句,"异常"的框架就放到堆栈上面,直到所有的 try 语句都完成。如果下一级的 try 语句没有对某种"异常"进行处理,堆栈就会展开,直到遇到有处理这种"异常"的 try 语句。

97、一个".java"源文件中是否可以包括多个类(不是内部类)?有什么限制?

可以。必须只有一个类名与文件名相同。

98、MVC 的各个部分都有那些技术来实现?如何实现?

MVC 是 Model-View-Controller 的简写。"Model"代表的是应用的业务逻辑(通过 JavaBean, EJB 组件实现), "View" 是应用的表示面(由 JSP 页面产生), "Controller" 是提供应用的处理过程控制(一般是一个 Servlet),通过这种设计模型把应用逻辑,处理过程和显示逻辑分成不同的组件实现。这些组件可以进行交互和重用。

99、java 中有几种方法可以实现一个线程?用什么关键字修饰同步方法? stop()和 suspend()方法为何不推荐使用?

有两种实现方法,分别是继承 Thread 类与实现 Runnable 接口

用 synchronized 关键字修饰同步方法

反对使用 stop(),是因为它不安全。它会解除由线程获取的所有锁定,而且如果对象处于一种不连贯状态,那么其他线程能在那种状态下检查和修改它们。结果很难检查出真正的问题所在。suspend()方法容易发生死锁。调用 suspend()的时候,目标线程会停下来,但却仍然持有在这之前获得的锁定。此时,其他任何线程都不能访问锁定的资源,除非被"挂起"的线程恢复运行。对任何线程来说,如果它们想恢复目标线程,同时又试图使用任何一个锁定的资源,就会造成死锁。所以不应该使用 suspend(),而应在自己的 Thread 类中置入一个标志,指出线程应该活动还是挂起。若标志指出线程应该挂起,便用 wait()命其进入等待状态。若标志指出线程应当恢复,则用一个 notify()重新启动线程。

100、java 中有几种类型的流? JDK 为每种类型的流提供了一些抽象类以供继承,请说出他们分别是哪些类?

字节流,字符流。字节流继承于 InputStream OutputStream,字符流继承于 InputStreamReader OutputStreamWriter。在 java.io 包中还有许多其他的流,主要是为了提高性能和使用方便。

7.16

Java 面试笔试题大汇总(3)

01、java 中会存在内存泄漏吗,请简单描述。

会。如: int i,i2; return (i-i2); //when i 为足够大的正数,i2 为足够大的负数。结果会造成溢位,导致错误。

102、java 中实现多态的机制是什么?

方法的重写 Overriding 和重载 Overloading 是 Java 多态性的不同表现。重写 Overriding 是父类与子类之间多态性的一种表现,重载 Overloading 是一个类中多态性的一种表现。

103、垃圾回收器的基本原理是什么?垃圾回收器可以马上回收内存吗?有什么办法主动通知虚拟机进行垃圾回收?

对于 GC 来说,当程序员创建对象时,GC 就开始监控这个对象的地址、大小以及使用情况。

通常，GC 采用有向图的方式记录和管理堆(heap)中的所有对象。通过这种方式确定哪些对象是"可达的"，哪些对象是"不可达的"。当GC 确定一些对象为"不可达"时，GC 就有责任回收这些内存空间。可以。程序员可以手动执行 System.gc()，通知 GC 运行，但是 Java 语言规范并不保证 GC 一定会执行。

104、静态变量和实例变量的区别？

```
static i = 10; //常量
```

```
class A a; a.i = 10; //可变
```

105、什么是 java 序列化，如何实现 java 序列化？

序列化就是一种用来处理对象流的机制，所谓对象流也就是将对象的内容进行流化。可以对流化后的对象进行读写操作，也可将流化后的对象传输于网络之间。序列化是为了解决在对对象流进行读写操作时所引发的问题。

序列化的实现：将需要被序列化的类实现 Serializable 接口，该接口没有需要实现的方法，implements Serializable 只是为了标注该对象是可被序列化的，然后使用一个输出流(如：FileOutputStream)来构造一个 ObjectOutputStream(对象流)对象，接着，使用 ObjectOutputStream 对象的 writeObject(Object obj)方法就可以将参数为 obj 的对象写出(即保存其状态)，要恢复的话则用输入流。

106、是否可以从一个 static 方法内部发出对非 static 方法的调用？

不可以,如果其中包含对象的 method(); 不能保证对象初始化.

107、写 clone()方法时，通常都有一行代码，是什么？

Clone 有缺省行为，super.clone(); 他负责产生正确大小的空间，并逐位复制。

108、在 JAVA 中，如何跳出当前的多重嵌套循环？

用 break; return 方法。

109、List、Map、Set 三个接口，存取元素时，各有什么特点？

List 以特定次序来持有元素，可有重复元素。Set 无法拥有重复元素,内部排序。Map 保存 key-value 值，value 可多值。

110、J2EE 是什么？

J2EE 是 Sun 公司提出的多层(multi-tiered),分布式(distributed),基于组件(component-base)的企业级应用模型 (enterprise application model).在这样的一个应用系统中，可按照功能划分为不同的组件，这些组件又可在不同计算机上，并且处于相应的层次(tier)中。所属层次包括客户层(client tier)组件,web 层和组件,Business 层和组件,企业信息系统 (EIS)层。

111、UML 方面

标准建模语言 UML。用例图,静态图(包括类图、对象图和包图),行为图,交互图(顺序图,合作图),实现图。

112、说出一些常用的类，包，接口，请各举 5 个

常用的类：BufferedReader BufferedWriter FileReader FileWriter String Integer

常用的包：java.lang java.awt java.io java.util java.sql

常用的接口：Remote List Map Document NodeList

113、开发中都用到那些设计模式?用在什么场合？

每个模式都描述了一个在我们的环境中不断出现的问题，然后描述了该问题的解决方案的核心。通过这种方式，你可以无数次地使用那些已有的解决方案，无需在重复相同的工作。主要用到了 MVC 的设计模式。用来开发 JSP/Servlet 或者 J2EE 的相关应用。简单工厂模式等。

114、jsp 有哪些动作?作用分别是什么？

JSP 共有以下 6 种基本动作 jsp:include: 在页面被请求的时候引入一个文件。jsp:useBean: 寻找或者实例化一个 JavaBean。jsp:setProperty: 设置 JavaBean 的属性。jsp:getProperty:

输出某个 JavaBean 的属性。jsp: forward: 把请求转到一个新的页面。jsp:plugin: 根据浏览器类型为 Java 插件生成 OBJECT 或 EMBED 标记。

115、Anonymous Inner Class (匿名内部类) 是否可以 extends(继承) 其它类, 是否可以 implements(实现) interface(接口)?

可以继承其他类或完成其他接口, 在 swing 编程中常用此方式。

116、应用服务器与 WEB SERVER 的区别?

