

1. 请说出 ArrayList, Vector, LinkedList 的存储性能和特性 ?

解答: 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。

2. List、Map、Set 三个接口存储元素时各有什么特点?

解答:

1) List 是有序的 Collection, 使用此接口能够精确的控制每个元素插入的位置。用户能够使用索引 (元素在 List 中的位置, 类似于数组下标) 来访问 List 中的元素, 这类似于 Java 的数组。

2) Set 是一种不包含重复的元素的 Collection, 即任意的两个元素 e1 和 e2 都有 e1.equals(e2)=false, Set 最多有一个 null 元素。

3) Map 接口: 请注意, Map 没有继承 Collection 接口, Map 提供 key 到 value 的映射

3、abstract class 和 interface 有什么区别 ?

解答: 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。新型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, 所有成员变量都是 public static final 的。一个类可以实现多个接口, 当类实现接口时, 必须实现接口的所有方法。抽象类在 Java 语言中表示的是一种单继承的关系, 对于 interface 来说则不然, 并不要求 interface 的实现者和 interface 定义在概念本质上是一致的, 仅仅是实现了 interface 定义的契约而已; 抽象类中可以定义自己的成员变量, 也可以包含非抽象的方法, 而在接口中只能有静态的常量, 所有方法必须是抽象的; 实现抽象类时可以只实现其中的部分方法, 而要是实现一个接口的话就必须实现这个接口中的所有抽象方法。

4. JSP 页面之间传递参数的方法有哪些?

解答:

1) request

2) session

3) application

4) 提交表单

5) 超链接

5. forward 和 redirect 的区别

解答: forward 是容器中控制权的转向, 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，并且从浏览器的地址栏中可以看到跳转后的链接地址。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。

6.Java 反射机制的作用？

解答：Java 反射机制的作用是：

- 1) 在运行时判断任意一个对象所属的类。
- 2) 在运行时构造任意一个类的对象。
- 3) 在运行时判断任意一个类所具有的成员变量和方法。
- 4) 在运行时调用任意一个对象的方法

7. 你是怎么理解 java 的泛型的？

解答：在 Java SE 1.5 之前，没有泛型的情况下，通过对类型 `Object` 的引用来实现参数的“任意化”，“任意化”带来的缺点是要做显式的强制类型转换，而这种转换是要求开发者对实际参数类型可以预知的情况下进行的。对于强制类型转换错误的情况，编译器可能不提示错误，在运行的时候才出现异常，这是一个安全隐患。

泛型是 Java SE 1.5 的新特性，泛型的本质是参数化类型，也就是说所操作的数据类型被指定为一个参数。这种参数类型可以用在类、接口和方法的创建中，分别称为泛型类、泛型接口、泛型方法。

泛型的好处是在编译的时候检查类型安全，并且所有的强制转换都是自动和隐式的，提高代码的重用率。

1. 编写程序，完成文件复制功能

```

/**
 * 复制单个文件
 * @param oldPath String 原文件路径 如：c:/fqf.txt
 * @param newPath String 复制后路径 如：f:/fqf.txt
 * @return boolean
 */
public void copyFile(String oldPath, String newPath) {
    try {
        int bytesum = 0;
        int byteread = 0;
        File oldfile = new File(oldPath);
        if (oldfile.exists()) { //文件存在时
            InputStream inStream = new FileInputStream(oldPath); //读入原文件
            FileOutputStream fs = new FileOutputStream(newPath);
            byte[] buffer = new byte[1444];
            int length;
            while ((byteread = inStream.read(buffer)) != -1) {
                bytesum += byteread; //字节数 文件大小
                System.out.println(bytesum);
                fs.write(buffer, 0, byteread);
            }
            inStream.close();
        }
    }
    catch (Exception e) {
        System.out.println("复制单个文件操作出错");
        e.printStackTrace();
    }
}
}

```

2. Servlet 创建过程及生命周期 Servlet 在容器中运行时，其实例的创建及销毁等是由容器进行控制。

Servlet 的创建有两种方法。

- 1) . 客户端请求对应的 Servlet 时，创建 Servlet 实例；大部分 Servlet 都是这种 Servlet。
- 2) . 通过在 web.xml 中设置 load-on-startup 来创建 servlet 实例，这种实例在 Web 应用启动时，立即创建 Servlet 实例

Servlet 的运行都遵循如下生命周期：

- 1) . 创建 Servlet 实例。
- 2) . Web 容器调用 Servlet 的 init() 方法，对 Servlet 进行初始化。
- 3) . Servlet 初始化后，将一直存在于容器中，用于响应客户端请求。根据客户端的请求方式通过 Servlet 中 service() 方法去相应的 doXXX() 方法；
- 4) . Web 容器销毁 Servlet 时，调用 Servlet 的 destroy() 方法，通常在关闭 Web 容器之时销毁 Servlet。

3. 用 sql 语句分页：

Mysql 数据库：

```

SELECT TOP 页大小 * FROM table1 WHERE id NOT IN (
    SELECT TOP 页大小 *( 页数 -1) id FROM table1 ORDER BY id
) ORDER BY id

```

Oracle 数据库：

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。

截取 ID 的方法

```
select * from emp a, (select empno,rownum as num from emp)b where  
a.empno=b.empno and b.num between 5 and 7;
```

三层嵌套

```
SELECT * FROM ( SELECT A.*, rownum r FROM ( SELECT * FROM emp ) A WHERE  
rownum <=7 ) B WHERE r >5;
```

4. list.map.set 的存储特点？

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

5. final,finally,finaliz 的区别

final 一修饰符（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 一再异常处理时提供 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 一方法名。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。

6. arraylist 和 vector 的区别？

1). 同步性 :Vector 是线程安全的，也就是说是同步的，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

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

1) . ArrayList 采用的是数组形式来保存对象的，这种方式将对象放在连续的位置中，所以最大的缺点就是插入删除时非常麻烦。LinkedList 采用的将对象存放在独立的空间中，而且在每个空间中还保存下一个链接的索引。但是缺点就是查找非常麻烦，要从第一个索引开始。

2) . ArrayList 和 Vector 都是用数组方式存储数据，此数组元素数要大于实际的存储空间以便进行元素增加和插入操作，他们都允许直接用序号索引元素，但是插入数据元素涉及到元素移动等内存操作，所以索引数据快而插入数据慢。

3) . Vector 使用了 synchronized 方法（线程安全），所以在性能上比 ArrayList 要差些。

4) . LinkedList 使用双向链表方式存储数据，按序号索引数据需要前向或后向遍历数据，所以索引数据慢，是插入数据时只需要记录前后项即可，所以插入的速度快。

8. HASHMAP,HASHTABLE区别

- 1.Hashtable 是 Dictionary 的子类，HashMap是 Map接口的一个实现类；
- 2.Hashtable 中的方法是同步的，而 HashMap中的方法在缺省情况下是非同步的。
3. 在 HashMap中，null 可以作为键，这样的键只有一个；可以有一个或多个键所对应的值为 null 。

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

Session 用于保存每个用户的专用信息。每个客户端用户访问时，服务器都为每个用户分配一个唯一的会话 ID (Session ID)。她的生存期是用户持续请求时间再加上一段时间 (一般是 20 分钟左右)。Session 中的信息保存在 Web服务器内容中，保存的数据量可大可小。当 Session 超时或被关闭时将自动释放保存的数据信息。由于用户停止使用应用程序后它仍然在内存中保持一段时间，因此使用 Session 对象使保存用户数据的方法效率很低。对于小量的数据，使用 Session 对象保存还是一个不错的选择。

Cookie 用于保存客户浏览器请求服务器页面的请求信息，程序员也可以用它存放非敏感性的用户信息，信息保存的时间可以根据需要设置。如果没有设置 Cookie 失效日期，它们仅保存到关闭浏览器程序为止。如果将 Cookie 对象的 Expires 属性设置为 Minvalue，则表示 Cookie 永远不会过期。Cookie 存储的数据量很受限制，大多数浏览器支持最大容量为 4K,因此不要用来保存数据集及其他大量数据。由于并非所有的浏览器都支持 Cookie，并且数据信息是以明文文本的形式保存在客户端的计算机中，因此最好不要保存敏感的，未加密的数据，否则会影响网站的安全性。

session 工作原理

- (1) 当有 Session 启动时，服务器生成一个唯一值，称为 Session ID (好像是通过取进程 ID 的方式取得的)。
- (2) 然后，服务器开辟一块内存，对应于该 Session ID。
- (3) 服务器再将该 Session ID 写入浏览器的 cookie。
- (4) 服务器内有一进程，监视所有 Session 的活动状况，如果有 Session 超时或是主动关闭，服务器就释放改内存块。
- (5) 当浏览器连入 IIS 时并请求的 ASP内用到 Session 时，IIS 就读浏览器 Cookie 中的 Session ID。
- (6) 然后，服务检查该 Session ID 所对应的内存是否有效。
- (7) 如果有效，就读出内存中的值。
- (8) 如果无效，就建立新的 Session。

10. String 和 stringbuffer 进行字符串连接时的区别？

String 对项内容是不可改变的，StringBuffer 是可以改变的，且高效；

11. 什么叫项目？

用有限的资源、有限的时间为特定客户完成特定目标的一次性工作

12. .J2EE J是什么意思？ 2 是什么意思 EE 是什么意思？ Struts 的拦截器你是怎么做的，是独立开发的，能够独立开发的到。然后问有哪几种拦截？

Java 2 Platform , Enterprise Edition

继承 AbstractInterceptor 类，覆盖 intercept() 方法

有 struts 自己的拦截器 如 timer ,i18n,scop,servletconfig,token
还有自定义拦截器

13. 简单介绍下 java ? Spring 的 AOP, IOC 的讲述 对 struts2 的了解 , 1 , 2 的比较 xml
的了解 J2ee 的 webserviced 的协议 ?

Spring AOP: 代理机制 Spring 提供的自动代理机制

Spring 的 IoC 来实组件之间的依赖关系注入 , 使控制层与业务实现分离 , 即客户通过
调用业务委托接口来调用具体的业务组件的实例 , 避免控制层与业务层的藕合带来的维护
或升级困难。由 Spring 为 DAO 生成代理对象来为 DAO 添加事务服务 , 由 IoC 容器 DAO 的代
理实例注入到业务组件中 , 业务组件通过 DAO 的委托接口调用 DAO 对象 , 使得上层组件不直
接依赖于 DAO 的实现类 , 使得持久层更换或修改不影响上层组件。

14. Collections, collection 的区别

Collection 是个 java.util 下的接口 , 它是各种集合结构的父接口。 Collections 是个
java.util 下的普通类 , 它包含有各种有关集合操作的静态方法。 Collections 是针对集
合类的一个帮助类 , 它提供一系列的静态方法实现对各种集合的搜索 , 排序 , 线程安全化
等操作

15. 如何释放过度使用的 undo 表空间

1. 启动 SQLPLUS 并用 sys 登陆到数据库。 2. 查找数据库的 UNDO 表空间名 3. 确认
UNDO 表空间 ; SQL> select name from v\$tablespace; 4. 检查数据库 UNDO 表空间占
用空间情况以及数据文件存放位置 ; 5. 查看回滚段的使用情况 , 哪个用户正在使用回
滚段的资源 , 如果有用户最好更换时间 6. 检查 UNDO Segment 状态

16. 数据库里面游标 , 索引是怎么用的 ? 在 oracle 数据库中 , 用 get 和 load 方法 这 2
个方法来查没有的数据 那么分别会反馈什么样的结果 ?

declare cur cursor keyset for

get 返回 null, load classnotfoundException

17. . 在 Oracle 中数据库中的一个表中 , 这个表没有主键 id 也没有特定标示来查数据 , 怎
么查 ?
rowid

18. 简述 private 、 proteced 、 public 、 internal 修饰符的访问权限

	同一个类	同一个包	不同包的子
类	不同包非子类		
private	*		
default	*	*	
protected	*	*	
*			
public	*	*	
*		*	

19. 概述反射和序列化

Reflection 是 Java 被视为动态（或准动态）语言的一个关键性质。这个机制允许程序在运行时透过 Reflection APIs 取得任何一个已知名称的 class 的内部信息，包括其 modifiers（诸如 public, static 等等）、superclass（例如 Object）、实现之 interfaces（例如 Cloneable），也包括 fields 和 methods 的所有信息，并可于运行时改变 fields 内容或唤起 methods。本文借由实例，大面积示范 Reflection APIs。

20. 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

21. ORACLE 中 not 和 and 和 or 优先级由高到低？

not and or

22. java 是如何进行异常处理的

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

23. struts2 必备包

commons-fileupload-1.2.1.jar

freemarker-2.3.13.jar

ognl-2.6.11.jar

struts2-core-2.1.6.jar

xwork-2.1.2.jar

24. dao 是什么及作用

dao 是数据访问对象 DAO 负责管理与数据源的连接来获取和储存其中的数据

25. ibatis 中的 #与\$的区别

在 Ibatis 中我们使用 SqlMap 进行 Sql 查询时需要引用参数，在参数引用中遇到的符号 #和\$之间的区分为，#可以进行与编译，进行类型匹配，而 \$不进行数据类型匹配，例如：

select * from table where id = #id#，其中如果字段 id 为字符型，那么 #id# 表示的就是 'id' 类型，如果 id 为整型，那么 #id# 就是 id 类型。

select * from table where id = \$id\$，如果字段 id 为整型，Sql 语句就不会出错，但是如果字段 id 为字符型，那么 Sql 语句应该写成 select * from table where id = '\$id\$ '

26. struts2 的实现原理

- 1、客户端初始化一个指向 Servlet 容器（例如 Tomcat）的请求
- 2、这个请求经过一系列的过滤器（Filter）（这些过滤器中有一个叫做 ActionContextCleanUp 的可选过滤器，这个过滤器对于 Struts2 和其他框架的集成很有帮助，例如： SiteMesh Plugin）；
- 3、接着 FilterDispatcher 被调用，FilterDispatcher 询问 ActionMapper 来决定这个请求是否需要调用某个 Action；
- 4、如果 ActionMapper 决定需要调用某个 Action，FilterDispatcher 把请求的处理交给 ActionProxy；
- 5、ActionProxy 通过 Configuration Manager 询问框架的配置文件，找到需要调用的 Action 类；
- 6、ActionProxy 创建一个 ActionInvocation 的实例。
- 7、ActionInvocation 实例使用命名模式来调用，在调用 Action 的过程前后，涉及到相关拦截器（Interceptor）的调用。
- 8、一旦 Action 执行完毕，ActionInvocation 负责根据 struts.xml 中的配置找到对应的返回结果。返回结果通常是（但不总是，也可能是另外的一个 Action 链）一个需要被表示的 JSP 或者 FreeMarker 的模版。在表示的过程中可以使用 Struts2 框架中继承的标签。在这个过程中需要涉及到 ActionMapper

27. 简述 spring 的事务传播行为和 隔离级别

spring 的事务传播行为：Spring 在 TransactionDefinition 接口中规定了 7 种类型的事务传播行为，它们规定了事务方法和事务方法发生嵌套调用时事务如何进行传播：

PROPAGATION_REQUIRED 如果当前没有事务，就新建一个事务，如果已经存在一个事务中，加入到这个事务中。这是最常见的选择。

PROPAGATION_SUPPORTS 支持当前事务，如果当前没有事务，就以非事务方式执行。

PROPAGATION_MANDATORY 使用当前的事务，如果当前没有事务，就抛出异常。

PROPAGATION_REQUIRES_NEW 新建事务，如果当前存在事务，把当前事务挂起。

PROPAGATION_NOT_SUPPORTED 以非事务方式执行操作，如果当前存在事务，就把当前事务挂起。

PROPAGATION_NEVER 以非事务方式执行，如果当前存在事务，则抛出异常。

PROPAGATION_NESTED 如果当前存在事务，则在嵌套事务内执行。如果当前没有事务，则执行与 PROPAGATION_REQUIRED 类似的操作。

Spring 的隔离级别

- 1、Serializable：最严格的级别，事务串行执行，资源消耗最大；
- 2、REPEATABLE_READ 保证了一个事务不会修改已经由另一个事务读取但未提交（回滚）的数据。避免了“脏读取”和“不可重复读取”的情况，但是带来了更多的性能损失。
- 3、READCOMMITTED 大多数主流数据库的默认事务等级，保证了一个事务不会读到另一个并行事务已修改但未提交的数据，避免了“脏读取”。该级别适用于大多数系统。
- 4、Read Uncommitted：保证了读取过程中不会读取到非法数据。

28. 写一段把本地文件 fromfile 拷贝到本地文件 tofile 的程序

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



```

File dir = new File("f:/jtd");
// 得到 d:/jtd 下的所有文件对象
File[] files = dir.listFiles();
for (File file : files) {
    String fileName = file.getName(); // 得到文件名
    int index = fileName.indexOf(".");// 的索引位置
    String fileName2 = fileName.substring(0, index); // 不带后缀名
    的文件名

    String houZui = fileName.substring(index);
    if (houZui.equals(".jtd")) {
        // "f:/java/" + fileName2 + ".java"
        // 要移到的地方
        file.renameTo(new File("f:/java/" + fileName2 +
".java"));
    }
}
}
}
}

```

29. 写出删除表中重复记录的语句 oracle

delete from people

where peopleid in (select peopleid from people group by peopleid having

count(peopleid) > 1) and rowid not in (select min(rowid) from people group by

30. java 类实现序列化的方法（二种）？如在 collection 框架中，要实现什么样的接口

java.io.Serializable 接口

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

31. Struts2 实现拦截器的原理？

实现原理：Struts2 拦截器是在访问某个 Action 或 Action 的某个方法、字段之前或之后实施拦截，并且 Struts2 拦截器是可插拔的，拦截器是 AOP 的一种实现。当请求 struts2 的 action 时，Struts 2 会查找配置文件，并根据其配置实例化相对的拦截器对象，然后串成一个列表，最后一个一个地调用列表中的拦截器。

32. sleep() 和 wait() 区别

sleep() 方法：线程主动放弃 CPU, 使得线程在指定的时间内进入阻塞状态，不能得到 CPU 时间，指定的时间一过，线程重新进入可执行状态。典型地，sleep() 被用在等待某个资源就绪的情形：测试发现条件不满足后，让线程阻塞一段时间后重新测试，直到条件满足为止。

wait() ：与 notify() 配套使用，wait() 使得线程进入阻塞状态，它有两种形式，一种允许指定以毫秒为单位的一段时间作为参数，另一种没有参数，当指定时间参数时对应的 notify() 被调用或者超出指定时间时线程重新进入可执行状态，后者则必须对应的 notify() 被调用

33. Servlet 的生命周期分为 3 个阶段 :? 和 CGI 的区别 ?

Servlet 的生命周期主要由 3 个过程组成。

(1)init() 方法：服务器初始化 servlet 。

(2)service() 方法：初始化完毕， servlet 对象调用该方法响应客户的请求。

(3)destroy() 方法：调用该方法消灭 servlet 对象。

其中，init() 方法只在 servlet 第一次被请求加载的时候被调用一次， 当有客户再请求 servlet 服务时，web 服务器将启动一个新的线程， 在该线程中， 调用 service 方法响应客户的请求。

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 。

34. forward 与 redirect 的区别 ?有哪些方式实现

1) . <jsp:forward> 重定向后 url 地址栏地址不变还是原来的地址；而 response.sendRedirect() 重定向后 url 地址栏地址显示的是请求后的新地址。

2) . <jsp:forward> 重定向的时候可以保存回话信息，因此可以使用 request 来进行参数传递， 在新页面 可以使用 request.getAttribute () 来得到 参数。而 response.sendRedirect() 不支持此通过 request 进行参数传递。它唯一的传值方式为 response.sendRedirect(“ example.jsp?aa=123 ”)，在新的页面通过 request.getParameter(“ aa ”)来得到参数值

35. 多线程、同步实现方法？

1) 实现线程有两种方法： 继承 Thread 类或者实现 Runnable 接口

2) 实现同步也有两种，一种是用同步方法，一种是用同步块 .. 同步方法就是在方法返回类型后面加上 synchronized, 比如：

```
public void synchronized add(){...}
```

同步块就是直接写 :synchronized (这里写需要同步的对象){...}

36. 集合类有哪些？有哪些方法？

集合类 ArrayList LinkedList HashSet HashMap

方法： add(),remove(),put(),addAll(),removeAll()

37. java 中实现多态的机制是什么

重写，重载

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。

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。

38. 静态的多态和动态的多态的区别

静态的多态：即为重载；方法名相同，参数个数或类型不相同。(overloading)

动态的多态：即为重写；子类覆盖父类的方法，将子类的实例传与父类的引用调用的是子类的方法 实现接口的实例传与接口的引用调用的实现类的方法。

39. 作用域 public,private,protected. 以及不写时的区别

public 整个 java 程序中都可以访问

protected 在其他包中不可以访问

friendly 只有当前包或当前内可以访问，其他都不行

private 只有当前内可以访问

不写的话默认是 protected

40. extends 和 implement 的不同

extends 是继承父类，只要那个类不是声明为 final 或者那个类定义为 abstract 的就能继承，JAVA中不支持多重继承，但是可以用接口来实现，这样就要用到 implements，继承只能继承一个类，但 implements 可以实现多个接口，用逗号分开就行了 比如
class A extends B implements C,D,E

41. join 与 left join 的区别：

inner join(等值连接) 只返回两个表中联结字段相等的行

left join(左联接) 返回包括左表中的所有记录和右表中联结字段相等的记录

right join(右联接) 返回包括右表中的所有记录和左表中联结字段相等的记录

42. DAO设计模式与 DELEGATE模式

DAO(Data Access Object) 模式实际上是两个模式的组合，即 Data Accessor 模式和 Active Domain Object 模式，其中 Data Accessor 模式实现了数据访问和业务逻辑的分离，而 Active Domain Object 模式实现了业务数据的对象化封装，一般我们将这两个模式组合使用。

DAO 模式通过对底层数据的封装，为业务层提供一个面向对象的接口，使得业务逻辑开发人员可以面向业务中的实体进行编码。通过引入 DAO模式，业务逻辑更加清晰，且富于形象性和描述性，这将为日后的维护带来极大的便利。试想，在业务曾通过 Customer.getName 方法获得客户姓名，相对于直接通过 SQL语句访问数据库表并从 ResultSet 中获得某个字符型字段而言，哪种方式更加易于业务逻辑的形象化和简洁化？

Business Delegate 起到客户端业务抽象化的作用。它抽象化，进而隐藏业务服务的实现。

使用 Business Delegate，可以降低表示层客户端和系统的业务服务之间的耦合程度。根据实现策略不同，Business Delegate 可以在业务服务 API 的实现中，保护客户端不受可能的变动性影响。这样，在业务服务 API 或其底层实现变化时，可以潜在地减少必须修改表示层客户端代码的次数。

43. 适配器模式与桥梁模式的区别

适配器模式把一个类的接口变换成客户端所期待的另一种接口，从而使原本因接口不匹配而无法在一起工作的两个类能够在一起工作。又称为转换器模式、变压器模式、包装模式（把已有的一些类包装起来，使之能有满足需要的接口）。适配器模式的用意是将接口不同而功能相同或者相近的两个接口加以转换，包括适配器角色补充一些源角色没有但目标接口需要的方法。就像生活中电器插头是三相的，而电源插座是两相的，这时需要一个三相变两

相的转换器来满足。

比如，在 Java I/O 库中使用了适配器模式，象 `FileInputStream` 是一个适配器类，其继承了 `InputStream` 类型，同时持有一个对 `FileDescriptor` 的引用。这是将一个 `FileDescriptor` 对象适配成 `InputStream` 类型的对象形式的适配器模式。`StringReader` 是一个适配器类，其继承了 `Reader` 类型，持有一个对 `String` 对象的引用。它将 `String` 的接口适配成 `Reader` 类型的接口。等等。

桥梁模式的用意是要把实现和它的接口分开，以便它们可以独立地变化。桥梁模式并不是用来把一个已有的对象接到不相匹配的接口上的。当一个客户端只知道一个特定的接口，但是又必须与具有不同接口的类打交道时，就应该使用桥梁模式。

比如，JDBC 驱动器就是一个桥梁模式的应用，使用驱动程序的应用系统就是抽象化角色，而驱动器本身扮演实现化角色。应用系统和 JDBC 驱动器是相对独立的。应用系统动态地选择一个合适的驱动器，然后通过驱动器向数据库引擎发出指令就可以访问数据库中的数据。

工厂模式

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。

44. 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。

(1) MVC 模式，在 J2EE 项目开发中主要用在表示层框架中，很好解决视图和流程控制。在项目中采用的 Struts、WebWork 等框架。

(2) DAO 模式，在项目开发中主要用在数据层，封装数据的访问操作，为业务层提供数据服务。

(3) IoC 模式，在项目开发中业务层有大量对象，他们之间存在依赖关系，可以使用 IoC 模式减少他们之间的代码耦合，提高系统的可扩展性。实际项目中使用的 Spring 框架来实现业务组件的装配。

(4) Observer 模式，在 Servlet 的监听器中应用到了观察者模式。

(5) singleton 单例模式和 Factory 工厂模式结合使用在项目中无需使用者了解过多的细节就可获取有关的对象实例。比如 Hibernate 项目中通过 SessionFactory 来获取 Session。

(6) bridge 模式，在项目中使用 JDBC 驱动访问数据库。

45. Hibernate 中离线查询与在线查询的区别

`Criteria` 和 `DetachedCriteria` 的主要区别在于创建的形式不一样，`Criteria` 是在线的，所以它是由 `Hibernate Session` 进行创建的；而 `DetachedCriteria` 是离线的，创建时无需 `Session`，`DetachedCriteria` 提供了 2 个静态方法 `forClass(Class)` 或 `forEntityName(Name)` 进行 `DetachedCriteria` 实例的创建。

46. C#数据类型有哪些？

`int` 整型 `double` 双精度型 `string` 字符串型 `char` 字符型 `bool` 布尔型

47. 抽象类能否被实例化？抽象类的作用是什么？

抽象类一般不能被实例化；

抽象类通常不是由程序员定义的，而是由项目经理或模块设计人设计抽象类的原因通常是为了规范方法名 抽象类必须要继承，不然没法用，作为模块设计者，可以把让底层程序员直接用得方法直接调用，而一些需要让程序员覆盖后自己做得方法则定义称抽象方法

48. ASP.NET页面传递值的方式有哪些

- 1 通过 URL链接地址传递
- 2 通过 post 方式。
- 3 通过 session
- 4 通过 Application
- 5 通过 Server.TransferSend.aspx:

49. 如何定义一个 MDI(.NET 技术)

定义 MDI父窗口应该是设置窗体的 MdiParent 属性 = true

50. WCF是什么？

Windows Communication Foundation (WCF) 是由微软发展的一组数据通信的应用程序开发接口，它是 .NET 框架的一部分，由 .NET Framework 3.0 开始引入，与 Windows Presentation Foundation 及 Windows Workflow Foundation 并行为新一代 Windows 操作系统以及 WinFX 的三个重大应用程序开发类库

51. 常用的包，类，接口，各举 5 个。

类 String Integer Long File Date

包 java.lang.*;java.io.*;java.util.*;java.swt.*;java.math.*;

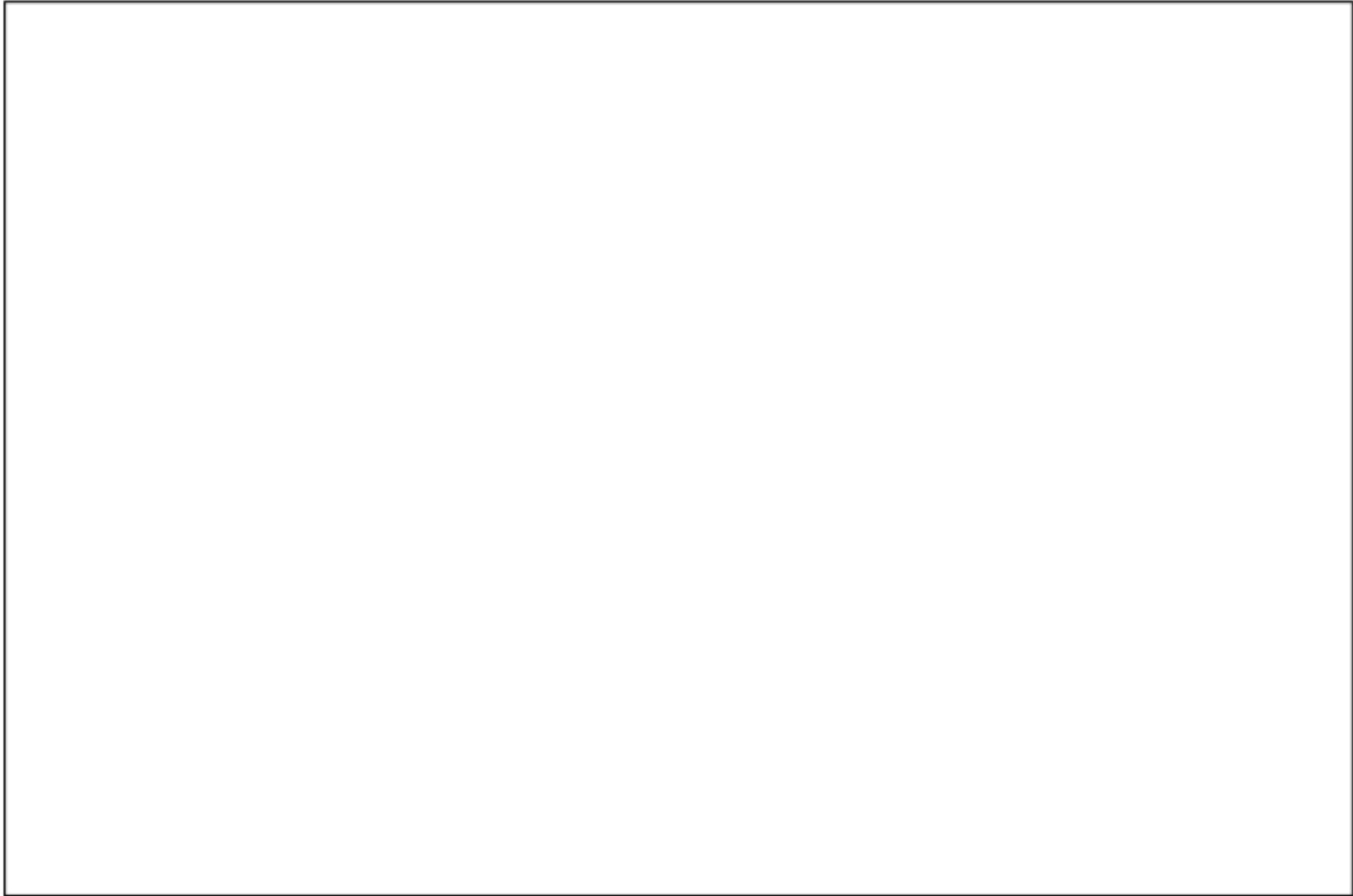
接口：Comparable ,Connection,Map,List,Runnable,

52. 网上购物系统的类图，用例图。

用例图：



类图：



53. 给你一个整数类型的数组，按从小到大的顺序进行排列！
public static void fun(int [] a,int c){

```

for(int i=0;i<a.length-1;i++){
    for(int j=i;j<a.length-1;j++){
        if(a[i]<a[j]){
            t=a[i];
            a[i]=a[j];
            a[j]=t;
        }
    }
    System.out.print(a[i]+" ");
}
}

```

54. 查询出 M 页的 N 行数据 (分页的实现 , 求第 M 页的记录数)

Select top N from table where id not in (select top (M-1)*N id from table)

55. LinkedList 与 ArrayList 内部是如何实现的 (更深入的问了 LinkedList 与 ArrayList 的区别)

ArrayList 的内部实现是基于内部数组 Object[] , 所以从概念上讲 , 它更像数组 , 但 LinkedList 的内部实现是基于组连接的记录 , 所以 , 它更像一个链表结构 , 所以 , 它们在性能上有很大的差别。

在 ArrayList 的前面或中间插入数据时 , 必须将其后的所有数据相应的后移 , 这样必然要花费较多时间 , 所以 , 当你的操作是在一系列数据的后面添加数据而不是在前面或中间 , 并且需要随机地访问其中的元素时 , 使用 ArrayList 会提供比较好的性能 ;

而访问链表中的某个元素时 , 就必须从链表的一端开始沿着连接方向一个一个元素地去查找 , 直到找到所需的元素为止 , 所以 , 当你的操作是在一系列数据的前面或中间添加或删除数据 , 并且按照顺序访问其中的元素时 , 就应该使用 LinkedList 了。

如果在编程中 , 两种情形交替出现 , 这时 , 可以考虑使用 List 这样的通用接口 , 而不用关心具体的实现 , 在具体的情形下 , 它的性能由具体的实现来保证

56. new ArrayList(10) 含义

在内存中连续分配了用来存在 Object 对象的数组 ;

57. 面向对象主要内容 ? 有什么作用 ?

就分析设计层面来说 , 面向对象技术提供了更高级的抽象能力以及更多的方法和工具 , 如设计模式 ; 就实现维护层面来说 , 面向对象技术使代码重用更容易 , 且使代码具有可扩展性 , 便于软件的维护和升级更新。

58. 如何移动一个数 使 $62-63=1$ 成立

$62-63=1$ 如何移动 2 的 6 次方等于 $64-63=1$ 把 2 移动到 6 的左下方

59. 递归实现 1,1,2,3,5,8, ? . 第 30 个数是多少 ?

```

public static int Foo(int i)
{
    if (i <= 0)

```



```

        return 0;
    else if(i > 0 && i <= 2)
        return 1;
    else return Foo(i - 1) + Foo(i - 2);
}
int i=Foo(30);
System.out.println(i);

```

60. JAVA实现向数据库添加一列

```

Connection con = null;
ResultSet rs = null;
Class.forName("com.microsoft.jdbc.sqlserver.SQLServerDriver");
String url="jdbc:microsoft:sqlserver://localhost:1433;DatabaseName=db_name";
Connection con = DriverManager.getConnection(url,"","");
StateManager sm =con.createStatement();
String sql = " alter table student add age int; " ;
rs = sm.excute(sql);

```

61. 写出删除表中重复记录的语句 oracle

```

delete from people
where peopleId in (select peopleId from people group by peopleId having
count(peopleId) > 1) and rowid not in (select min(rowid) from people group

```

62. 用什么方法使服务器关闭之后， session 所保存的信息不会丢失？
使用 cookie

63. 如果页面没有 <html></html> 标签，会出现什么错误代码？
没有什么变化

64. 抽象类是否可以没有抽象方法？为什么？

可以在 java 中用 abstract 关键字来修饰一个类时，这个类叫做抽象类。

抽象类中不一定要包含 abstract 方法，但一个类中包含了 abstract 方法，则这个类必须声明为 abstract 类。

65. servlet 中的 init 方法什么时候被调用？
容器首次创建实例的时候

66. 用 4 个 0，用你所知道的数学方法计算出 24
0 的阶乘等于 1 即 0 ! = 1 那么 4 个 0 就是 4 了
又 4 的阶乘为 2 4 4 ! = 2 4

67. linux 中进程的查看和调度分别用什么命令？
ps -A 查看 renice top

68. 集合类有哪些？有哪些方法？

集合类 ArrayList LinkedList HashSet HashMap

方法： add(),remove(),put(),addAll(),removeAll()

69. OSI 七层模型 tcp 四层模型是什么样的

OSI 七层参考模型：物理层、数据链路层、网络层、传输层、应用层、会话层、表示层

TCP/IP: 第一层网络接口层第二层 第二层 ??网间层 第三层 ??传输层 第四层 ??应用层

70. JAVA测试模式 (ZZ)：

模式（一） Main 模式

模式（二） toString 模式

模式（三） Equal 模式

模式（四） Internal Tester Class 模式

模式（五） Extern Tester Class 模式

71. 一学生表！有班及 id，学号！成绩！一求平均成绩！二求比平均成绩高的学生的所有信息

select id,avg(成绩) from table group by id

select * from table where 成绩 > (select 成绩 from (select id,avg(成绩) 成绩 from table group by id) as a)

72. jdk 的新特性

“JDK1.5”的一个重要主题就是通过新增一些特性来简化开发，这些特性包括泛型，for-each 循环，自动装包 / 拆包，枚举，可变参数，静态导入。使用这些特性有助于我们编写更加清晰，精悍，安全的代码。

73. j2ee 的规范？

SUN公司定义的一个开发分布式企业级应用的规范。它提供了一个多层次的分布式应用模型和一系列开发技术规范。多层次分布式应用模型是指根据功能把应用逻辑分成多个层次，每个层次支持相应的服务器和组件，组件在分布式服务器的组件容器中运行（如 Servlet 组件在 Servlet 容器上运行，EJB 组件在 EJB 容器上运行），容器间通过相关的协议进行通讯，实现组件间的相互调用 J2EE 规范定义了以下四个层次。1. 客户端层 (Client Tier) 2.Web 层 3. 业务层 (Business Tier) 4. 企业信息系统层

74. ajax 的工作原理？

Ajax 基本上就是把 JavaScript 技术和 XMLHttpRequest 对象放在 Web 表单和服务器之间。当用户填写表单时，数据发送给一些 JavaScript 代码而不是直接发送给服务器。相反，JavaScript 代码捕获表单数据并向服务器发送请求。同时用户屏幕上的表单也不会闪烁、消失或延迟。换句话说，JavaScript 代码在幕后发送请求，用户甚至不知道请求的发出。更好的是，请求是异步发送的，就是说 JavaScript 代码（和用户）不用等待服务器的响应。因此用户可以继续输入数据、滚动屏幕和使用应用程序。

然后，服务器将数据返回 JavaScript 代码（仍然在 Web 表单中），后者决定如何处理这些数据。它可以迅速更新表单数据，让人感觉应用程序是立即完成的，表单没有提交或刷新而用户得到了新数据。JavaScript 代码甚至可以对收到的数据执行某种计算，再发送另一

个请求，完全不需要用户干预！这就是 XMLHttpRequest 的强大之处。它可以根据需要自行与服务器进行交互，用户甚至可以完全不知道幕后发生的一切。结果就是类似于桌面应用程序的动态、快速响应、高交互性的体验，但是背后又拥有互联网的全部强大力量。

75. .Xml 特点？

XML是一种可扩展标记语言 (XML) 是 Web 上的数据通用语言。它使开发人员能够将结构化数据，从许多不同的应用程序传递到桌面，进行本地计算和演示。XML 允许为特定应用程序创建唯一的数据格式。它还是在服务器之间传输结构化数据的理想格式。

XML主要具有以下几个特点：

简洁有效

XML是一个精简的 SGML, 它将 SGML的丰富功能与 HTML的易用性结合到 Web应用种，它保留了 SGML的可扩展功能，这使得 XML从根本上有区别于 HTML。并且 XML种还包括可扩展格式语言 XSL(Extensible Style Language) 和可扩展链接语言 XLL (Extensible Linking Language) 使得 XML的显示和解析更加方便快捷。

易学易用

XML对 SGML进行了精简，它抛弃了 SGML中不常用的部分，方便用户编写 Web页面同时也给设计人员实现 XML浏览器降低了困难。

开放的国际化标准

XML是 W3C正式批准的，它完全可用于 Web和工具的开发。XML具有标准的名域说明方法，支持文档对象模型标准、可扩展类型语言标准、可扩展链接语言标准和 XML指针语言标准。使用 XML可以在不同的计算机系统间交换信息，而且还可以跨越国界和超越不同文化疆界交换信息。

高效可扩充

XML支持复用文档片断，使用者可以发明和使用自己的标签，也可以与他人共享，可延伸性大。在 XML中，可定义一组无限量的标准，可以有效地进行 XML文件的扩充。

76. oracle 中除了数据库备份，还有什么方法备份？

Oracle 数据库有三种标准的备份方法，它们分别是导出 / 导入 (EXP/IMP)、热备份和冷备份。导出备份是一种逻辑备份，冷备份和热备份是物理备份。

77. j2ee 的常用设计模式

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)

78. 由 aaa 字符串得到 1000 个 a 写个程序

```
StringBuilder sBuilder = new StringBuilder();
for(int i=0;i<333;i++){
    sBuilder.Append("aaa");
}
sBuilder.Append("aaa".subString(0,1));
```

79. servlet 与 cgi 的区别

CGI应用开发比较困难，因为它要求程序员有处理参数传递的知识，这不是一种通用的技能。CGI不可移植，为某一特定平台编写的 CGI 应用只能运行于这一环境中。每一个 CGI 应用存在于一个由客户端请求激活的进程中，并且在请求被服务后被卸载。这种模式将引起很高的内存、CPU开销，而且在同一进程中不能服务多个客户。Servlet 提供了 Java 应用程序的所有优势——可移植、稳健、易开发。使用 Servlet Tag 技术，Servlet 能够生成嵌于静态 HTML页面中的动态内容。Servlet 对 CGI 的最主要优势在于一个 Servlet 被客户端发送的第一个请求激活，然后它将继续运行于后台，等待以后的请求。每个请求将生成一个新的线程，而不是一个完整的进程。多个客户能够在同一个进程中同时得到服务。一般来说，Servlet 进程只是在 Web Server 卸载时被卸载。

80. SOCKET中有几种连接方式，区别？

Sockets 有两种主要的操作方式：面向连接的和无连接的。无连接的操作使用数据报协议，无连接的操作是快速的和高效的，但是数据安全性不佳。面向连接的操作使用 TCP 协议。面向连接的操作比无连接的操作效率更低，但是数据的安全性更高。

81. Socket 如何获取本地 ip 地址？

```
public InetAddress getLocalAddress()
```

82. truncate 与 delete 的区别？（delete from table 和 truncate table 的区别！）

truncate 是 DDL 语言，delete 是 DML 语言。DDL 语言是自动提交的，命令完成就不可回滚。truncate 的速度也比 delete 要快得多。

详细说明：

相同点：truncate 和不带 where 子句的 delete，以及 drop 都会删除表内的数据。

不同点：

1. truncate 和 delete 只删除数据不删除表的结构（定义）。

drop 语句将删除表的结构被依赖的约束（constraint），触发器（trigger），索引（index）；依赖于该表的存储过程 / 函数将保留，但是变为 invalid 状态。

2.delete 语句是 dml，这个操作会放到 rollback segment 中，事务提交之后才生效；如果有相应的 trigger，执行的时候将被触发。

truncate, drop 是 ddl，操作立即生效，原数据不放到 rollback segment 中，不能回滚。操作不触发 trigger。

3.delete 语句不影响表所占用的 extent，高水线（high watermark）保持原位置不动。

显然 drop 语句将表所占用的空间全部释放。

truncate 语句缺省情况下空间释放到 minextents 个 extent，除非使用 reuse storage；truncate 会将高水线复位（回到最开始）。

4. 速度，一般来说：drop > truncate > delete

5. 安全性：小心使用 drop 和 truncate，尤其没有备份的时候。否则哭都来不及。使用上，想删除部分数据行用 delete，注意带上 where 子句。回滚段要足够大。

想删除表，当然用 drop。

想保留表而将所有数据删除。如果和事务无关，用 truncate 即可。如果和事务有关，或者想触发 trigger，还是用 delete。

83. xml 有哪些解析技术，及区别？

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

84. 文件和目录 (i/o) 操作，怎么列出某目录下所有文件？某目录下所有子目录，怎么判断文件或目录是否存在？如何读写文件？

列出某目录下所有文件：调用 `listFile()`，然后判断每个 `File` 对象是否是文件可以调用 `isFile()`，判断是否是文件夹可以调用 `isDirectory()`，判断文件或目录是否存在：调用 `exists()` 方法，读写文件使用 `FileReader` 和 `FileWriter` 两个类即可

85. 怎么用 java 反射得到一个类的私有方法？

```
package test;
```

```
public class TestClass {
    private String testMethod(String value){
        return "test:" +value;
    }
}
```

```
try {
    // 得到 test.TestClass 类
    Class c=Class.forName("test.TestClass");
    // 得到 testMethod 的方法
    Method m=c.getDeclaredMethod("testMethod", new Class[]{String.class});
    // 打印完整的方法表示字符串
    System.out.println(m.toGenericString());
    // 调用这个方法
    Object obj=m.invoke(c.newInstance(), new Object[]{"method"});
    // 打印返回结果
    System.out.println(obj);
} catch (SecurityException e) {
    // TODOAuto-generated catch block
    e.printStackTrace();
} catch (IllegalArgumentException e) {
    // TODOAuto-generated catch block
    e.printStackTrace();
} catch (ClassNotFoundException e) {
    // TODOAuto-generated catch block
    e.printStackTrace();
} catch (NoSuchMethodException e) {
    // TODOAuto-generated catch block
    e.printStackTrace();
}
```

```

    } catch (IllegalAccessException e) {
        // TODOAuto-generated catch block
        e.printStackTrace();
    } catch (InvocationTargetException e) {
        // TODOAuto-generated catch block
        e.printStackTrace();
    } catch (InstantiationException e) {
        // TODOAuto-generated catch block
        e.printStackTrace();
    }
}

```

86. 写一个单例模式？描述工厂模式和单例优缺点 举例在什么情况下用

```

class Single{}
public class SingleFactory {
    private SingleFactory();
    private static Single single ;
    public static Single getSingle(){
        if ( single ==null ){
            single =new Single();
        }
        return single ;
    }
}

```

5. 简单工厂模式能够根据外界给定的信息，决定究竟应该创建哪个具体类的对象。通过它，外界可以从直接创建具体产品对象的尴尬局面中摆脱出来。
6. 外界与具体类隔离开来，耦合性低。
7. 明确区分了各自的职责和权力，有利于整个软件体系结构的优化。

缺点：工厂类集中了所有实例的创建逻辑，容易违反 GRASP 的高内聚的责任分配原则。虽然简单工厂模式能够适应一定的变化，但是它所能解决的问题是远远有限的。它所能创建的类只能是事先考虑到的，如果需要添加新的类，则就需要改变工厂类了。（这个问题在工厂方法模式将得到很好的解决）

应用情景 工厂类负责创建的对象比较少 客户只知道传入了工厂类的参数，对于如何创建对象（逻辑）不关心

87. JVM加载 class 文件原理？

所谓装载就是寻找一个类或是一个接口的二进制形式并用该二进制形式来构造代表这个类或是这个接口的 class 对象的过程。

在 Java 中，类装载器把一个类装入 Java 虚拟机中，要经过三个步骤来完成：装载、链接和初始化，其中链接又可以分成校验、准备、解析

装载：查找和导入类或接口的二进制数据；

链接：执行下面的校验、准备和解析步骤，其中解析步骤是可以选择的；

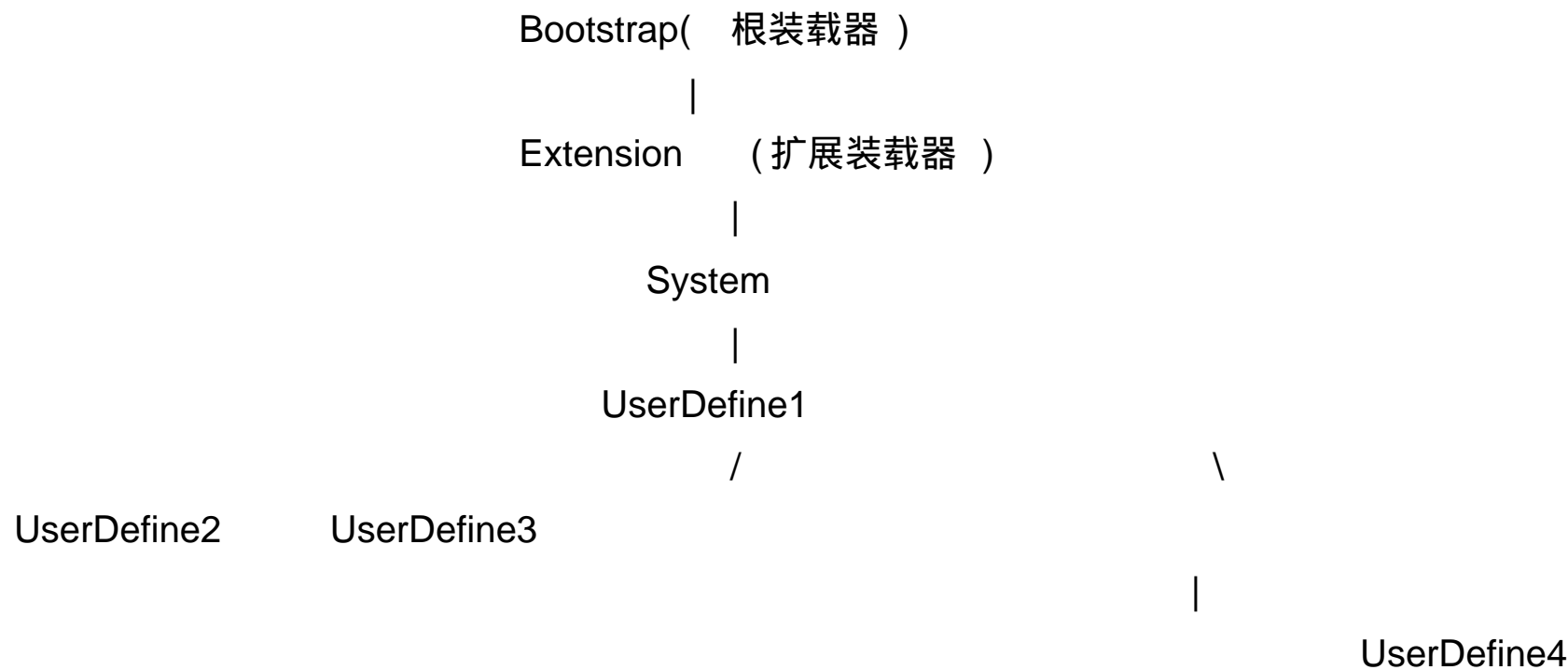
校验：检查导入类或接口的二进制数据的正确性；

- 准备：给类的静态变量分配并初始化存储空间；
- 解析：将符号引用转成直接引用；
- 初始化：激活类的静态变量的初始化 Java 代码和静态 Java 代码块

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

一个 Java 应用程序使用两种类型的类装载器：根装载器 (bootstrap) 和用户定义的装载器 (user-defined) 。

根装载器以某种默认的方式将类装入，包括那些 Java API 的类。在运行期间一个 Java 程序能安装用户自己定义的类装载器。 根装载器是虚拟机固有的一部分， 而用户定义的类装载器则不是，它是用 Java 语言写的，被编译成 `class` 文件之后然后再被装入到虚拟机，并像其它的任何对象一样可以被实例化。 Java 类装载器的体系结构如下所示：



Java 的类装载模型是一种代理 (delegation) 模型。当 JVM 要求类装载器 CL(ClassLoader) 装载一个类时 ,CL 首先将这个类装载请求转发给他的父装载器。 只有当父装载器没有装载并无法装载这个类时 ,CL 才获得装载这个类的机会。这样 , 所有类装载器的代理关系构成了一种树状的关系。树的根是类的根装载器 (bootstrap `ClassLoader`) , 在 JVM 中它以 "null" 表示。除根装载器以外的类装载器有且仅有一个父装载器。 在创建一个装载器时 , 如果没有显式地给出父装载器 , 那么 JVM将默认系统装载器为其父装载器

下面针对各种类装载器分别进行详细的说明：

根 (Bootstrap) 装载器：该装载器没有父装载器，它是 JVM 实现的一部分，从 `sun.boot.class.path` 装载运行时库的核心代码。

扩展 (Extension) 装载器：继承的父装载器为根装载器， 不像根装载器可能与运行时的操作系统有关，这个类装载器是用纯 Java 代码实现的，它从 `java.ext.dirs` (扩展目录) 中装载代码。

系统 (System or Application) 装载器：装载器为扩展装载器，我们都知道在安装 JDK 的 时 候 要 设 置 环 境 变 量 (CLASSPATH) , 这 个 类 装 载 器 就 是 从 `java.class.path`(CLASSPATH 环境变量) 中装载代码的，它也是用纯 Java 代码实现的，同时还是用户自定义类装载器的缺省父装载器。

小应用程序 (Applet) 装载器： 装载器为系统装载器，它从用户指定的网络上的特定目录装载小应用程序代码。

88. Tomcat 的 class 加载的优先顺序一览

1. 最先是 \$JAVA_HOME/jre/lib/ext/ 下的 jar 文件。
2. 环境变量 CLASSPATH 中的 jar 和 class 文
3. \$CATALINA_HOME/common/classes 下的 class
4. \$CATALINA_HOME/commons/endorsed 下的 jar 文件。
5. \$CATALINA_HOME/commons/i18n 下的 jar 文件。
6. \$CATALINA_HOME/common/lib 下的 jar 文件。
(JDBC驱动之类的 jar 文件可以放在这里 , 这样就可以避免在 server.xml 配置好数据源却出现找不到 JDBC Driver 的情况。)
7. \$CATALINA_HOME/server/classes 下的 class 文件。
8. \$CATALINA_HOME/server/lib/ 下的 jar 文件。
9. \$CATALINA_BASE/shared/classes 下的 class 文件。
10. \$CATALINA_BASE/shared/lib 下的 jar 文件。
11. 各自具体的 webapp /WEB-INF/classes 下的 class 文件。
12. 各自具体的 webapp /WEB-INF/lib 下的 jar 文件。

89. CS与 BS联系区别

- 1) java 适合开发 bs 结构 , cs 不是它的强项 C/S 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 客户机上只要安装一个浏览器 (Browser) , 如 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 WebServer 同数据库进行数据交互 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
- 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 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

90. Error 和 exception 的区别与联系

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 , 网络故障等。 不可能指望程序能处理的情况。

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 , 就不会发生的情况。 程序可以捕获到这些异常 , 并进行处理 。

91. 在编写页面的会用到哪些技术 ?

美化页面需要用到 CSS 页面交互使用 JavaScript 、 动态页面需要用到 JSP、 XML 等动态网页技术。

92. 手写个程序在本页面中心弹出一个窗体 , 里面有学生的成绩 , 可以修改学生的成绩 , 并

且可以关闭窗口 , 把每个要用的 jsp 页面显示出来。 (用 JavaScript)

到显示页面的 ShowStudentServlet

```
public void doGet(HttpServletRequest request, HttpServletResponse response)
```

```
        throws ServletException, IOException {
    request.setCharacterEncoding(    "gbk" );
    response.setCharacterEncoding(    "gbk" );
    response.setContentType(    "text/html" );
    StudentDao dao=  new StudentDao();
    List<Student> sts=dao.stlist();
    request.setAttribute(    "sts" , sts);
    request.getRequestDispatcher(    "/showStu.jsp" ).forward(request, response);
}
```

学生成绩的显示页面： showStu.jsp

```
<table >
    <tr >
        <th >学号 </ th >
        <th >姓名 </ th >
        <th >成绩 </ th >
        <th >修改 </ th >
    </ tr >
    <c:forEach    items ="${sts}"    var ="st" >
        <tr >
            <td >${st.sid}    </ td >
            <td >${st.name}    </ td >
            <td >${st.score}    </ td >
            <td ><button
onclick =" window.showModalDialog('UpdateStudentServlet?sid=    ${st.sid }    &rand='+Math.ra
ndom());location.reload();    ">修改 </ button ></ td >
        </ tr >
    </ c:forEach >
</ table >
```

请求修改的 UpdateStudentServlet

```
public void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    request.setCharacterEncoding(    "gbk" );
    response.setCharacterEncoding(    "gbk" );
    response.setContentType(    "text/html" );
    String s=request.getParameter(    "sid" );
    if (s!= null &&!s.equals(    "" )){
        StudentDao dao=  new StudentDao();
        Integer sid=Integer.    parseInt (s);
        Student st=dao.findByld(sid);
        request.setAttribute(    "st" , st);
        request.getRequestDispatcher(    "/updateStu.jsp" ).forward(request,
response);
    } else {
        throw new ServletException(    " 需要传递一个名为 sid 的int 类型参数 ");
    }
}
```

```
    }  
}
```

可以修改成绩的页面 :updateStu.jsp

```
<base href = "<%basePath %>" target = "_self" >  
<form action = "UpdateScoreServlet" method = "post" >  
    <table >  
        <tr >  
            <th>编号 </ th >  
            <td ><input type = "text" name = "sid" value = "${st.sid }"  
readonly = "readonly" /></ td >  
        </ tr >  
        <tr >  
            <th>姓名 </ th >  
            <td ><input type = "text" name = "name" value = "${st.name }"  
readonly = "readonly" /></ td >  
        </ tr >  
        <tr >  
            <th>分数 </ th >  
            <td ><input type = "text" name = "score" value = "${st.score }" /></ td >  
        </ tr >  
        <tr >  
            <td colspan = "2" ><input type = "submit" value = "修改 " /></ td >  
        </ tr >  
    </ table >  
</ form >
```

负责修改成绩的 UpdateScoreServlet

```
public void doPost(HttpServletRequest request, HttpServletResponse response)  
    throws ServletException, IOException {  
  
    request.setCharacterEncoding( "gbk" );  
    response.setCharacterEncoding( "gbk" );  
    response.setContentType( "text/html" );  
    String sid=request.getParameter( "sid" );  
    String score=request.getParameter( "score" );  
    if (sid!= null &&!sid.equals( "" )&&score!= null &&"".equals(score)){  
        StudentDao dao= new StudentDao();  
        Student st= new Student();  
        st.setSid(Integer.parseInt (sid));  
        st.setScore(Double.parseDouble (score));  
        dao.update(st);  
        response.getWriter().println( "<script type='text/javascript'>alert('修  
改成功 ');window.close();</script>" );  
    }  
}
```

93. 用 main 函数输出一到一百的和。

```
#include <stdio.h>

int main() {
    printf("sum:%d\n", sum());
    return 0;
}

int sum() {
    int i;
    int sum = 0;
    for (i = 1; i <= 100; i++)
        sum += i;
    return sum;
}
```

94. 查一下每门课程都大于 80 的学生姓名

学生表 student 分数表 grade

```
select s.name from student s where s.id not in(select g.studentid from grade g where g.marks<=80)
```

或者

```
select s.name from student s where not exists(select 1 from grade g where g.studentid=s.id and g.marks<=80)
```

95. LINUX 如何查看 CPU,IP, 内存？

cat /proc/cpuinfo 查看 CPU

cat /proc/meminfo 查看内存

/sbin/ifconfig 查看 IP 的

96. J2EE 系统访问速度慢，从哪些方面可以优化

J2EE 性能的优化包括很多方面的，要达到一个性能优良的系统，除了关注代码之外，还应该根据系统实际的运行情况，从服务器软硬件环境、集群技术、系统构架设计、系统部署环境、数据结构、算法设计等方面综合考虑

97. J2EE 访问速度慢，怎么样优化

- 1 使用缓冲标记 对于代码逻辑复杂的页面， 利用缓冲标记提高性能的效果比较明显； 反之，效果可能略逊一筹。
- 2 始终通过会话 Bean 访问实体 Bean 一些时候，使用实体 Bean 会导致程序性能不佳。如果实体 Bean 的惟一用途就是提取和更新数据，改成在会话 Bean 之内利用 JDBC 访问数据库可以得到更好的性能
- 3 选择合适的引用机制
- 4 在部署描述器中设置只读属性 实体 Bean 的部署描述器允许把所有 get 方法设置成“只读”。当某个事务单元的工作只包含执行读取操作的方法时， 设置只读属性有利于提高性能， 因为容器不必再执行存储操作

5 缓冲对 EJB Home 的访问

98. haShtable 的原理

原理：通过节点的关键码确定节点的存储位置，即给定节点的关键码 k ，通过一定的函数关系 H (散列函数)，得到函数值 $H(k)$ ，将此值解释为该节点的存储地址

99. struts 中的 prepare 怎么用？

prepare 是在 validate 拦截器之前执行

```
<s:checkboxlist list="users" listKey="id" listValue="name" name="userid"></s:checkboxlist>
```

在使用 struts2 checkboxlist,select 绑定 list 时，有时候会出现以下异常

The requested list key 'users' could not be resolved as a collection/array/map/enumeration/iterator type. Example: people or people.{name} - [unknown location] 这样的错误。是因为 list 的值为空

解决办法是把初始化 list 的工作放到 prepare 拦截器中

```
public class RoleAction extends ActionSupport implements Preparable{
    @Override
    public void prepare() throws Exception {
        // 初始化 list
    }
}
```

100. C#题目 contextmenu 控件如何使用 .. listview

ContextMenu 组件提供与选定对象相关的常用命令的菜单。可以通过向 MenuItems 集合中添加 MenuItem 对象来向快捷菜单中添加项。可以从快捷菜单中永久地移除项；但是在运行时隐藏或禁用项可能更为妥当。ListView 的 ContextMenu 属性设置为 ContextMenu 对象

101. 写一条 SQL 语句，查询姓张学生中平均成绩大于 75 的学生信息

```
select * from student where name in (select name from student
where name like ' 张%' group by name having avg(score) > 75)
```

102. 从以下方面比较 strut1 和 strut2: 线程模式、依赖、可测性、表达式语言、捕获输入、绑定值到页面、类型转换

线程模式：Struts1 Action 是单例模式并且必须是线程安全的，因为仅有 Action 的一个实例来处理所有的请求。单例策略限制了 Struts1 Action 能作的事，并且要在开发时特别小心。Action 资源必须是线程安全的或同步的。

？Struts2 Action 对象为每一个请求产生一个实例，因此没有线程安全问题。（实际上，servlet 容器给每个请求产生许多可丢弃的对象，并且不会导致性能和垃圾回收问题）

Servlet 依赖：

？Struts1 Action 依赖于 Servlet API，因为当一个 Action 被调用时 HttpServletRequest 和 HttpServletResponse 被传递给 execute 方法。

？Struts 2 Action 不依赖于容器，允许 Action 脱离容器单独被测试。如果需要，Struts2 Action 仍然可以访问初始的 request 和 response。但是，其他的元素减少或者消除了直接访问 HttpServletRequest 和 HttpServletResponse 的必要性。

可测性：

? 测试 Struts1 Action 的一个主要问题是 execute 方法暴露了 servlet API (这使得测试要依赖于容器)。一个第三方扩展 - - Struts TestCase - - 提供了一套 Struts1 的模拟对象 (来进行测试)。

? Struts 2 Action 可以通过初始化、设置属性、调用方法来测试, “依赖注入”支持也使测试更容易。

捕获输入:

? Struts1 使用 ActionForm 对象捕获输入。所有的 ActionForm 必须继承一个基类。因为其他 JavaBean 不能用作 ActionForm, 开发者经常创建多余的类捕获输入。动态 Bean (DynaBeans) 可以作为创建传统 ActionForm 的选择, 但是, 开发者可能是在重新描述 (创建) 已经存在的 JavaBean (仍然会导致有冗余的 javabean)。

? Struts 2 直接使用 Action 属性作为输入属性, 消除了对第二个输入对象的需求。输入属性可能是有自己 (子) 属性的 rich 对象类型。Action 属性能够通过 web 页面上的 taglibs 访问。Struts2 也支持 ActionForm 模式。rich 对象类型, 包括业务对象, 能够用作输入 / 输出对象。这种 ModelDriven 特性简化了 taglib 对 POJO 输入对象的引用。

表达式语言:

? Struts1 整合了 JSTL, 因此使用 JSTL EL。这种 EL 有基本对象图遍历, 但是对集合和索引属性的支持很弱。

? Struts2 可以使用 JSTL, 但是也支持一个更强大和灵活的表达式语言 - - "Object Graph Notation Language" (OGNL)。

绑定值到页面 (view):

? Struts 1 使用标准 JSP 机制把对象绑定到页面中来访问。

? Struts 2 使用 "ValueStack" 技术, 使 taglib 能够访问值而不需要把你的页面 (view) 和对象绑定起来。ValueStack 策略允许通过一系列名称相同但类型不同的属性重用页面 (view)。

类型转换:

? Struts 1 ActionForm 属性通常都是 String 类型。Struts1 使用 Commons-Beanutils 进行类型转换。每个类一个转换器, 对每一个实例来说是不可配置的。

? Struts2 使用 OGNL 进行类型转换。提供基本和常用对象的转换器。

103. struts 的工作原理可分为如下 8 步。

1. 读取配置 (初始化 ModuleConfig 对象)

Struts 框架总控制器 (ActionServlet) 是一个 Servlet, 在 web.xml 中被配置成一个自动启动的 Servlet。读取配置文件 struts-config.xml 的配置信息, 为不同的 Struts 模块初始化相应的 ModuleConfig 对象。

2. 用户请求

用户提交表单或调用 URL 向 WEB 应用程序服务器提交一个请求, 请求的数据用 HTTP 协议上传给 WEB 服务器。

3. 填充 FormBean

(*do 请求) 从 ActionConfig 中找出对应该请求的 Action 子类, 如有对应的 Action 且这个 Action 又一个相应的 ActionForm, ActionForm 被实例化并用 HTTP 请求的数据填充其属性, 并保存在 ServletContext 中, 这样他们就可以被其它 Action 对象或 JSP 调用。如果没有对应的 Action, 控制器则直接转发给 JSP 或静态页面。

4. 派发请求

控制器根据配置信息 `ActionConfig` 将请求派发到具体的 `Action`，相应的 `FormBean` 一并传给这个 `Action` 的 `execute()` 方法。

5. 处理业务

`Action` 一般只包含一个 `execute` 方法，它负责执行相应的业务逻辑。执行完毕后返回一个 `ActionForward` 对象，控制器通过该 `ActionForward` 对象来进行转发工作。

6. 返回响应

`Action` 根据业务处理的不同结果返回一个响应对象给总控制器，该目标响应对相对应一个具体的 JSP 页面或另一个 `Action`。

7. 查找响应

总控制器根据业务功能 `Action` 返回的目标响应对象找到对应的资源对象，通常是一个具体的 JSP 页面。

8. 响应用户

JSP 将结果展现给用户。

104. Spring 工作原理

内部最核心的就是 IOC 了，动态注入，让一个对象的创建不用 `new` 了，可以自动的生产，这其实就是利用 `java` 里的反射，反射其实就是在运行时动态的去创建、调用对象，`Spring` 就是在运行时，跟 `xml` `Spring` 的配置文件来动态的创建对象，和调用对象里的方法的 还有一个核心就是 AOP 这个就是面向切面编程，可以为某一类对象 进行监督和控制（也就是在调用这类对象的具体方法的前后去调用你指定的 模块）从而达到对一个模块扩充的功能。这些都是通过配置类达到的

`Spring` 目的：就是让对象与对象（模块与模块）之间的关系没有通过代码来关联，都是通过配置类说明 管理的（`Spring` 根据这些配置 内部通过反射去动态的组装对象）要记住：`Spring` 是一个容器，凡是在容器里的对象才会有 `Spring` 所提供的这些服务和功能

105. Hibernate 的初始化

读取 `Hibernate` 的配置信息 - 创建 `Session Factory`

1) 创建 `Configuration` 类的实例。

它的构造方法：将配置信息 (`Hibernate config.xml`) 读入到内存。

一个 `Configuration` 实例代表 `Hibernate` 所有 `Java` 类到 `Sql` 数据库映射的集合。

2) 创建 `SessionFactory` 实例

把 `Configuration` 对象中的所有配置信息拷贝到 `SessionFactory` 的缓存中。

`SessionFactory` 的实例代表一个数据库存储源，创建后不再与 `Configuration` 对象关联。

缓存 (cache): 指 `Java` 对象的属性（通常是一些集合类型的属性 - 占用内存空间。

106. 在 `main` 方法中将字符串中的。数字排序并输出 `STRING` `A="56.89.5.3.75.98.98.26.15.44"`

```
String s= " 56.89.5.3.75.98.98.26.15.44 " ;
```



```
String s1[]=s. split ( " . " );
Integer ii[]=new Integer[s1.length];
For(int i=0;i<s1.length;i++){
    ii[i]=Integer.parseInt(s1[i]);
}
Arrays.sort(ii);
for(Integer o: ii){
    System.out.println(o+ " s " );
}
```

96Oracle 冷备份的通常步骤

- 1 正常关闭数据库 2 备份所有重要的文件到备份目录（数据文件、控制文件、重做日志文件等）
- 3 完成备份后启动数据库用冷备份进行恢复时，只需要将所有文件恢复到原有位置，就可以启动数据库了
4. 关闭数据库 SQL>shutdown 5 备份文件到备份的目录 6 然后启动数据库 # sqlplus
"/as sysdba"SQL>startup 冷备份完毕！！

107. servlet 创建过程以及 request , response , session 的生命周期？

Servlet 的创建过程：

第一步

```
public class AAA extends HttpServlet{
    实现对应的 doxxx 方法
}
```

第二步：

在 web.xml 中配置

```
<servlet>
    <servlet-name></servlet-name>
    <servlet-class></servlet-class>
</servlet>
```

```
<servlet-mapping>
    <servlet-name></servlet-name>
    <url-pattern></url-pattern>
</servlet-mapping>
```

servlet 的生命周期：

servlet 容器创建 servlet 的一个实例

容器调用该实例的 init() 方法

如果容器对该 servlet 有请求，则调用此实例的 service() 方法

容器在销毁本实例前调用它的 destroy() 方法

销毁并标记该实例以供作为垃圾收集

一旦请求了一个 servlet ，就没有办法阻止容器执行一个完整的生命周期。

容器在 servlet 首次被调用时创建它的一个实例， 并保持该实例在内存中， 让它对所有的请

求进行处理。容器可以决定在任何时候把这个实例从内存中移走。在典型的模型中，容器为每个 servlet 创建一个单独的实例，容器并不会每接到一个请求就创建一个新线程，而是使用一个线程池来动态的将线程分配给到来的请求，但是从 servlet 的观点来看，效果和为每个请求创建一个新线程的效果相同。

一旦请求提交给容器，容器会自动创建相应的 request 、response ,一旦回应完毕则 request 、response 自动销毁。客户端第一次请求时，容器会建立相应的会话，直到会话超时，会话随即销毁。

108. 手写个单例模式？每个公司基本都考

```
public class danli {  
    private static danli dl;  
    private danli(){  
        System.out.println("    单例模式 ");  
    }  
    public static danli getconnection()  
    {  
        if(dl==null)  
        {  
            dl=new danli();  
        }  
        return dl;  
    }  
}
```

109. 解释一下 mvc 以及熟悉的 mvc 框架

答:m 代表模型层， v 代表视图层， c 代表控制层，也就是把一个整体分割成不同的模块，各负责自己的功能，分工明确，提高代码的重用性和方便维护。

在 jsp 设计模式二中，jsp 用来做视图层，servlet 是控制器，dao 则处理相关业务成为模型层。

在 struts2.0, 其中 m是 action,c 是拦截器，v 是 jsp.