应用服务器: Weblogic、Tomcat、Jboss

WEB SERVER: IIS、Apache

117、BS 与 CS 的联系与区别。

C/S 是 Client/Server 的缩写。服务器通常采用高性能的 PC、工作站或小型机, 并采用大型数据库系统, 如 Oracle、Sybase、Informix 或 SQL Server。客户端需要安装专用的客户端软件。

B/S 是 Browser/Server 的缩写, 客户机上只要安装一个浏览器 (Browser), 如 Netscape Navigator 或 Internet Explorer, 服务器安装 Oracle、Sybase、Informix 或 SQL Server 等数据库。在这种结构下, 用户界面完全通过 WWW 浏览器实现, 一部分事务逻辑在前端实现, 但是主要事务逻辑在服务器端实现。浏览器通过 Web Server 同数据库进行数据交互。

C/S 与 B/S 区别:

1. 硬件环境不同:

C/S 一般建立在专用的网络上, 小范围里的网络环境, 局域网之间再通过专门服务器提供连接和数据交换服务。

B/S 建立在广域网之上的, 不必是专门的网络硬件环境, 例与电话上网, 租用设备. 信息自己管理. 有比 C/S 更强的适应范围, 一般只要有操作系统和浏览器就行

2. 对安全要求不同

C/S 一般面向相对固定的用户群, 对信息安全的控制能力很强. 一般高度机密的信息系统采用 C/S 结构适宜. 可以通过 B/S 发布部分可公开信息.

B/S 建立在广域网之上, 对安全的控制能力相对弱, 可能面向不可知的用户。

3. 对程序架构不同

C/S 程序可以更加注重流程, 可以对权限多层次校验, 对系统运行速度可以较少考虑.

B/S 对安全以及访问速度的多重的考虑, 建立在需要更加优化的基础之上. 比 C/S 有更高的要求 B/S 结构的程序架构是发展的趋势, 从 MS 的 .Net 系列的 BizTalk 2000 Exchange 2000 等, 全面支持网络的构件搭建的系统. SUN 和 IBM 推的 JavaBean 构件 技术等, 使 B/S 更加成熟.

4. 软件重用不同

C/S 程序可以不可避免的整体性考虑, 构件的重用性不如在 B/S 要求下的构件的重用性好.

B/S 对的多重结构, 要求构件相对独立的功能. 能够相对较好的重用. 就入买来的餐桌可以再利用, 而不是做在墙上的石头桌子

5. 系统维护不同

C/S 程序由于整体性, 必须整体考察, 处理出现的问题以及系统升级. 升级难. 可能是再做一个全新的系统

B/S 构件组成, 方面构件个别的更换, 实现系统的无缝升级. 系统维护开销减到最小. 用户从网上自己下载安装就可以实现升级.

6. 处理问题不同

C/S 程序可以处理用户面固定, 并且在相同区域, 安全要求高需求, 与操作系统相关. 应该都是相同的系统

B/S 建立在广域网上, 面向不同的用户群, 分散地域, 这是 C/S 无法作到的. 与操作系统平台关系最小.

7. 用户接口不同

C/S 多是建立在 Window 平台上, 表现方法有限, 对程序员普遍要求较高

B/S 建立在浏览器上, 有更加丰富和生动的表现方式与用户交流. 并且大部分难度减低, 减低开发成本.

8. 信息流不同

C/S 程序一般是典型的中央集权的机械式处理, 交互性相对低

B/S 信息流向可变化, B-B B-C B-G 等信息、流向的变化, 更像交易中心。

118、Linux 下线程, GDI 类的解释。

Linux 实现的就是基于核心轻量级进程的 "一对一" 线程模型, 一个线程实体对应一个核心轻量级进程, 而线程之间的管理在核外函数库中实现。

GDI 类为图像设备编程接口类库。

119、Struts 的应用(如 Struts 架构)

Struts 是采用 Java Servlet/JavaServer Pages 技术, 开发 Web 应用程序的开放源码的 framework。采用 Struts 能开发出 基于 MVC(Model-View-Controller) 设计模式的应用构架。

Struts 有如下的主要功能: 一. 包含一个 controller servlet, 能将用户的请求发送到相应的 Action 对象。二. JSP 自由 tag 库, 并且在 controller servlet 中提供关联支持, 帮助开发员创建交互式表单应用。三. 提供了一系列实用对象: XML 处理、通过 Java reflection APIs 自动处理 JavaBeans 属性、国际化的提示和消息。

120、JDO 是什么?

JDO 是 Java 对象持久化的新的规范, 为 java data object 的简称, 也是一个用于存取某种数据仓库中的对象的标准 API。JDO 提供了透明的对象存储, 因此对开发人员来说, 存储数据对象完全不需要额外的代码(如 JDBC API 的使用)。这些繁琐的例行工作已经转移到 JDO 产品提供商身上, 使开发人员解脱出来, 从而集中时间和精力在业务逻辑上。另外, JDO 很灵活, 因为它可以在任何数据底层上运行。JDBC 只是面向关系数据库(RDBMS) JDO 更通用, 提供到任何数据底层的存储功能, 比如关系数据库、文件、XML 以及对象数据库(ODBMS) 等等, 使得应用可移植性更强。

121、内部类可以引用他包含类的成员吗? 有没有什么限制?

一个内部类对象可以访问创建它的外部类对象的内容

122、Web Service 名词解释。JSDDL 开发包的介绍。JAXP、JAXM 的解释。SOAP、UDDI、WSDL 解释。

Web Service Web Service 是基于网络的、分布式的模块化组件, 它执行特定的任务, 遵守具体的技术规范, 这些规范使得 Web Service 能与其他兼容的组件进行互操作。

JAXP(Java API for XML Parsing) 定义了使用 DOM, SAX, XSLT 的通用的接口。这样在你的程序中你只要使用这些通用的接口, 当你需要改变具体的实现时候也不需要修改代码。

JAXM(Java API for XML Messaging) 是为 SOAP 通信提供访问方法和传输机制的 API。

WSDL 是一种 XML 格式, 用于将网络服务描述为一组端点, 这些端点对包含面向文档信息或面向过程信息的信息进行操作。这种格式首先对操作和信息进行抽象描述, 然后将其绑定到具体的网络协议和信息格式上以定义端点。相关的具体端点即组合成为抽象端点(服务)。SOAP 即简单对象访问协议(Simple Object Access Protocol), 它是用于交换 XML 编码信息的轻量级协议。

UDDI 的目的是为电子商务建立标准; UDDI 是一套基于 Web 的、分布式的、为 Web Service

提供的、信息注册中心的实现标准规范，同时也包含一组使企业能将自身提供的 Web Service 注册，以使别的企业能够发现的访问协议的实现标准。

7.17

JAVA 代码查错

1.

```
abstract class Name {  
    private String name;  
    public abstract boolean isStupidName(String name) {}  
}
```

大侠们，这有何错误？

答案：错。abstract method 必须以分号结尾，且不带花括号。

2.

```
public class Something {  
    void doSomething () {  
        private String s = "";  
        int l = s.length();  
    }  
}
```

有错吗？

答案：错。局部变量前不能放置任何访问修饰符 (private, public, 和 protected)。final 可以用来修饰局部变量

(final 如同 abstract 和 strictfp, 都是非访问修饰符, strictfp 只能修饰 class 和 method 而非 variable)。

3.

```
abstract class Something {  
    private abstract String doSomething ();  
}
```

这好像没什么错吧？

答案：错。abstract 的 methods 不能以 private 修饰。abstract 的 methods 就是让子类 implement(实现)具体细节的，怎么可以用 private 把 abstract method 封锁起来呢？(同理，abstract method 前不能加 final)。

4.

```
public class Something {  
    public int addOne(final int x) {  
        return ++x;  
    }  
}
```

这个比较明显。

答案：错。int x 被修饰成 final，意味着 x 不能在 addOne method 中被修改。

5.

```
public class Something {  
    public static void main(String[] args) {  
        Other o = new Other();  
        new Something().addOne(o);  
    }  
}
```

```

}
public void addOne(final Other o) {
    o.i++;
}
}
class Other {
    public int i;
}

```

和上面的很相似，都是关于 **final** 的问题，这有错吗？

答案: 正确。在 `addOne` method 中，参数 `o` 被修饰成 **final**。如果在 `addOne` method 里我们修改了 `o` 的 **reference**

(比如: `o = new Other();`)，那么如同上例这题也是错的。但这里修改的是 `o` 的 **member variable** (成员变量)，而 `o` 的 **reference** 并没有改变。

6.

```

class Something {
    int i;
    public void doSomething() {
        System.out.println("i = " + i);
    }
}

```

有什么错呢？看不出来啊。

答案: 正确。输出的是 `"i = 0"`。`int i` 属于 **instant variable** (实例变量，或叫成员变量)。**instant variable** 有 **default value**。`int` 的 **default value** 是 `0`。

7.

```

class Something {
    final int i;
    public void doSomething() {
        System.out.println("i = " + i);
    }
}

```

和上面一题只有一个地方不同，就是多了一个 **final**。这难道就错了吗？

答 案: 错。`final int i` 是个 **final** 的 **instant variable** (实例变量，或叫成员变量)。**final** 的 **instant variable** 没有 **default value**，必须在 **constructor** (构造器) 结束之前被赋予一个明确的值。可以修改为 `"final int i = 0;"`。

8.

```

public class Something {
    public static void main(String[] args) {
        Something s = new Something();
        System.out.println("s.doSomething() returns " + doSomething());
    }
    public String doSomething() {
        return "Do something ...";
    }
}

```


看上去很完美。

答案: 错。看上去在 main 里 call doSomething 没有什么问题, 毕竟两个 methods 都在同一个 class 里。但仔细看, main 是 static 的。static method 不能直接 call non-static methods。可改成 "System.out.println ("s.doSomething() returns " + s.doSomething());"。同理, static method 不能访问 non -static instant variable。

9.

此处, Something 类的文件名叫 OtherThing.java

```
class Something {
    private static void main(String[] something_to_do) {
        System.out.println("Do something ...");
    }
}
```

这个好像很明显。

答案: 正确。从来没有人说过 Java 的 Class 名字必须和其文件名相同。但 public class 的名字必须和文件名相同。

10.

```
interface A {
    int x = 0;
}
class B {
    int x = 1;
}
class C extends B implements A {
    public void pX() {
        System.out.println(x);
    }
    public static void main(String[] args) {
        new C().pX();
    }
}
```

答案: 错误。在编译时会发生错误(错误描述不同的 JVM 有不同的信息, 意思就是未明确的 x 调用, 两个 x 都匹配(就象在同时 import java.util 和 java.sql 两个包时直接声明 Date 一样)。对于父类的变量, 可以用 super.x 来明确, 而接口的属性默认隐含为 public static final。所以可以通过 A.x 来明确。

11.

```
interface Playable {
    void play();
}
interface Bounceable {
    void play();
}
interface Rollable extends Playable, Bounceable {
    Ball ball = new Ball("PingPang");
}
```

```

class Ball implements Rollable {
private String name;
public String getName() {
return name;
}
public Ball(String name) {
this.name = name;
}
public void play() {
ball = new Ball("Football");
System.out.println(ball.getName());
}
}

```

这个错误不容易发现。

答 案: 错。"interface Rollable extends Playable, Bounceable"没有问题。interface 可继承多个 interfaces, 所以这里没错。问题出在 interface Rollable 里的 "Ball ball = new Ball ("PingPang");"。任何在 interface 里声明的 interface variable (接口变量, 也可称成员变量), 默认为 public static final。也就是说 "Ball ball = new Ball("PingPang");"实际上是 "public static final Ball ball = new Ball("PingPang");"。在 Ball 类的 Play()方法 中, "ball = new Ball("Football");"改变了 ball 的 reference, 而这里的 ball 来自 Rollable interface, Rollable interface里的 ball 是 public static final的, final 的 object 是不能被改变 reference 的。因此编译器将在 "ball = new Ball("Football");"这里显示有错。

7.18

JAVA 编程题

1. 现在输入 n 个数字, 以逗号, 分开; 然后可选择升或者降序排序; 按提交键就在另一页面显示按什么排序, 结果为, 提供 reset

```

import java.util.*;
public class bycomma{
public static String[] splitStringByComma(String source){
if(source==null||source.trim().equals(""))
return null;
StringTokenizer commaToker = new StringTokenizer(source,",");
String[] result = new String[commaToker.countTokens()];
int i=0;
while(commaToker.hasMoreTokens()){
result[i] = commaToker.nextToken();
i++;
}
return result;
}
public static void main(String args[]){
String[] s = splitStringByComma("5,8,7,4,3,9,1");
int[] ii = new int[s.length];
for(int i = 0; i<ii.length; i++) ii[i] = Integer.parseInt(s[i]);
}
}

```

```

}
Arrays.sort(ii);
//asc
for(int i=0;i<ii.length;i++){
    System.out.println(ii[i]);
}
//desc
for(int i=ii.length-1;i>=0;i--){
    System.out.println(ii[i]);
}
}
}
}

```

2. 金额转换，阿拉伯数字的金额转换成中国传统的形式如：（¥1011）—>（一千零一拾一元整）输出。

```

package test.format;
import java.text.NumberFormat;
import java.util.HashMap;
public class SimpleMoneyFormat {
    public static final String EMPTY = "";
    public static final String ZERO = "零";
    public static final String ONE = "壹";
    public static final String TWO = "贰";
    public static final String THREE = "叁";
    public static final String FOUR = "肆";
    public static final String FIVE = "伍";
    public static final String SIX = "陆";
    public static final String SEVEN = "柒";
    public static final String EIGHT = "捌";
    public static final String NINE = "玖";
    public static final String TEN = "拾";
    public static final String HUNDRED = "佰";
    public static final String THOUSAND = "仟";
    public static final String TEN_THOUSAND = "万";
    public static final String HUNDRED_MILLION = "亿";
    public static final String YUAN = "元";
    public static final String JIAO = "角";
    public static final String FEN = "分";
    public static final String DOT = ".";

    private static SimpleMoneyFormat formatter = null;
    private HashMap chineseNumberMap = new HashMap();
    private HashMap chineseMoneyPattern = new HashMap();
    private NumberFormat numberFormat = NumberFormat.getInstance();

    private SimpleMoneyFormat() {