110. 解释一下 IOC, 以及 spring 的举例

IOC 称为控制反转，也叫依赖注入，ioc 是 Spring 的核心组件，它通过配置文件，将需要创建的对象以池的方式管理，将实例注入到需要的对象中，是对象依赖于注入而不依赖于实现，解决了各个组件的耦合度，使得项目在后期的维护和扩展上非常方便。如在 ssh 框架整合中，我们将 datasource 对象注入给 sessionFactory ,再将 sessionFactory 注入给 dao 组件，再将 dao 组件注入给 struts 的 Action 组件，在将 action 对象注入给 struts 的拦截器。

111. Oracle 和 Sqlserver 的区别？数据库里面的语言符号表示？

a)体系结构：

ORACLE 的文件体系结构为：

数据文件 .DBF (真实数据) / 日志文件 .RDO / 控制文件 .CTL / 参数文件 .ORA

SQL SERVER 的文件体系结构为：

.MDF (数据字典) / .NDF (数据文件) / .LDF (日志文件)

b) 存储结构：

ORACLE 存储结构：在 ORACLE 将存储单位分为块、区、段、表等；块的大小可设置 (OLTP 块和 DSS 块)；将连续的块组成区，可动态分配区 (区的分配可以是等额的也可以是自增长的)可减少空间分配次数；ORACLE 里表可以分为多个段，段由多个区组成，每个段可指定分配在哪个表空间里 (段的类型分为：数据段、索引段、回滚段、临时段、CASH 段。ORACLE 里还可对表进行分区，可按照用户定义的业务规则、条件或规范，物理的分开磁盘上的数据。这样大大降低了磁盘争用的可能性。

SQL SERVER 存储结构：以页为最小分配单位，每个页为 8K(不可控制，缺乏对页的存储情况的分析机制)，可将 8 个连续的页的组成一个‘扩展’，以进一步减少分配时所耗用的资源。(分配缺乏灵活性)，在 SQL SERVER 里数据以表的方式存放，而表是存放在数据库里。真实数据与数据字典存放在一起。对系统参数信息无安全机制。

c) 操作系统：

Oracle 可在所有主流平台上运行；但 SQL Sever 只在 Window 平台上的表现很好；

d) 运行速度与安全性：Oracle 高于 SQLServer;

e) 价格：Oracle 高于 SQLServer;

112. 浏览器页面与 TOMCat 的交互过程？

当一个 JSP 页面第一次被访问的时候，JSP 引擎将执行以下步骤：

(1) 将 JSP 页面翻译成一个 Servlet，这个 Servlet 是一个 java 文件，同时也是一个完整的 java 程序

(2) JSP 引擎调用 java 编译器对这个 Servlet 进行编译，得到可执行文件 class

(3) JSP 引擎调用 java 虚拟机来解释执行 class 文件，生成向客户端发送的应答，然后发送给客户端

以上三个步骤仅仅在 JSP 页面第一次被访问时才会执行，以后的访问速度会因为 class 文件已经生成而大大提高。当 JSP 引擎收到一个客户端的访问请求时，首先判断请求的 JSP 页面是否比对应的 Servlet 新，如果新，对应的 JSP 需要重新编译。

113. 用 C 编写将一个 100 以内的自然数分解质因数

```
/* 100 以内素数 */
```

```
#include<stdio.h>
```

```
main()
```

```
{
```

```
int i,j;
```

```
for(i=2;i<100;i++)
```

```
{
```

```
for(j=2;j<i;j++)
```

```
{
```

```
if(i%j==0)
```

```
break;
```

```
}
```

```
if(i==j)
```

```
{
```

```

    printf("%d ",i);
}
}
}

/* 分解质因数 */
main()
{
    int n,i;
    printf( "please input a number:\n ");
    scanf( "%d",&n);
    printf( "%d= ",n);
    for(i=2;i <=n;i++)
        while(n!=i)
        {
            if(n%i==0)
            {
                printf( "%d* ",i);
                n=n/i;
            } else{ break; }
        }
    printf( "%d ",n);
    getch();
}

```

114. 用一个方法查出宜个数值类型数组的最大值，用递归方式实现

方法 1

```

public class Test1 {
    public static int a(int[] i,int j){
        if(i.length-1>j){
            if(i[j]>i[j+1]){
                i[j+1]=i[j];
            }
            return a(i,j+1);
        }else{
            return i[i.length-1];
        }
    }
}

```

方法 2 -- 非递归

```

    public static int test(int []num) { int x=0; int
log
num.Length;for(intt=0;t<log;t++){ if(num[t]>x){ x=num[t]; } }return x;}
=

```

方法 3 --- 递归 不改变原数组中的元素

```
public static int getMax(int[]a, int index,int max){
    int len = a.length;
    if(len==1){
        return a[len-1];
    }
    if(index==0){
        max = a[index];
    }
    if(index==len){
        return max;
    }
    if(max<a[index]){
        max = a[index];
    }
    index++;
    return getMax(a,index,max);
}
// 测试
int max = getMax(new int[]{2,5,18,3,38,10,2},0,0);
System.out.println(max);
```

115. JSP 与 SERVLET区别

JSP 在本质上就是 SERVLET但是两者的创建方式不一样 .Servlet 完全是 JAVA 程序代码构成，擅长于流程控制和事务处理，通过 Servlet 来生成动态网页很不直观 .JSP 由 HTML代码和 JSP 标签构成，可以方便地编写动态网页 . 因此在实际应用中采用 Servlet 来控制业务流程，而采用 JSP 来生成动态网页 . 在 struts 框架中，JSP 位于 MVC设计模式的视图层，而 Servlet 位于控制层 .JSP 是 Servlet 技术的扩展，本质上就是 Servlet 的简易方式。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 是 Java 和 HTML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。

116. xmlHttpRequest.open() 里面有几个方法 ？

XMLHttpRequest.open()

初始化 HTTP 请求参数

语法

open (method, url, async, username, password)

method 参数是用于请求的 HTTP 方法。值包括 GET、POST和 HEAD。

(大小写不敏感)。

POST: 用 "POST"方式发送数据 ，可以大到 4MB

GET: 用 "GET"方式发送数据 ，只能 256KB

如果请求带有参数的化实用 POST方式，POST方式将参数放置在页面的隐藏控件内

没有参数使用 GET方式

对于请求的页面在中途可能发生更改的，也最好用 POST方式

)

url 参数是请求的主体。大多数浏览器实施了一个同源安全策略，并且要求这个 URL 与包含脚本的文本具有相同的主机名和端口。

async 参数指示请求使用应该异步地执行。如果这个参数是 false，请求是同步的，后续对 send() 的调用将阻塞，直到响应完全接收。

如果这个参数是 true 或省略，请求是异步的，且通常需要一个 onreadystatechange 事件句柄。

username 和 password 参数是可选的，为 url 所需的授权提供认证资格。如果指定了，它们会覆盖 url 自己指定的任何资格。

117. java 优缺点

先说优点吧，比较显见一点

1. 指针 .

任何一个学过 C 或者 C++的人都知道指针的操作对于他们是很重要的，为什么，指针能够支持内存的直接操作，这样的直接操作能够带来的是效率上的提高。但是任何一个东西都是有副作用的，指针给程序员带了的混乱也是无可比拟的。对于一个不是很强劲的 C 程序员而言，指针是要命的（对我而言也是，我几乎不会写 C代码）。而 JAVA就为我们省去了这样的麻烦，或者说 JAVA完整的限制了对内存的直接操作，让程序员能够安心去搞业务逻辑，而不用管自己后院的那堆内存有没有人来收拾。

2. 垃圾回收

其实垃圾回收也是 JAVA 对于内存操作的限制之一，也大大解放了程序员的手脚。但是也正是这样的一个内存保姆的存在导致 JAVA 程序员在内存上几乎没有概念... 一个纯粹的 JAVA 程序员对于内存泄露这样的问题是从来没有概念的，因为他从来不用担心，因为大多数情况下即便他大手大脚也不会有什么问题的。

3. 平台无关性

平台无关性的真正意义在于程序的可移植性高，能够在多个平台平滑迁移。这点就特别适合网络应用。

好吧，然后我们再来看看缺点吧：

1 效率

1.1 当然首先就是内存操作限制带来安全的同时带来的负面效果了.....

1.2 其次呢，就是面对对象的一个很重要的特点所带来的——多态导致的。

1.3 平台无关性带来的。由于需要平台无关，所以 JAVA是不能直接调用 OS的 API 的，JAVA需要调用自己的 API 来隐式地调用 OS的 API，大家可想而知这样的效果。多一层调用，

必然多一层代码需要加载到栈内了。

2 逆向编译问题

由于 JAVA程序是动态连接的，从一个类到另外一个类的引用是符号化的。在静态连接的可执行程序中，类之间的只是直接的指针或者偏移量，`int`。相反地，在 `JAVAclass` 文件中，指向另一个类的引用通过字符串清楚地标明了所指向的类的名字。如果引用指向一个字段的话，这个字段的名称和描述符会被详细说明。如果引用指向一个成员变量，这个成员变量的名称和描述符（方法的返回类型，方法参数的数量和类型）也会被详细说明。

3 线程调度问题

Java 虽然是一个总进程下的多线程，但是线程间的调度没有非常的清晰，由于为了能够支持多种系统的不同的线程支持策略，JAVA只能采取这样笼统而又松散的线程规范。

118. 写一个单例 Singleton 代码

```
public class Singleton {
    private Singleton();
    private static Singleton singleton ;
    @SuppressWarnings("unused")
    private static Singleton getSingleton(){
        if ( singleton ==null ){
            singleton =new Singleton();
        }
        return singleton ;
    }
}
```

119. Mvc 各部分用什么实现？

视图层用 `jsp` , `javascript`

作用 -- 显示数据，接受用户输入数据

控制层用 `servlet`

作用 -- 接收视图层数据，传输 给业务逻辑层（即模型层）

模型层用普通 `javaBean`

作用 -- 业务类的实现，如：数据库操作

120. ejb 是基于哪些技术实现的，和 `javabean` 的区别？

EJB 包括 `Session Bean`、`Entity Bean`、`Message Driven Bean`，基于 `JNDI`、`RMI`、`JAT` 等技术实现 EJB 是一个关于用 `JAVA` 语言开发的可部署的服务器端组件的组件体系结构。它是一个技术协议，能使组件运行于任何应用服务器，专门用来解决商务问题 `JAVABEAN`是 `JAVA` 类，是由属性、事件和方法组成的 `JAVA`组件，它们可以用来组成 `JAVA`应用程序

121. 适配器模式解释：

把一个类的接口变换成客户端所期待的另一种接口，从而使原本因接口原因不匹配而无

法一起工作的两个类 能够一起工作。适配类可以根据参数返还一个合适的实例给客户端

122. 进程和线程的区别：

线程是一种操作系统对象，代表着一个进程中要被执行的代码的路径。每一个 WIN32 应用程序至少有一个线程 -- 通常称为主线程或默认线程 -- 但应用程序可以自由地创建其他线程来执行其他任务！

进程是程序的一次动态执行过程，它对应了从代码加载、执行到执行完毕的一个完整过程，这个过程也是进程本身从产生、发展到消亡的过程 线程是比进程更小的执行单位。一个进程在其执行过程能够中，可以产生 多个线程，形成多条执行线索。每条线索，即每个线程也有它自身的产生、存在和消亡过程，也是一个动态的概念

123.

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)

124. 写一个程序，把一个文件的数组按对角线做对称变换，并输出！

一个正方形里面全数字，写一个程序，成对角线转变！ 我做的这个是 3 行 3 列的对角互换，也许转换规则不一样

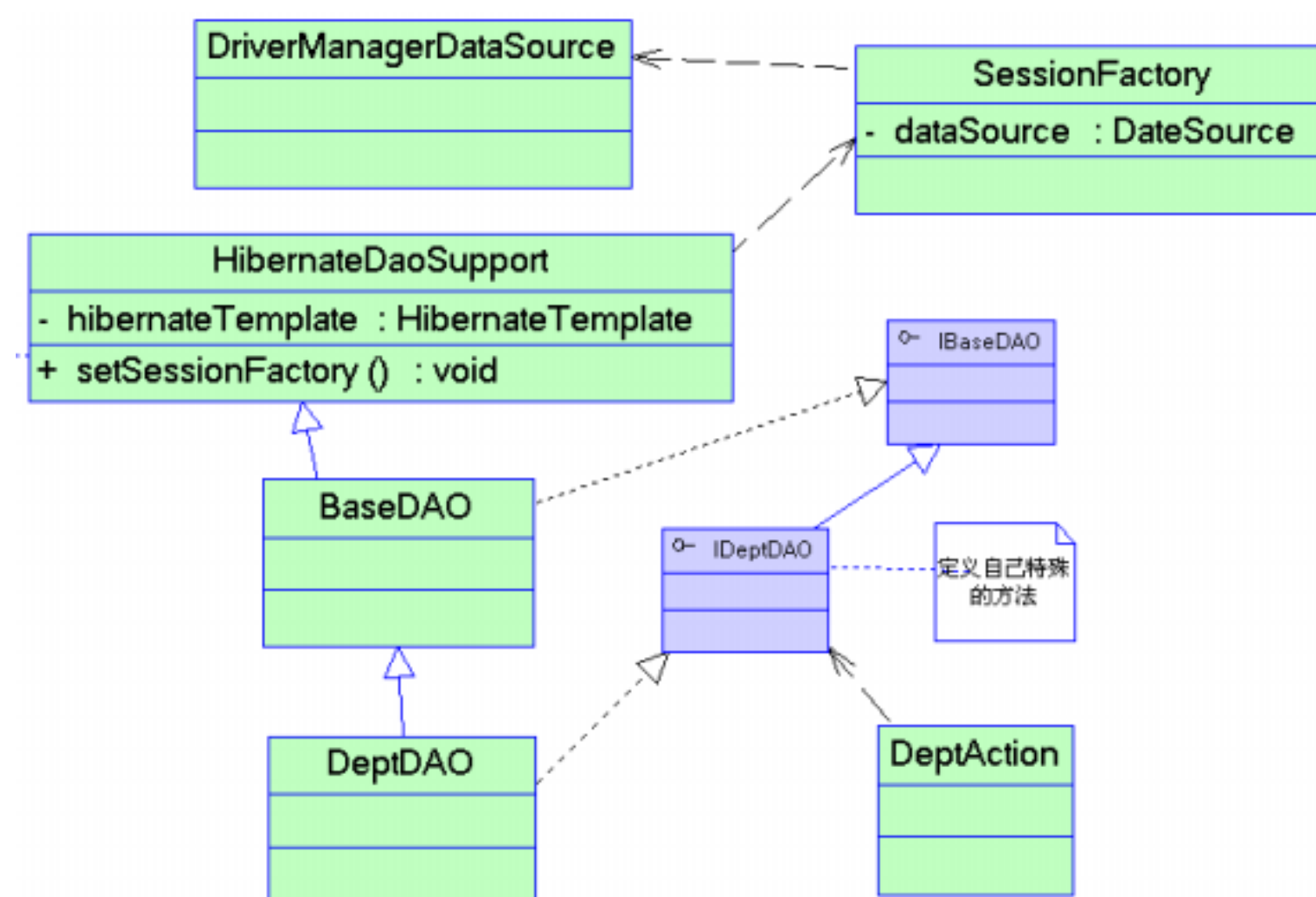
```
public class testMain {
    public static void main(String[] args) {
        int a[][]=new int[3][3];
        int c=1;
        // 初始化数据
        for(int i=0;i<3;i++){
            for(int j=0;j<3;j++){
                a[i][j]=c++;
            }
        }
        System.out.println("    转换之前： ");
        for(int i=0;i<3;i++){
            for(int j=0;j<3;j++){
                System.out.print("a["+i+"]"+"["+j+"]="+a[i][j]+" ");
            }
            System.out.println("\n");
        }
        for(int i=0;i<3;i++){
            for(int j=0;j<3;j++){
                if((i+1<3&&j+1<3)&&i==j&&i!=0&&i!=3-i){
                    int temp=a[i-1][j-1];
                    a[i-1][j-1]=a[i+1][j+1];
                    a[i+1][j+1]=temp;
                    temp=a[i-1][j+1];
```

```

        a[i-1][j+1]=a[i+1][j-1];
        a[i+1][j-1]=temp;
    }
}
}
System.out.println("    转换之后 : ");
for(int i=0;i<3;i++){
    for(int j=0;j<3;j++){
        System.out
print("a["+i+"]"+"["+j+"]="+a[i][j]+" ");
    }
    System.out.println("\n");
}
}
}

```

125. 再 SSH框架中，用流程表示处理过程。从 Request, Response



126. 人民币小写转成大写，写个 java 程序和思路（写 10100.00 转成汉字壹万零壹百整）

```

public class Money {
    public static void main(String[] args) {
        // 金额大小写转换
        float money = 2038.0f;
        int m1, m2, m3, m4, m5, m6, temp;
        temp = (int) (money * 100);
        System.out.println(money);

        m1 = temp % 1000000 / 100000;
    }
}

```

```

m2 = temp % 100000 / 10000;
m3 = temp % 10000 / 1000;
m4 = temp % 1000 / 100;
m5 = temp % 100 / 10;
m6 = temp % 10;

char c1, c2, c3, c4, c5, c6;
c1 = transe(m1);
c2 = transe(m2);
c3 = transe(m3);
c4 = transe(m4);
c5 = transe(m5);
c6 = transe(m6);
String str = "";
if (m1 != 0) {
    str += c1 + "千";
}
if (m2 != 0) {
    str += c2 + "百";
}
if (m1 != 0 && m2 == 0 && (m3 != 0 || m4
!= 0)) {
    str += "零";
}
if (m3 != 0) {
    str += c3 + "十";
}
if (m2 != 0 && m3 == 0 && m4 != 0) {
    str += "零";
}
if (m4 != 0) {
    str += c4;
}
str += "元";
if (m5 == 0 && m6 == 0) {
    str += "整";
}
if (m5 != 0) {
    str += c5 + "角";
}
if (m5 == 0 && m6 != 0) {
    str += "零";
}
if (m6 != 0) {

```

```

        str    +=  c6  +  "分";
    }

    System.out.print(str    +  "\n");
    System.out.println(".....");

}

public  static  char  transe(int    m)  {
    char  r  =  ' ';
    switch  (m)  {
    case  0:
        r  =  '零';
        break;
    case  1:
        r  =  '壹';
        break;
    case  2:
        r  =  '贰';
        break;
    case  3:
        r  =  '叁';
        break;
    case  4:
        r  =  '肆';
        break;
    case  5:
        r  =  '伍';
        break;
    case  6:
        r  =  '陆';
        break;
    case  7:
        r  =  '柒';
        break;
    case  8:
        r  =  '捌';
        break;
    case  9:
        r  =  '玖';
        break;

    }
    return  r;
}

```

```

    }
}

```

127. 针对一个分期付款，总期为 1 年，给定分期金额，期数和开始还款时间，计算出各期还款日期。

```

package demo;

import java.util.Calendar;
import java.util.Date;

public class TestDemo {

    // 分期付款，总期为 1 年，给定分期金额，期数和开始还款时间
    // 计算出各期还款日期
    public void huankuan(double amount,int num,Date start){
        int period = 365/num; // 一年中分期间隔天数

        Calendar cal = Calendar.getInstance();
        cal.set(Calendar.YEAR, start.getYear()+1900);
        cal.set(Calendar.MONTH, start.getMonth());
        cal.set(Calendar.DATE, start.getDate());

        for(int i=1;i<=num;i++){
            System.out.println("第" + i + "期还款日期: " + cal.getTime().toLocaleString());
            cal.add(Calendar.DATE, period);
        }
    }

    public static void main(String[] args) {
        TestDemo demo = new TestDemo();
        demo.huankuan(20000.00, 1, new Date());
    }
}

```

128. 用 java 写从 1 加到 100

sum = (1+i)*i/2 i=100 最简单的方法

```

public class Hello5050{
    public static void main(String[] args){
        for (int i=1;i<=100;i++){
            {
                System.out.println(++i);
            }
        }
    }
}

```

```
}  
}
```

129. SSH的优点和缺点

优点： ssh 把数据库，实体，配置都整合，封装

层次的结构（偏于 mvc 模式）

而且框架的融合，代码的耦合都很不错

对于编程的持续化，扩展化，维护化都很棒，SSH 已经把底层实现封装好了，开发人员可以更专注于业务处理，

缺点： 出现错误容易搞的复杂

如果某个框架不是很熟悉的话（特别是 spring ）那就很麻烦

运行速度比较慢

130. 用 JDBC来实现访问数据库记录可以采用下面的几个步骤：

- 1、 通过驱动器管理器获取连接接口。
- 2、 获得 Statement 或它的子类。
- 3、 限制 Statement 中的参数。
- 4、 执行 Statement 。
- 5、 查看返回的行数是否超出范围。
- 6、 关闭 Statement 。
- 7、 处理其它的 Statement
- 8、 关闭连接接

131. overload 和 override 的区别

override 是方法重载，用在同一个类中，是几个方法的名字相同，返回值相同，但是参数列表不同， 举例来说就像构造函数， 可以有多个构造函数，并且每个的参数列表都不同，这样可以用多种方式构造对象。

overload 是方法覆盖，用在父子类中，是方法名字相同，参数列表也相同，声明形式都相同，但是子类方法的权限不允许小于父类，不允许抛出比父类更多的异常

132. 最常见的 runtime exception 运行时异常？

数字转换异常，类异常 ClassCastException NumberFormatException

ArrayIndexOutOfBoundsException: 数组越界 ClassCastException: 类型转换错误

ArithmeticException : 算术错误，典型的 就是 0 作为除数的时候。

IllegalArgumentException : 非法参数 IndexOutOfBoundsException

NullPointerException

SystemException ArrayStoreException

133. 描述工厂模式和单例优缺点 举例在什么情况下用。

单例 保证一个类只有单一的实例， 也就是说你无法通过 New或 CreateInstance 来创建这个类的一个新实例 好处：当一个对象在程序内部只能有一个实例的时候， 它可以保证我们不会重复创建， 而是始终指向同一个对象。 缺点就是 它就是在内存上共享， 都可以去访问它， 而且多个用户访问的都是同一个实例，会造成线程不安全。

```

/**
 *   IOC 模式简单实例
 */

/**
 *   运行类
 */
public class MainClass {
    /**
     *   主函数
     */
    public static void main(String[] args) {
        try {
            PrinterFactory.createPrinter().printByString("Hello
World~!");
        } catch (Exception ex) {
            System.out.println(ex.toString());
        }
    }
}

/**
 *   Printer 接口
 */
interface IF_Printer {
    /**
     *   接口 printByString 方法声明
     */
    public void printByString(String str);
}

/**
 *   MyPrinter 实现 Printer 接口
 */
class MyPrinter implements IF_Printer {
    public void printByString(String str) {
        System.out.println(str);
    }
}

/**
 *   IF_Printer 对象工厂，用于创建实现接口的类对象
 */
class PrinterFactory {

```

```

    /**
     * 工厂方法，返回 IF_Printer 接口实例
     */
    public static IF_Printer createPrinter() throws InstantiationException,
        ClassNotFoundException, IllegalAccessException {
        String str = "MyPrinter";// 通过字符串寻找实现接口的类，字符串可从文件中读取获得，从而实现 IOC 模式
        return (IF_Printer) Class.forName(str).newInstance();// 返回 IF_Printer 接口实例
    }
}

```

134. 创建一个序列号，从 1000 开始，每次增加 1，最大到 999999

```
create sequence newstudno increment by 1
```

```
start with 10001 maxvalue 99999
```

135. 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。

```

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);          //??? 这个前面不要加 Thread 吗
            t.start();
        }
    }
    private synchronized void inc(){          //??? 为什么要加锁呢 不加可以吗
        j++;
        System.out.println(Thread.currentThread().getName()+"-inc:"+j);
    }          //???currentThread() 有什么作用呢是什么方法
    //???getname 呢

    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();      }      }      }      }

```

136. 十六进制的 216 转换十进制是多少：

216 是 16 进制，转 10 进制：

$= 2 \times 16^2 + 1 \times 16^1 + 6 \times 16^0$

$= 512 + 16 + 6$

$= 536$

137. Java 中的 XML 解析方式： dom 和 jdom 解析

Java 中处理 XML 文档的标准 API 有两种，即 XML 的简单 API (SAX, Simple API for XML) 和文档对象模型 (DOM Document Object Model)，其他的 API，如 JDOM, dom4j 等。基于 DOM 的 XML 分析器将一个 XML 文档转换成一个对象模型的集合（通常称 DOM 树），应用程序正是通过对这个对象模型的操作，来实现对 XML 文档数据的操作。通过 DOM 接口，应用程序可以在任何时候访问 XML 文档中的任何一部分数据。

基于树的处理的优点和缺点

优点

树在内存中是持久的，因此可以修改它以便应用程序能对数据和结构作出更改

可以在任何时候在树中上下导航，而不是像 SAX 那样是一次性的处理

DOM API 是读写的，而不象 SAX 那样是只读的。

缺点

在内存中构建一个 DOM 树开销比较大

创建一棵 DOM 树可能是一个缓慢的过程

DOM 不象 SAX 那样可以提供很多 XML 文档中的信息。比如非分析实体、标记、属性类型和 DTD 中的声明。

SAX 是一个基于事件的 API。解析器向一个事件处理程序发送事件，比如元素开始和元素结束，而事件处理器则处理该信息。应用程序本身就能够处理该数据。原始的文档仍然保留完好无损

优点：

可以解析任意大小的文件；

适合创建自己的数据结构；

适合小信息子集；

简单；

快速

缺点：

不能对文档做随机存取；

难以实现复杂的查询；
不能使用文档类型定义（ DTD）；
不可获取词法信息；
SAX是只读的；
当前的浏览器不支持 SAX；

JDOM是 java document object model 的简称， JDOM兼顾了 DOM和 SAX的优点，它会提供适配器用来选择具体的 XML解析器。 JDOM是一个源代码开发的项目，它基于树型结构，利用纯 Java 的技术对 XML文件实现解析、生成、序列化以及多种操作。在 JDOM 中，XML元素就是 Element 的实例，XML属性就是 Attribute 的实例，XML文档本身就是 Document 的实例。因此创建一个新 JDOM对象就如在 Java 语言中使用 new 操作符一样容易。 JDOM使用标准的 Java 编码模式。只要有可能，它使用 Java new 操作符而不故弄玄虚使用复杂的工厂化模式，使对象操作即便对于初学用户也很方便。

138. UML几种图及用途？

单一的图形不可能包含系统所学的所有信息，更不可能描述系统的整体结构，所以系统通常是从多个不同的方面来描述的，分别对应 UML的几种视图：

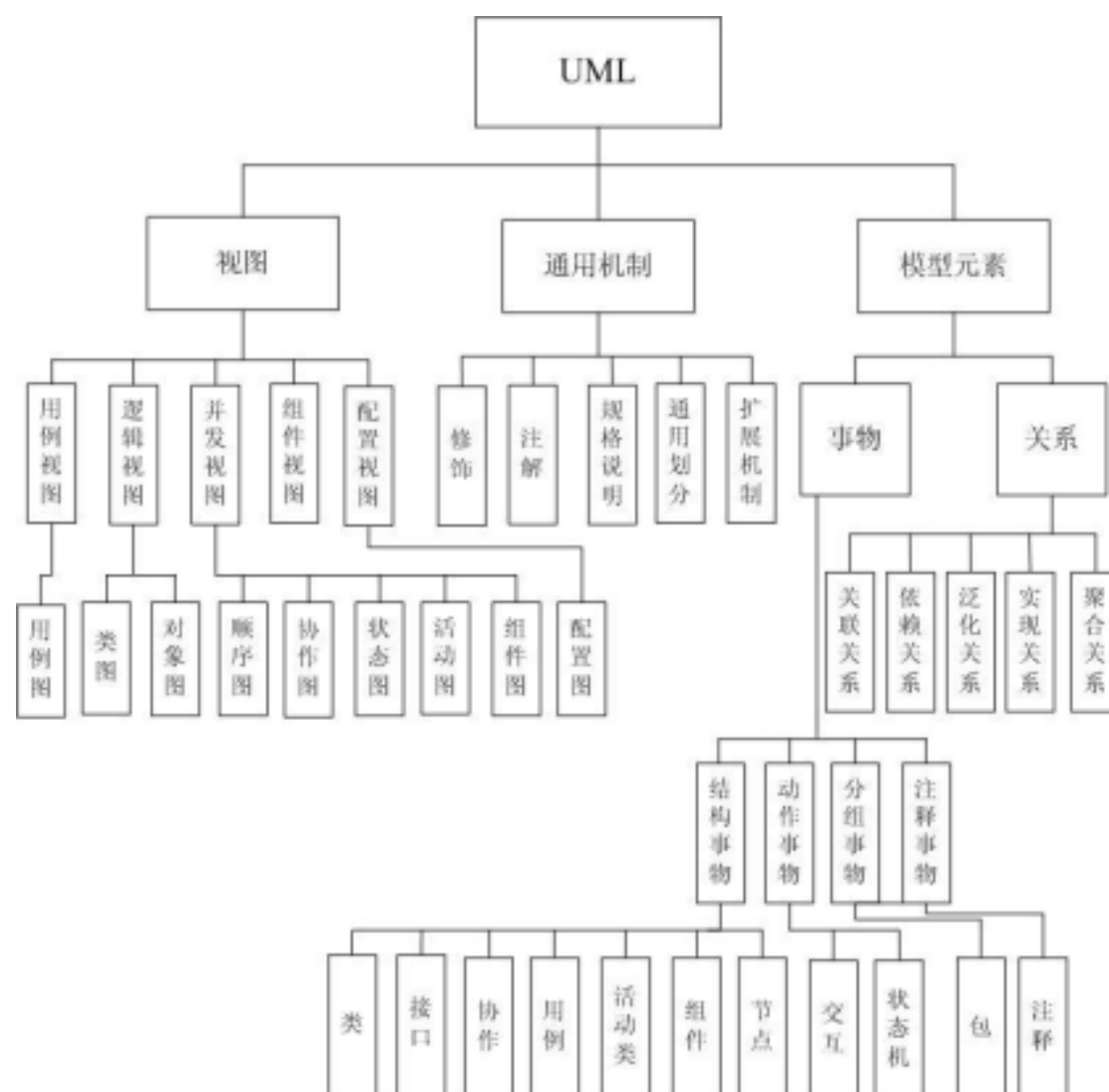
用例视图 -- 从使用者角度描述系统功能

逻辑视图—描述系统内部如何实现

组件视图—描述系统有哪些组件 / 模块构成

并发视图—描述系统内部的通信与并发问题

配置视图—描述系统软硬件的配置与位置关系



用例图—描述系统的参与者与用例之间的关系

类图—描述系统中的概念及它们之间的关系

对象图—描述系统中实例及它们之间关系

状态图—描述系统中对象所具备的状态，及触发状态改变的事件

时序图—描述对象之间发送消息的时间顺序

协作图—描述对象之间协作关系

活动图—描述执行过程中涉及的活动及状态变化

组件图—描述系统组件之间的关系

配置图—描述系统的硬件及软件的物理结构

139. 列举你接触过的框架，说明特点和原理

struts 框架具有组件的模块化，灵活性和重用性的优点，同时简化了基于 MVC 的 web 应用程序的开发，缺点： Taglib 是 Struts 的一大优势，但对于初学者而言，却需要一个持续学习的过程，甚至还会打乱你网页编写的习惯 Struts 将 MVC 的 Controller 一分为三，在获得结构更加清晰的同时，也增加了系统的复杂度，工作原理：用一个 servlet 拦截用户的请求，并分发到具体的 action 中去处理，再将返回的 response 进行封装，这是处理页面流程的机制，另外还有一些标签，非常容易在页面进行显示。

Hibernate 是一个开放源代码的对象关系映射框架，它对 JDBC 进行了非常轻量级的对象封装，使得 Java 程序员可以随心所欲的使用对象编程思维来操纵数据库。Hibernate 可以应用在任何使用 JDBC 的场合，既可以在 Java 的客户端程序使用，也可以在 Servlet/JSP 的 Web 应用中使用，最具革命意义的是，Hibernate 可以在应用 EJB 的 J2EE 架构中取代 CMP，完成数据持久化的重任。

Spring 也表示是一个开源框架，是为了解决企业应用程序开发复杂性由 Rod Johnson 创建的。框架的主要优势之一就是其分层架构，分层架构允许您选择使用哪一个组件，同时为 J2EE 应用程序开发提供集成的框架。Spring 使用基本的 JavaBean 来完成以前只可能由 EJB 完成的事情。然而，Spring 的用途不仅限于服务器端的开发。从简单性、可测试性和松耦合的角度而言，任何 Java 应用都可以从 Spring 中受益。

140. Struts Hibernate Spring 在程序中用到哪些

典型的 J2EE 三层结构，分为表现层、中间层（业务逻辑层）和数据服务层。三层体系将业务规则、数据访问及合法性校验等工作放在中间层处理。客户端不直接与数据库交互，而是通过组件与中间层建立连接，再由中间层与数据库交互。

表现层是传统的 JSP 技术，自 1999 年问世以来，经过多年的发展，其广泛的应用和稳定的表现，为其作为表现层技术打下了坚实的基础。

中间层采用的是流行的 Spring+Hibernate，为了将控制层与业务逻辑层分离，又细分为以下几种。

Web 层，就是 MVC 模式里面的“C”（controller），负责控制业务逻辑层与表现层的交互，调用业务逻辑层，并将业务数据返回给表现层作组织表现，该系统的 MVC 框架采用 Struts。

Service 层（就是业务逻辑层），负责实现业务逻辑。业务逻辑层以 DAO 层为基础，通过对 DAO 组件的正面模式包装，完成系统所要求的业务逻辑。

DAO 层，负责与持久化对象交互。该层封装了数据的增、删、查、改的操作。

PO, 持久化对象。通过实体关系映射工具将关系型数据库的数据映射成对象, 很方便地实现以面向对象方式操作数据库, 该系统采用 Hibernate 作为 ORM 框架。

Spring 的作用贯穿了整个中间层, 将 Web 层、Service 层、DAO 层及 PO 无缝整合, 其数据服务层用来存放数据。

一个好的框架可以让开发人员减轻重新建立解决复杂问题方案的负担和精力; 它可以被扩展以进行内部的定制化; 并且有强大的用户社区来支持它。框架通常能很好的解决一个问题。然而, 你的应用是分层的, 可能每一个层都需要各自的框架。仅仅解决 UI 问题并不意味着你能够很好的将业务逻辑和持久性逻辑和 UI 组件很好的耦合。

141. 你对面向对象思想的理解?