```

```
numberFormat.setMaximumFractionDigits(4);
numberFormat.setMinimumFractionDigits(2);
numberFormat.setGroupingUsed(false);
```

```
chineseNumberMap.put("0", ZERO);
chineseNumberMap.put("1", ONE);
chineseNumberMap.put("2", TWO);
chineseNumberMap.put("3", THREE);
chineseNumberMap.put("4", FOUR);
chineseNumberMap.put("5", FIVE);
chineseNumberMap.put("6", SIX);
chineseNumberMap.put("7", SEVEN);
chineseNumberMap.put("8", EIGHT);
chineseNumberMap.put("9", NINE);
chineseNumberMap.put(DOT, DOT);
```

```
chineseMoneyPattern.put("1", TEN);
chineseMoneyPattern.put("2", HUNDRED);
chineseMoneyPattern.put("3", THOUSAND);
chineseMoneyPattern.put("4", TEN_THOUSAND);
chineseMoneyPattern.put("5", TEN);
chineseMoneyPattern.put("6", HUNDRED);
chineseMoneyPattern.put("7", THOUSAND);
chineseMoneyPattern.put("8", HUNDRED_MILLION);
}
```

```
public static SimpleMoneyFormat getInstance() {
    if (formatter == null)
        formatter = new SimpleMoneyFormat();
    return formatter;
}
```

```
public String format(String moneyStr) {
    checkPrecision(moneyStr);
    String result;
    result = convertToChineseNumber(moneyStr);
    result = addUnitsToChineseMoneyString(result);
    return result;
}
```

```
public String format(double moneyDouble) {
    return format(numberFormat.format(moneyDouble));
}
```

```

public String format(int moneyInt) {
    return format(numberFormat.format(moneyInt));
}

public String format(long moneyLong) {
    return format(numberFormat.format(moneyLong));
}

public String format(Number moneyNum) {
    return format(numberFormat.format(moneyNum));
}

private String convertToChineseNumber(String moneyStr) {
    String result;
    StringBuffer cMoneyStringBuffer = new StringBuffer();
    for (int i = 0; i < moneyStr.length(); i++) {
        cMoneyStringBuffer.append(chineseNumberMap.get(moneyStr.substring(i, i + 1)));
    }
    //拾佰仟万亿等都是汉字里面才有的单位，加上它们
    int indexOfDot = cMoneyStringBuffer.indexOf(DOT);
    int moneyPatternCursor = 1;
    for (int i = indexOfDot - 1; i > 0; i--) {
        cMoneyStringBuffer.insert(i, chineseMoneyPattern.get(EMPTY + moneyPatternCursor));
        moneyPatternCursor = moneyPatternCursor == 8 ? 1 : moneyPatternCursor + 1;
    }

    String fractionPart = cMoneyStringBuffer.substring(cMoneyStringBuffer.indexOf("."));
    cMoneyStringBuffer.delete(cMoneyStringBuffer.indexOf("."), cMoneyStringBuffer.length());
    while (cMoneyStringBuffer.indexOf("零拾") != -1) {
        cMoneyStringBuffer.replace(cMoneyStringBuffer.indexOf("零拾"), cMoneyStringBuffer.indexOf("零拾") + 2, ZERO);
    }
    while (cMoneyStringBuffer.indexOf("零佰") != -1) {
        cMoneyStringBuffer.replace(cMoneyStringBuffer.indexOf("零佰"), cMoneyStringBuffer.indexOf("零佰") + 2, ZERO);
    }
    while (cMoneyStringBuffer.indexOf("零仟") != -1) {
        cMoneyStringBuffer.replace(cMoneyStringBuffer.indexOf("零仟"), cMoneyStringBuffer.indexOf("零仟") + 2, ZERO);
    }
    while (cMoneyStringBuffer.indexOf("零万") != -1) {
        cMoneyStringBuffer.replace(cMoneyStringBuffer.indexOf("零万"), cMoneyStringBuffer.indexOf("零万") + 2, TEN_THOUSAND);
    }
}

```

```

while (cMoneyStringBuffer.indexOf("零亿") != -1) {
cMoneyStringBuffer.replace(cMoneyStringBuffer.indexOf("零亿 "), cMoneyStringBuffer.indexOf("零
亿") + 2, HUNDRED_MILLION);
}
while (cMoneyStringBuffer.indexOf("零零") != -1) {
cMoneyStringBuffer.replace(cMoneyStringBuffer.indexOf("零零 "), cMoneyStringBuffer.indexOf("零
零") + 2, ZERO);
}
if (cMoneyStringBuffer.lastIndexOf(ZERO) == cMoneyStringBuffer.length() - 1)
cMoneyStringBuffer.delete(cMoneyStringBuffer.length() - 1, cMoneyStringBuffer.length());
cMoneyStringBuffer.append(fractionPart);

result = cMoneyStringBuffer.toString();
return result;
}

private String addUnitsToChineseMoneyString(String moneyStr) {
String result;
StringBuffer cMoneyStringBuffer = new StringBuffer(moneyStr);
int indexOfDot = cMoneyStringBuffer.indexOf(DOT);
cMoneyStringBuffer.replace(indexOfDot, indexOfDot + 1, YUAN);

```

J2EE 面试笔试题大汇

1、 一个“.java”源文件中是否可以包括多个类（不是内部类）？有什么限制？

可以。必须只有一个类名与文件名相同。

2、MVC的各个部分都有那些技术来实现?如何实现?

MVC 是 Model—View—Controller的简写。“Model”代表的是应用的业务逻辑（通过JavaBean, EJB组件实现），“View”是应用的表示面（由JSP页面产生），“Controller”是提供应用的处理过程控制（一般是一个Servlet），通过这种设计 模型把应用逻辑，处理过程和显示逻辑分成不同的组件实现。这些组件可以进行交互和重用。

3、java中有几种方法可以实现一个线程？用什么关键字修饰同步方法？stop()和suspend()方法为何不推荐使用？

有两种实现方法，分别是继承Thread类与实现Runnable接口

用synchronized关键字修饰同步方法

反对使用stop()，是因为它不安全。它会解除由线程获取的所有锁定，而且如果对象处于一种不连贯状态，那么其他线程能在那种状态下检查和修改它们。结果 很难检查出真正的问题所在。suspend()方法容易发生死锁。调用suspend()的时候，目标线程会停下来，但却仍然持有在这之前获得的锁定。此时，其他任何线程都不能访问锁定的资源，除非被“挂起”的线程恢复运行。对任何线程来说，如果它们想恢复目标线程，同时又试图使用任何一个锁定的资源，就会造成死锁。所以不应该使用suspend()，而应在自己的Thread类中置入一个标志，指出线程应该活动还是挂起。若标志指出线程应该挂起，便用wait()命其进入等待状态。若标志指出线程应当恢复，则用一个notify()重新启动线程。

4、java中有几种类型的流？JDK为每种类型的流提供了一些抽象类以供继承，请说出他们分别是哪些类？

字节流，字符流。字节流继承于InputStream OutputStream，字符流继承于InputStreamReader OutputStreamWriter。在java.io包中还有许多其他的流，主要是为了提高性能和使用方便。

5、java中会存在内存泄漏吗，请简单描述。

会。如：int i, i2; return (i-i2); //when i 为足够大的正数, i2为足够大的负数。结果会造成溢位，导致错误。

6、java中实现多态的机制是什么？

方法的重写Overriding和重载Overloading是Java多态性的不同表现。重写Overriding是父类与子类之间多态性的一种表现，重载Overloading是一个类中多态性的一种表现。