面向对象 (Object Oriented, OO) 是当前计算机界关心的重点, 它是 90 年代软件开发方法的主流。面向对象的概念和应用已超越了程序设计和软件开发, 扩展到很宽的范围。如数据库系统、交互式界面、应用结构、应用平台、分布式系统、网络管理结构、CAD 技术、人工智能等领域。

面向对象的特征

(1) 对象唯一性。

每个对象都有自身唯一的标识, 通过这种标识, 可找到相应的对象。在对象的整个生命期中, 它的标识都不改变, 不同的对象不能有相同的标识。

(2) 分类性。

分类性是指将具有一致的数据结构 (属性) 和行为 (操作) 的对象抽象成类。一个类就是这样一种抽象, 它反映了与应用有关的重要性质, 而忽略其他一些无关内容。任何类的划分都是主观的, 但必须与具体的应用有关。

(3) 继承性。

继承性是子类自动共享父类数据结构和方法的机制, 这是类之间的一种关系。在定义和实现一个类的时候, 可以在一个已经存在的类的基础之上来进行, 把这个已经存在的类所定义的内容作为自己的内容, 并加入若干新的内容。

继承性是面向对象程序设计语言不同于其它语言的最重要的特点, 是其他语言所没有的。

在类层次中, 子类只继承一个父类的数据结构和方法, 则称为单重继承。

在类层次中, 子类继承了多个父类的数据结构和方法, 则称为多重继承。

在软件开发中, 类的继承性使所建立的软件具有开放性、可扩充性, 这是信息组织与分类的行之有效的方法, 它简化了对象、类的创建工作量, 增加了代码的可重用性。

采用继承性, 提供了类的规范的等级结构。通过类的继承关系, 使公共的特性能够共享, 提高了软件的重用性。

(4) 多态性 (多形性)

多态性是指相同的操作或函数、过程可作用于多种类型的对象上并获得不同的结果。不同的对象，收到同一消息可以产生不同的结果，这种现象称为多态性。多态性允许每个对象以适合自身的方式去响应共同的消息。多态性增强了软件的灵活性和重用性。

数组和链表的理解，及优缺点？

142. 链表是一种常见的数据组织形式，它采用动态分配内存的形式实现。需要时可以用 new 分配内存空间，不需要时用 delete 将已分配的空间释放，不会造成内存空间的浪费。

A 从逻辑结构来看

A-1. 数组必须事先定义固定的长度（元素个数），不能适应数据动态地增减的情况。当数据增加时，可能超出原先定义的元素个数；当数据减少时，造成内存浪费。

A-2. 链表动态地进行存储分配，可以适应数据动态地增减的情况，且可以方便地插入、删除数据项。（数组中插入、删除数据项时，需要移动其它数据项）

B 从内存存储来看

B-1.（静态）数组从栈中分配空间，对于程序员方便快捷，但是自由度小

B-2. 链表从堆中分配空间，自由度大但是申请管理比较麻烦。

=====

===

数组中的数据在内存中的按顺序存储的，而链表是随机存储的！

要访问数组中的元素可以按下标索引来访问，速度比较快，如果对他进行插入操作的话，就得移动很多元素，所以对数组进行插入操作效率很低！

由于链表是随机存储的，链表在插入，删除操作上有很高的效率（相对数组），如果要访问链表中的某个元素的话，那就得从链表的头逐个遍历，直到找到所需要的元素为止，所以链表的随机访问的效率就比数组要低

数组在内存中开辟连续的一块区域，如果一个数据要两个内存单元，一组 5 个数据 10 个单元就够了，无需标记其地址，因为数组定义时候标顶了第一个原址的地址，其他四个都知道了。

链表可以是连续的，也可以是不连续的，但一般都是不连续的，尽管在内存中是连续的，我们也不把他当作是连续的，而是把他当作是不连续的，因为如果把他当作是连续的，不如当作是数组了，在某些情况下。一链 5 个数据，如果每个数据本身用 2 个内存单元，那么 10 个单元是不够的，因为每个数据都要表示出下个数据在哪里，所以一个数据本身用 2 个单元，再用 1 个单元表示此链下一个数据在什么地址。

143. 写几个 java.lang.Object 类中的方法名称
equals(); toString(); getClass(); hashCode(); clone() finalize()
notify() wait() notify()

144. js 中加 VAR与不加 VAR的区别？
在全局作用域中可以不使用 var 声明变量，但在声明局部变量时，一定要使用 var 语句。
加了 var: 在局部函数中加了 var 说明是局部变量，只对该函数起作用。
不加 var: 在局部函数中不加的话，说明是给这个全局变量重新赋值。其结果为重新赋值后的。

145. Struts 1 的 Action 类与 Struts 2 的 Action 类区别？

Struts 1 要求 Action 类要扩展自一个抽象基类。Struts 1 的一个共有的问题是面向抽象类编程而不是面向接口编程。

Struts 2 的 Action 类实现了一个 Action 接口，连同其他接口一起实现可选择和自定义的服务。

Struts 1 Action 类是单例类，因只有一个示例控制所有的请求。

Struts 2 Action 对象每一个请求都实例化对象，所以没有线程安全的问题。

146. jsp 有哪些的动作？分别有什么作用？

答：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。

147. 用 oracle 写向 mine 表中差入 miname字段？

```
ALTER TABLE MINE ADD MINNAME VARCHAR2(10);
```

148. oracle 有哪几个对象？

Oracle 数据库对象有表、索引、视图、同义词、序列、存储过程、触发器、PL/SQL 块等

149. 在 oracle 中，有 4 个大对象 (lobs) 类型可用

分别是 blob,clob,bfile,nclob。

150. 定义两个变量 a 和 b，不使用第三个变量，使两个值交换

```
public class testMain {  
    public void test(int a,int b){  
        System.out.println("    交换前 a = "+a);  
        System.out.println("    交换前 b = "+b);  
  
        a=a+b;  
        b=a-b;
```

```

        a=a-b;

        System.out.println("        交换后 a = " +a);
        System.out.print("        交换后 b = "+b);
    }
    public static void main(String args[]){
        new testMain().test(10,13);
    }
}

```

151. 题目： JS 树形菜单

```
<%@ page language="java" import="java.util.*" pageEncoding="gbk"%>
```

```
<%
```

```
String path = request.getContextPath();
```

```
String                                basePath                                =
```

```
request.getScheme()+"://"+request.getServerName()+":"+request.getServerPort()+p
ath+"/";
```

```
%>
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
```

```
<html>
```

```
<head>
```

```
<base href="<%=basePath%>">
```

```
<title>My JSP 'scriptTree.jsp' starting page</title>
```

```
<script type="text/javascript">
```

```
function fun1(){
```

```
window.showModalDialog("${pageContext.request.contextPath}/user.do?method=login
", "700");
```

```
    window.dialogHeight=800;
```

```
}
```

```
function fun(isd){
```

```
    var tablds=document.getElementById(isd);
```

```
    if(tablds.style.display=='none'){
```

```
        tablds.style.display='inline';
```

```
    }else{
```

```
        tablds.style.display='none';
```

```
    }
```

```
}
```

```
</script>
```

```
</head>
```

```

<body>
  <table border="2" borderColor="green" width="75%">
    <tr>
      <td onclick="fun('tableId1');">树型菜单 </td>
      <td>菜单名称 </td>
    </tr>
    <tr>
      <td colspan="2">
        <table id="tableId1" border="2" borderColor="red">
          <tr style="display:inline">
            <td>kongjian </td>
          </tr>
          <tr style="display:inline">
            <td><a href="">xiaoxin </a> </td>
          </tr>
        </table>
      </td>
    </tr>
  </table>
</body>
</html>

```

152. 办公自动化系统用到的技术，分哪些层，有哪些包

使用技术 :DWR + Struts + Hibernate + Spring+Ajax+JavaScript

系统架构分为 : View 层、WEB层、Service 层、DAO层、持久层 (po 层)

153. 解释下 mvc，客户端的请求被发送到哪 ？

.jsp 是 v 层(视图层 view) Servlet 是 c 层(控制层 ,Control) Bean Bin 都是 M 层(后台类,Model)

m 层(模型层)

154. 使用了哪些数据库事务 ？

BeginTransaction 、Commit 和 Rollback

.

155. 对数据库的访问是怎么实现的

将对持久层数据库的基本添加，修改，查找等操作提取到 BaseDAO 中，采用 JavaBean 对数据进行封装，以便对持久层的数据能够很好的处理，实现 BaseDAO 设计对数据库访问的便捷。业务组件通过 DAO 的委托接口调用 DAO 对象，使得上层组件不直接依赖于 DAO 的实现类。

回答步骤：

1. 回答什么是抽象类、什么是接口。
2. 回答抽象类和接口各自特点的区别。
3. 回答他在应用场合的区别。

答：

含有 `abstract` 修饰符的 `class` 即为抽象类，`abstract` 类不能创建的实例对象。含有 `abstract` 方法的类必须定义为 `abstract class`，`abstract class` 类中的方法不必是抽象的。`abstract class` 类中定义抽象方法必须在具体 (Concrete) 子类中实现，所以，不能有抽象构造方法或抽象静态方法。如果的子类没有实现抽象父类中的所有抽象方法，那么子类也必须定义为 `abstract` 类型。

接口 (`interface`) 可以说成是抽象类的一种特例，接口中的所有方法都必须是抽象的。接口中的方法定义默认为 `public abstract` 类型，接口中的成员变量类型默认为 `public static final`。

<< 回答了什么是抽象类、什么是接口 >>

下面比较一下两者的语法区别：

1. 抽象类可以有构造方法，接口中不能有构造方法。
2. 抽象类中可以有普通成员变量，接口中没有普通成员变量
3. 抽象类中可以包含非抽象的普通方法，接口中的所有方法必须都是抽象的，不能有非抽象的普通方法。
4. 抽象类中的抽象方法的访问类型可以是 `public`，`protected` 和 (默认类型，虽然 eclipse 下不报错，但应该也不行)，但接口中的抽象方法只能是 `public` 类型的，并且默认即为 `public abstract` 类型。
5. 抽象类中可以包含静态方法，接口中不能包含静态方法
6. 抽象类和接口中都可以包含静态成员变量，抽象类中的静态成员变量的访问类型可以任意，但接口中定义的变量只能是 `public static final` 类型，并且默认即为 `public static final` 类型。
7. 一个类可以实现多个接口，但只能继承一个抽象类。

<<对比了抽象类和接口各自特点 >>

下面接着再说说两者在应用上的区别：

接口更多的是在系统架构设计方法发挥作用，主要用于定义模块之间的通信契约，对架构组件进行解藕命名其组件实现依赖于抽象而不依赖于实现。而抽象类在代码也可以在架构设计上实现方面发挥作用，可以实现代码的重用。如果对一个组件接口的实现可以通过抽象类实现方法体，再通过子类进行派生，选择重写实现，这也是一种适配器模式的应用。

<<架构上的作用 >>

156. 数据库优化的方案

建立主键，为数据库创建索引，建立存储过程，触发器，可提高查询速度。

157. 冒泡

```
public static void Bubble(int a[]){
    for(int i=0;i<a.length-1;i++){
        for(int j=a.length-1;j>i;j--){
            if(a[j]<a[j-1]){
                a[j]=a[j]+a[j-1];
```

```

        a[j-1]=a[j]-a[j-1];
        a[j]=a[j]-a[j-1];
    }
}
}
}

```

158. 常用的设计模式有哪些？并写出一段程序代码

Factory(工厂模式) , Adapter(适配器模式) , Singleton(单例模式) , State(状态模式) , Observer(观察者模式) 等。

单例模式

```

public class Singleton{
    private static Singleton instance=null;
    private Singleton(){}
    public static Singleton getInstance(){
        if(instance==null){
            instance=new Singleton();
        }
        return instance;
    }
}

```

159. 数据库有哪几种结构？

从数据管理的角度看，数据库通常采用三级模式结构，这是数据库管理系统的内部结构；从数据库最终用户的角度看，数据库的结构可分为集中式结构、分布式结构、客户 / 服务器结构、并型结构，这是数据库的外部的体系

160. css 是什么？ ABSTRAC是什么意思？（层叠样式表，抽象）

Css:（层叠样式表，抽象）

abstract 修饰符可以用于类、方法、属性、事件和索引指示器（ indexer ），表示其为抽象成员，abstract 不可以和 static 、 virtual 一起使用 声明为 abstract 成员可以不包括实现代码，但只要类中还有未实现的抽象成员（即抽象类），那么它的对象就不能被实例化，通常用于强制继承类必须实现某一成员

161. 你对 Oracle 有哪些了解？（大型数据库，安全性高，稳定性高，功能强大，效率低，价格贵，操作麻烦）

162. 你大学所学习的哪些课程？（会哪些 JAVA技术？学过 C 语言， c++,c#,java ）

163. BBS论坛运用了那些技术，如果你单独开发需要多久，你能够独立完成整个代码的编写吗？（jsp 技术， ajax 技术，通信技术， ）

164. 你对博客网有什么认识（典型的网络新事物，指网上写作的一种特定形式和格式。由按时间倒序排列的文档组成的栏目，两侧通常还可以有补充材料，频繁更新，一般大量使用链接）

165. 测试覆盖最强的是 __d_____, 最弱的是 __c____

A. 条件覆盖 B. 条件及判定覆盖 C. 语句覆盖 D. 条件组合覆盖

166. DFD是面向 __c__ 分析方法的描述工具 (DFD数据流程图)

A. 数据结构 B. 数据流 C. 对象 D. 构件

167. 从终端用户的角度来看 , 事务是一个 __B__

A. 对象 B. 原子 C. 操作 D. 子程序

168. 构造器是否可以被重写 ? 不能

169. char 是否可以装一个汉字 ? 可以

```
170. public class Test {  
    public static void add(Integer i){  
        int val=i.intValue();  
        val+=3;  
        i=new Integer(val);  
    }  
    public static void main(String[] args) {  
        Integer i=new Integer(0);  
        add(i);  
        System.out.println(i.intValue());  
    }  
}
```

打印的结果是 0

171. outmermory 是个什么问题? 你是怎么解决的?

内存溢出 解决方法: 手动设置 Heap size 修改 TOMCAT_HOME/bin/catalina.bat

在 “ echo "Using CATALINA_BASE: \$CATALINA_BASE" ” 上面加入以下行:

JAVA_OPTS="-server -Xms800m -Xmx800m -XX:MaxNewSize=256m"

172. &和&&的区别

&, 按位与, 是位运算符

&&, 与, 是逻辑运算符

173. 数组有没有 length() 这个方法, String 有没有 length() 这个方法

数组没有 length() 方法, 但有 length 属性

String 有 length() 方法。

174. String s=new String(“ xyz ”) 创建了几个对象

2 个 string 对象, 一个是 =null 的 s, 一个是 = “ xyz ” 的 string

175. 谈谈 struts 的整个体系?

Struts 的目的是为了减少在运用 MVC设计模型来开发 Web应用的时间。你仍然需要学习和应用该架构, 不过它将可以完成其中一些繁重的工作 Struts 跟 Tomcat、Turbine 等诸多 Apache 项目一样, 是开源软件, 这是它的一大优点, 使开发者能更深入的了解其内部实现机制。

除此之外, Struts 的优点主要集中体现在两个方面: Taglib 和页面导航。 Taglib 是 Struts 的标记库, 灵活动用, 能大大提高开发效率。另外, 就目前国内的 JSP 开发者而言,

除了使用 JSP 自带的常用标记外，很少开发自己的标记，或许 Struts 是一个很好的起点。

关于页面导航，我认为那将是今后的一个发展方向，事实上，这样做，使系统的脉络更加清晰。通过一个配置文件，即可把握整个系统各部分之间的联系，这对于后期的维护有着莫大的好处。尤其是当另一批开发者接手这个项目时，这种优势体现得更加明显。

MVC 即 Model-View-Controller 的缩写，是一种常用的设计模式。MVC 减弱了业务逻辑接口和数据接口之间的耦合，以及让视图层更富于变化。Struts 是 MVC 的一种实现，它将 Servlet 和 JSP 标记（属于 J2EE 规范）用作实现的一部分。Struts 继承了 MVC 的各项特性，并根据 J2EE 的特点，做了相应的变化与扩展

176. 域名格式要求

英文 26 个字母和 10 个阿拉伯数字以及横杠 "-" 可以用作域名。字母的大小写没有区别；每个层次最长不能超过 26 个字母；首字或结尾字不能是 "." 和 "-"；域名不能包括有空格；中文暂时不能用作域名。

177. AJAX 有什么好处？

用于需要查询数据库才能做的页面校验，再比如实时刷新，再比如良好的用户界面以及快速的响应等等 无刷新、减少网络流量

178. Webservice 有什么好处？

跨平台的可互操作性 跨防火墙的通信 应用程序集成 软件和数据重用

179. JAVA 的缓冲机制？ BufferdStream

180. 如何在 javascript 中实现继续？首先 js 是单线程的，现在基本除了 XHR 的 callback 可以另开一个线程以外，还无法实现多线程，其次 timeout 是在本次 js 代码运行完开始计时 timeout 的，所以在一个耗时的函数之外 timeout 或者 interval 一个函数运行本身就是荒谬的，ie 下面 alert，js 函数执行也会阻塞 interval 执行

181. XMLHTTP 的 hibernate 的 save 与 saveOrUpdate 的区别？ get 与 load 的区别？

答案：Transient 状态 -- 离线状态，Persient 状态 -- 持久状态，Detached 状态 -- 脱管状态

1. save () 方法，调用 save 方法时，首先会在 session 缓存中查找保存对象如果实体对象已经处于 Persient 状态，直接返回，否在实行 sql 操作，并将保存的实体对象加入 session 缓存（save 方法不会把实体加入二级缓存的），最后对存在的级联关系进行递归处理。

2. saveOrUpdate () 方法：和 save 方法一样首先在 session 缓存中查找，判断对象是否为保存状态，如果对象处于 Persient，不执行操作，处于 Transient 执行 save 操作，处于 Detached 调用 save 将对象与 session 重新关联（简单的说就是该方法会先看该对象是否已经存在，如果已经存在就更新，否则新增保存）。

3. get 与 load 的区别？load 会先从缓存中查询，如果不存在再到数据库查询；而 get 是直接 从数据库中查询；load 的效率会快点（因为他从缓存中查询）

182. 连接池的原理

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量的池连接。当客户端程序需要连接时，吃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，池驱动就建立一定新的连接

183. 用 javascript 编写脚本小程序，实现点击全选（ ）

```
function selectAll(){
    var item = document.getElementsByName("ids");
    for( var j=0;j<item.length;j++){
        if(!item[j].disabled){
            item[j].checked = true
        }
    }
}
<input type="checkbox" name="ids" value="${N.oid}">
<input type="button" value="    全选 " onclick="selectAll()">
```

184. 说一下垃圾回收的原理，可以直接从内存中回收吗 ？

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和进行清楚回收， 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

185. java 的异常有哪几种，有什么区别。

第一种方式：直接在函数头中 throws SomeException ，函数体中不需要 try/catch

第二种方式：使用 try/catch ，在 catch 中进行一定的处理之后（如果有必要的话）抛出某种异常

第三种方法：使用 try/catch/finally ，在 catch 中进行一定的处理之后（如果有必要的话）抛出某种异常

186. 通过部署描述文件（ web.xml ）可以配置哪些功能？

监听器。。过滤器。。也可以用来验证 XML文件的正确性

187. 说一下 mvc原理，（关于 mvc）

1 、模型（ Model ）

模型是应用程序的主体部分。模型表示业务数据，或者业务逻辑 。

2、视图（ View ）

视图是应用程序中用户界面相关的部分，是用户看到并与之交互的界面。

3、控制器（ controller）

控制器工作就是根据用户的输入，控制用户界面数据显示和更新 model 对象状态。

MVC式的出现不仅实现了功能模块和显示模块的分离，同时它还提高了应用系统的可维护性、可扩展性、可移植性和组件的可复用性

188. servlet 在 MVC中充当的角色 ： 中心控制器

189. 什么是 BS和 CS

B/S 结构是 Browser/Server 的简称

即浏览器 / 服务器结构，程序和数据库放在 SERVER端，客户端通过

浏览器来访问数据，进行数据交换。

B 其实就是 BROWSE 即浏览器，S 就是 SERVER 即服务器

也就是说，用户可以用浏览器就可以享受 SERVER 提供的服务，也就是我们上网的啦

而 CS 呢，C 就是 CLIENT 即客户端，S 就是 SERVER 服务端，

也就是说，用户要使用 SERVER 的服务，需要装上相应的客户端软件，像 QQ 就是这样的。

C/S 又称 Client/Server 或客户 / 服务器模式。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。浏览器通过 Web Server 同数据库进行数据交互。

C/S 的优点是能充分发挥客户端 PC 的处理能力，很多工作可以在客户端处理后再提交给服务器。对应的优点就是客户端响应速度快。

缺点主要有以下几个：

只适用于局域网。而随着互联网的飞速发展，移动办公和分布式办公越来越普及，这需要我们的系统具有扩展性。这种方式远程访问需要专门的技术，同时要对系统进行专门的设计来处理分布式的数据。客户端需要安装专用的客户端软件。首先涉及到安装的工作量，其次任何一台电脑出问题，如病毒、硬件损坏，都需要进行安装或维护。特别是有很多分部或专卖店的情况，不是工作量的问题，而是路程的问题。还有，系统软件升级时，每一台客户机需要重新安装，其维护和升级成本非常高。

对客户端的操作系统一般也会有限制。可能适应于 Win98，但不能用于 win2000 或 Windows XP。或者不适用于微软新的操作系统等等，更不用说 Linux、Unix 等。

B/S 最大的优点就是可以在任何地方进行操作而不用安装任何专门的软件。只要有一台能上网的电脑就能使用，客户端零维护。系统的扩展非常容易，只要能上网，再由系统管理员分配一个用户名和密码，就可以使用了。甚至可以在线申请，通过公司内部的安全认证 (如 CA 证书) 后，不需要人的参与，系统可以自动分配给用户一个账号进入系统。

190. 页面中使用 javascript 的利与弊？ (益端：能减少一些简单的逻辑业务，提高页面的效率，弊端：对于有些不能识别 Javascript 的浏览器存在一定的安全隐患)

191. 在 Linux/Unix 下创建一个角色帐号，自己拥有一切权限，系统用户拥有访问权限，其他用户没有任何权限。

192. SQL 中有个 PreparedStatement 对象，用 Java 实现连接 Oracle 数据库，运用 PreparedStatement 对象。 [ConnDB.java](#)

```
package demo.bean;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;

public class ConnDB {

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

```

        try{
            String driver = "oracle.jdbc.driver.OracleDriver";
            String url="jdbc:oracle:thin:@localhost:1521:yf";
            String username = "scott";
            String password = "tiger";
            DriverManager.registerDriver(new
oracle.jdbc.OracleDriver());
            Connection con = DriverManager.getConnection(url,
username, password);
            PreparedStatement stm = con.prepareStatement("select
sysdate from dual");

            ResultSet rs = stm.executeQuery();
            while(rs.next()){
                System.out.print(rs.getString(1));
            }

        }catch(Exception e){
            e.printStackTrace();
        }
    }
}

```

193. 数据库索引的优点和缺点

一、为什么要创建索引呢（优点）？

这是因为，创建索引可以大大提高系统的性能。

- 第一，通过创建唯一性索引，可以保证数据库表中每一行数据的唯一性。
- 第二，可以大大加快数据的检索速度，这也是创建索引的最主要的原因。
- 第三，可以加速表与表之间的连接，特别是在实现数据的参考完整性方面特别有意义。
- 第四，在使用分组和排序子句进行数据检索时，同样可以显著减少查询中分组和排序的时间。
- 第五，通过使用索引，可以在查询的过程中，使用优化隐藏器，提高系统的性能。

二、建立方向索引的不利因素（缺点）但是，为表中的每一个列都增加索引，是非常不明智的缺点：

- 1 创建索引和维护索引要耗费时间，这种时间随着数据量的增加而增加。
- 2 索引需要占物理空间，除了数据表占数据空间之外，每一个索引还要占一定的物理空间，如果要建立聚簇索引，那么需要的空间就会更大。
- 3 当对表中的数据进行增加、删除和修改的时候，索引也要动态的维护，这样就降低了数据的维护速度。

194. struts2.0 中几种常用的标签

<s:if test=""></s:if> <s:form action=""></s:form>

<s:checkbox></s:checkbox>----- 复 选 框 <s:url></s:url>----- 创 建 url

<s:textarea></s:textarea>----- 文本域输入框

<s:autocompleter></s:autocompleter> <s:a href=""></s:a>----- 超链接，类似于
html 里的 <a>

195. 介绍下 spring

Spring 提供了管理业务对象的一致方法并且鼓励了注入对接口编程而不是对类编程的良好习惯。Spring 的架构基础是基于使用 JavaBean 属性的 Inversion of Control 容器是一个解决了许多在 J2EE 开发中常见的问题的强大框架，还提供了可以和总体的 IoC 容器集成的强大而灵活的 MVC web 框。

196. 介绍下 ajax

AJAX 全称为 “ Asynchronous JavaScript and XML ” (异步 JavaScript 和 XML), 是指一种创建交互式网页应用的网页开发技术

197. 介绍下 hibernate

Hibernate 是连接 Java 应用程序和关系数据库的中间件。它对 JDBC API 进行了封装，负责 Java 对象的持久化。在分层的软件架构中它位于持久化层，封装了所有数据访问细节，使业务层可以专注于实现业务逻辑

198. servlet 和 jsp 页面过滤器 Filter 的作用及配置

过滤器可附加到一个或多个 servlet 或 JSP 页面上，并且可以检查进入这些资源的请求信息
过滤器要好处。 1 它以一种模块化的或可重用的方式封装公共的行为 2 利用它能够将高级访问决策与表现代码相分离

199. 学 JAVA的都看了什么书？

基础很差的人我建议用那本 JavaSE 6.0 编程指南 《最新 Java2 核心技术 卷 1、卷 2》了，不要怕代码难懂，这个一定要克服，当然某些代码 (比如经常提及的画小球、你又不是搞图象的 :)) 就可以跳过去。这当中可以穿插阅读 《Java 实效编程百例》，最后建议大家阅读 《Java 编程思想》 thinking in JA VA 第三版中文版 Core Java (java 核心技术)

200. 函数和过程的区别？

过程和函数区别如下：

- 1、标识符不同：函数的标识符为 FUNCTION 过程为： PROCEDURE
- 2、语法不同：函数中一般不用变量形参，用函数名直接返回函数值；而过程如有返回值，则必须用变量形参返回。
- 3、函数在定义时一定要进行函数的类型说明，过程则不进行过程的类型说明。
- 4、调用方式不同：函数的调用出现在表达式中，过程调用，由独立的过程调用语句来完成。
- 5、过程一般会被设计成求若干个运算结果，完成一系列的数据处理，或与计算无关的各种操作；而函数往往只为了求得一个函数值。

201. Javascript 如何实现继承？

1. 采用对象冒充方式：

原理：构造函数使用 this 关键字给所有属性和方法赋值 ，因为构造函数只是一个函数 ，所以可以使 ClassA 的构造函数成为 classB 的方法，然后调用它。这样 classB 就会收到 classA 的构造函数中定义的属性和方法 。例子：

2. 例：

```
function classA(name)
{
    this.name=name;
    this.showName=function(){alert(this.name);}
}
function classB(name)
{
    this.newMethod = classA;
    this.newMethod(name);
}
obj = new classA("hero");
objB = new classB("dby");
obj.showName(); // print hero
objB.showName(); // print dby
```

说明 classB 继承了 classA 的方法 .

202. 将字符串 str (例： abcrepefd) 中的 rep部分替换成 with字符串 (不能用 jdk 自带的 replace 方法)

答案如下：

```
public void replace(String str,String rep,String with){
    try{
        int i = str.indexOf(rep);
        String s1 = str.substring(0, i);
        String s2 = str.substring(i, rep.length()+i);
        String s3 = str.substring(rep.length()+i, str.length());
        s2 = with;
        String mes = s1+s2+s3;
        System.out.println("    替换前： "+str);
        System.out.println("    替换后： "+mes);
    }catch(Exception e){
        System.out.println("    字符串 "+str+" 中不含有 "+rep+" 字符串！ ");
    }
}
```

203. Java Reflection 是什么？

答：

1.JAVA 反射，Reflection 是 Java 程序开发语言的特征之一，它允许运行中的 Java 程序对自身进行检查，或者说 " 自审 "，并能直接操作程序的内部属性。例如，使用它能获得 Java 类中各成员的名称并显示出来；

2. 一个简单的例子

```
import java.lang.reflect.*;
public class DumpMethods {
    public static void main(String args[]) {
        try {
```

```

Class c = Class.forName(args[0]);
Method m[] = c.getDeclaredMethods();
for (int i = 0; i < m.length; i++)
System.out.println(m.toString());
}
catch (Throwable e) {
System.err.println(e);
}
}}

```

204. 1 到 11 相加是奇数还是偶数 ?
偶数

205. 一个圆上有 6 个点，可以连多少条直线？
15 条线段

206. Stack 堆栈，实现进栈，出栈

```

package t1;

public class mystack {
    private Object[] data;
    private int top=-1;
    private int size;
    public mystack()
    {
        data=new Object[5];
        size=5;
    }
    public mystack(int size)
    {
        data=new Object[size];
        this.size=size;
    }
    public void push(Object obj)
    {
        if(this.isfull())
        {
            return ;
        }
        top++;
        data[top]=obj;
    }
    public Object pop() {
        if(this.isempty())

```

```

        {
            return null;
        }
        Object obj=data[top];
        top--;
        return obj ;
    }

    public boolean isfull()
    {
        if(top==data.length)
        {
            return true;
        }
        else
        {
            return false;
        }
    }
    public boolean isempty()
    {
        if(top==-1)
        {
            return true;
        }
        else
        {
            return false;
        }
    }
}

```

207. 每周的工作效率要提高 24%，每周工作 6 天，那么每天工作效率要提高多少？（ 6%，4%,24% ）
 答： 24%

208. 前一天是星期一，又是第二个工作日，那么今天是星期几？
 答：星期三

209. Hibernate 中的 HQL 和 criteria 的区别？
 答：

1.QBC(Query by Criteria) 查询对查询条件进行了面向对象封装，符合编程人员的思维方式；

2. HQL(Hibernate Query Language) 查询提供了更加丰富的和灵活的查询特性，在涵盖 Criteria 查询的所有功能的前提下，提供了类似标准 SQL 语句的查询方式，同时也提供了更加面向对象的封装；

210. Web service 是什么？它是采用什么通信机制，它的通信协议是什么？

答： Webservice 是跨平台，跨语言的远程调用技术；

1. 它的通信机制实质就是 xml 数据交换；
2. 它采用了 soap 协议（简单对象协议）进行通信

211. Struts 是怎么实现 MVC 的？

Struts 提供 MVC三层模式：

MVC就是 model view controller.

- 1、view 视图层一般运用 jsp 或者 html 进行实现，展现给用户看；
- 2、controller 控制层 struts 有个文件叫 struts-config.xml，该文件里面定义个处理后的跳转页面；
- 3、model 模型层理解不深，可能是逻辑模型

212. java 对象初始化顺序

1. 类的初始化（ initialization class & interface ）
2. 对象的创建（ creation of new class instances ）

顺序：应为类的加载肯定是第一步的，所以类的初始化在前。大体的初始化顺序是：

类初始化 -> 子类构造函数 -> 父类构造函数 -> 实例化成员变量 -> 继续执行子类构造函数的语句

213. 一个字符串中可能存在 A-Z 的全角字符，写一个方法把里面的全角字符转变成半角字符？

答：采用建立字典表进行查找转换

```
public static String translate(String s){
    String qj = " A B C D E F G H I J K L M N O P Q R S T U V W X Y Z ";
    String bj = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
    StringBuffer sb = new StringBuffer();
    for(int i=0;i<s.length();i++){
        char c = s.charAt(i);
        int pos = qj.indexOf(c);
        if(pos>=0){
            System.out.println(c + "," + pos);
            sb.append(bj.charAt(pos));
        }else{
            sb.append(c);
        }
    }
    return sb.toString();
}
```

214. 写一个方法，传入一个 int 型的数字，把它的四个字节码取出来，并且把它按大小顺序通过控制台输出？

```
public static void main(String[] args) {
    int num = -800000000;
    String str = Integer.toBinaryString(num); // 获得 num 的二进制
    if(num>=0) { // 如果输入的数为正数，位数可能不足 32 位，要补 0；负数肯定是 32 位
        if(str.length()<32) { // 二进制不足 32 位，就在前面补 0
            int n0 = 32-str.length(); // 看差几个 0
            String temp = "";
            for(int i=0;i<n0;i++) {
                temp = temp + "0"; // 拼 0
            }
            str = temp + str;
        }
    }
    String s1 = str.substring(0, 8);
    String s2 = str.substring(8, 16);
    String s3 = str.substring(16, 24);
    String s4 = str.substring(24, 32);

    System.out.println(str);
    System.out.println(s1);
    System.out.println(s2);
    System.out.println(s3);
    System.out.println(s4);

    int n1=Integer.parseInt(s1,2);// 以二进制把字符串解析为 10 进制的数
    int n2=Integer.parseInt(s2,2);
    int n3=Integer.parseInt(s3,2);
    int n4=Integer.parseInt(s4,2);
    System.out.println(n1);
    System.out.println(n2);
    System.out.println(n3);
    System.out.println(n4);    // 整数大小自己比较吧

}
```

215. 类的工厂模式怎么实现？

首先确定这些类的父类或共同的接口，担任抽象产品的角色，比如

```
public interface IServer{
    public void calc(int a,int b);
}
```

然后，这些类的具体实现继承某个父类或实现共同的接口，担任具体产品角色，比如 AddServer 类和 SubServer 类

```
public class AddServer implements IServer{
    public void calc(int a,int b){
        System.out.println(a+b);
    }
}
```

```
public class SubServer implements IServer{
    public void calc(int a,int b){
        System.out.println(a-b);
    }
}
```

最后，涉及一个类充当工厂的角色，根据客户端的要求，提供具体的产品，比如

```
public class ServerFactory {
    private static ServerFactory factory;
    private ServerFactory(){}

    public static ServerFactory getInstance(){
        if(factory == null)
            factory = new ServerFactory();
        return factory;
    }