7、垃圾回收器的基本原理是什么？垃圾回收器可以马上回收内存吗？有什么办法主动通知虚拟机进行垃圾回收？

对于GC来说，当程序员创建对象时，GC就开始监控这个对象的地址、大小以及使用情

况。通常，GC采用有向图的方式记录和管理堆(heap)中的所有对象。通过这种方式确定哪些对象是“可达的”，哪些对象是“不可达的”。当GC确定一些对象为“不可达”时，GC就有责任回收这些内存空间。可以。程序员可以手动执行System.gc()，通知GC运行，但是Java语言规范并不保证GC一定会执行。

8、静态变量和实例变量的区别？

```
static i = 10; //常量
    class A a;    a.i =10;//可变
```

9、什么是 java 序列化，如何实现 java 序列化？

序列化就是一种用来处理对象流的机制，所谓对象流也就是将对象的内容进行流化。可以对流化后的对象进行读写操作，也可将流化后的对象传输于网络之间。序列化是为了解决在对对象流进行读写操作时所引发的问题。

序列化的实现：将需要被序列化的类实现Serializable接口，该接口没有需要实现的方法，implements Serializable只是为了标注该对象是可被序列化的，然后使用一个输出流(如：FileOutputStream)来构造一个ObjectOutputStream(对象流)对象，接着，使用ObjectOutputStream对象的writeObject(Object obj)方法就可以将参数为obj的对象写出(即保存其状态)，要恢复的话则用输入流。

10、是否可以从一个 static 方法内部发出对非 static 方法的调用？

不可以，如果其中包含对象的method()；不能保证对象初始化。

11、写 clone() 方法时，通常都有一行代码，是什么？

Clone 有缺省行为，super.clone()；他负责产生正确大小的空间，并逐位复制。

12、在 JAVA 中，如何跳出当前的多重嵌套循环？

用 break; return 方法。

13、List、Map、Set 三个接口，存取元素时，各有什么特点？

List 以特定次序来持有元素，可有重复元素。Set 无法拥有重复元素，内部排序。Map 保存 key-value 值，value 可多值。

14、J2EE 是什么？

J2EE 是 Sun 公司提出的多层(multi-tiered)，分布式(distributed)，基于组件(component-base)的企业级应用模型 (enterprise application model)。在这样的一个应用系统中，可按照功能划分为不同的组件，这些组件又可在不同计算机上，并且处于相应的层次(tier)中。所属层次包括客户层(client tier)组件，web 层和组件，Business 层和组件，企业信息系统 (EIS) 层。

15、UML 方面

标准建模语言 UML。用例图，静态图(包括类图、对象图和包图)，行为图，交互图(顺序图，合作图)，实现图。

16、说出一些常用的类，包，接口，请各举 5 个

常用的类: `BufferedReader` `BufferedWriter` `FileReader` `FileWirtter` `String`
 `Integer`

常用的包: `java.lang` `java.awt` `java.io` `java.util` `java.sql`

常用的接口: `Remote` `List` `Map` `Document` `NodeList`

17、开发中都用到那些设计模式?用在什么场合?

每个模式都描述了一个在我们的环境中不断出现的问题,然后描述了该问题的解决方案的核心。通过这种方式,你可以无数次地使用那些已有的解决方案,无需在重复相同的工作。主要用到了MVC的设计模式。用来开发JSP/Servlet或者J2EE的相关应用。简单工厂模式等。

18、jsp有哪些动作?作用分别是什么?

JSP 共有以下6种基本动作 `jsp:include`: 在页面被请求的时候引入一个文件。 `jsp:useBean`: 寻找或者实例化一个JavaBean。 `jsp:setProperty`: 设置JavaBean的属性。

`jsp:getProperty`: 输出某个JavaBean的属性。 `jsp:forward`: 把请求转到一个新的页面。 `jsp:plugin`: 根据浏览器类型为Java插件生成OBJECT或EMBED标记。

17、Anonymous Inner Class (匿名内部类) 是否可以 extends (继承) 其它类, 是否可以 implements (实现) interface (接口)?

可以继承其他类或完成其他接口, 在 swing 编程中常用此方式。

18、应用服务器与 WEB SERVER 的区别?

应用服务器: `Weblogic`、`Tomcat`、`Jboss`

WEB SERVER: `IIS`、`Apache`

19、BS 与 CS 的联系与区别。

C/S 是 Client/Server 的缩写。服务器通常采用高性能的 PC、工作站或小型机, 并采用大型数据库系统, 如 `Oracle`、`Sybase`、`Informix` 或 `SQL Server`。客户端需要安装专用的客户端软件。

B/S 是 Brower/Server 的缩写, 客户机上只要安装一个浏览器 (Browser), 如 `Netscape Navigator` 或 `Internet Explorer`, 服务器安装 `Oracle`、`Sybase`、`Informix` 或 `SQL Server` 等数据库。在这种结构下, 用户界面完全通过 WWW 浏览器实现, 一部分事务逻辑在前端实现, 但是主要事务逻辑在服务器端实现。浏览器通过 Web Server 同数据库进行数据交互。

C/S 与 B/S 区别:

1. 硬件环境不同:

C/S 一般建立在专用的网络上, 小范围里的网络环境, 局域网之间再通过专门服务器提供连接和数据交换服务。

B/S 建立在广域网之上的, 不必是专门的网络硬件环境, 例与电话上网, 租用设备。信息自己管理。有比 C/S 更强的适应范围, 一般只要有操作系统和浏览器就行

2. 对安全要求不同

C/S 一般面向相对固定的用户群, 对信息安全的控制能力很强。一般高度机密的信息系统采用 C/S 结构适宜。可以通过 B/S 发布部分可公开信息。

B/S 建立在广域网之上, 对安全的控制能力相对弱, 可能面向不可知的用户。

3. 对程序架构不同

C/S 程序可以更加注重流程, 可以对权限多层次校验, 对系统运行速度可以较少考虑。

B/S 对安全以及访问速度的多重的考虑, 建立在需要更加优化的基础之上。比 C/S 有更高的要求 B/S 结构的程序架构是发展的趋势, 从 MS 的. Net 系列的 BizTalk 2000 Exchange 2000 等, 全面支持网络的构件搭建的系统。SUN 和 IBM 推的 JavaBean 构件 技术等, 使 B/S 更加成熟。

4. 软件重用不同

C/S 程序可以不可避免的整体性考虑, 构件的重用性不如在 B/S 要求下的构件的重用性好。

B/S 对的多重结构, 要求构件相对独立的功能。能够相对较好的重用。就买来的餐桌可以再利用, 而不是做在墙上的石头桌子

5. 系统维护不同

C/S 程序由于整体性, 必须整体考察, 处理出现的问题以及系统升级。升级难。可能是再做一个全新的系统

B/S 构件组成, 方面构件个别的更换, 实现系统的无缝升级。系统维护开销减到最小。用户从网上自己下载安装就可以实现升级。

6. 处理问题不同

C/S 程序可以处理用户面固定, 并且在相同区域, 安全要求高需求, 与操作系统相关。应该都是相同的系统

B/S 建立在广域网上, 面向不同的用户群, 分散地域, 这是 C/S 无法作到的。与操作系统平台关系最小。

7. 用户接口不同

C/S 多是建立的 Window 平台上, 表现方法有限, 对程序员普遍要求较高

B/S 建立在浏览器上, 有更加丰富和生动的表现方式与用户交流。并且大部分难度减低, 减低开发成本。

8. 信息流不同

C/S 程序一般是典型的中央集权的机械式处理, 交互性相对低

B/S 信息流向可变化, B-B B-C B-G 等信息、流向的变化, 更像交易中心。

20、Linux 下线程, GDI 类的解释。

Linux 实现的就是基于核心轻量级进程的“一对一”线程模型, 一个线程实体对应一个核心轻量级进程, 而线程之间的管理在核外函数库中实现。

GDI 类为图像设备编程接口类库。

21、Struts 的应用 (如 Struts 架构)

Struts 是采用 Java Servlet/JavaServer Pages 技术, 开发 Web 应用程序的开源码的 framework。采用 Struts 能开发出 基于 MVC (Model-View-Controller) 设计模式的应用构架。Struts 有如下的主要功能: 一. 包含一个 controller servlet, 能将用户的请求发送到相应的 Action 对象。二. JSP 自由 tag 库, 并且在 controller servlet 中提供关联支持, 帮助开发员创建交互式表单应用。三. 提供了一系列实用对象: XML 处理、通过 Java reflection APIs 自动处理 JavaBeans 属性、国际化的提示和消息。

22、Jdo 是什么?

JD0是 Java对象持久化的新的规范,为 java data object的简称,也是一个用于存取某种数据仓库中的对象的标准化 API。JD0提供了透明的对象存储,因此对开发人员来说,存储数据对象完全不需要额外的代码(如 JDBC API的使用)。这些繁琐的例行工作已经转移到 JD0产品提供商身上,使开发人员解脱出来,从而集中时间和精力在业务逻辑上。另外,JD0很灵活,因为它可以在任何数据底层上运行。JDBC只是面向关系数据库(RDBMS) JD0更通用,提供到任何数据底层的存储功能,比如关系数据库、文件、XML以及对象数据库(ODBMS)等等,使得应用可移植性更强。

23、内部类可以引用他包含类的成员吗?有没有什么限制?

一个内部类对象可以访问创建它的外部类对象的内容

24、WEB SERVICE 名词解释。JSWDL开发包的介绍。JAXP、JAXM的解释。SOAP、UDDI,WSDL 解释。

Web ServiceWeb Service是基于网络的、分布式的模块化组件,它执行特定的任务,遵守具体的技术规范,这些规范使得 Web Service 能与其他兼容的组件进行互操作。

JAXP(Java API for XML Parsing) 定义了 Java 中使用 DOM, SAX, XSLT 的通用的接口。这样在你的程序中你只要使用这些通用的接口,当你需要改变具体的实现时候也不需要修改代码。

JAXM(Java API for XML Messaging) 是为 SOAP通信提供访问方法和传输机制的 API。WSDL是一种 XML 格式,用于将网络服务描述为一组端点,这些端点对包含面向文档信息或面向过程信息的信息进行操作。这种格式首先对操作和信息进行抽象描述,然后将其绑定到具体的网络协议和消息格式上以定义端点。相关的具体端点即组合成为抽象端点(服务)。

SOAP即简单对象访问协议(Simple Object Access Protocol),它是用于交换 XML编码信息的轻量级协议。

UDDI 的目的是为电子商务建立标准;UDDI是一套基于Web的、分布式的、为Web Service提供的、信息注册中心的实现标准规范,同时也包含一组使企业能将自身提供的 Web Service注册,以使别的企业能够发现的访问协议的实现标准。

7.20

java 面试基础题

以下题目需要当场编写,并有本人讲解

- 1、 写一个通用方法把任何一个类似“abcdefg”的字符串倒叙排。
- 2、 把字符数组String[] str={"6","5","3","1","2","4"}中的字符数字按照从小到大调整位置。

两种方法

- 3、 用杨辉三角规律打一个数字三角型。
- 4、 把List容器中的数字元素“6” ” 5” ” 3” ” 1” ” 2” ” 4”，按照从大到小排序进行重新存放。
- 5、 有一个接口，接口中有一个抽象的方法，然后一个普通的类 implements 该接口，但该类并不自己去实现从接口中继承的抽象方法，如何来完成，举个例子。
- 6、 写一个通用的方法，该方法有一个参数，当给该方法传递一个浮点数字，该方法进行四舍五入的保留两位小数的处理，然后在返回处理之后的浮点数字，两种方法实现。
- 7、 单态（单实例）模式类，两种实现方法实现，并当场解释为什么？
- 8、 有一个接口和接口的子类，接口中有一个 print 的方法，子类实现该方法，而且子类中有一个自己声明的方法 make，编程运用上行机制调用子类中的方法 print,同时解释在上行时能否调用 make,为什么？
- 9、 面向对象实现表现在哪些方面，举例说明，并解释面向对象的这些方面对开发有什么意义。
- 10、 对一个 Map 类型的容器中元素“6” ” 5” ” 3” ” 1” ” 2” ” 4” 进行排序。
- 11、 对一个 Map 类型的容器中元素 {new Box(1,2,6), new Box(2,3,4), new Box(3,4,5), new Box(5,6,7)} 进行排序,其中 Box 构造函数中参数分别为（长，宽，高），具体按照 Box 实例长的降序和升序分别排序。
- 12、 写一个通用的方法（该方法有一个 String 目录参数），使用 IO 输入输出知识实现递归调用把传递给该方法的目录下的文件，子文件夹，以及子文件夹下的子 文件夹和文件，子文件夹下的子文件夹和文件、、、的名称和类型（是文件还是文件夹，最好有一定的层次感）打印到控制台上。

7.21

【人力资源】Java 程序员面试测试题

Java 程序员面试测试题

姓名：_____ 电话：_____

Email: _____

1、下面是本公司将要开发的办公后台系统：

请分析与用户及权限相关的对象有哪些？

每个对象有哪些属性、方法、接口？

请给出用户及权限相关的数据结构。

形式:

数据表 1: 数据表名称 (如用户信息表)

字段名称

字段描述

与其他数据表中字段的关联关系

...

1 您对创建该后台系统还有什么其他方面建议?

2、下面是本公司将要开发的留言系统前台界面:

写出校验 email 的 Javascript 脚本

写出保存留言的 JSP 程序代码

参数如下:

oracle9i 数据库服务器地址: local

数据库连接帐号: admin

数据库连接密码: enet.com.cn

数据库名称: enet

数据表名称: feedback

数据表 feedback 的结构如下:

字段名称

字段描述

ID

字段编号

comments

您的留言

type

意见性质

username

您的姓名

email

联系 Email

telephone

联系电话

referurl

用于记录访问留言页面前的页面路径

请您画出处理留言后台的流程图

您对该留言系统还有什么其他的建议?