    // 根据客户给出的名字返回对应的产品
    public IServer getServer(String name){
        if("Add".equals(name)){
            return new AddServer();
        }else if("Sub".equals(name)){
            return new SubServer();
        }else{
            return null;
        }
    }

    public static void main(String args[]){
        IServer addserver = ServerFactory.getInstance().getServer("Add");

        addserver.calc(5,3);
    }
}
```

216. Web 容器里面的对象存活周期？

Web 容器里面的对象的存活周期由 JVM 控制。某个对象有容器负责实例化或应用负责实例化，如果某个对象得不到任何引用，则由 JVM 的垃圾收集器负责销毁。

217. struts2 中 action 的对象的生命周期？

用 spring 来控制 action 的生命周期 Struts2 支持通过拦截器堆栈（Interceptor Stacks）为每一个 Action 创建不同的生命周期。堆栈能够根据需求和不同的 Action 一起使用。

218. annotation 的使用方法和用途主要分为几类？

答 1、内建 Annotation —— Java5.0 版在 java 语法中经常用到的内建 Annotation：
2、开发者自定义 Annotation：由开发者自定义 Annotation 类型；
3、使用第三方开发的 Annotation 类型

219. 说说你对 IOC 的理解

答：1. 控制反转（Inversion of Control，英文缩写为 IoC）是一种设计模式；
2. 控制反转还有一个名字叫做依赖注入（Dependency Injection）。简称 DI；
应用控制反转，对象在被创建的时候，由一个调控系统内所有对象的外界实体，将其所依赖的对象的引用，传递给它。也可以说，依赖被注入到对象中。

220. 从 action 类，servlet 依赖，action 执行的 control、线程模式几个方面谈 struts1 和 struts2 的区别。

答：从 action 类上分析：

1. Struts1 要求 Action 类继承一个抽象基类。Struts1 的一个普遍问题是使用抽象类编程而不是接口。
2. Struts2 Action 类可以实现一个 Action 接口，也可实现其他接口，使可选和定制的服务成为可能。Struts2 提供一个 ActionSupport 基类去实现常用的接口。Action 接口不是必须的，任何有 execute 标识的 POJO 对象都可以用作 Struts2 的 Action 对象。

从 Servlet 依赖分析：

3. Struts1 Action 依赖于 Servlet API，因为当一个 Action 被调用时 HttpServletRequest 和 HttpServletResponse 被传递给 execute 方法。
4. Struts2 Action 不依赖于容器，允许 Action 脱离容器单独被测试。如果需要，Struts2 Action 仍然可以访问初始的 request 和 response。但是，其他的元素减少或者消除了直接访问 HttpServletRequest 和 HttpServletResponse 的必要性。

从 action 线程模式分析：

5. Struts1 Action 是单例模式并且必须是线程安全的，因为仅有 Action 的一个实例来处理所有的请求。单例策略限制了 Struts1 Action 能作的事，并且要在开发时特别小心。Action 资源必须是线程安全的或同步的。
6. Struts2 Action 对象为每一个请求产生一个实例，因此没有线程安全问题。（实际上，servlet 容器给每个请求产生许多可丢弃的对象，并且不会导致性能和垃圾回收问题）

221. 如何理解 struts.jsf, struts2 的工作原理

答：1. JSF 和 STRUTS 工作原理大致相同，有一个中心控制器即一个 servlet，

2.JSF: 用一个 JAVABEAN里的属性绑定页面表单值 ,而 struts 用 ActionForm 对应表单值;

3.JSF 在一个 JAVABEAN里处理从页面送来的个值 ,而 struts 则是把 actionForm 的值送给对象的 ACTION来处理,两者都是 MVC模式的 .

222. 整合 spring 与 struts 的方法,那种最好,为什么?

答: 1. 第一种方法 :

Struts 的 Action 继承 Spring 的 ActionSupport 类,并在 Action 中获取 Spring 的 ApplicationContext 。这是最简单的一种整合方式, 但有三个缺点: 第一, Struts 与 Spring 紧密耦合,不能改换到其他 IoC 容器;第二,难以使用 Spring AOP 特性;第三,对于需要使用 DispatchAction 的 Struts 应用无能为力。

2. 第二种方法 :

在 Struts 的配置文件中,以 Spring 的 DelegatingRequestProcessor 类代替 Struts 的 RequestProcessor 类,并在 Spring 的配置文件中定义与 Struts 配置文件中 <action-mappings> 对应的 bean,从而将 Struts 的 Action 与 Spring 分开,并把 Struts 的动作置于 Spring 的控制之下。这种整合方式的优点是将不再依赖 Spring 这个特定的 IoC 容器,但必须依赖 Struts 的 RequestProcessor 类。

3. 第三种方法 :

通过 Spring 的 DelegatingActionProxy 类代理 Struts 的动作,即在 Struts 配置文件中,定义 <action-mappings> 的 type 属性全部改为 DelegatingActionProxy ,而不是具体的类名,并在 Spring 配置文件中定义与 Struts 动作映射对应的 bean,从而将 Struts 的 Action 与 Spring 分开,并把 Struts 的动作置于 Spring 的控制之下。无疑,这是最灵活的一种整合方式。

223. springframework 与 struts 的不同之处

答: 1. SpringFramework 是一个采用了反转控制 (InversionofControl, loc) 策略的基于 J2EE 的轻量级应用 框架。核心是 IoC 容器,对于其它应用,如 数据库 访问,日志等, SpringFramework 多使用现有的、成熟的框架。它采用了模块化的方式,各模块可以共同使用,也可以单独使用其中的一个模块。

2. struts2 利用成熟的 struts1 和 webwork 整合使之操作更加方便 ;

224. spring 使用了哪些设计模式,这样用有什么好处?

Dao 模式 通过实现 DAO,我们达了解耦合的目的 ,使的程序更加的健壮 ,但复杂性增加了;

225. spring 对多种 ORM框架提供了很好的支持,结合事务管理描述 spring 中使用 Hibernate 的方法。

答: spring 中采用 aop 模式注入 hibernate 的 sessionFactory 和事务管理 ,在 dao 中调用 .Spring 集成 hibernate 有两种方式 ,一是 dao 层只与 hibernate 有关,不使用任何 spring 的 api,然后把 dao 注入到 ioc 容器 .二是使用 spring 的 hibernateDaoSupport.事务管理也可以只使用

hibernate 的事务管理 .

226. struts 中 action 是全局的还是局部的，为什么？
是全局

227. Hibernate 有哪几种查询数据的方法？

答:hibernate 在查询方式上有三种方式 :HQL SQL QBC

228. http 和 ftp 的默认使用端口是什么？

答：http : 80 ftp : 21

229. Ping 命令的作用？

答：Ping 命令的主要作用是通过发送数据包并接收应答信息来检测两台计算机之间的网络是否连通。当网络出现故障的时候，可以用这个命令来预测故障和确定故障地点

230. 构造器的名能不能和类的名字相同？

答：可以！并且一定要相同

231. 在一个主方法类可不可以调用一个非静态的方法？

答：可以

232. 一个类中可不可以有 2 个公共的方法？

答：可以

233. Internet 的最顶级的商业域名叫什么？

答：.com

234. GC 是什么，为什么要使用它？

垃圾回收 (garbage collection, GC) 一个跟踪过程，它传递性地跟踪指向当前使用的对象的所有指针，以便找到可以引用的所有对象，然后重新使用在此跟踪过程中未找到任何堆内存。公共语言运行库垃圾回收器还压缩使用中的内存，以缩小堆所需要的工作空间 因为你没有足够多内存，并且，你挺懒，不去自己清理内存，所以就有了 GC

235. 编程题：有一个数据文件 :123 34 17 651234 345.... 这些数据都是随机产生的，编写程序读出该文件，并将其以从大到小的顺序输出到另一个文件中。

```
public void readtext(){
    File file = new File("D:\\test.txt");
    List list= new ArrayList();
    try {
        BufferedReader br=new BufferedReader(new FileReader(file));
        String data = "";
        String line = null;
        while ( (line = br.readLine()) != null) {
            data = data.concat(line);
        }
    }
```

```

        StringTokenizer stoken = new StringTokenizer(data, " ");
        while (stoken.hasMoreTokens()) {
            int i = Integer.parseInt(stoken.nextToken());
            list.add(i);
        }
    } catch (Exception ex) {}
    String[] str = new String[list.size()];
    for(int i=0;i<list.size();i++){
        str[i]=list.get(i);
    }
    Object iTemp= null;
    for(int i=1;i<list.size();i++) {
        for(int j=list.size()-1;j>=i;j--) {
            if(str[j]>str[j-1]) {
                iTemp = str[j-1];
                str[j-1] = str[j];
                str[j] = iTemp;
            }
        }
    }
    String result = "";
    for(int i=0;i<str.length;i++){
        result +=str[i]+" ";
    }
    // 将 result 写入另外一个文件即可。
}

```

236. 从一到十九共十九个数，打印出利用这十九个整数任意多个相加等于 20 所以可能性，每个数字在同一个算式中只出现一次。

```

public void test(){
    Integer[] a = new Integer[19];
    for(int i=1;i<20;i++){
        a[i-1]=i;
    }
    for(int i=0;i<18;i++){
        for(int j=18-i;j<18;j++)
            if(a[i]+a[j]==20)
                System.out.println(a[i]+"+"+a[j]+"="+20);
    }
}

```

237. JSP 内置对象作用，如何取 Cookie 的方法

使用 request 对象的 getCookies() 方法取 cookies

(1) HttpServletRequest 类的 Request 对象

作用：代表请求对象，主要用于接受客户端通过 HTTP 协议连接传输到服务器端的数据。

(2) HttpServletResponse 类的 Response 对象

作用：代表响应对象，主要用于向客户端发送数据

(3) JspWriter 类的 out 对象

作用：主要用于向客户端输出数据；

Out 的基类是 JspWriter

(4) HttpSession 类的 session 对象

作用：主要用于来分别保存每个用户信息，与请求关联的会话；

会话状态维持是 Web 应用开发者必须面对的问题。

(5) ServletContext 类的 application 对象

作用：主要用于保存用户信息，代码片段的运行环境；

它是一个共享的内置对象，即一个容器中的多个用户共享一个 application 对象，故其保存的信息被所有用户所共享。

(6) PageContext 类的 PageContext 对象

作用：管理网页属性，为 JSP 页面包装页面的上下文，管理对属于 JSP 中特殊可见部分中已命名对象的访问，它的创建和初始化都是由容器来完成的。

(7) ServletConfig 类的 Config 对象

作用：代码片段配置对象，表示 Servlet 的配置。

(8) Object 类的 Page (相当于 this) 对象

作用：处理 JSP 网页，是 Object 类的一个实例，指的是 JSP 实现类的实例，即它也是 JSP 本身，只有在 JSP 页面范围之内才是合法的。

(9)Exception

作用：处理 JSP 文件执行时发生的错误和异常

238. 写几个线程安全类，不安全的，支持排序的类名？

线程安全类： Vector 、 Hashtable 、

不安全的类： ArrayList 、 HashMap LinkedList

支持排序的类： TreeSet

JDK 也提供一个对集合类进行线程安全处理的类 Collections 类，可以对一些非线程安全的类进行同步处理。

239. 将 ISO8859-1 字符串转成 GB2312 编码，语句为？

String s=new String(text.getBytes(" iso8859-1 "), " gb2312 ").

240. 说出你用过的 J2EE 标准的 WEB 框架和他们之间的比较？

答：用过的 J2EE 标准主要有： JSP&Servlet 、 JDBC JNDI、JavaMail 、 JTA、XML RMI、WebServices 等。这些标准主要提供了对于 WEB 框架的全方位的支持与整合，通过 Web 容器来管理这些中间件，使得 Web 框架开发实现者，只需注重应用系统的实现，而不必关注与业务不相关的逻辑，大大降低了开发成本和开发周期。同时这些中间件有良好的扩展性、平台不相关性，也使得这个标准成为大多数 Web 框架所采用。

241. 类库有哪些？

Xerces , JDOM dom4j XOM jConfig XStream

242. <jsp:include> 与 <%@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，不能实现参数共享。

243. 怎么样手动启动 Oracle 数据库？

答：第一步：启动 Oracle 在 OS下的监听程序。第二步：启用 OS下的实例服务。第三步打开数据库。

244. WEB SERVICE的理解

WebService 主要是为了使原来各孤立的站点之间的信息能够相互通信、共享而提出的一种接口。Web Service 所使用的是 Internet 上统一、开放的标准，如 HTTP、XML、SOAP(简单对象访问协议)、WSDL等，所以 Web Service 可以在任何支持这些标准的环境(Windows,Linux)中使用。注：SOAP协议(Simple Object Access Protocol, 简单对象访问协议)，它是一个用于分散和分布式环境下网络信息交换的基于 XML的通讯协议。在此协议下，软件组件或应用程序能够通过标准的 HTTP协议进行通讯。它的设计目标就是简单性和扩展性，这有助于大量异构程序和平台之间的互操作性，从而使存在的应用程序能够被广泛的用户访问。

245. 写一个递归编程

```
public class Test {
    static int multiply(int n){
        if(n==1||n==0)
            return n;
        else
            return n*multiply(n-1);
    }

    public static void main(String[] args){
        System.out.println(multiply(10));
    }
}
```

246. AWT和 Swing 的区别？

一般 swing 组件是从 awt 组件扩展的，swing 是轻量组件 而 awt 是重量组件，AWT是比较依赖操作系统组件，功能少，Swing 不依赖，组件多，功能多 窗口风格更多样化 Swing 是由纯 JAVA CODE所写的，解决了 JAVA因窗口类而无法跨平台的问题，使窗口功能也具有跨平台与延展性的特性，而且 SWING不需占有太多系统资源，并且一些组件也采用一些更开放式的开发方式，支持更多的扩展。

247. 哪几个方法可以实现一个线程？

一种是声明 Thread 的子类，重载 Thread 类的方法 run。另一种途径是声明一个类，该类实现 Runnable 接口。然后再实现方法 run。

248. STOP()和 SUSPEND(不推荐使用的原因？

因为 stop 本质上就是不安全的。 停止线程会导致解锁它已锁定的所有监视程序。 suspend 从本质上就是易于死锁的一种操作。

249. JAVA事件有哪些模式？

1. 事件直接驱动模式。它的第一个要求就是性能的要求，需要直接而且快，是必须经常使用的，主要适合于迅速处理 前台的命令，往往是系统架构的重要部分，也是流程控制的主要模式。
2. 监控式事件模式是借助第三者来监控和触发事件，特点是： 有一个观察者置身事外在定期独立运行着。

250. 用 javascript 实现用户登录验证的代码 .

```
<script language=javascript>
function checkSubmit()
{if ((document.form1.name.value)=="")
{
window.alert (' 姓名必须填写 ');
document.form1.name.select();
document.form1.name.focus();
return false;
}
else
return true;
}
}</script>
<form name="form1" onsubmit="javascript:return checkSubmit()">
<input type="text" name="name">
</form>
<%
```

251. uml 三要素 .

事物、关系、图

252. EJB 规范规定的 EJB 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。

253. TRUNCAT和 DELETE有以下几点区别？

- 1、TRUNCAT在各种表上无论是大的还是小的都非常快。如果有 ROLLBACK命令 DELETE将被撤销，而 TRUNCAT则不会被撤销。

2、TRUNCAT是一个 DDL语言，向其他所有的 DDL语言一样，他将被隐式提交，不能对TRUNCAT使用 ROLLBACK命令。

3、TRUNCAT将重新设置高水平

254. http 协议， tcpip 协议解释

答：HTTP: 超文本传输协议 (Hypertext Transfer Protocol)

TCP/IP (Transmission Control Protocol/Internet Protocol) 的简写，中文译名为传输控制协

255. // 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”。

```
public static void main(String[] args) throws Exception{
    String str = "    我 a 爱中华 abc def";
    String str = "    我 ABC汉";
    int num = trimGBK(str.getBytes("GBK"),5);
    System.out.println(str.substring(0,num) );
}

public static int trimGBK(byte[] buf,int n){
    int num = 0;
    boolean bChineseFirstHalf = false;
    for(int i=0;i<n;i++)
    {
        if(buf[i]<0 && !bChineseFirstHalf){
            bChineseFirstHalf = true;
        }else{
            num++;
            bChineseFirstHalf = false;
        }
    }
    return num;
}
```

256. java 程序每天 12 点打印 " hello "

```
public static void main(String[] args) {
    Date date = new Date();
    System.out.println(date.getHours());
    while (true) {
        while (date.getHours() == 12) {
            if (date.getMinutes() == 0) {
                if (date.getSeconds() == 0) {
```


WebService 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。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。它是 Webservice 所采用的主要协议之一。

261. 介绍几种 WEB服务器，问 Tomcat 里面的文件组成？

IIS 、apache、Tomcat

bin: 启动,关闭 Tomcat 的命令。

common/lib: 网络编程的 jar 文件。

conf: 配置文件。

logs: 日志文件。

server: 自带的 web应用 (三个)。

shared: 所有 web应用都可以访问的内容。

temp: 临时。

webapps: 默认站点文件夹。

work:jsp 生成的类。

Tomcat 是一个开放源代码、运行 servlet 和 JSP Web应用软件的基于 Java 的 Web应用软件容器。Tomcat Server 是根据 servlet 和 JSP 规范进行执行的，因此我们就可以说 Tomcat Server 也实行了 Apache-Jakarta 规范且比绝大多数商业应用软件服务器要好。

262. ThreadLocal 在项目中的实际意义？

ThreadLocal 和其它同步机制相比从另一个角度来解决多线程的并发访问，它为每一个线程维护一个和该线程绑定的变量的副本，从而隔离了多个线程的数据，每一个线程都拥有自己的变量副本，从而也就没有必要对该变量进行同步了。还提供了线程安全的共享对象，在编写多线程代码时，可以把不安全的整个变量封装进 ThreadLocal

ThreadLocal 可以大量减少参数的传递，可以使代码简洁，但一个线程会绑定多个自己定义的局部对象，ThreadLocal 是抽象在线程上的对象创建工厂，目前的 Tomcat5 使用了线程池，一个线程处理一个 request，这样 ThreadLocal 对象可以抽象的绑定在 request 生命周期，不会存在线程危机，而且线程池也平衡了这些 ThreadLo

263. 不同的数据库里面有相同的表实现同步更新（从多方面考虑）

-- 用触发器可以实现两个数据表之间的数据同步更新

```
For Example:
Create Trigger room_texas_change_id
on room
for update
as if update(room.cstname)
set texas.cstname=room.cstname
where texas.id=room.id。
```

264. ejb 是基于哪些技术实现的，和 javabean 的区别？

EJB 包括 Session Bean 、Entity Bean 、Message Driven Bean ，基于 JNDI、RMI、JAT 等技术实现 EJB 是一个关于用 JAVA 语言开发的可部署的服务器端组件的组件体系结构。它

是一个技术协议，能使组件运行于任何应用服务器，专门用来解决商务问题 JAVABEANS是 JAVA类，是由属性、事件和方法组成的 JAVA组件，它们可以用来组成 JAVA应用程序

265. 适配器模式解释：

把一个类的接口变换成客户端所期待的另一种接口，从而使原本因接口原因不匹配而无法一起工作的两个类能够一起工作。适配类可以根据参数返还一个合适的实例给客户端

266. Gc是什么，为什么要有 GC

GC是垃圾收集的意思（GarbageCollection），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。

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。

267. switch 可以作用于 byte 吗？可以作用于 long，可以作用于 string 吗？

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。

268. 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？

也许你的答案是在 return 之前，但往更细地说，我的答案是在 return 中间执行，请看下面程序代码的运行结果：

```
public class Test {
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        System.out.println(new Test().test());
    }
    static int test()
    {
        int x = 1;
        try
        {
            return x;
        }
        finally
        {
            ++x;
        }
    }
}
```

```
}
```

----- 执行结果 -----

1

运行结果是 1，为什么呢？主函数调用子函数并得到结果的过程，好比主函数准备一个空罐子，当子函数要返回结果时，先把结果放在罐子里，然后再将程序逻辑返回到主函数。所谓返回，就是子函数说，我不运行了，你主函数继续运行吧，这没什么结果可言，结果是在说这话之前放进罐子里的。

269. 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？对。

如果对象要保存在 HashSet 或 HashMap 中，它们的 equals 相等，那么，它们的 hashCode 值就必须相等。

如果不是要保存在 HashSet 或 HashMap，则与 hashCode 没有什么关系了，这时候 hashCode 不等是可以的，例如 ArrayList 存储的对象就不用实现 hashCode，当然，我们没有理由不实现，通常都会去实现的。

270. 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？

引用传递。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。

271. 在 Oracle 中数据库中的一个表中，这个表没有主键 id 也没有特定标示来查数据，怎么查？

[题意不明：考虑从面试者的角度能是考伪列和 rownum 的内容]
考虑使用 Oracle 的伪列 rowid 来查询

272. 概述反射和序列化

反射 Reflection 是 Java 程序开发语言的特征之一，它允许运行中的 Java 程序对自身进行检查；能直接操作程序的内部属性和方法。

简单来说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，我们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。而把 Java 对象转换为字节序列的过程称为对象的序列化。

273. 简述 WebService 和 Remoting 的特点和运用场景

WebService 主要是为了使原来各孤立的站点之间的信息能够相互通信、共享而提出的一种接口。Web Service 所使用的是 Internet 上统一、开放的标准，如 HTTP、XML、SOAP (简单对象访问协议)、WSDL 等，所以 Web Service 可以在任何支持这些标准的环境 (Windows, Linux) 中使用。Remoting 依赖于特定语言的远程调用机制，比如 Java 的 RMI/IIOP 它的设计目标就是简单性和扩展性，这有助于大量异构程序和平台之间的互操作性，从而使存在的应用程序能够被广泛的用户访问。

274. dao 是什么及作用

DAO组件主要提供数据库访问操作，主要是对各自数据表的 CRUD方法，在一般设计过程中为了考虑松耦合的问题，业务逻辑层中不应当牵涉到任何持久层实际操作，所以可以将各数据源表的操作特性封装到 DAO组件中去，这样增强了系统的可读性、可维护性。

275. Inserted 和 deleted 的含义？

SQLServer 的 DML触发器中两个临时表

inserted 表反映插入或更新操作时插入的记录

deleted 表反映删除或更新操作时删除的记录

276. 3 个框架在项目在项目当中的用， basedao 是用来做什么的。

DAO组件主要提供数据库访问操作，针对不同数据源表进行了封装，这样可以提供其它层的访问接口，使得组件之间解耦，而 BaseDAO是这些所有不同 DAO的公共特性的封装，针对一些可以进一步进行细化的 DAO组行的进一步抽象，使组件的移植性更为灵活。

277. Oracle 中除了数据库 [XX]备份，还有什么方法备份？

1. 导出 / 导入（ EXP/IMP）。

2. 热备份。热备份是在数据库运行的情况下，采用 archive log mode方式备份数据库的方法。所以，如果你有昨天夜里的一个冷备份而且又有今天的热备份文件，在发生问题时，就可以利用这些资料恢复更多的信息。热备份要求数据库在 Archive log 方式下操作，并需要大量的档案空间。一旦数据库运行在 archive log 状态下，就可以做备份了。

3. 冷备份。冷备份发生在数据库已经正常关闭的情况下，当正常关闭时会提供给我们一个完整的数据库。冷备份时将关键性文件拷贝到另外的位置的一种说法。

278. jsp 与 servlet 中请求转发的区别

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，但此转发是连续的转发，在转发过程中请求参数也连续转发到目标地址中。

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。前者对请求是连续的，后者是重新发送一个新的请求，所以只能重新发送新的请求参数，原有的请求参数将不再连续到目标页面。

279. 分页的优化需考虑哪些

sql 语句的优化：ID 排除法、三层嵌套等

表的优化：适当使用索引

280. web 运用程序的稳定、安全需要考虑哪些

Web服务器的性能考虑主要有：并发用户数、事务安全、负载均衡、时段流量、网络带宽

安全性：

关键数据的保护，例如用户数据等

功能服务的正常提供。

网站的防攻击能力

对异常灾害的恢复能力

程序性能：

响应请求并运行得出结果的平均时间

错误的检测和拦截

扩展性

281. 问 spring 的 AOP, mvc 到底是怎么优化程序的结构？

SpringAOP 主要提供了 Pointcut 、 Aspects 等以及它们如何被织入应用程序、代理的方式等进行优化的控制。而 Spring MVC 里面前端控制器叫做 DispatcherServlet 。里面充当 Action 的组件叫做 Controller ，返回的视图层对象叫做 ModelAndView ，提交和返回都要经过过滤。从而提高程序可读性和稳定性。

282. 笔试： 1.LinkedList 和 ArrayList 内部是如何实现的？ new ArrayList (10) 意思

ArrayList 的内部实现是基于内部数组 Object[] ，所以从概念上讲，它更像数组，但 LinkedList 的内部实现是基于一组连接的记录，所以，它更像一个链表结构，所以，它们在性能上有很大的差别。

new ArrayList (10) 构造一个初始容量为 10 的空列表

283. 定义一个变量， Var v =new Object (), 用 js 编写的简便写法是什么？

var v = {};

284. 网上购物系统的类图，用例图。

主要的类有：商品分类 商品 订单 订单项 商品评论

主要用例有：查看商品详情 发表评论 登录 注册 购物车管理 订单生成 查询订单等

285. 给你一个整数类型的数组，按从小到大的顺序进行排列！

Array.sort(int a[]);// 默认为升序排列

286. 查询出 M 页的 N 行数据 。

答：

SQL实现： select top N from a where id not in

(select top 每页显示 * (M-1) id from students order by id) order by id

MYSQL:select * from a limit m,n

287. 把二叉树从小到大遍历出来怎么实现？

```
public class Node {
    public int value;
    public Node left;
    public Node right;
    public void store( int value)
    {
        if (value< this .value)
        {
```

```

        if (left == null )
        {
            left = newNode();
            left.value=value;
        }
        else
        {
            left.store(value);
        }
    }
    else if (value> this .value)
    {
        if (right == null )
        {
            right = new Node();
            right.value=value;
        }
        else
        {
            right.store(value);
        }
    }
}

public boolean find( int value)
{
    System. out .println("happen " + this .value);
    if (value == this .value)
    {
        return true ;
    }
    else if (value> this .value)
    {
        if (right == null ) return false ;
        return right.find(value);
    } else
    {
        if (left == null ) return false ;
        return left.find(value);
    }
}

public void preList()
{

```

```

        System.out.print( this.value + ",");
        if (left!= null ) left.preList();
        if (right!= null ) right.preList();
    }

    public void middleList()
    {
        if (left!= null ) left.preList();
        System.out.print( this.value + ",");
        if (right!= null ) right.preList();
    }

    public void afterList()
    {
        if (left!= null ) left.preList();
        if (right!= null ) right.preList();
        System.out.print( this.value + ",");
    }

    public static void main(String [] args)
    {
        int [] data = new int [20];
        for ( int i=0;i<data.length;i++)
        {
            data[i] = ( int )(Math.random()*100) + 1;
            System.out.print(data[i] + ",");
        }
        System.out.println();

        Node root = new Node();
        root.value = data[0];
        for ( int i=1;i<data.length;i++)
        {
            root.store(data[i]);
        }

        root.find(data[19]);

        root.preList();
        System.out.println();
        root.middleList();
        System.out.println();
        root.afterList();
    }
}

```

288. mysql 分页怎么实现， oracle 分页怎么实现，他们之间的区别？

Ajax 是一个 RIA 应用技术，解决富 Web 客户端问题，提高用户体验的

jms 是 Java 的消息服务系统，解决 Java 分布式应用消息异步通信

mysql 中的 limit 关键字查询出指定区间的记录进行分页。

oracle 中通过 rownum 来记录结果集记录编号实现分页

289. 描述 struts 体系结构？对应各个部分的开发工作主要包含哪些？

按 MVC 设计的 Web 架构，

其中 Model 为 ActionForm 组件的设计

View 为 Struts 标签的使用

Controller 为 Action 组件的设计和 ActionServlet 的配置

290. 组合优于继承。你是否同意这句话，并说明

继承和组合都可以实现代码的重用 (Reuse)，为什么组合要优于继承呢？

通过继承，子类能够获得与父类相同的功能和属性，父类拥有了某种功能，子类就有了；但是如果子类需要新的功能，直接在父类里面增加新的功能，那么会导致所有的子类同时拥有这种功能，这实际上是将类的功能与类紧紧的耦合在一起，是一种紧密耦合系统，由于关系相当密切，导致改动一处而动全身，实际上不利于代码的重用。而组合恰好相反，她把类的各项功能分割开，互不干涉。一个类的功能，实际上是各种子功能的组合，比如鸭子的功能就是飞行功能和发出叫声功能的组合，无论是何种鸭子，都是这样。同时到底是怎样飞行，怎样叫，又可以通过面向接口的编程实现多态。

51. javascript 取浏览器的版本 里面还有哪些对象？

```
<script type="text/javascript">
```

```
function getBrowser(){
```

```
var bro = navigator.userAgent.toLowerCase();
```

```
if(/msie/.test(bro)) return 'IE' + bro.match(/msie ([\d.]*)/)[1]
```

```
else if(/navigator/.test(bro)) return 'NS' + bro.match(/navigatorV([\d.]*)/)[1]
```

```
else if(/chrome/.test(bro)) return 'CR' + bro.match(/chromeV([\d.]*)/)[1]
```

```
else if(/safari/.test(bro)) return 'SF' + bro.match(/versionV([\d.]*)/)[1]
```

```
else if(/opera/.test(bro)) return 'OP' + bro.match(/versionV([\d.]*)/)[1]
```

```
else if(/firefox/.test(bro)) return 'FF' + bro.match(/firefoxV([\d.]*)/)[1]
```

```
}
```

```
</script>
```

291. 如何运行本地的 EXE 文件？

双击运行，或者在 DOS 窗口中通过命令运行

292. struts 中的 prepare 怎么用？

首先 Action 组件实现 Preparable 接口的 prepare() 方法，在配置 Action 组件时应用 PrepareInterceptor 拦截器。

293. Hrbernate 的二级缓存讲述。

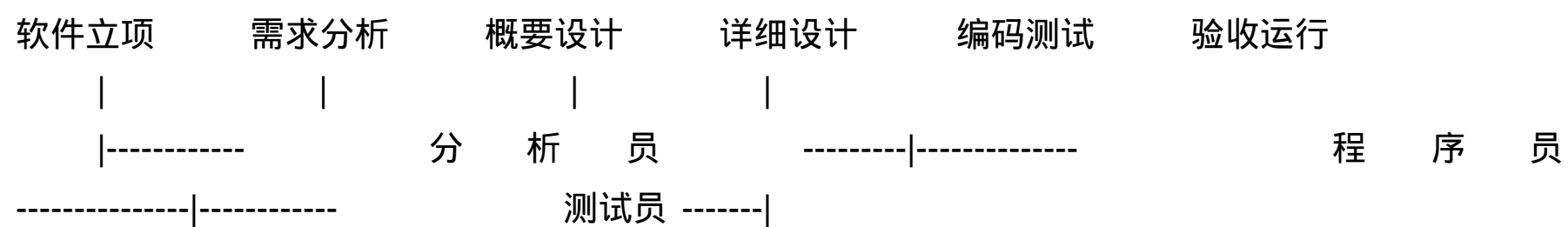
hibernate 的 session 提供了一级缓存，每个 session，对同一个 id 进行两次 load，不会发送两条 sql 给数据库，但是 session 关闭的时候，一级缓存就失效了。

二级缓存是 SessionFactory 级别的全局缓存，它底下可以使用不同的缓存类库，比如 ehcache、oscache 等。hibernate 在执行任何一次查询的之后，都会把得到的结果集放到缓存中，缓存结构可以看作是一个 hash table，key 是数据库记录的 id，value 是 id 对应的 pojo 对象。当用户根据 id 查询对象的时候（load、iterator 方法），会首先在缓存中查找，如果没有找到再发起数据库查询。

294. 如果做个招聘网站性别只招男的，年龄 20-30，如果不匹配就不能让他投简历，怎么实现？填身份证时如何让该人信息全部显示出来？

使用 JavaScript 做客户端检验，填身份证时使用 Ajax 获取人员信息；

295. 软件的开发过程、以及角色？



296. io 流 InputStream 与 Reader 的区别，怎样将字符流转换成字节流

InputStream 字节输入流、Reader 字符输入流

OutputStreamWriter 是字符流通向字节流的桥梁

InputStreamReader 是字节流通向字符流的桥梁

297. 启动线程调用那个方法？为什么要调用 start（）方法？若调用 run（）方法会出现什么情况？

面试题上有

298. Struts2.0 若在一个方法中返回的是 null，表明什么？struts.xml 还需要配置吗？为什么？struts 有那些标签？struts.xml 中有那些标签？

方法返回 null，表示没有视图结果回应，当然自己通过流直接输出响应结果，则 struts.xml 只需对 Action 进行映射配置，而无须进行结果配置。

299. Hibernate 运行的流程？优缺点？为什么 insert 比 select 快？可以批量 insert 一些数据吗？

Hibernate 应该是建立在 JDBC 基础上的一个开源的 ORM 映射框架，很好的封装了实体映射实现，提供友好的实体操作接口，以及基于 HQL 的查询接口，在开发中可以减少编写 SQL 相关代码，提高效率和减少出错。

以面向对象的方式操作数据库，进行增删改很方便，但执行复杂的查询时则很费事麻烦。

select 查询可能需要执行多条 SQL 语句，牵涉到加载策略所以执行需要消耗很多时间，相比 insert 操作要慢些。

hibernate 可批量 insert 一些数据，配置文件中设置 hibernate.jdbc.batch_size 参数，来指定每次提交 SQL 的数量。

300. sqlserver 查询 怎样取得 JAVA虚拟机默认的编码方式？

可通过 `Charset.defaultCharset()` 来获取

301. 智力题：

1、有一辆火车以 15 公里 / 小时从洛杉矶开往纽约，一辆火车以 20 公里 / 小时从纽约开往洛杉矶，一只鸽子以 30 公里 / 小时从洛杉矶开往纽约，鸽子和两辆火车同时出发，鸽子遇到火车转头，向另一辆火车飞，直到两辆火车相遇，问鸽子飞了多少公里？

假设洛杉矶到纽约的距离为 s

那小鸟飞行的距离就是 $(s/(15+20))*30$ 。

2、有两根不均匀的香，烧完 1 小时，问如何知道 15 分钟

把香称重，取出总重量的 $1/4$

3 一群人在一黑屋子，有黑白 2 种帽子，每人可以看其他人的帽子，至少 1 顶黑帽子。如果知道自己是黑帽子，打自己一巴掌。第一次开灯，无人，关灯。第二开灯，无人，关灯。第三次，听到劈劈啪啪的声音。问有几顶黑帽子。

302. 什么是 联通图的生成树？

什么是哈夫曼树？什么是平衡二叉树？写一个深度优先排序，写一个广度优先排序？写出 C 语言对链表 HEAD 的排序？

连通图的生成树——是连通图的一个极小连通子图，它含有图中全部顶点，但只有足以构成一棵树的 $n-1$ 条边。

哈夫曼树又称最优树（二叉树），是一类带权路径最短的树。构造这种树的算法最早是由哈夫曼 (Huffman)1952 年提出，这种树在信息检索中很有用。

如果任意节点的左右子树的深度相差不超过 1，那这棵树就是平衡二叉树。

303. 哪些不是 JAVA关键字： TRUE sizeof native 是关键字

sizeof 不是 大写的 TRUE不是

304. 怎样知道给出的两条线段是否相交，用伪码实现。

假设线段 AB,A(X1,Y1),B(X2,Y2); 线段 CD,C(X3,Y3),D(X4,Y4);

1. 判断两斜率 相等则不相交

2. 若斜率不等

305. ORACL数据库用到了哪些方面，怎样优化数据库。 SQL语句过长时，怎样优化？

(1) 使用分区这种手段。所谓分区就是动态地将表中的记录分离到若干不同的表空间上，使数据在物理上被分割开来，便于维护、备份、恢复、事务及查询性能。

(2) 重建索引，及时回收那些未使用空间，提高空间利用率。

(3) 调整 SGA参数

sql 语句长：用存储过程！

306. JAVASCRIPT的一些常用东西：如提示框如何实现的：MVC

使用 jQuery 等第三方的插件 Dialog 即可，无需重复造轮子。

307. 两个 SQL语句 一个 ArrayList 里面存了一些 Student 对象，Student 对象有 Name 和 Age 两人属性，此类里面有 set 和 get 方法，要求以 Age 来降序排序。

要求 Student 实现 Comparable 接口的 compareTo() 方法来按照 age 判断对象的大小，并且大于时返回 -1，小于时返回 1，相等则返回 0。

308. 有一些奇数和偶数，无序的，要求写一个方法，实现将所有的奇数放在全部的偶数前面
主要代码如下：主要采用快速排序的算法。

```
int[] n = {1,2,5,6,4,7,10,9,3,8};
int m = n[0];
int j = n.length -1;
int i= 0;
while(i<j){
    while(i<j && n[j]%2==0){j--;}
    if(j>=i){
        n[i++] = n[j];
    }

    while(i<j && n[i]%2!=0){i++;}
    if(i<=j){
        n[j--] = n[i];
    }
}
n[i] = m;

for(int k=0;k<n.length;k++){
    System.out.print(n[k] + " ");
}
```

309. 你对工作流和表单的理解

一个工作流包括一组活动及它们的相互顺序关系，还包括过程及活动的启动和终止条件，以及对每个活动的描述。 一个完整的工作流管理系统通常由工作流引擎、 可视工作流设计、流程操作、工作流客户端程序、流程监控、表单设计工具、与表单的集成以及与应用程序的集成八个部分组成。 工作流引擎作为工作流管理系统的核心部分， 主要提供了对于工作流定义的解析以及流程流转的支持。

表单用来显示查询或输入的业务数据。

310. Classpath 怎么设置和它的意义是什么？

比如， set CLASSPATH=C:\Java\jdk1.6.0_10\lib\dt.jar;.

设置 class 文件的搜索路径

311. 用 java 写一个读取盘符的名字