7.22

java 程序员面试必备问答汇总

java程序员面试必备问答汇总

第一，谈谈 final, finally, finalize 的区别。

final 用于声明属性，方法和类，分别表示属性不可变，方法不可覆盖，类不可继承。finally 是异常处理语句结构的一部分，表示总是执行。finalize 是 Object 类的一个方法，在垃圾收集器执行的时候会调用被回收对象的此方法，可以覆盖此方法提供垃圾收集时的其他资源回收，例如关闭文件等。

第二，Anonymous Inner Class (匿名内部类) 是否可以 extends (继承) 其它类，是否可以 implements (实现) interface (接口)?

可以继承其他类或完成其他接口，在 swing 编程中常用此方式。

第三，Static Nested Class 和 Inner Class 的不同，说得越多越好 (面试题有的很笼统)。

Static Nested Class 是被声明为静态 (static) 的内部类，它可以不依赖于外部类实例被实例化。而通常的内部类需要在外部类实例化后才能实例化

第四，& 和 && 的区别。

& 是位运算符，表示按位与运算，&& 是逻辑运算符，表示逻辑与 (and)。

第五，HashMap 和 Hashtable 的区别。

HashMap是Hashtable的轻量级实现（非线程安全的实现），他们都完成了Map接口，主要区别在于HashMap允许空（null）键值（key），由于非线程安全，效率上可能高于Hashtable。

第六，Collection 和 Collections 的区别。

Collection是集合类的上级接口，继承与他的接口主要有Set 和 List.Collections是针对集合类的一个帮助类，他提供一系列静态方法实现对各种集合的搜索、排序、线程安全化等操作。

第七，什么时候用 assert。

1.4 新增关键字（语法），用于测试boolean表达式状态，可用于调试程序。使用方法 `assert`，表示如果表达式为真（true），则下面的语句执行，否则抛出 `AssertionError`。另外的使用方式 `assert < boolean表达式>:`，表示如果表达式为真，后面的表达式忽略，否则后面表达式的值用于 `AssertionError` 的构建参数。注意编译时要增加 `-source 1.4` 参数，否则报错。] 运行时要增加 `-ea` 参数，否则 `assert` 行被忽略

第八，GC是什么？为什么要有GC？

GC是垃圾收集的意思（Garbage Collection），内存处理是编程人员容易出现问题的地方，忘记或者错误的内存回收会导致程序或系统的不稳定甚至崩溃，Java 提供的GC功能可以自动监测对象是否超过作用域从而达到自动回收内存的目的，Java语言没有提供释放已分配内存的显示操作方法。

第九，`String s = new String("xyz");` 创建了几个String Object?
两个

第十，`Math.round(11.5)` 等於多少？`Math.round(-11.5)` 等於多少？

`Math.round(11.5)=12`
`Math.round(-11.5)=-11`
round方法返回与参数最接近的长整数，参数加1/2后求其floor。

第十一，`short s1 = 1; s1 = s1 + 1;` 有什么错？

`short s1 = 1; s1 = s1 + 1;` （s1+1运算结果是int型，需要强制转换类型）
`short s1 = 1; s1 += 1;` （可以正确编译）

第十二, `sleep()` 和 `wait()` 有什么区别?

`sleep` 是线程类 (`Thread`) 的方法, 导致此线程暂停执行指定时间, 给执行机会给其他线程, 但是监控状态依然保持, 到时后会自动恢复。调用 `sleep` 不会释放对象锁。`wait` 是 `Object` 类的方法, 对此对象调用 `wait` 方法导致本线程放弃对象锁, 进入等待此对象的等待锁定池, 只有针对此对象发出 `notify` 方法 (或 `notifyAll`) 后本线程才进入对象锁定池准备获得对象锁进入运行状态。

第十三, Java 有没有 `goto`? 没有 很十三的问题, 如果哪个面试的问到这个问题, 我劝你还是别进这家公司。

第十四, 数组有没有 `length()` 这个方法? `String` 有没有 `length()` 这个方法?

数组没有 `length()` 这个方法, 有 `length` 的属性。

`String` 有 `length()` 这个方法。

第十五, `Overload` 和 `Override` 的区别。Overloaded 的方法是否可以改变返回值的类型?

方法的重写 `Overriding` 和重载 `Overloading` 是 Java 多态性的不同表现。重写 `Overriding` 是父类与子类之间多态性的一种表现, 重载 `Overloading` 是一个类中多态性的一种表现。如果在子类中定义某方法与其父类有相同的名称和参数, 我们说该方法被重写 (`Overriding`)。子类的对象使用这个方法时, 将调用子类中的定义, 对它而言, 父类中的定义如同被“屏蔽”了。如果在一个类中定义了多个同名的方法, 它们或有不同的参数个数或有不同的参数类型, 则称为方法的重载 (`Overloading`)。Overloaded 的方法是可以改变返回值的类型。

第十六, `Set` 里的元素是不能重复的, 那么用什么方法来区分重复与否呢? 是用 `==` 还是 `equals()`? 它们有何区别?

`Set` 里的元素是不能重复的, 那么用 `iterator()` 方法来区分重复与否。`equals()` 是判读两个 `Set` 是否相等。`equals()` 和 `==` 方法决定引用值是否指向同一对象 `equals()` 在类中被覆盖, 为的是当两个分离的对象的內容和类型相配的话, 返回真值。

第十七, 给我一个你最常见到的 `runtime exception`。

ArithmeticException, ArrayStoreException, BufferOverflowException, BufferUnderflowException, CannotRedoException,

CannotUndoException, ClassCastException, CMMException, ConcurrentModificationException, DOMException,

EmptyStackException, IllegalArgumentException, IllegalMonitorStateException, IllegalPathStateException,

IllegalStateException, ImagingOpException, IndexOutOfBoundsException,

MissingResourceException, NegativeArraySizeException, NoSuchElementException,

NullPointerException, ProfileDataException, ProviderException, RasterFormatException, SecurityException,

SystemException, UndeclaredThrowableException, UnmodifiableSetException, UnsupportedOperationException

第十八, error和exception有什么区别?

error 表示恢复不是不可能但很困难的情况下的一种严重问题。比如说内存溢出。不可能指望程序能处理这样的情况。

exception 表示一种设计或实现问题。也就是说, 它表示如果程序运行正常, 从不会发生的情况。

第十九, List, Set, Map是否继承自Collection接口?

List, Set 是, Map不是

第二十, abstract class和interface有什么区别?