```
new File("").getAbsolutePath().substring(0, 1);
```

312. Deprecated API 是什么意思？

deprecated 是指已经过时，但是为了向前兼容起见仍然保留的方法，这些方法可能会在以后取消支持。你应当改用较新的方法。一般在 API 里面会说明你应当用什么方法来代替之。

313. 如何去掉 JAVA的 内存机制？

是不是垃圾回收机制，使用 System.gc() 强制回收。

314. 缺少 IDE 如何编译 JAVA代码？

使用 JDK的 javac.exe 直接手工编译

或使用 Apache 的 ANT脚本构建

315. THHP与 HTTPS有什么区别？

HTTP为普通 WEB协议、HTTPS为加密 WEB协议。默认 http 为 tcp80 端口。https 为 tcp443 端口

316. Session 的具体用法？

Session 用来保存每一个用户的专有信息，比如像用户登录验证、购物车。

1.ibatis 中的 #与 \$的区别

1.# 是把传入的数据当作字符串，如 #field# 传入的是 id, 则 sql 语句生成是这样， order by "id", 这当然会报错。

2.\$ 传入的数据直接生成在 sql 里,如 #field# 传入的是 id, 则 sql 语句生成是这样， order by id, 这就对了。

3.# 方式能够很大程度防止 sql 注入。

4.\$ 方式无法防止 sql 注入。

5.\$ 方式一般用于传入数据库对象。例如传入表名。

6. 一般能用 #的就别用 \$。

317. 请列举 session 、 cookie 、 viewstated 的应用范围

Session 服务器端维护，某个用户活动时间 + 延迟时间（默认 20 分钟），

Cookie 客户端维护，程序可指定生命周期，

ViewState 小量数据，一个 web页面的生命期

318. 用什么方法使服务器关闭之后， session 所保存的信息不会丢失？

将 session 信息保存到数据库中或文件中

319. B2C和 C2C你们是怎么区别的？

Business（企业）、Customer（消费者）

B2C的 B 是 Business 企业，2 则是 to 的谐音，C 是 Customer 消费者，所以 B2C是企业对消费者的电子商务模式。这种形式的电子商务一般以网络零售业为主，主要借助于 Internet 开展在线销售活动。B2C简而言之就是商家与顾客之间的商务活动，也就是通常所说的“网上购物网站”，企业、商家可充分利用电子商城提供的网络基础设施、支付平台、安全平台、管理平台等共享资源有效地、低成本地开展自己的商业活动。

C2C就是消费者（consumer）与消费者（consumer）之间的电子商务。打个比方，比如一个消费者的有一台旧电脑，通过网上拍卖，把它卖给另外一个消费者，这种交易类型就称

为 c2c 电子商务。主要是指网上拍卖。 c2c 模式的特点就是大众化交易，因为是个人与个人之间的交易！

320. 用最好效率写一个 2^{17} 等于多少？
($2^{<4}$) + 2

321. 用 4 个 0，用你所知道的数学方法计算出 24
0 的阶乘等于 1 即 $0! = 1$ 那么 4 个 0 就是 4 了
又 4 的阶乘为 24 $4! = 24$

322. 10 怎样计算等于 24？
 $(10 \times 10 - 4) \div 4 = 24$

323. 有一个长方形蛋糕，切掉长方形一块，怎样一刀切下去成大小相等两块
将完整的蛋糕的中心与被切掉的那块蛋糕的中心连成一条线。这个方法也适用于立方体

324. 九个球不知道哪个轻重，让你称？
4 个和 4 个称，若相等则剩下一个是比较重要 (2) 若不相等，则往那边不相等的那 4 个球中
取两个分为一组，则左 2 个，右 2 个，则有左边或右边一个是比较重的，从任意一边取一个投入另一方，并记住这个球，从其它 4 个取两个放入另外一边，并记住这两个球，若从一边取走的放到另一边后，那边重，则这球比较重要，若从一边取走的放到另一边后，那边轻，则为另外一边所剩的一个球发

一：单选题

1. 下列哪一种叙述是正确的 (D)
A . abstract 修饰符可修饰字段、方法和类
B . 抽象方法的 body 部分必须用一对大括号 {} 包住
C . 声明抽象方法，大括号可有可无
D . 声明抽象方法不可写出大括号

2. 如下代码

```
public class Test {  
    public int aMethod() {  
        static int i = 0;  
        i++;  
        return i;  
    }  
    public static void main (String args[]) {  
        Test test = new Test();  
        test.aMethod();  
        int j = test.aMethod();  
        System.out.println(j);  
    }  
}
```

输出结果是什么？ D

- A. 0
- B. 1
- C. 2
- D. 编译失败

3. 下列哪种说法是正确的 (D)

- A . 实例方法可直接调用超类的实例方法
- B . 实例方法可直接调用超类的类方法
- C . 实例方法可直接调用其他类的实例方法
- D . 实例方法可直接调用本类的类方法

4. 如下代码 :

```
class Super {  
    public Integer getLenght() { return new Integer(4); }  
}  
  
public class Sub extends Super {  
    public Long getLenght() { return new Long(5); }  
    public static void main(String[] args) {  
        Super sooper = new Super();  
        Sub sub = new Sub();  
        System.out.println(sooper.getLenght().toString() + "," +  
            sub.getLenght().toString() );  
    }  
}
```

输出是什么 ? A

- A. 4,4
- B. 4,5
- C. 5,4
- D. 5,5
- E. 编译失败 .----- 重定时不能改变返回类型

5. 在 Servlet 处理请求的方式为 : C

- A. 以进程的方式
- B. 以程序的方式
- C. 以线程的方式
- D. 以响应的方式

6. JDBC 中 , 用于表示数据库连接的对象是 : B

- A. Statement
- B. Connection
- C. DriverManager
- D. PreparedStatement

7. 用于调用存储过程的对象是 : C

- A. ResultSet
- B. DriverManager
- C. CallableStatemet
- D. PreparedStatement

8. 按照 MVC 设计模式， JSP 用于实现： B
- A.Model
 - B.View
 - C.Controller
 - D.容器
9. 如下代码
- ```
10. public Object m() {
11. Object o = new Float(3.14F);
12. Object [] oa = new Object[1];
13. oa[0] = o;
14. o = null;
15. oa[0] = null;
16. print 'return 0';
17. }
```
- 当 Float 对象在第 11 行被创建后，什么时候能够被垃圾回收？ C
- A. 13 行以后 .
  - B. 14 行以后 .
  - C. 15 行以后 .
  - D. 16 行以后 .
10. 如下代码：
- ```
class Base {  
    Base() { System.out.print("Base"); }  
}  
  
public class Alpha extends Base {  
    public static void main( String[] args ) {  
        new Alpha(); ----- 调用父类无参的构造方法  
        new Base();  
    }  
}
```
- 结果是什么？ B
- A. Base
 - B. BaseBase
 - C. 编译失败 .
 - D. 代码运行但没有输出 .
 - E. 运行时抛出异常
11. 在 J2EE 中属于 Web 层的组件有： A
- A. Servlet
 - B.EJB
 - C.Applet
 - D.HTML
12. 在 Servlet 的生命周期中，容器只调用一次的方法是： C
- A.service
 - B.getServletConfig

- C.init
- D.destroy

13. 以下关于异常的说法正确的是： **D**

- A. 一旦出现异常，程序运行就终止了
- B. 如果一个方法申明将抛出某个异常，它就必须真的抛出那个异常
- C. 在 catch 子句中匹配异常是一种精确匹配
- D. 可能抛出系统异常的方法是不需要申明异常的

14. 如下代码：

```
public class Foo {  
    public static void main(String[] args) {  
        try {  
            return;  
        } finally {  
            System.out.println( "Finally" );  
        }  
    }  
}
```

输出结果是什么 ？ **A**

- A. Finally
- B. 编译失败
- C. 代码正常运行但没有任何输出
- D. 运行时抛出异常

15. jsp 指令中 isELIgnored="boolean" 的意思是 (**C**)

- A. 决定是否实现 Servlet 的单线程模式，
- B. 决定改页面是否是一个错误处理页面，
- C. 决定是否支持 EL 表示，
- D. 没有具体的含义。

16. 下面关于 Java 的引用，说法正确的是： **B**

- A. 应用实际上就是指针
- B. 应用本身是 Primitive ????????????????????????????????
- C. 引用就是对象本身
- D. 一个对象只能被一个引用所指引

17. 对于 import java.util 包，以下那种说法是错误的 ？ **C**

- A. Vector 类属于 java.util 包
- B. Vector 类放在 ... / java/util/ 目录下
- C. Vector 类放在 java.util 文件中
- D. Vector 类是 Sun 公司的产品

18. 在一个线程中 Sleep (1000) 方法，将使得该线程在多少时间后获得对 CPU 的控制 (假设睡眠过程中不会有其他事件唤醒该线程) ？ **C**

- A. 正好 1000 毫秒
- B. 1000 毫秒不到
- C. =>1000 毫秒
- D. 不一定

19. 在 WEB-INF 目录下，必须存放的文件为 : B
- A.class 文件
 - B.web.xml
 - B.jar 文件
 - D.html 文件
20. 下面关于垃圾收集的说法正确的是 : D
- A. 一旦一个对象成为垃圾，就立刻被收集掉。
 - B. 对象空间被收集掉之后，会执行该对象的 finalize 方法
 - C. finalize 方法和 C++ 的析构函数是完全一回事
 - D. 一个对象成为垃圾是因为不再有引用指着它，但是线程并非如此。
21. 下面的语句会产生什么样的输出 ? A
- ```
System.out.println(4&7);
```
- A.4
  - B.5
  - C.6
  - D.7
  - E.0
22. 下面的程序中，temp 的最终值是什么 ? B
- ```
long temp=(int)3.9;
temp%=2;
```
- A .0
 - B .1
 - C .2
 - D .3
 - E .4
23. 下面哪个不是 JAVA 关键字 A
- A integer
 - B double
 - C float
 - D default
24. Which method is used by a Servlet to place its session ID in a URL that is written to the servlet 's response output stream? B
- A. The encodeURL method of the HttpServletRequest interface.
 - B. The encodeURL method of the HttpServletResponse interface.
 - C. The rewriteURL method of the HttpServletRequest interface.
 - D. The rewriteURL method of the HttpServletResponse interface.
25. 每个使用 Swing 构件的程序必须有一个 (). D
- A. 按钮
 - B. 标签
 - C. 菜单
 - D. 容器
26. 下列标识符 (名字) 命名原则中，正确的是 () D
- A. 类名的首字母小写
 - B. 变量和方法名的首写字母大写
 - C. 接口名的首写字母小写
 - D. 常量完全大写

27. 类 Test1 定义如下：

```
1. public class Test1{  
2.     public float aMethod ( float a , float b ) { return 0;}  
3.  
4. }
```

将以下哪种方法插入行 3 是不合法的。（ B ）

- A、 public float aMethod (float a , float b , float c) { return 0;}
- B、 public float aMethod (float c , float d) { return 0;}
- C、 public int aMethod (int a , int b) { return 0;}
- D、 private float aMethod (int a , int b , int c) { return 0;}

28. 哪个语句创建了一个数组实例 ？ A

- A. int[] ia = new int [15];
- B. float fa = new float [20];
- C. char[] ca = Some String ;
- D. int ia [][] = {4, 5, 6} {1, 2, 3};

29. 以 public 修饰的类如： public class Car{ 则 Car() A

- A、可被其它程序包中的类使用 B、仅能被本程序包中的类使用
- C、不能被任意其它类使用 D、不能被其它类继承

30. 下列哪个属于容器型构件 （ ） C

- A .JEdit B.JButton C.JPanel D.JTextField

31. Java 程序的执行过程中用到一套 JDK 工具，其中 java.exe 是指 （ ） B

- A.Java 文档生成器 B.Java 解释器
- C.Java 编译器 D.Java 类分解器

32. 下列关于构造方法的叙述中，错误的是 （ ） C

- A. Java 语言规定构造方法名与类名必须相同
- B. Java 语言规定构造方法没有返回值，但不用 void 声明
- C. Java 语言规定构造方法不可以重载
- D. Java 语言规定构造方法只能通过 new 自动调用

33. 构造函数何时被调用 （ ） B

- A. 类定义时 B. 创建对象时
- C. 调用对象方法时 D. 使用对象的变量时

34. break 语句 （ ） D

- A、只中断最内层的循环 B、只中断最外层的循环
- C、借助于标号，可以实现任何外层循环中断 D、只中断某一层的循环

35. Java 语言中下面哪个可以用作正确的变量名称 （ ） B

- A、3D B、name C、extends D、implements

36. 在 JavaScript 中如何验证一个数据是否是数字 （ D ）

- A. 如果用 Integer.parseInt(value) 有误就不是数字
- B. int l = value 若报错就不是数字，
- C. 没有方法验证
- D. 利用 isNaN(value) 返回的 boolean 进行判断
-----返回一个 Boolean 值，指明提供的值是否是保留值 NaN（不是数字）。

37. 下面哪一个不能作 JSP 的服务器 （ D ）

- A. IBM WebSphere B. BEA WebLogic C. Tomcat D. pws

38. 下面哪项不是 JSP 操作指令 (**D**) ----- ??????????????????????
A. getProperty B. forward
C. include D. import
39. 如果希望控件在界面上按表格行分列排列，应使用下列那种布局管理器？ (**B**)
A、BoxLayout B、GridLayout C、BorderLayout D、FlowLayout
40. 下面哪一项不是 Java 类访问控制关键字 (**C**)
A. public B. protected
C. this D. private
41. 在配置 tomcat 虚拟目录时，需要打开哪个文件？ (**C**)
A. index.jsp B. web.xml C.server.xml D. 以上都不是
42. 下面哪个不是表单标记 (**D**)
A. INPUT B.RADIO C.CHECKBOX D. TR
43. 下面哪项不是 response 对象的方法 (**C**)
A. setContentType(String contentTypestr)
B. setHeader(String headername,String headervalue)
C. getParameter(String str)
D. sendError(int errorcode)
44. 以下 (**B**) 是编写 Servlet 必须导入的包。
A.java.sql.* B.java.servlet.* C.java.util.* D.java.lang.*
45. 下面哪项不属于 SQL 语句的子类 (**D**)
A. 数据定义语言 (DDL)
B. 数据查询语言 (DQL)
C. 事务控制语言 (TCL)
D. 数据插入语言 (DIL)
46. 选出所有正确的答案
- ```
public class Outer{
 public void someOuterMethod() {
 // Line 3
 }
 public class Inner{}
 public static void main(String[]argv) {
 Outer o = new Outer();
 // Line 8
 }
}
```

Which instantiates an instance of Inner?                      A

- A. new Inner(); // At line 3  
B. new Inner(); // At line 8  
C. new o.Inner(); // At line 8  
D. new Outer.Inner(); // At line 8

## 二：多选题

- 下列属于 jsp 中注释的有 ( **AD** )
  - A . <%-- 与 --%>
  - B . /
  - C . /\*\* 与 \*\*/
  - D . <!-- 与 -->
- 按照学生平均成绩 (avg\_grade) 将 students 表中的数据检索出来, 下面 SQL语句正确的是 ( **ACD** )
  - A) SELECT \* FROM students ORDER BY avg\_grade
  - B) SELECT \* FROM students GROUP BY avg\_grade ASC
  - C) SELECT \* FROM students ORDER BY avg\_grade DESC
  - D) SELECT \* FROM students ORDER by avg\_grade asc
- 下列是 jsp 作用域的通信对象的有 ( **ABC** )
  - A . Session
  - B. application
  - C.pageContext
  - D.cookie
- 在接口中以下哪条定义是正确的 ? ( **AB** )
  - A. void methoda();
  - B. public double methoda();
  - C. public final double methoda();
  - D. static void methoda(double d1);
  - E. protected void methoda(double d1);
- 下面哪个语句正确地声明一个整型的二维数组 ? **CDE**
  - A. int a[][] = new int[][] ;
  - B. int a[10][10] = new int[][] ;
  - C. int a[][] = new int[10][10] ;
  - D. int [][]a = new int[10][10] ;
  - E. int []a[] = new int[10][10] ;
- 下面哪些不是 java 的简单数据类型 ? **BC**
  - A. short
  - B. Boolean
  - C. Double
  - D. float
- 下面哪些语句能够正确地生成 5 个空字符串 ? **AB**
  - A.String a[]=new String[5] ; for(int i=0 ; i<5 ; a[i++]= ; )
  - B.String a[]={ , , , , ; }
  - C.String a[5] ;
  - D.String[5]a ;
  - E.String []a=new String[5] ; for(int i=0 ; i<5 ; a[i++]=null) ;
- 下列说法错误的有 ( **BCD** )
  - A . 数组是一种对象
  - B . 数组属于一种原生类
  - C . int number[]={31,23,33,43,35,63}
  - D . 数组的大小可以任意改变

9. 不能用来修饰 interface 的有 ( **ACD** )  
A . private   B . public   C . protected   D . static

10. 下列说法错误的有 ( **ACD** )  
A . 在类方法中可用 this 来调用本类的类方法  
B . 在类方法中调用本类的类方法时可直接调用  
C . 在类方法中只能调用本类中的类方法  
D . 在类方法中绝对不能调用实例方法

11. 如下代码

```
class A {
 A() {}
}
```

```
class B extends A {
}
```

哪两个说明是正确的？      **BD**

- A. B 类的构造器应该是 public.  
B. B 类的构造器应该没有参数  
C. B 类的构造器应该调用 this().  
D. B 类的构造器应该调用 super().
12. 下列标识符不合法的有 ( **ACD** )  
A . new   B . \$Usdollars   C . 1234   D . car.taxi

13. 在接口中以下哪条定义是正确的 ?      **AB**

- A. void methoda();  
B. public double methoda();  
C. public final double methoda();  
D. static void methoda(double d1);  
E. protected void methoda(double d1);

14. 以下哪四个能使用 throw 抛出 ?      **ADEF**

- A. Error  
B. Event  
C. Object  
D. Throwable  
E. Exception  
F. RuntimeException

15. javax.Servlet 的包中，属于类的是 :      **BC**

- A. Servlet  
B. GenericServlet  
C. ServletException  
D. ServletContext

16. 如下代码 :

```
public class X {
 public X aMethod() { return this;}
}

public class Y extends X {
```

}

哪两个方法能加到 Y 类的定义中 ? CE

- A. public void aMethod() {}
- B. private void aMethod() {}
- C. public void aMethod(String s) {}
- D. private Y aMethod() { return null; }
- E. public X aMethod() { return new Y(); }

17. 对该段代码描述正确的有 <% int i =

Integer.parseInt(request.getParameter( " value " )) %>( BDC )

- A) 不会有错
- B) 当 value 与 int 类型不匹配时回报错
- C) 当 value = " " 时会报错
- D) 为了安全起见应该将该段代码放在 try{} 和 catch(){} 之间

18. 选出所有正确的答案

Which two are equivalent? BG

- A. <%= YoshiBean.size%>
- B. <%= YoshiBean.getSize()%>
- C. <%= YoshiBean.getProperty("size")%>
- D. < JSP.getProperty id="YoshiBean" param="size"/>
- E. <jsp:getProperty name="YoshiBean" param="size"/>
- F. <jsp:getProperty id="YoshiBean" property="size"/>
- G. <jsp:getProperty name="YoshiBean" property="size"/>

19. 下面哪三个描述是正确的 ? BCE

- A. 默认构造器初始化方法变量
- B. 默认构造器有和它所在类相同的访问修饰词
- C. 默认构造器调用其父类的无参构造器
- D. 如果一个类没有无参构造器 ,编译器会它创建一个默认构造器
- E. 只有当一个类没有任何构造器时 , 编译器会为它创建一个默认构造器

20. 哪二种声明防止方法覆盖 ? AD

- A. final void methoda() {}
- B. void final methoda() {}
- C. static void methoda() {}
- D. static final void methoda() {}
- E. final abstract void methoda() {}

### 三：简答题

1. 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。

2. 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 。

3. 如何利用 ServletContext 和 ServletConfig 对象获得初始化参数

```
String psw = config.getInitParameter("psw");
ServletContext ss = config.getServletContext();
String ppp = ss.getInitParameter("name");
```

4. 如何用 css 约束一个层不可见？

```
<div style="display:none"> </div> ----- 表示不可见
<div style="display:block"> </div>----- 表示可见
```

5. 在表单中存在一组名称为 checkbox1 的多选框，如何才能取得这些多选框内的值？

在页面中用 javascript 方式取值

```
<script >
 function sss() {
 var str = document.getElementsByName("checkbox1");
 var checkValue = new Array();
 for (var i=0;i<str.length;i++) {
 if (str[i].checked) {
 alert(str[i].value);
 }
 }
 }
}

</ script >
Function sss()
 Var str=document.getElementsByName(" checkbox1 ");
 Var checkValue=new Array()
 For(var i=0;i<str.length;i++){
 If(str[i].checked){
 Alert(str[i].value);
 }
 }
}
```

在 Servlet 中取值

```
String str[] = request.getParameterValues("checkbox1");
```

6. 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，新建连接的数量由配置参数决定。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。

7. 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

8. 一个页面在打开时，需要处理一些事情（比如弹出一个广告页），需要定义窗口的什么事件？

```
<script >
window.onload= function () { ... }
</ script >
```

或是

```
<body onload="">
```

9. HTML的 Form 和 XFORM的区别

10. 描述 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, 目标地址可以接收 request 请求参数，然后把结果发给浏览器，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哪个地址，浏览器会重新进行请求，此时不能用 request 传值，浏览器的地址栏会变成新的地址

11. Static 变量是什么含义

static 是静态变量，就是变量值不随函数执行结束而消失，下次调用同一函数时，上次所赋予的值仍存在。

12. 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。

13. 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。

14. 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

## 四：编程题

1. 请编写一 JSP 程序，它提供一个征求客户意见的文本区，在客户提交意见后，它能将意见及其他客户以前提交的意见一起显示出来。

2. 写一个算法对 1, 8, 5, 2, 4, 9, 7 进行顺序排列。

```
public class OrderbyArray {
 // 冒泡排序 Bubble Sort 最简单的排序方法是冒泡排序方法
 public int [] orderArray(int [] array){
 for (int i=0;i<array.length ;i++){
 for (int j=i;j<array.length ;j++){
 if (array[i]>array[j]){
 int s = array[i];
 array[i] = array[j];
 array[j] = s;
 }
 }
 }
 return array;
 }

 public static void main(String[] args) {
 int [] array = {1,8,5,2,4,9,7};
 OrderbyArray order = new OrderbyArray();
 array = order.orderArray(array);
 }
}
```

3. 写一段 Jdbc 连接 Oracle 的程序 ,并实现数据查询。

创建一个连接数据库的工具类

```
import java.sql.*;

public class DbUtil {
```



```

public static Connection getConnection(){
 String driver = "" ;
 String url = "" ;
 String name = "scot" ;
 String psw = "123" ;
 Connection conn = null ;
 try {
 Class.forName(driver);
 conn = DriverManager.getConnection(url,name,psw);
 } catch (ClassNotFoundException e) {
 e.printStackTrace();
 } catch (SQLException e) {
 e.printStackTrace();
 }
 return conn;
}
}
import java.sql.*;
public class SearchInfo {
 public void searchInfo(int id){
 Connection conn = null ;
 PreparedStatement pstat = null ;
 ResultSet res = null ;
 String sql = "select * from users where id=?" ;
 conn = DbUtil.getConnection();
 try {
 pstat = conn.prepareStatement(sql);
 pstat.setInt(1, id);
 res = pstat.executeQuery();
 while (res.next()){
 String name = res.getString("name");
 }
 } catch (SQLException e) {
 e.printStackTrace();
 }
 }
}

```

4. 请利用 JSP 编写一简易的购物车程序， 它包括一购物界面及往购物车中添加商品与删除商品功能，并能将用户当前已购的所有商品显示出来。

5. 写一个 Singleton（单例模式）出来。

```

public class LazySingleton {
 private static LazySingleton instance = null;

```

```
// 默认的私有的构造方法 ,保证外界无法直接实例化
private LazySingleton() {
}
```

```
// 静态方法 ,返回此类的唯一实例
public static LazySingleton getInstance() {
 if (instance == null) {
 instance = new LazySingleton();
 }
 return instance;
}
```

```
public void pp(){
 System.out.println("lazy ok");
}
}
```

6. 用循环控制语句打印输出： 1+3+5+ ?? .+99= ? 结果。

```
int sum = 0;
for (int i=1;i<100;i+=2){
 sum = sum+i;
}

System.out.println("1+3+5+...+99 = " +sum);
```

7. 请编写一 JSP 程序，它能根据输入的订单号查寻出该订单的详细信息，包括订单用户名及价钱，其中订单号以请求参数的形式传给该 JSP。订单信息都保存在 SQLSERVER 数据库中。并要求将查寻到的订单信息以表格的形式显示给用户。

## IBM 笔试题

注：IBM 笔试题一小时之内完成，题目全部用英文描述，这里用中文表述

### 一、名词解释

此题大概有 15 个左右，还有一些记不清了

- 1、Eclipse
- 2、J2EE
- 3、EJB
- 4、Ajax
- 5、Web service
- 6、

### 二、找出以下代码问题

```
public class test{

public void print (String str) {
 char[] s=str ;
```

```
}
```

```
}
```

```
public class a{
```

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

```
}
```

三、写出以下输出

```
public class Parent{
 public void method () {
 System.out.println("parent");
 }
 public static void smethod(){
 Sysyem.out.println("parent");
 }
}
```

```
public class Child extends parent{
 public void method () {
 System.out.println("child");
 }
 public static void smethod(){
 Sysyem.out.println("child");
 }
}
```

```
public class test{
 public static void print(String [] args){
 Child c=new Child();
 Parent p=(Parent)c;
 p.method();
 p.smethod();
 }
}
```

四、编程题（两题选其一）

1、IBM 全球有 35 万员工，名字由 26 个字母组成，长度不一。

1) 请设计一个算法，能够快速查找出要查询的名字。

2) 写出此算法的时间复杂度

3) 如果对此算法进行测试，请写出测试用例

2、用熟悉的 c/c++/java 写出一个完成的函数，此函数能够将一个值插入一个排好序的二叉树中

五、Unix 命令题

此题包含 Unix 操作系统的常用命令如建目录、系统启动、磁盘使用情况察看等，大概个以上

15

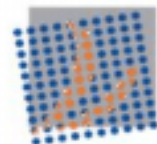
六、英文短文（两题选其一，不少于 200 字）

- 1、描述一下你理想中的职业
- 2、描述一下你心目中的中国文化以及它对于日常工作的影响



工业和信息化部人才交流中心

Talent Exchange Center, Ministry of Industry and Information Technology



国信蓝点软件学院  
GUOXIN BLUEDOT EDU

## JAVA 面试题 -CORE JAVA 部分

1. 在 `main(String[] args)`方法内是否可以调用一个非静态方法？

答案：不能

2. 同一个文件里是否可以有两个 `public` 类？

答案：不能

3. 方法名是否可以与构造器的名字相同？

答案：可以。

```
public class Test
{
 public Test(String iceboy)
 {
 System.out.println(iceboy);
 }
 public void Test(String iceboy)
 {
 System.out.println(iceboy);
 }
 public static void main(String[] args)
 {
 Test a = new Test("abc");//输出 “ abc ”
 a.Test("iceboy");//输出 “ iceboy ”
 }
}
```

4. 初始化了一个没有 `run()` 方法的线程类 ,是否会出错？

答案：不会。

第一种方法：直接继承 `Thread` 类。

```
public class Test
```

```

{
 public static void main(String[] args)
 {
 ThreadClass t = new ThreadClass();
 t.start();
 System.out.println("end");//输出 “ end ”
 }
}
class ThreadClass extends Thread //Thread 类已经实现了空的 run() 方法。
{
}

```

第二种方法：实现 **Runnable** 接口

```

public class Test
{
 public static void main(String[] args)
 {
 ThreadClass t = new ThreadClass();
 Thread thread = new Thread(t);
 thread.start();
 System.out.println("end");
 }
}
class ThreadClass implements Runnable
{
 public void run() //必须有此方法否则编译报错。 它是 Runnable 接口中的抽象方法。
 {
 System.out.println("Threads");
 }
}

```

4 . 局部内部类是否可以访问非 **final** 变量？

答案：不能访问局部的，可以访问成员变量（全局的）。

```

class Out
{
 private String name = "out.name";
 void print()
 {
 final String work = "out.local.work";//若不是 final 的则不能被 Animal 使用。
 int age=10;
 class Animal
 //定义一个局部内部类 .只能在 print() 方法中使用 .
 //局部类中不能使用外部的非 final 的局部变量 .全局的可以 .
 {

```

```

 public void eat()
 {
 System.out.println(work);//ok
 //age=20;error not final
 System.out.println(name);//ok.
 }
 }
 Animal local = new Animal();
 local.eat();
}
}

```

5 . 选择语句 **case** 中，允许使用的值有哪些？      答案： **int,short,char,byte**都在 **int** 范围之内，且是整数 )

6 . **Math,String** 是不可继承的。（ **final** 类 ）

**Instanceof** 后面跟的应该是 **OBJECT** 。

构造器可以是私有的。（ **private** ）

**=**与 **==**意义是完全不同的。一个是赋值，一个是等于。

全局变量可以不进行初始化， 如果使用一个局部变量， 则这个局部变量要被初始化。

7 . 在 **try-catch-finally** 块中的退出语句。

```

public class Test
{
 public static void main(String[] args)
 {
 int a=1;
 try
 {
 a=a/0;
 }catch(Exception e)
 {
 System.out.println("catch");
 return;//当 return 时， finally 中的语句会执行。
 //System.exit(0);//若用上这句， finally 中的语句不会执行。 直接返回，
 退出程序。
 }
 finally //当没有 System.exit(0)时，无论是否发生异常它都会执行。
 {
 System.out.println("finally");
 }
 }
}

```

注： **try-catch-finally** 块的顺序不能调换。

8 . 下面都是正确的 **main** 方法签名。

```

public static void main(String[] args)

```

**public static final void main(String[] args)**  
**static public void main(String[] args)**  
**static public synchronized void main(String[] args)**  
**static public abstract void main(String[] args)**错误

9 . **if(-0.0 == 0.0)**是相等还是不等？

答案：相等。

10 . 一个抽象类是否可以没有抽象方法？

答案：可以。

11 . **RandomAccessFile**类继承 **Object** , 并且实现了 **DataInput** 和 **DataOutput** 接口。

答案：正确

12 . **Collection**与 **Collections**的区别？

答案：**Collection**是一个接口，但 **Collections**却是一个辅助类，里面有很多静态的工具方法。而且很有用的。如：**reverse(List list)** ;**sort(List list,**

**Comparator c)**等。**Collections**没有实现任何接口。它直接继承了 **Object**。

13 . **class Child extends Parents{}**

**class Parents{}**

是否可以这样声明类，在一个文件中？

答案：可以。无所谓的。

14. 数组，无论是当前的，还是类等级的，都会被初始化。

**String** 是被初始化为 **null**,不是空字符。

**null**, “ ”, “ ”,都是不同的。

“**continue**”只能在一个循环里 (如 **for,do,while**),它不能在 **case**语句中出现。

**Primitive(int,char,long**等)数组是不能互相赋值的，即使它们本身可以。

一个 **Constructor**可以抛出任何异常。

初始化块是按照声明的顺序执行的。

所有关于 **NaN(Not a Number)** 和 **non-NaN** 的比较，都返回 **false**.这条很重要。

**==**会产生编译错误，如果两边类型不匹配的话。

**equals()** 返回 **false** 如果对象类型不同，但不产生编译错误。

15 . **Java** 成员变量默认初始化的值。

成员变量类型      取值

**byte-0      short-0      int-0      long-0L**

**char-" 0000"      float-0.0F      double-0.0D      boolean-false**

所有引用类型      **null**

16. **integer**和 **long** 操作 /和% 的话，可能会抛出 **ArithmeticException** , 比如除 0。但是 **float** 与

**double**不会，即使是除以 0。

**double a=0;a=a/0**则 **a** 等于 **NaN**。

17 . 普通内部类不可以拥有静态变量，但静态内部类可以。

**File** 类没有 任何 处理文件内容的方法。

**InputStream** 和 **OutputStream** 是 抽象类，**DataInput** 和 **DataOutput** 是

接口。 **DataInputStream** 实现了 **DataInput** 接口。

18 . 面向对象的特征有哪些方面 ？

答案：最基本特征：封装，继承，多态。

其他特征：抽象关联，聚合，组合，内聚，耦合

19 . **String** 是最基本的数据类型吗 ？ **String** 和 **StringBuffer** 的区别？

答案： **String** 不是一最基本的数据类型。

**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

20 . **int** 和 **Integer** 有什么区别

答案： **int** 是基本类型。 **Integer** 是引用类型。 **Integer** 可以把 **String** 转换成 **int**。

21.运行时异常与一般异常有何异同？

答案：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。

22 . 说出一些常用的类，包，接口，请各举 5 个？

答案：常用类 -**System,ArrayList,FileInputStream,Thread,Socket.**

常用的包 -**java.io,java.util,java.sql,java.javax.naming,java.net**

常用接口 -**Collection,Connection, Cloneable, Comparable, Serializable**

23.说出 **ArrayList,Vector, LinkedList** 的存储性能和特性 .

答案：**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。

24 . 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。

注：因为这 4 个线程共享 **J**，所以线程类要写到内部类中。

加线程：每次对 **j** 加一。

减线程：每次对 **j** 减一。

```
public class TestThreads
{
 private int j=1;
 //加线程
 private class Inc implements Runnable
 {
 public void run()
```



```

 {
 for(int i = 0;i < 10;i++)
 {
 inc();
 }
 }
 }
//减线程
private class Dec implements Runnable
{
 public void run()
 {
 for(int i = 0;i < 10;i++)
 {
 dec();
 }
 }
}
//加 1
private synchronized void inc()
{
 j++;

```

```

System.out.println(Thread.currentThread().getName()+"-inc:"+j);
}
//减 1
private synchronized void dec()
{
 j--;

```

```

System.out.println(Thread.currentThread().getName()+"-dec:"+j);
}
//测试程序
public static void main(String[] args)
{
 TestThreads test = new TestThreads();
 //创建两个线程类
 Thread thread = null;
 Inc inc = test.new Inc();
 Dec dec = test.new Dec();
 //启动 4 个线程
 for(int i = 0;i < 2;i++)
 {
 thread = new Thread(inc);

```

```

 thread.start();
 thread = new Thread(dec);
 thread.start();
 }
}
}

```

25 . 数组转换问题。

```

Object[] object = new Person[2];
Person [] person = new Person [3];
person = (Person [])object;//可以转换
int[] i = new int[2];
long[] l = new int[3];
i = (long[])l;//不可以转换

```

26 . 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。

**Server.java**源代码

```

import java.net.*;
import java.io.*;

```

```

class Server

```

```

{
 public Server()
 {
 BufferedReader br = null;
 PrintWriter pw = null;
 try
 {
 ServerSocket server = new ServerSocket(8888)建立服务器端
 Socket socket = server.accept()监听客户端
 //得到该连接的输入流
 br = new BufferedReader(new
InputStreamReader(socket.getInputStream()));
 //得到该连接的输出流
 pw = new PrintWriter(socket.getOutputStream(),true);
 //先读后写
 String data = br.readLine();
 System.out.println(data);//输出到控制台
 pw.println(data);//转发给客户端
 }catch(Exception e)
 {
 e.printStackTrace();
 }
 finally
 {

```

```

 try
 {
 //关闭读写流
 br.close();
 pw.close();
 }catch(Exception e)
 {}
 }
}
}
public static void main(String[] args)
{
 Server server = new Server();
}
}

```

**Client.java:源代码**