声明方法的存在而不去实现它的类被叫做抽象类 (abstract class), 它用于要创建一个体现某些基本行为的类, 并为该类声明方法, 但不能在该类中实现该方法的情况。不能创建 abstract 类的实例。然而可以创建一个变量, 其类型是一个抽象类, 并让它指向具体子类的一个实例。不能有抽象构造函数或抽象静态方法。Abstract 类的子类为它们父类中的所有抽象方法提供实现, 否则它们也是抽象类。取而代之, 在子类中实现该方法。知道其行为的其它类可以在类中实现这些方法。接口 (interface) 是抽象类的变体。在接口中, 所有方法都是抽象的。多

继承性可通过实现这样的接口而获得。接口中的所有方法都是抽象的，没有一个有程序体。

接口只可以定义 `static final` 成员变量。接口的实现与子类相似，除了该实现类不能从接口定义中继承行为。当类实现特殊接口时，它定义（即将程序体给予）所有这种接口的方法。然后，它可以在实现了该接口的类的任何对象上调用接口的方法。由于有抽象类，它允许使用接口名作为引用变量的类型。通常的动态联编将生效。引用可以转换到接口类型或从接口类型转换，`instanceof` 运算符可以用来决定某对象的类是否实现了接口。

第二十一，`abstract` 的 `method` 是否可同时是 `static`，是否可同时是 `native`，是否可同时是 `synchronized`？

都不能

第二十二，接口是否可继承接口？抽象类是否可实现 (`implements`) 接口？抽象类是否可继承实体类 (`concrete class`)？

接口可以继承接口。抽象类可以实现 (`implements`) 接口，抽象类可继承实体类，但前提是实体类必须有明确的构造函数。

第二十三，启动一个线程是用 `run()` 还是 `start()`？

启动一个线程是调用 `start()` 方法，使线程所代表的虚拟处理机处于可运行状态，这意味着它可由 JVM 调度并执行。这并不意味着线程就会立即运行。`run()` 方法可以产生必须退出的标志来停止一个线程。

第二十四，构造器 `Constructor` 是否可被 `override`？

构造器 `Constructor` 不能被继承，因此不能重写 `Overriding`，但可以被重载 `Overloading`。

第二十五，是否可以继承 `String` 类？

`String` 类是 `final` 类故不可以继承。

第二十六，当一个线程进入一个对象的一个 `synchronized` 方法后，其它线程是否可进入此对象的其它方法？

不能，一个对象的一个 `synchronized` 方法只能由一个线程访问。

第二十七, try {}里有一个return语句, 那么紧跟在这个try后的 finally {}里的 code会不会被执行, 什么时候被执行, 在return前

还是后?

会执行, 在return前执行。

第二十八, 编程题: 用最有效率的方法算出 2 乘以 8 等於几?

有 C 背景的程序员特别喜欢问这种问题。

2 << 3

第二十九, 两个对象值相同(x.equals(y) == true), 但却可有不同的 hash code, 这句话对不对?

不对, 有相同的 hash code。

第三十, 当一个对象被当作参数传递到一个方法后, 此方法可改变这个对象的属性, 并可返回变化后的结果, 那么这里到底是值传递还是引用传递?

是值传递。Java 编程语言只由值传递参数。当一个对象实例作为一个参数被传递到方法中时, 参数的值就是对该对象的引用。对象的内容可以在被调用的方法中改变, 但对象的引用是永远不会改变的。

第三十一, switch是否能作用在byte上, 是否能作用在 long上, 是否能作用在 String上? switch(expr1) 中, expr1是一个整数表达式。因此传递给 switch 和 case 语句的参数应该是 int、short、char 或者 byte。

long, string 都不能作用于 switch。

第三十二, 编程题: 写一个 Singleton 出来。

Singleton 模式主要作用是保证在 Java 应用程序中, 一个类 Class 只有一个实例存在。

一般 Singleton 模式通常有几种形式:

第一种形式: 定义一个类, 它的构造函数为 private 的, 它有一个 static 的 private 的该类变量, 在类初始化时实例化, 通过一个

public 的 getInstance 方法获取对它的引用, 继而调用其中的方法。

```

public class Singleton {
    private Singleton() {}
    //在自己内部定义自己一个实例，是不是很奇怪？
    //注意这是 private 只供内部调用
    private static Singleton instance = new Singleton();
    //这里提供了一个供外部访问本 class 的静态方法，可以直接访问
    public static Singleton getInstance() {
        return instance;
    }
}

```

第二种形式:

```

public class Singleton {
    private static Singleton instance = null;
    public static synchronized Singleton getInstance() {
        //这个方法比上面有所改进，不用每次都进行生成对象，只是第一次
        //使用时生成实例，提高了效率！
        if (instance==null)
            instance=new Singleton();
        return instance;
    }
}

```

其他形式:

定义一个类，它的构造函数为 private 的，所有方法为 static 的。

一般认为第一种形式要更加安全些

第三十三: set, map, list 之间有什么区别和共同点

set —— 其中的值不允许重复，无序的数据结构
list —— 其中的值允许重复，因为其有序的数据结构
map —— 成对的数据结构，键值必须具有唯一性

第三十四: Collection, ArrayList 还有 Vector 有什么区别和共同点

ArrayList 是 List 接口的一个可变长数组实现。实现了所有 List 接口的操作，并允许存储 null 值。除了没有进行同步，ArrayList 基本等同于 Vector。在 Vector 中几乎对所有的方法都进行了同步，但 ArrayList 仅对 writeObject 和 readObject 进行了同步，其它比如 add(Object)、remove(int) 等都没有同步。因此，Vector 是线程安全的，但 ArrayList 不是。这使得 ArrayList 要比 Vector 快速。对于一些最新的 JVM，两个类在速度上的差异可以忽略不计：严格地说，对于这些 JVM，这两个类在速度上的差异 小于比较这些类性能的测试所显示的时间差异。

简单说 :

ArrayList——实现的是数组功能，不同步，效率高

Vector——实现的是数组的功能，同步，控制简单

第三十五：post与get的差别与共同点

Post与Get的不同：Post在发送前要进行编码处理，然后分组发送，发送过程相对安全，适合大数据量的处理；而Get方法不进行任何处理，一次性发出，适合小数据量交换。除非你肯定你提交的数据可以一次性提交，否则请尽量用Post方法。

第三十六：public, private, protected, friendly, internal 的作用范围

private 同一个类中

protected 同一类中，同一包中，不同包中的该类子类

public 同一类中，同一包中，不同包中的该类子类，不同包中的该类的非子类

friendly 同一个类中，同一包中

internal 表示在同一个应用程序（Application）或类库（Library）中都可以使用，不过这个我们基本上不用

补充：

abstract与final不能同时修饰一个类

abstract与private static final native不能并列修饰同一个方法

abstract类中不能有private修饰的域和方法

static方法中不能处理非static的域

java 面试题

1. 写出下面代码的结果

```
public class TestString{
    public static void link(String a){
        a += "World";
    }

    public static void main(String[] args){
        String a = "Hello";
        link(a);
        System.out.println(a);
    }
}
```

```
    }  
}
```

答：这道题考两个方面：

- 1). String 对象的内容是不能改变的, `a+="World"`并不是把 a 所指对象改变, 而是先生成一个临时 String 对象, 其值为 `a+"World"`, 然后在把这个临时对象赋给 a.
- 2). Java 中函数参数传递方式为 Call by value, link 方法中会生产 a 的一个拷贝, 同样指向 a 所指的对象, 综合以上两点, 得出结果为 Hello

2. 写出下面代码的结果

```
System.out.println("ja"+"va" == "java");
```

答：“=”用来比较对象的引用, 而 `equals()` 用来比较对象的内容, 但是如果是字符串常量, 用“=”也可以比较内容

是否相等, “ja”+“va”和“java”都是字符串常量, 因此结果为 true

同理, 下面代码结果也是 true

```
final String str = "java";
```

```
System.out.println(str=="java")
```