```

import java.net.*;
import java.io.*;

```

```

class Client

```

```

{
 public Client()
 {

```

```

 BufferedReader br = null;

```

```

 PrintWriter pw = null;

```

```

 try

```

```

 {

```

Socket socket = new Socket("localhost",8888);//与服务器建立连接，服务器要先启

```

 //得到 Socket的输入与输出流

```

```

 br = new BufferedReader(new
InputStreamReader(socket.getInputStream()));

```

```

 pw = new PrintWriter(socket.getOutputStream(),true);

```

```

 //先写后读

```

```

 pw.println("Client: 你好！");

```

```

 String data = null;

```

```

 while(true)

```

```

 {

```

```

 data = br.readLine();

```

```

 if(data!=null) break;

```

```

 }

```

```

 System.out.println(data);

```

```

 }catch(Exception e)

```

```

 {

```

```

 e.printStackTrace();

```

```

 }
 finally
 {
 try
 {
 br.close();
 pw.close();
 } catch (Exception e)
 {}
 }
}

public static void main(String[] args)
{
 Client c = new Client();
}
}

```

27 . 谈谈 **final**, **finally**, **finalize** 的区别。

答案：**final** 用于声明属性，方法和类，分别表示属性不可变，注意：如果是基本类型说明变量

量本身不能改变，如果是引用类型，说明它不能指向其他的对象了。但对对象还是可以改变的。

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，表示无论是否出现异常总是执行。

**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。

28 . **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)?

答案：可以继承其他类或完成其他接口，在 **swing** 编程中常用此方式。

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

答案：**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。

30. **&** 和 **&&** 的区别。

答案：**&** 是位运算符，表示按位与运算，**&&** 是逻辑运算符，表示逻辑与 ( **and** )。

另外，**&** 也可以用于逻辑运算，与 **&&** 的区别在于 **&** 是非短路运算符，**&&** 是短路运算符

31. **HashMap** 和 **Hashtable** 的区别。

答案：**HashMap** 是 **Hashtable** 的轻量级实现 ( 非线程安全的实现 ) ，他们都实现了 **Map** 接口，

主要区别在于 **HashMap** 允许空 ( **null** ) 键值 ( **key** ) 与空值 ( **value** ) , 由于

非线程安全，效率上可能高于 **Hashtable**,**Hashtable**不允许有空（**null**）键值（**key**）与空值（**value**）。

32 . 什么时候用 **assert**?

答案：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 <boolean 表达式>**，表示如果表达式为真（**true**），则下面的语句执行，否则抛出 **AssertionError**。

33 . **GC** 是什么？为什么要有 **GC**?

答案：**GC** 是垃圾收集的意思（**Gabage 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式操作方法。

34 . **String s = new String("xyz")**创建了几个 **String Object**?

答案：两个，一个是对象，一个是对象的引用。

35 . **Math.round(11.5)**等於多少？**Math.round(-11.5)**等於多少？

答案：**Math.round(11.5)==12**

**Math.round(-11.5)==-11**，总是返回接近 0 的数。

36 . **short s1 = 1; s1 = s1 + 1**有什么错？**short s1 = 1; s1 += 1**有什么错？

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

**short s1 = 1; s1 += 1**没有错，**s1==2**）

37.**sleep()** 和 **wait()** 有什么区别？

答案：**sleep**是线程类（**Thread**）的方法，导致此线程暂停执行指定时间（休息），把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，时间到了会继续运行。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。

38 . **Java** 有没有 **goto** 语句？

答案：没有

39 . 数组有没有 **length()**这个方法？**String** 有没有 **length()**这个方法？

答案：数组没有 **length()**这个方法，有 **length**的属性。

**String** 有 **length()**这个方法。

40 . 数组是不是基本类型？

答案：不是。

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

答案：方法的重写 **Overriding** 和重载 **Overloading** 是 **Java** 多态性的不同表现。

重载 **Overload** 是一个类中多态性的一种表现，方法名必需相同，参数列表必需不同（长度不同或类型不同），与返回值类型没有关系。构造器也可以重载。方法名和参数列表都相同，只有返回类型不相同则是方法重复定义。编译出错。**Overloaded** 的方法是可以改变返回值的类型。

重写 **Override** 是父类与子类之间多态性的一种表现。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。

42. **Set**里的元素能重复吗？那么用什么方法区分是否重复？

答案：**Set**里的元素是不能重复的，如果重复就重复的就不加到里面。用 **iterator()**方法来区分重复与否。应该覆盖 **equals()**判断两个 **Set**是否相等。

43. “**==**”与 **equal**有何区别？

答案：“**==**”比较的是内存地址，**equal**比较的是内容本身。

44. 给我一个你最常见到的 **runtime exception**。

答案：**IndexOutOfBoundsException** 数组越界  
**NullPointerException** 空指针异常  
**SystemException** 系统异常

45. **error**和 **exception**有什么区别？

答案：**Error**表示系统级的错误和程序不必处理的异常，我们无法处理它。  
**Exception**表示是可以捕捉或者需要程序进行处理的异常。

46. **List**, **Set**, **Map**是否继承自 **Collection**接口？

答案：**List**, **Set**是。**Map**不是

47. **abstract class**和 **interface**有什么区别？

答案：一个只能继承一个抽象类，但却可以实现多个接口。抽象类中可以有也可以没有抽象方法。并且可以定义和常规类一样的变量和方法。而接口中所有的方法都是抽象的，所有的变量都是静态不可修改的。

48. **abstract**的 **method**是否可同时是 **static**，是否可同时是 **native**，是否可同时是 **synchronized**？

答案：都不能。

49. 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答 (**concrete class**)？

答案：接口可以继承接口。接口间继承 (**extends**)，不能实现 (**implements**)。抽象类可以实现 (**implements**)接口，但接口不能实现抽象类。抽象类间也用继承 (**extends**)。抽象类是否可继承实体类，但前提是实体类必须有无参的构造函数。

50. 启动一个线程是用 **run()**还是 **start()**？有什么区别？

答案：启动一个线程是调用 **start()**方法，这并不意味着线程就会立即运行，只是进入了可运行状态。直接调用 **run()**方法不会产生线程，而是把它当作普通的方法调用，马上执行。

51. **Constructor**是否可被 **override**？

答案：不可以，但可以重载。

52. **Class.forName()**的作用？

答：调用该访问返回一个类名为指定字符串的类的对象。

53. 文件读写的基本类有哪些？

答案：**FileInputStream**, **FileOutputStread**, **File**, **IOException**等。

54 . 写一个冒泡排序的程序。

```
public class Sort
{
 public static void main(String[] args)
 {
 int[] array =
{2,3,4,1,43,432,1,344,234,2,3,43,32,434,3432,43,2432,432,4,43};
 int num = 0;
 /*
 冒泡排续
 */
 for(int i = 0;i < array.length;i++)
 {
 /*
 每下底一个元素 , 则调换的次数减一。
 注意 : j<array.length-i-1,不减一则数组下标越界。
 */
 for(int j = 0;j<array.length-i-1;j++)
 {
 int temp = 0;
 /*
 两两比较若前面的大于后面的则进行调换。
 */
 if(array[j] > array[j+1])
 {
 temp = array[j];
 array[j] = array[j+1];
 array[j+1] = temp;
 }
 }
 }
 /*
 输出排序后的数组
 */
 System.out.println("***** 已排序的数组 *****");
 for(int i = 0;i < array.length;i++)
 {
 System.out.print(array[i]+" ");
 num++;
 if(num == 5)//每行输出 5 个。
 {
 System.out.println();
 num = 0;
 }
 }
 }
}
```

```

 }
}
}

```

55 . 串行化的注意事项以及如何实现串行化？ ( **Serializable** )

答案： \*\*\*\*\*

56 . 线程的基本概念、线程的基本状态以及状态之间的关系

答案： \*\*\*\*\*

57 . 线程的同步、如何实现线程的同步？

答案： \*\*\*\*\*

58 . 几种常用的数据结构及内部实现原理。

答案： \*\*\*\*\*

59 . **Socket**通信 (**TCP**、**UDP** 区别及 **Java** 实现方式 )

答案： \*\*\*\*\*

60 . **Java** 的事件委托机制和垃圾回收机制

答案： \*\*\*\*\*

61 . 作用域 **public,private,protected** 以及不写时的区别。

答案： **private**-类可见，不写 -包可见

**protected**包可见，并子类可见 **public**-都可见

62 . **Java** 的国际化

答案： \*\*\*\*\*

63 . 怎样让程序到后台去运行？

答案： \*\*\*\*\*

64 . 怎么把一个文件中的东西做全文替换。

答案： \*\*\*\*\*

65 . **ArrayList** 和 **Vector** 的区别？

答案：就 **ArrayList** 与 **Vector** 主要从二方面来说 .

同步性 :**Vector** 是线程安全的，也就是说同步的，而 **ArrayList** 是线程序不安全的，不是同步的

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。

66 . **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。

67 . 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**

68 . 垃圾回收机制 ,如何优化程序？

答案： \*\*\*\*\*

69 . **float** 型 **float f=3.4** 是否正确？

答:不正确。类型不匹配 ,应该用强制类型转换，如下所示： **float f=3.4f;**

70. **Java** 中异常处理机制，事件机制？

答案： \*\*\*\*\*

71.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；

**import java.util.StringTokenizer;**



```

import java.util.Arrays;
public class Test
{
 public static void main(String[] args)
 {
 String s = "2,654,24,6554,3234,3544,666,354,5435,11";
 StringTokenizer tok = new StringTokenizer(s,",");
 //返回有多少个被分割元素
 int count = tok.countTokens();
 int[] result = new int[count];
 //把分割得到的数字存到数组中去。
 for(int i = 0;i<count;i++)
 {
 String temp = (String)tok.nextElement();
 result[i] = Integer.parseInt(temp);
 }
 //排序
 Arrays.sort(result);
 for(int i = 0;i<result.length;i++)
 {
 if(i==0)
 {
 System.out.print(result[i]);
 continue;//第一个元素的前面不打印“ , ”
 }
 System.out.print(","+result[i]);
 }
 System.out.println();//输出 : 2,11,24,354,654,666,3234,3544,5435,6554
 }
}

```

72 . 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。

答案 : \*\*\*\*\*

73. 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 ?

答案 : 能 , 一个对象的 **synchronized** 方法只能由一个线程访问。但其他线程可以同时访问这个对象的非 **synchronized** 方法

74. **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 ?

答案 : 会执行 , 在 **return** 前执行。

75 . 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。

答案 :  $2 \ll 3$  。如果是 2 的 10 次方则是 :  $1 \ll 10$  .

76. 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？

答案：两个对象的值是否相同，根据的是 **equals** 方法，而 **hash code** 是由 **hashCode()**方法决定。因此值相同的对象 **hash code**可能不同。

77．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 ？

答案：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。

78。编程题：写一个 **Singleton**出来。

方式一：醉汉式

```
public class Singleton
```

```
{
```

```
 private static Singleton test = new Singleton();//静态的。保留自身的引
```

用。

```
 //必须是私有的构造函数
```

```
 private Singleton(){}
 public static Singleton getInstance()
 {
 return test;
 }
}
```

```
 //公共的静态的方法。
```

方式二：懒汉式

```
public class Singleton
```

```
{
```

```
 private static Singleton test = null;//静态的。保留自身的引用。
```

```
 //必须是私有的构造函数
```

```
 private Singleton(){}
 public static Singleton getInstance()
 {
 if(test == null)
 {
 test = new Singleton();
 }
 return test;
 }
}
```

```
 //公共的静态的方法。
```

79.描述一下 **JVM** 加载 **class**文件的原理机制 ？

答案：\*\*\*\*\*

80．试举例说明一个典型的垃圾回收算法？

答案：\*\*\*\*\*

81．请写一个 **java** 程序实现线程连接池功能？

答案：\*\*\*\*\*

- 82 . 给定一个 C 语言函数，要求实现在 java 类中进行调用。  
答案：\*\*\*\*\*
- 83 . 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。  
答案：\*\*\*\*\*
- 84 . 列出某文件夹下的所有文件；  
答案：\*\*\*\*\*
- 85 . 实现从文件中一次读出一个字符的操作；  
答案：\*\*\*\*\*
- 86 . 列出一些控制流程的方法；  
答案：\*\*\*\*\*
- 87 . 多线程有哪些状态？  
答案：\*\*\*\*\*
- 88 . 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。  
答案：\*\*\*\*\*
- 99 . 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。  
答案：\*\*\*\*\*
- 100 . 调用系统命令实现删除文件的操作；  
答案：\*\*\*\*\*
- 101 . C++ 或 Java 中的异常处理机制的简单原理和应用。  
答案：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。
- 102 . Java 的接口和 C++ 的虚类的相同和不同处。  
答案：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。
- 103 . 垃圾回收的优点和原理。并考虑 2 种回收机制。  
答案：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。

104 . 描述一下你最常用的编程风格。

答案：\*\*\*\*\*

105 . heap和 stack有什么区别。

答案：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

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

答案：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

107 . JAVA 类实现序列化的方法？在 collection中，实现比较要实现什么样的接口？

答案：java 类实现序列化需要实现 java.io.Serializable接口。

在 collection 中实现比较要实现 java.lang.Comparable 接口或 java.util.Comparator 接口。

## NHN CHINA 公司笔试题

1 . 在这写代码

```
Var array1 = [10,20];
Var array2 = [8,5];
Var result = array1.add(array2);
Alert(result);
输出结果 18 , 25
```

2 . 解析 dom tree

```
<html>
 <body>
 <input type= text name= username请填写姓名 >
 </body>
</html>
```

3 . 要求按按钮后 my word 向下移 10px

```
Function fun(){
 //在这填写代码
}
<style>
 #name{
 Color:red,
 Height:100px,
 Weight:110px
 Left:50px;
```

```

 Top:50px;
 }
</style>
<div id= name >
 My word
</div>
<input type= button onclick= fun() >

```

4 . 给出一段代码去替代    addEventListener 和 attachEvent  
Function myAttachEvent(o,evt,fun){};

```
-----Alert(o.Xclient+o.Yclient);
```

```
myAttachEvent(document, ' click ',onClick);
```

5 . 用 js 写段代码，当按按钮后自动在    div 层中添加组建

```
<javascript>
```

```
 //在这填写代码
```

```
</javascript>
```

```
<div /*todo something*/>
```

```
<select /*todo something*/></select>
```

```
</div>
```

```
<input type= button id= bu onclick= fun(/*todo something*/) value= add.... >
```

6 . 当选中后背景色改变

```
<javascript>
```

```
 //在这填写代码
```

```
</javascript>
```

```
<body>
```

```
<select /*todo something*/>
```

```
 <option value= red >red</option>
```

```
 <option value= gray >gray</option>
```

```
 <option value= blue >blue</option>
```

```
</select>
```

```
</body>
```

7 . 给出一些 js 代码，说出他们适用于什么浏览器

```
 //这写代码忘了
```

```
(1)var xhr = new XMLHttpRequest();
```

```
.
```

```
.
```

```
.
```

8. 说出一下 4 个 this 中哪个 this 与其他三个不同

//这写代码忘了

9.编写一个函数 (addAll()), 实现如何结果 :

addAll(1,8); //结果 :9

addAll(5,8,-7); 结果 :6

10, 结果相同的语句

var o = {};

Var o = [];

Var o = //;

Var o = new Object();

Var o = new Array();

Var o = new reg(); // 这个不正确 ,忘怎么写的了

11.邮箱的正则表达式

12,实现数字每 1 秒增加 10 个

//这里写 Js 代码

<input type= text id= num value= 0 >

<input type= button onclick= addNum() />

T4game 呈天游软件公司笔试题目

大部分时 J2SE 的知识点。

1. JAVA 的基本数据类型有哪几种? String 是基本数据类型吗? 它们之间的区别是什么?
2. 怎样将数值类型转化为数字类型 ( Interger、Double )怎样将数字类型转化为字符串类型? 怎样取得小数点的前两位, 并且进行四舍五入?
3. JSP的内置对象有那些, 他们的方法是什么?
4. Servlet 的生命周期是什么?
5. Socket ( TCP、UDP ) 和 JAVA 中的通信有什么却别, 它们是怎样实现的?
6. 启动一个线程的方式 run() 还是 start ( ) ?
7. 当一个数据类型的长度超过了 long 型, 请设计一种数据结构来存储这个数据。
8. 内部类是以什么方式实现的?
9. JAVA 三大特征是什么?

10 你希望公司能给你什么回报?

11.你进入公司后将为公司创造怎样的价值?

12.你玩过什么游戏, 到几级了?

13.游戏的分类有那些?

14.你不喜欢和什么样的人在一起工作?

15.你了解什么历史典故？可以谈一个吗？

## 笔试

### 基础部分

#### A . 面向对象

1. 谈谈你对多态的理解。
2. 你是如何使用 `public,private,protected,default` 这些访问修饰符的？
3. `final` 修饰引用类型和原始类型时有什么不同？
4. java 中是值传递还是引用传递？
5. 请你例举出两个使用重载的原因。
6. 你在编程时，何时使用组合，何时使用继承？
7. 请你分析一下 JAVA 的加载过程。
8. 接口可以包含字段吗？如果可以，它们默认的修饰符是什么？
9. `static` 的方法和一般的方法有什么区别 ？
10. 低聚合低耦合，还是高聚合低耦合？

#### B . 异常处理

1. `Error` 和 `Exception` 继承自哪个类？
2. 在子类中覆盖父类的方法，如何处理父类方法声明的异常。
3. 例举一下你经常使用的异常类。

#### C . 类型问题

1. RTTI 是什么？
2. `Class.forName( “ 类名 ” )` 与直接调用构造方法新建对象有什么不同？
3. `boolean.class` 与 `Boolean.type` 有什么不同？

4. java 在何时获得对象的确切类型？

#### D . 数据结构

1. 数组和容器的区别？

2. 方法的返回值可以是数组吗？

3. java 一共有几种容器，例举出来。

4. ArrayList 和 LinkedList 的底层实现有什么不同？

5. 常用的几种排序算法的时间复杂度？

6. 用 java 代码实现一个双向链表。

#### E . 多线程：

1. JAVA 中是单进程多线程还是多进程多线程？

2. 列举出一种出现死锁的情况？ java 是否解决了死锁问题？

3. 如何解决死锁问题？

4. 一个对象被锁定之后，其他的方法是否可以访问这个对象？

#### JSP/SERVLET：

1. jsp 的内置对象及其作用

2. servlet 的生命周期

3. 转发和重定向的区别

4. Filter 和 Servlet 的区别。

5. 何时使用转发何时使用重定向

#### JDBC：

1. 实现一个数据库的连接。

2. 简述链接数据库的步骤。



数据库：

1. 连接查询，视图
2. 数据表的设计（一对一，一对多，多对多）
3. 为什么会发生事务死锁，解决方法？

上机题：

1. 设计一个程序，在页面输入 SQL 语句查询数据库，并在另一个页面显示查询结果
2. 写程序判断一个 IP 地址是否合法？
3. 写程序获取一百个不重复的随机数。
4. 计算字符串“ 23743298 ”基数位的和，偶数位的和。

## 爱嘉途公司笔试题

主要做出境旅游网站，用的是 RUBY ON RAILS。关于 RUBY ON RAILS

只是在上课听老师稍提出了一下。主考官很和善，主要问的是：1：说出

PROTOTYPE 里的一个类或方法（这是 AJAX 里的一个框架，但是为 RUBY ON RAILS 而产生的，后来被 AJAX 延用了）

2：WEBLOGIC 的内部架构及相关知识。

3：平时的 SVN 怎么用的？还用过其它的插件没有？

4：除 JAVASCRIPT 外你还知道哪种动态脚本语言？

5：HASH 的工作原理

6：怎么遍历一个树形数组？

### 北京安托系统集成有限公司笔试题

1. #include <filename.h> 和 #include "filename.h" 有什么区别
2. const 有什么作用？
- 3, 4 记不清了
5. c++ 改错题
6. C++语言里面的 String 类，让重写普通构造函数，拷贝构造函数，析构函数，还有一个改值函数
7. 双向链表里面，在 P 后面的值之前插入 s，用 C语言写出逻辑代码
8. 用 VB语言实现冒泡排序
9. 将“销售部”里面工资小于 600 的员工的工资提高 %10; 写出 SQL语句。
10. 五段英译汉。

### 北京东华合创数码科技股份有限公司

#### 笔试试卷（Java 语言部分）

姓名：\_\_\_\_\_

联系方式：\_\_\_\_\_

1. which won't cause a compiler warning or error (多选)

- a) float f = 1.3;
- b) char c = ?a ' ;
- c) byte b = 257;
- d) boolean b = null;
- e) int i = 10;

2. 请选择下面这段代码的输出结果？

```
int i = 0;
switch (i) {
case 0:
 System.out.println("zero");
case 1:
 System.out.println("one");
case 2:
 System.out.println("two");
 break;
case 3:
 System.out.println("three");
```

}

- 1) zero
- 2) zero,one
- 3) zero,one,two
- 4) zero,one,two,three

```
3. public class Test{
 private static int j=0;
 public static boolean methodB(int k){
 j+=k;
 return true;
 }
 public static void methodA(int i){
 boolean b;
 b=i>10&methodB(1);
 b=i>10&&methodB(2);
 }
 public static void main(String args){
 methodA(0);
```

**17)**

```
 }
 }
```

what is the value of j at **line 17**?

- 1 ) 0
- 2 ) 1
- 3 ) 2
- 4 ) 3

4. If we execute the code below with `java Test Red Green Blue` , what is the result?

```
public class Test{
 public static void main(String[] args){
 String foo=args[1];
 String bar=args[2];
 String baz=args[3];
 }
}
```

what is the value of baz?

- A. baz has value of ""
- B. baz has value of null

- C. baz has value of "Red"
- D. baz has value of "Blue"
- E. baz has value of "Green"
- F. the code does not compile
- G. the program throw an exception

5. 请选择下面这段代码的输出结果 ？

```
1)public class Test{
2)public static void main(String[] args){
3) class Foo{
4) public int i=3;
5) }
6)Object o=(Object)new Foo();
7) Foo foo=(Foo)o;
8)System.out.println(foo.i);
9) }
10) }
```

- A.compile error at line 6
- B.compile error at line 7
- C.compile error at line 8
- D.print out 3

6.

```
int index=1;
String[] test=new String[3];
String foo=test[index];
```

what is the result of foo?

- A. ""
- B.null
- C.throw a Exception
- D.not compile

7. 下面的五个选择中哪两个描述是正确的？（多选）

- A. static inner class requires a static initializer
- B. A static inner class requires an instance of the enclosing class
- C. A static inner class has no reference to an instance of the enclosing class
- D. A static inner class has accesss to the non-static member of the other class
- E. static members of a static inner class can be referenced using the class name of the static inner class

8. 请选择下面这段代码的输出结果 ？

```
class A{
```

```

public int getNumber(int a){
 return a+1;
}
}
class B extends A{
 public int getNumber(int a, char c){
 return a+2;
 }
 public static void main(String[] args){
 B b=new B();
 System.out.println(b.getNumber(0));
 }
}

```

- A. compilation succeeds and 1 is printed
- B. compilation succeeds and 2 is printed
- C. An error at line 8 cause compilation to fail
- D. An error at line 14 cause compilation to fail

9. class ExceptionTest{

```

 public static void main(String args[]){
 try{
 methodA();
 }
 catch(IOException e){
 System.out.println("caught IOException");
 }
 catch(Exception e){
 System.out.println("caught Exception");
 }
 }
}

```

If methodA() throws a IOException, what is the result?

If we change the sequence of catch,what                      ?s the result

10. 请写出下面这段代码的输出结果    ?

```

public class Test{
 public static void main(String[] args){
 StringBuffer a=new StringBuffer("A");
 StringBuffer b=new StringBuffer("B");
 }
}

```

```

 operate(a,b);
 System.out.println(a+","+b);
 }
 public static void operate(StringBuffer x, StringBuffer y){
 x.append(y);
 y=x;
 }
}

```

11. 请写出下面这段代码的输出结果 ？

```

class Shape{
 Shape(int i){
 System.out.println("This is Shape" + i);
 }
}
public class Circle extends Shape{
 static Shape s1 = new Shape(1);
 Shape s2 = new Shape(3);
 Circle(int i){
 super(i);
 System.out.println("This is Circle" + i);
 }

 public static void main(String args[]){
 Circle c1 = new Circle(2);
 }
}

```

12. 数组有没有 length() 这个方法 ？ String 有没有 length() 这个方法 ？

## 北京伽文信息技术公司笔试题

请写出十种以上的 java 开源软件，并用一句话说明功能

写出你认为做为一名程序员必须要阅读的三本书

简述 abstract class 和 interface 的区别

构筑工具 ant 使用的缺省 xml 文件名是什么

简述什么是 SOA

简述什么是工作流

XML 在 java 中有哪些解析方法

说说下面语句是否有错误，并指出错在哪里

1)

```
import java.util.List;
```

...

```
List list = new List();
```

2)

```
public class MyString extends String{
```

```
.....
```

```
}
```

3)

```
public class MyServlet extends HttpServlet{
```

```
 private map myInput;
```

```
 public void dGet(HttpServletRequest request,HttpServletResponst response)throws
```

```
Exception{
```

```
 myInput.put(request.getParameter("command"));
```

```
 }
```

```
}
```

4)

```
public class MyData extends MyData1,MyData2{
```

```
.....
```

```
}
```

WAR 包中 jar 一般放在哪个目录中

简述 JUnit 和单元测试的作用

GOF 的设计模式中提到多少中设计模式，分为哪三种类型

你任务软件开发人员的哪个素质最重要，沟通能力、技术能力、业务需求能力

尽可能多的列出一个企业“机构管理”应具备哪些功能

列举权限管理中有哪些表、表关系、各表哪些功能

## 孙文中总结笔试题

- 1 写出十种开源框架，用一句话说明他们的作用。
- 2 什么是 SOP。
- 3 什么是工作流。
- 4 多少种设计模式，三种分类。
- 5 找错，都很简单

有一道题认真就行

```
Class MyServlet extends HttpServlet {
 Private map myMap;
 Public void doPost (req,res) throws Exception {
 myMap.put(req.getParamater(name));
 }
}
```

以上有哪些错误，还会问到用 hashTable 和 hashMap，和线程安全的问题。

- 6 设计简单的权限控制的表和结构 (user,role,menu,function)。

(写汽车租赁的就行，然后他问的时候给他讲解一下)

- 7 你认为在“部门管理”中有哪些功能

- 8 junit 相关(谈的时候会问到一些，使用的版本，怎么用，什么时候用，居然连标签首字母大小写都问了，哈哈)

剩下的都是一些简单的东西，忘记了

单独问的时候，问到了关于 js 构建对象有几种方式，是什么， json 用得是什么格式的

这家感觉考得比较基础，技术部人问的时候相关的比较多。

## 畅子伟总结

名词解释： Ajax 和 AOB

描述什么环境下用一下三种模式

单例模式：确保某一个类只有一个实例，而且自行实例化并向整个系统提供这个实例。

代理模式：

工厂模式： 有时候需要单例或者创建时隐藏过程细节或者加一些扩展， 比如日志记录等， 需要用工厂模式。

你经常去的有关 IT 技术的网站有哪些，它们各有什么特点？

改错

A.

```
public class Mylist extends list{

}
```

设计一个 JUnit 单元测试



（这题用简单的 Helloworld 实现就行了 判断字符串中是否包含空字符）  
6.tomcat 的缺省端口号是什么？修改端口号在那个目录下的那个文件中？

张强总结：

车源找货源，或是货源找车源，以一放为用户设计需求分析，用文字或是图。

卿佐晨总结：

找错 ,多处错误

```
public class MyFile1 implements Runnable {

 public void run() {
 while(true){// 死循环不知道算不算错
 try {
 FileReader fr = new FileReader(new File("a.txt"));// 这具体的文件路径记不清了 ,可能有错
 String line = fr.readLine();//FileReader 没有 readLine()
 System.out.println(line);
 } catch (FileNotFoundException e) {

 e.printStackTrace();
 }
 sleep(1000);// 该处应为 Thread.sleep(1000);
 }
 }
}
```

北京伽文信息技术有限公司                      地址：北京市海淀区中关村东路                      123 号 都市网景大厦    B  
座 1401    电话： 01062525698

路线：知春路翠宫饭店南或者北三环联想桥向北路西

## 北京润乾软件笔试题

问答题

1. 排序方法有多种包括有：冒泡排序，插入排序，快速排序、选择排序。请用任何一种语言或伪代码（最好用 java 语言），写出选择排序。
2. 你熟悉 javascript 语言吗？如果 table 表格的属性中有个 id，如果 id= ‘ td1 ’，请问你如何获取第 4 行第 3 个单元格对象？
3. 数组有没有 length()这个方法？String 有没有 length()这个方法？

4. 假如你知道北京润乾公司，你如何解说明白给陌生人找到本公司确切的地址：
- 1) 要求用文字叙述
  - 2) 要求用图形绘画出来
5. 在 Servlet 中完成正确接受中文及向显示页面响应的代码。 试写一段程序 ,用来接收从 SERVLET 用 POST 和 GET 方法传来得中文参数
- ```
public void service (HttpServletRequest request,HttpServletResponse response)
{
    ??????????????
    ..
}
```
- 6.基于 J2EE 的 Web 应用的目录结构，以及各目录下的部署文件
- 7.完成数据库查询：

员工表：（ emp 表）

| empId | empName | empSal(员工薪水) |
|-------|---------|----------------|
| 1 | ABC | 1200 |
| 2 | BCD | 2000 |

薪水级别表：（ sal 表）

| minSal | maxSal | salClass(薪水级别) |
|--------|--------|------------------|
| 1000 | 1500 | 1 |
| 1500 | 2200 | 2 |

问题：请查询出员工号为 “ 1 ” 的薪水级别？

8. 假如有 5 个信封中，分别装有 5，10，20，50，100，每个信封中只能装有一张，每个人只能持有一个信封，且只知道自己信封中钱数，当你知道自己持有的信封为 10 元时，你会和别人的信封交换吗？为什么？

复试时的面试题（技术 ）

- 1.谈谈 struts 的作用
- 2.谈谈 ajax 的作用
- 3.你了解那些设计模式？说说在什么情况下必须使用你了解的设计模式
- 4.说说软件的开发周期
- 5.谈谈 arraylist 与 hashtable 的区别

孟庆欢总结面试题

- 1.站在客户角度怎么去理解 MVC 模式和单例模式
- 2.Spring 和 EJB 有什么区别？各有什么优点？
- 3.ArrayList 和 hashtable 有什么区别？
- 4.谈谈你的项目经验和项目中的角色

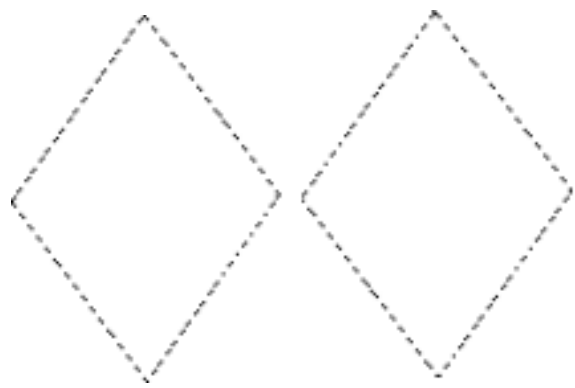
5.Ajax 的优点是什么？

北京润乾软件技术有限公司 地址：北京市海淀区上地信息路 19 号商服中心
402 室 邮编：100085 010-51295366 网址：<http://www.runqian.com.cn>
公交车站：上地环岛 华联商场旁边

北京市东方爱婴咨询面试题

吴安民总结（ 2009-2-19）

1. 写出三个你常用的技术论坛，并写出用户名。如果你要搜索不懂的技术会选择哪个搜索引擎？怎样搜？（比如，你不知道怎么把正在运行的 java 写的动态网站静态化，在搜索引擎里该怎么写？写出关键字。 ）
2. 你最喜欢的和你最熟悉的编程语言是什么？为什么？
3. 写出你熟悉的编码方式和命名规范。
4. 英文翻译题。（多线程的问题）
5. 一个应用程序的开发环境是 windowsXP+eclipse+Tomcat，生产环境是 linux+Tomcat,在开发环境下正确，但在生产环境下出现异常，请问你用什么样的方式来检查？
6. 如果给你一个任务，开发 800 客服电话，要求记录通话记录和回访记录，你决定用怎样的工作流程来完成？
7. 一个网站有免费会员跟付费会员，他们都能积分，积分都能抽奖，这些信息都放在一个会员表里面，积分的回报率不同，如果你遇到这样的情况你该怎么解决？
8. 脑筋急转弯：有由 8 根火柴组成的两个菱形，只动两根使其变成一个菱形。



北控三兴笔试题

1. 写出 jsp 的 9 各内置对象
2. 一道 javascript 题：

```
<% var a;  
    for(int i= 0;i<5;i++){a=a+2;}%>  
<script>
```

```
    <%=a>;  
    alert(--a-1);  
</script>
```

... 还有一点记不清楚了

反正也是 alert() 出一个值

3. 用数据库语句写出 sqlserver 和 orcal 的左外连接语句
4. 递归的算法求出 n 的阶乘
5. m,n 的最大公约数
6. 一个员工表，字段有员工号，员工姓名，员工的年龄。写一个 SQL 语句，要求删除年龄大于 30，并且姓李的员工信息。
7. web.xml 中都要配置那些信息，试举例说明。
8. 解释数据库中关系，事务,和事务的原子性
9. 给出父类子类算结果的程序题：

```
class A {  
    new A(){  
        System.out.print(      2  );  
    }  
}
```

```
class B extends A{
```

```

        public B(){
            System.out.print(        b    );
        }
    }
    class C{
        public static void mian(String[] args){
            A    a = new B();
            a = new B();
        }
    }
}

```

请写出程序运行的结果：

10. 猴子有 100 个香蕉 ,到家有 50 米的距离 ,猴子每走一米就要吃一个香蕉 ,一次最多只能带 50 只香蕉 ,问猴子最多能带回家多少香蕉 ?请详细给出如何搬的。

技术面试：

针对简历上上的技术发问， 问的都不算很深。 问问你做过的项目， 项目中所用到的技术。你负责的那块是怎么做的。

特别还问道，会不是使用 oracle 做管理。。

对数据库要求好像蛮多，怀疑要招 dba , , , 只是猜测。。。

博思永拓笔试题

- 1、找出程序中的错误，并修改

```

public class Test
{
    public void aMethod(String str)
    {
        System.out.println("Hello" + str);
    }
    public static void main(String[] args)
    {
        String s = "Tom";
        Test.aMethod(s);
    }
}

```

- 2、找出程序中的错误，并修改

```

interface Test1
{

```

```

        void show();
    }
    interface Test2
    {
        void show();
    }
    interface Test3 extends Test1,Test2
    {
        Deom d = new Deom();
    }

    public class Test implements Test3
    {
        public void show()
        {
            Deom d=new Deom();
        }
    }

```

3、试图编译运行下面的代码会发生什么情况

```

public class Q {
    public static void main(String argv[]){
        int anar[]=new int[]{1,2,3};
        System.out.println(anar[1]);
    }
}

```

博彦科技笔试题

尹俊总结：

一．选择题：

我记不清了，选择题有 5-6 道，有两到是计算机基础方面的题， 不是编程题。 都是用英语问的，选项也是英语
判断数据类型 和 int 有关的（选择题，也是用英语问的）

二．填空题：

```

Int sum ( int a[],int n )
{
    if (n>0) return_____；
    else return_____；
}

```

三．问答题：

String 和 StringBuffer 的区别 （用英语问的）

abstract class 和 interface 有什么区别 （用英语问的）

list set map 之间的关系 （用英语问的）

黑盒测试和白盒测试 （用英语问的）

单元测试 junit 有关的（用英语问的）

数组有没有 length() 这个方法？String 有没有 length() 这个方法？（英语问的）

么雪娜总结：

应聘博彦科技有限公司软件测试实习工程师面试：

笔试内容：（半个小时）

专业知识部分：

1.xcopy F:\ backup E:\

以上这句命令实现的是什么功能

2.ps/2 的键盘接口不能识别，分析可能的原因

3. 显示器如果不亮了，分析可能的原因

4. 英文 Windows 操作系统下如何正确显示中文字符

5. 你是如何理解工作组和域的概念的（选作题）

必做题还有一道，抱歉想不起来了。

英文水平部分：

1. 有关 MS/DOS的一段文章（英译汉）

2. 有关 MSN的有关文章（汉译英）

面试部分（半个小时）

以下是能想起来的：

1. 四级成绩

2. 做的项目中遇到过什么问题，你是如何解决的

3. 自认为自己是一个办事认真，有责任心的人么

4. 如何看待加班这个问题

5. 在使用计算机时发生的一些小问题，能自己解决么

6. 能否保证上班时间

7. 如果给你个任务，你在自己看不懂的情况下你会怎么做

8. 假如让你测试别人做的软件，你会怎么去测

9. 能否用英文描述自己的功能模块

10. 以前是否自己写过文档

希望能对以后的学员有所帮助

长峰科技面试总结

王涛总结：

（注意：以下问题他们会根据你的回答继续提问）

- 1、请做一个自我介绍。
- 2、请谈谈你最熟悉的技术？
- 3、你为什么从上一家单位辞职？
- 4、解释一下 Spring
- 5、谈谈 Struts
- 6、描述 JSP 和 Servlet 的区别、共同点、各自应用的范围
- 7、Oracle 数据库存储过程的特征
- 8、你熟悉的 Web 容器有哪些，如何在 tomcat 中配置数据库连接池、如何处理以 get 方式提交过来的字符编码
- 9、对面向对象的理解？
- 10、项目中用到什么框架？
- 11、项目中用什么方式进行的数据库操作？
- 12、用什么做的页面显示，JavaScript 的相关问题。
- 13、面向对象的三大特征，封装，继承，多态在什么地方用过。
- 14、介绍一下 MVC 模式
- 15、你是否熟悉 Webwork 简单介绍一下
- 16、如何整合 Struts 和 Spring，如何整合 Spring 和 Hibernate
- 17、hibernate 的优缺点分别是什么
- 18、请说出 struts 框架的几大组件？
- 19、Spring 是什么？优点是什么？
- 20、什么是 AOP，AOP 的作用是什么？
- 21、你是否熟悉 iBatis
- 22、谈一下 hibernate 中的多对多关系，以及他是怎么实现的
- 23、Weblogic 的热发布
- 24、EJB 容器提供的服务
- 25、Dom 与 Sax 相比它们的优缺点是什么？

创娱天下笔试题

- 1.阐述你对 mvc 的理解，结合框架？
- 2.列举同步和异步在实际项目中的应用？
- 3.Abstract 和 interface 的实际应用？

- 4.常用设计模式，结合框架？
- 5.java 比 .Net；和 php 的优势？
- 6．阐述所用到的框架的优缺点？
- 7．打印自然数 N 以内的所有质数？
- 8．阐述你对单点登录的理解？

姓名： _____

日期： _____

创原天地 Java 笔试题目

1. 下列结论不正确的是（ 5 分） **cd**

- A . call by value 不会改变实际参数的数值
- B . call by reference 能改变实际参数的参考地址 (reference)
- C . call by reference 不能改变实际参数的参考地址 (reference)

- D . call by reference 能改变实际参数的内容**

2. 检查以下代码，并选择答案（ 5 分） **c**

```
public class Test{
    public void printStr(final String str)
    {
        str =str+ " ! "
    }
    public static void main(String args[])
    {
        Test test=new Test();
        Test.printStr( " Hello world " );
    }
}
```

- A: 执行打印出 Hello world!
- B: 执行期间抛出异常
- C:编译不通过

- D:以上答案都不是**

3. 下面关于变量及其范围的陈述哪些是不对的（ 5 分） **b**

- A. 实例变量是类的成员变量。
- B. 实例变量用关键字 static 声明。
- C. 在方法中定义的局部变量在该方法被执行时创建

- D. 局部变量在使用前必须被初始化**

4. 请写出如下代码的运行结果 (5 分)

```

public class Test extends TT{
    public static void main(String args[]){
        Test t = new Test("Tom");
    }
    public Test(String s){
        super(s);
        System.out.println("How do you do?");
    }
    public Test(){
        this("I am Tom");
    }
}
class TT{
    public TT(){
        System.out.println("What a pleasure!");
    }
    public TT(String s){
        this();
        System.out.println("I am "+s);
    }
}

```

5. 请写出如下代码的运行结果 (5 分)

```

class Aclass{
    void go(){
        System.out.println("Aclass");
    }
}
public class Bclass extends Aclass{
    void go{
        System.out.println("Bclass");
    }
}
public static void main(String args[]){
    Aclass a=new Aclass();
    Aclass a1=new Bclass();
    a.go();
    a1.go();
}

```

6. 请写出如下代码的运行结果 (5 分)

| | |
|---------------------------------------------------|-----------------------------------------|
| class Useful { | |
| public void f() {System.out.println(| “ Nowin f method of super class ”);} |
| public void g() {System.out.print(| “ Now in g method of super class ”);} |
| } | |
| class MoreUserfuld extends Useful { | |
| pubic void f() {System.out.println(| “ Nowin f method of derived class ”);} |
| pubic void g() {System.out.println(| “ Nowin g method of derived class ”);} |
| public void u() {}{System.out.println(| “ Nowin u method of derived class ”);} |
| public void v() {}{System.out.println(| “ Nowin v method of derived class ”);} |
| pubic void w() {}{System.out.println(| “ Nowin w method of derived class ”);} |
| } | |
| public class Test { | |
| public static void main(String[] args) { | |
| Useful[] x = { new Useful(), new MoreUserful() }; | |
| x[0].f(); | |
| x[1].g(); | |
| ((MoreUserful)x[1]).u(); | |
| } | |

}

7. 请说明 Java 中访问修饰符 public、protected、private 和缺省修饰符修饰数据成员和成员方法时所限定的访问范围（ 10 分）

| | 在同一类中 | 同一包中 | 不同包中 | 同一包的子类中 | 不同包子类中 |
|-----------|-------|------|------|---------|--------|
| public | yes | yes | yes | yes | yes |
| protected | yes | yes | no | yes | yes |
| package | yes | yes | no | yes | no |
| private | yes | no | no | no | no |

编写程序： 创建一个链表（ ArrayList 或 LinkedList ），并把如下字符串（ red , green , yellow , black , white , blue , cyan , grey ）增加到链表中，然后用迭代子在控制台打印输出链表中的所有元素（ 20 分）

```
8. public class test {  
  
    public static void main(String[] args) throws Exception{  
  
        ArrayList al= new ArrayList();  
  
        al.add( " red");al.add( " green");al.add( " yellow...");  
  
        Iterator it = al.iterator();  
  
        while(it.hasNext()){  
  
            system.out.println(it.next());  
  
        }  
    }  
}
```

9. 请用 Struts 完成以下描述的功能（ 20 分）

用户从 IE 客户端输入字符串 Hello , 并提交给 HellAction , 在 Action 中给 Hello 转换成 hello world! 并显示给用户

下面为处理流程的配置文件



```
<action path = "HelloWorld" type =  
"hello.HelloAction" name = "HelloForm" scope =  
"request" validate = "true" input = "hello.jsp" >  
  <forward name="SayHello" path="sayhello.jsp" />  
</action>
```

表单 HelloForm.java

```
public Class HelloForm{  
    private String content;  
  
    public String getContent(){  
        return this.content;  
    }  
  
    public void setContent(String content){  
        this.content = content;  
    }  
}
```

请实现 **hello.jsp** 、 **HelloAction.java** 、 **sayhello.jsp**

Hello.jsp:

```
<%@ page contentType="text/html; charset=UTF-8" language="java" %>  
  
<html>  
  
  <head><title>Simple jsp page</title></head>  
  
  <body>  
  
    <form action= " HelloWorld.do " name= " HelloForm " method= " post " >  
  
      <input id= " msg " name= " msg " type= " text " />  
  
      <input type= " button " value= " submit " onclick= " submit() " />  
  
    </form>  
  
  </body>  
  
</html>
```

HelloAction.java:

```
public class HelloWorld extends Action
{
    public ActionForward execute(ActionMapping actionMapping, ActionForm actionForm,
        HttpServletRequest servletRequest, HttpServletResponse servletResponse) throws
        Exception
    {
        String msg= servletRequest.getParameter(        “ msg” );

        If(msg.equals(    “ Hello  ” )){

            servletRequest.setAttribute("msg", "hello World ");

            return actionMapping.findForward("    SayHello  ");

        }

    }
}
```

Sayhello.jsp:

```
<%@ page contentType="text/html;charset=UTF-8" language="java" %>

<html>

    <head><title>Simple jsp page</title></head>

    <body>

        <font>${msg}</font>

    </body>

</html>
```

10. 现在有一名为 UserTab.java 的 POJO 对象，请编程这个对象的数据写入到数据库对应的表 USERTAB 中（ 20 分）
表 UERTAB 的表结构如下

表 USERTAB		
USERID	INTEGER	PK
USERNAME	VARCHAR2(20)	
ADDRESS	VARCHAR2(100)	

```
public class UserTab
{
    private java.lang.Integer userID
    private String userName;
```

```

        private String address;

        public UserTab(java.lang.Integer userID)
        {
            this.userID=userID;
        }

        public String getUserName()
        {
            return this.userName;
        }

        public void setUserName(String userName)
        {
            this.userName = userName;
        }

        public String getAddress()
        {
            return this.address;
        }

        public void setAddress(String address)
        {
            this.address=address;
        }

        public java.lang.Integer getUserID()
        {
            return this.userID;
        }

        public void setUserID(java.lang.Integer userID)
        {
            this.userID =userID;
        }
    }

```

请实现方法 insertUser(int userID,String userName,String address) 通过 UserTab.java 的 POJO对象将用户信息插入到数据库的表 USERTAB 中

```

public class UserDao{
    private Session session
    public UserDao(Session session){
        this.session =session;
    }
    // 请实现此方法
    public void insertUser(int userID,String userName,String address){
        UserTab user=new UserTab();
        user.set userID(      userID      ); .....
        Configuration cfg = new Configuration().configure();
        SessionFactory sf = cfg.buildSessionFactory();
        Session session = sf.openSession();
        Transaction tx = session.beginTransaction();
        session.save(user);
        tx.commit();
    }
}

```

}

11. 写作题目：请描述一个项目的开发过程（ 20 分）

项目过程

1、项目启动

- 1) 项目组成员（公司成员、客户成员）
- 2) 制定项目预期目标
- 3) 制定项目计划周期
- 4) 建立好项目组成员沟通机制

2、需求调研

- 1) 创建调研计划、协调调研时间
- 2) 收集客户资料，获取客户需求
所有的资料都需要保留一份，资料中存疑的需要及时询问
- 3) 编写需求文档
重点描述出客户的业务流程和性能要求。
采用 Word、Excel、Rose 等形式。
- 4) 需求变更记录
- 5) 确定开发环境和运行环境
- 6) 扩展性要求
- 7) 与旧系统的接驳要求。
- 8) 估算出项目工作量

本阶段需要一套需求管理系统来进行需求的管理。

本阶段的需求文档也是用户测试的依据。

3、系统设计 /详细设计

一个系统可以分为基础平台和应用模块两部分。

- 1) 选择基础平台， 无论是采用第三方平台还是自行开发平台，都需要深入了解， 查看是否符合要求。
- 2) 应用模块设计（针对业务流程）
- 3) 中间件的采用或自行开发，需要深入了解。
- 4) 用户界面的设计
如果用户界面设计完毕并确认，即可初步写出用户使用手册、管理员使用手册。
- 5) 变更记录

本阶段的系统设计是集成测试的依据。

4、程序开发

创建开发任务计划表、开发计划日程表

- 1) 优先编写测试用例

- 2) 按照编码规范编写代码
 - 3) 按照文档注释规范注释
- 以上形成开发文档。

本阶段需要一套版本管理系统。

本阶段的测试用例也是单元测试的依据。
如果能做到，最好每日构建。

5、测试

本阶段需要一套 Bug 管理系统，形成需求、设计、开发、测试互动。

- 1) 编写测试计划和测试方案
- 2) 功能测试
 - 单元测试、集成测试
- 3) 性能测试
 - 集成测试、压力测试

如果能做到，最好能进行自动化测试。
如果能做到，做分析统计工作。

最后形成测试报告。

6、试用、培训、维护

本阶段需要解决：

- 1) 解决异地修改和公司修改的同步问题。
- 2) 用户测试中的 Bug 修改问题，按照级别分为
 - a) 程序 Bug
 - b) 设计变更
 - c) 需求变更

尽量按照 a b c 的顺序来进行修改，尽量避免 b、c 级的修改。

最后形成安装手册、维护记录。

大唐笔试题

简答题：

- 1、String 与 stringBuffer 的区别，你在什么情况下使用这两个类？
- 2、abstract class 与 interface 的区别，分别在什么情况下运用？
- 3、你常见的设计模式有哪些？写出工厂模式及单例模式
- 4、xml 解析方式有哪些？各有什么有缺点？
- 5、手写一个排序算法。
- 6、简述 hibernate 工作机制及常用类及方法
- 7、private 与 final 修饰符
- 8、面向对象的特征有哪些方面？

- 9、java 中跳转的实现方式。之间的区别 ？
- 10、运行时异常与一般异常的区别？
- 11、你对事物控制的理解？
- 12、运用匿名内部类启动一个线程，并打印出字符串 ” Running “ 。
- 13、MVC 的各个部分都有那些技术来实现 ？如何实现 ？
- 14、ArrayList 、Vector、HashMap、Hashtable 中那些是线程安全的。 HashMap、Hashtable 那个可以有空的 key 及 value

数据库方面：

sql 语句：

exists 与 in 执行效率上那个更好，为什么？

union 与 union all 方面（具体问题记不太清了）

delete、drop、Truncate 方面（具体问题记不太清了）

表 A 字段 m , n , x , y , 其中 m,n 为 number 类型 ,x 为字符类型 , y 为日期类型

- 1、创建表 A 并且 m 不能重复， n 初始值为 0，m、n、y 不能为空
- 2、修改表 A 把 n 的初始化值改为 1
- 3、更新表 A 中 m 值在 5 到 15 之间的数据，将 n 该为 ” datang “， y 为系统当前日期。
- 4、假设 y 为员工的生日，查询出生日相同人的 x。

当乐网笔试题

1. 说说 vector ，arraylist 和 linkedlist 的异同。
2. 接口和抽象类的区别，开发中应该如何选择。
3. 设有 n 种不同面值的硬币，各硬币的面值存于数组 T [1:n] 中。现要用这些面值的硬币来找钱。用编程的方式实现找 k 分钱，所需要的最少的硬币数。
4. 写出一个购物车的接口，只要写出方法和参数即可。
5. 如果你快要下班时，接到上次下达的一个任务，不知道任务的重要程度，而你又和同学约好了去聚会，你给怎么办？
6. 写一个你认为效率最高的数组排序方法。

定谊科技面试题

基础题 基本都是蓝点发 122 到里面的几道

其他基础题是：

- 1.DOM 事件冒泡用 JAVASCRIPT 代码实现

2.DOM 节点数量用 JAVASCRIPT 实现

问答题：

- 1.一个 5 升水桶一个 3 升水桶怎么计算出 4 升水？
- 2.12 个小球 其中只有一个小球重量 可能轻可能重，用天平不能超过三次 找到这个重量不一的小球？
- 3.冒泡排序法

选择题是英文的我没记

东方国信面试题总结

那人事好象是姓马，女的。我办入职是她办的。（不好意思。只能提供这些了）

下边是我面试东方国信的题，没有机试。可能有出入，我是凭印象写的

1.有两个表：

表 userlist

telephone	rent	accounts
10041	aaaa	12.05
10042	aaaa	22.12
10043	ccc	33
10044	bbb	25.01

表 charger

telephone	fee01	fee02	fee03
10041	11	22	9
10042	12	10	10
10043	24	33	12

写 sql 语句，查询结果为：

telephone	user(int)	rent	accounts	fee01	fee02	fee03
10041	2	aaaa	34.17	23	32	19
10043	1	ccc	24	24	33	10
10044	1	bbb	25.01			

2.叙述 js 中 class 的创建，以及 js 中怎么样扩展 property 文件。

3.还有一个是一段代码，找错。是关于运算优先级的。

4.有如下 html

```
<html>
  <table>
    <tr onClick=      insertTr()      >
      <td></td>
```

```
        </tr>
    </table>
```

<html>

如果在点击后，添加一个 <tr>

5.写出在 js 中常用的导入文件。

6.写一个函数，实现如下功能：

1 元钱用 5 角，2 角，1 角，5 分，2 分，1 分，可以有多少种组合方式。

例 2 分可以有两种组合方式，2 个 1 分和 1 个 2 分。1 分有一种组合方式，1 个 1 分。函数

只需返回组合方式的个数，不需显示具体怎样组合的。

恩世教育投资有限公司笔试题

一、选择题

```
1 . public class ExSuper {
    String name;
    String nick_name;
    public ExSuper(String s, String t) {
        name = s;
        nick_name = t;
    }
    public String toString() {
        return name;
    }
}

public class Example extends ExSuper{
    public Example(String s, String t) {
        super(s, t);
    }
    public String toString(){
        return name+" a.k.a " +nick_name;
    }
    public static void main(String[] args) {
        ExSuper a=new ExSuper("First" ,"1st");
        ExSuper b=new Example("Second","2nd");
        System.out.println( "a is "+a.toString());
        System.out.println( "b is " +b.toString());
    }
}
```

A 编译时会出现异常。

B 运行结果为：

a is First

b is second

C 运行结果为：

a is First
b is Second a.k.a 2nd

D 运行结果为：

a is First a.k.a 1nd
b is Second a.k.a 2nd

```
2. abstract class MineBase{  
    abstract void amethod();  
    static int i;  
}
```

```
Public class Mine extends MineBase {  
    public static void main(String argv[]){  
        int [] ar= new int [5];  
        for ( i =0; i <ar. length ; i ++){  
            System. out .println(ar[ i ]);  
        }  
    }  
}
```

- A 打印 5 个 0。 B 编译出错，数组 ar[] 必须初始化。
C 编译错误，Mine 应声明为 abstract。 D 出现 IndexOutOfBounds 的例外。

```
3. public class Foo{  
    public static void main(String[] srgs){  
        try{  
            return;}  
        finally{System.out.println( " Finslly " );  
        }  
    }  
}
```

- A 程序运行正常，但不输出任何结果。 B 程序正常运行，并输出“ Finall ”。
C 编译能通过，但运行时会出现一个例外 D 因为没有 catch 语句块，所以不能通过编译。

4. 以下 sql 语句正确的是：

- A select studentid,depart,count(*) from student group by depart;
B select studentid,count(*) from student;
C select depart,max(avg(age)) from student group by depart;
D select studentid,avg(score),max(score) from score group by studentid;

5. <jsp:useBean> 标签中 scope 的范围由小到大是

- A page session request application B application page request session
C application request session page D page request session application

6. 要让不同用户共享一个 JavaBean 的对象则该对象的范围应该设置为：

- A session B page C application D request

7. 关于自定义标签的使用不正确的是：

- A 必须在 JSP 页面中使用 <@taglib> B 必须经过 web.xml 映射
C 必须存在 .tld 文件 D 自定义标签的类必须是 public 的。

8. 关于外键约束不正确的是：

- A 外键约束与唯一键约束不能同时存在与一个字段
 B 子表外键列的值必须在主表参照列值的范围内，或者为空
 C 主表外键值被子表参照时，主表记录不允许被删除
 D 外键参照的是主表的主键或者唯一键
9. 下面的类，正确的描述是：
- ```
public class Example{
 public static void main(String[] arge){
 static int x[]=new int[15];
 system.out.println(x[5]);
 }
}
```
- A 编译错误 B 编译正确，执行错误 C 输出 0 D 输出是 null
10. 关于 Oracle 中存储过程与函数错误的是：
- A 函数必须定义返回类型 B 存储过程不能定义返回类型  
 C 函数可以用 out 类型参数 D 存储过程可以用 in out 类型参数
- 11 . 关于 Servlet 错误的是：
- A 在 MVC 模式中 Servlet 一般用作控制器  
 B Tomcat5.0 中 Servlet 必须通过 web.xml 映射访问  
 C Servlet 中可以直接使用内置对象 out 调用 out.println() 进行输出，而不用声明  
 D Servlet 中可以跳转到另一个 Servlet

## 二、简答题

- 1.Vector 与 ArrayList 有什么区别，在 Java.util 包中还有哪两个类有类似区别？
- 2.response.sendRedirect() 与 <jsp:forward> 都能跳转，它们有什么区别？
- 3.<jsp:include> 与 <%@ include %> 都能包含有什么区别？
4. 存在学生表，课程表，分数表三个表：  
 Create table student(studentid number(10)primary key,name  
 varchar2(30),depart varchar2(20);  
 create table lesson(lesson number(10)primary key,lessonname  
 varchar2(20));  
 create table score(studentid number(10)foreign key references  
 student(studentid),lessonid number(10) foreign key references  
 lesson(lessonid),score number(3));  
 写一个 SQL 语句求出考试科目名称为 “ Java 语言 ” 的科目中分数为前 10 名的学生的 id , 姓名 , 专业。
- 5.struts 框架应用了那些设计模式？
6. 请阐述 spring IOC 思想。
- 7 阐述 hibernate 的延迟加载机制。
8. 设置本用户每小时第 15 , 30 , 59 分钟都将目录 ~aint/info/log 及目录下所有文件大小记录到日志文件 /tmp/chechdir.log 中。
9. 查看所有 oracle 用户的进程有那些。
10. 列出目录 /usr/lib 下所有文件，包括子目录及子目录下文件。
11. 查找目录 /var/adm 下是否有文件 messages , 如果有的话同时列出文件大小 , 日期等

属性。

### 三、编程题

1. 一个字符串中可能包含 a-z 中的多个字符，字符也可能重复，例如：  
String data= " aabcexmkduyruieiopxzkkkasdfjxjdsds " ;  
写一个程序，对于给定一个这样的字符串求出字符串出现次数最多的那个字母以及出现的次数（若次数最多的字母有多个，则全部求出）。
2. 输入一个正整数，若该数能用几个连续正整数之和表示，则输出所有可能的正整数序列。
- 3 有两个字符串 str1 和 str2 ，写一个函数实现在 str1 中查找 str2 的初始位置。要求不区分大小写。

### 四，编程题

1. 公司工资查询系统正待开发，现有状况是：使用 OurSQL 数据库（假想），已建立了数据库 employees\_system, 其中的一个表 salary\_table 的结构如下：

| SSN char(10) | Em_name char(10) | Age int(4) | Em_salary float(8) |
|--------------|------------------|------------|--------------------|
| 1            | Wangli           | 32         | 3500.80            |
| .....        | .....            | .....      | .....              |

数据库的驱动也已经装入，它的 jdbc 驱动程序类为 gov.edu.oo.oursql.Driver, 创建连接的字符串为 " jdbc:oursql:// " +serverName+ " / " +databaseName, 数据库服务器 OurSQL 驻留在本地。

要求用 Java 语言编写一个应用程序，能够实现如下功能：

- （1）使用 JDBC API 建立到 OurSQL 数据库的连接。
- （2）执行查询操作，打印出工资大于 2000 元的员工的姓名和工资，并按工资多少排序。
- （3）给年龄大于 20 的员工加薪，每人加 200 元。

### 五，智力题

1. 有一位警长，抓了三个逃犯。现警长决定给它们一次机会。他拿出 3 顶黑帽子，两顶白帽子，然后往这三个逃犯头上每人戴一顶帽子，每个逃犯只能看到另外两个逃犯的颜色，不能看到自己帽子的颜色，而且不能进行通讯，不能进行讨论，只能靠自己的推理推出来，如果猜出来，放一条生路，否则处死。

警长先问第一逃犯，结果第一逃犯猜错了，被杀掉了。

警长问第二个逃犯，结果还是猜错了，同样被杀掉了。

警长再问第三个逃犯，结果第三个逃犯猜对了。

说明一下，每个逃犯再回答问题时，其他逃犯是听不到的。

为什么第三个一定能猜中，请你给出答案。

北京恩世教育投资有限责任公司 地址：朝阳区夏家园半岛国际公寓底商二层 电话：  
84411199 网址：<http://www.cnnuage.com>  
路线：地铁十号线太阳宫站 C出口斜对面

方标电信笔试题

1. Java有几种内存？每一种内存有什么功能？哪一种内存速度快？
2. 对 java5.0 范型的新特性
3. 怎样手动的调用垃圾回收器
4. 线程有几种方式
5. Java INO 的工作过程
6. Java开源的软件，使用过什么版本。。在你的项目中你用到了哪些。 你的项目能达到什么级别（大概意思）

大概意思

1. spring 有几部分组成
2. 依赖注入和范型的区别
3. hibernate中对几个主流的数据库（ oracle,mysql,sqlserver) 的分页的实现

三个任选一题

1. 将一个二位的字符转换为十六进制数
2. 用线程实现一个计数器
3. 实现一个服务器。。。。。。。。。。记不好了

- 1, Java体系中有哪几种内存结构，分别用来存取什么？谁的速度快？
- 2, Java的垃圾回收机制，手动回收的方法有哪几种？
- 3, Java5.0 后引入了泛型，谈谈你对它的理解？
- 4, 谈谈你对 Java4.2后的 NIO 的理解。
- 5, 线程的新技术。
- 6, spring 的几大模块？
- 7, spring 中的 DI 与泛型的区别和相似之处？
- 8, Hibernate 在 oracle Mysql 数据库分页中所用到的几个关键词？
- 9, 把一个 2 位字符转换成一个 16 进制数，并打印出。
- 10, 写出一个非阻塞的计数器。
- 11, 把 windows 任务管理器中的 CPU 曲线图变成一条直线。

## 方正笔试题

- 1、java 中普遍运用了类和对象的概念，谈一下你对对象的认识
- 2、关系型数据库中的 3nf、4nf、5nf、bcnf 哪个在实际开发中更常用，说一下你的认识

(这道题我不会，都没听说过，题目大概是这个意思 )

3、解释一下瀑布模型、迭代极限模式

(这个也不会。。)

4、写一个电话号码的正则表达式，按照 区号 --座机号 --分机号的格式写

5、假如让你设计一个查询的方法 `public String [] chaxun(Object obj)`

如果查询不到的话你觉得返回一个什么值合适 A、String[0] B、null

(这道题后来项目经理给我讲了、应该选 A，说是一种设计模式的体现)

6、比较以下两种算法 谁优谁略

A、`for(Iterator i=a.iterator();i.hasNext();){dosomething(i.next())}`

B、

`while(i.hasNext())`

{

`dosomething(i.next());`

}

(这道题我也答错了 应该是选 A，因为 A 产生的 i 是临时变量，出了 for 循环就没有了 不会占用过多内存 ....我想到临时变量的问题了，但是内存的问题没有想到)

7、写代码运行结果的题

class a

{

`public void say()`

{

`System.out.print("hello ");`

}

`public void name()`

{

`System.out.print("jerry");`

}

}

class b extends a

{

`public void say()`

{

`System.out.print("byebye ");`

}



```

 public void name()
 {
 System.out.print("tom");
 }
 }

 public class mym{

 public static void main(String [] args)
 {
 a aa=new b();
 b.say();
 b.name();
 }

 }

```

写出运行后的结果

8、给你几个表 然后让你根据条件写 sql 语句，是关于银行账户的， sql 语句都不难，都是做一些表之间相互关联的查询操作，但是给出的表比较多，给了 6、7 个表，乍一看比较乱

给出的表包括：

银行表 -- ( 字段 ) 分行名称，分行地址

客户表 --( 字段 ) 客户姓名，客户住址

账户表 --( 字段 ) 账户号，金额。。

还有一些表 都是相关的 实在记不清了

题目是：1、查询所有在北京分行有账户的客户姓名

2、查询所有在北京的分行名称

3、查询所有在北京有业务的客户

9、List a=new Vector() ;

List a=new Vector() ;

哪个更好 请说明原因

10、本公司用的是 struts+hibernate 的框架，结合本公司的框架结构，说说 struts 在请求响应的全过程（可以用图表示）

11、结合自己的开发经验 谈谈你认为开发时应该编写什么样的代码

就这些了 还有一些 c 语言的题，题目代码很多，里面涵盖了指针，时间有限我也没有细看  
另外还有一些网络的题 关于协议什么的

方正简答题

- 1.不用 new 怎么创建对象？
- 2、java 中的引用和 C#中的指针最大的区别是什么？
- 3、你在什么时候用到 xml 的文件，说说你的实现方法？
- 4、简述什么是 ajax？

## 飞速康盛笔试题

1.  
写一个方法，入参为一个字符串，判断该字符串中是否有大于 3 的个数字，大于 3 个返回 True,否则 False,如 "1a2b3c4d" 返回 Ture , "1a2b3ccc" 返回 False。

2.  
还有一些理论题

1》PreparedStatement和 Statement的区别

3.  
题库里的题，基础的没多少，好像有后面的框架里提到的 EJB 方面的题，可以多留意一下。

4.  
写一个方法： 将作为入参传入的字符串进行分析， 验证其中是否有非法字符 （非法字符串固定的，中间用，号分割比如 "!,@,#,\$" ）

5.  
选择题： 全是关于数据库的 给你一些 SQL 语句 从其中选择出 SQL 语句 执行效率最高的 以及 最浪费资源的 和 最慢的

6.简答题：

- 1) 一个类 被 finaly 修饰 代表这个类怎么样了？
- 2 ) 谈谈 struts 框架的优缺点？
- 3) 页面间如何传递数据？
- 4) 优化代码： for(i=0;i<v.size();i++){

其中 v 代表 一个 vector

7.编程题：

- 1) 写一个 javabean 出来
- 2) 设计一个方法 入参是 2 个字符串 第一个代表 用户名 第 2 个代表 非法字符

如果用户名中包含 非法字符 就返回 false 否则返回 true

## 工商银行 2008 年笔试试题

一、50 道公务员类型题单选（ 50 分）

（总体感觉比公务员题简单）

几道实事题、 几道工行背景题（考你：工行电话银行号码 95588，工行引进三家国际战略投资高盛集团、安联集团、美国运通公司）

几道法律题（很简单，不用复习）

几道经济题（都特基本，不会的知识点可以用排除法搞定）

二、两篇英文阅读 10 个单选（ 10 分）

难度相当于四级考试前两篇阅读难度（比较简单）

（监考很松，见到不少同学在考场内偷用文曲星查单词）

三、案例分析（ 20 分）

英国巴林银行倒闭案

两个问题：巴林银行内控管理哪里有漏洞？给予我们什么启示？（大概意思吧）

四、命题作文（ 20 分）

《如何打造有核心竞争力的银行》 800 字以上

## 光声世纪信息科技有限公司笔试题

值传递和引用传递的区别

解释方法的重载

列举常见的集合类并加以说明

用 new 关键字创建对象，然后构造方法抛出异常，请问

该类对象是否创建，为什么？

反射机制有什么作用？

JavaScript对表格进行排序

编程

判断某一天是否为周末，以 2008 为例子

格式为 20080212

显示输出结果

Servlet 的生命周期

ArrayList ,LinkedList 的区别

怎么在 javaScrip 模拟一个 java 中的类

Try()

{

Return;

}

Catch()

Finally{

}

上面 return 会执行吗？

如果执行是在 finally 之前还是 之后

用数组实现一个栈，当溢出时栈的大小自动加倍

说出常用 xml 文件解析常用的工具

hashMap 和 hashTable 的区别

人民币大小写的转换（耿老师讲过）

抽象类的用法（是耿老师讲的例子圆面积和矩形面积）

ArcGis 实现地图的放大和缩小

【GIS 的技术题】

北京光声世纪信息科技发展有限公司 地址：北京市海淀区复兴路甲

38 号嘉德公寓 1301-1306 招聘苗小姐电话： 88212485-1801 网

址：[www.topgis.com.cn](http://www.topgis.com.cn)

16 日上午 9 点半

### 北京广聚福企业商务服务有限公司笔试题

1. ArrayList, Vector, LinkedList 的区别
2. java 中的 GC 指的是什么，有什么作用
3. 抽象类与接口有什么区别
4. 两个对象值相同，但却可以有不同的 hashCode, 这句话对不对？
5. String string = new String("xyz") 创建了几个 StringObject？
6. 写一个单例模式出来
7. Hibernate 中 Session 是如何工作的
8. ajax 的应用在那些方面？

### 广通信达笔试题

1. 请你分别画出 OSI 的七层网络结构图和 TCP/IP 的四层结构图。
2. 请问什么是 IP 地址？按照网络规模的大小，一般把网络地址分成几类？子网掩码的作用是什么？
3. 请你简述 VPN 和使用的协议。
4. 请问什么是 SNMP？
5. 请用 SQL 语句创建一个数据库用户并加以说明，数据库可以为 ORACLE/MYSQL/SQLSERVER。
6. 有如下三张表  
表名：student ( 学生表 )  
编号 sno varchar  
姓名 name varchar  
  
表名：course ( 课程表 )  
编号 cno varchar  
名称 name varchar

表名：course-selection(选课表)

课程编号 cno varchar

学生编号 sno varchar

1) 针对 student 表，写出典型的插入，更新，删除 SQL 语句。

2) 请用 SQL 语句查出没有选择课程名为“大学物理”的所有学生信息。

7.请说明在 linux 中如何打开和关闭 Telnet 服务。

8.你是 root 系统管理员一个 Unix 程序的属性如下：

-rwxr-xr-x 1 svradm users 43505 Jul 26 17:43 socketserver 这个 socket 服务器需要绑定 810 tcp 端口，如果要让 svradm 用户运行这个程序，如何进行设置？

9.一个 J2EE 标准的 WEB 应用中，在应用发布的根目录下一定要存在的目录名是 \_\_\_\_\_ 当前应用所用到的类文件必须放到 \_\_\_\_\_ 目录下，所用到的类库文件（一般为 .jar 文件）必须放到 \_\_\_\_\_ 目录下。

10.有如下的类定义：

```
public class Parent{
 int change(){.....}
}
class Child extends Parent{

}
```

以下语句那些可以填在 处？

```
public int change(){.....}
int change (int i){.....}
private int change(){.....}
abstract ubt change(){.....}
String change(){.....}
String change(int i){.....}
```

11.var a=10;var b=20;var c=10;alert(a=b);alert(a==b);alert(a==c) 结果是什么？

12.有如下的 HTML 定义：

```
<form name="a">
 <select name="a" size="1" onchange="_sel(this)">
 <option value="a">1</option>
 <option value="b">2</option>
 <option value="c">3</option>
 </select>
</form>
```

请问，如何通过 Javascript 脚本获取表单 <select> 域的选择部分的文本并输出？

13.现在有一个用户软件系统需要你来升级，在升级操作中你需要注意那些问题？如果在升级过程中发生的错误，你该如何处理？

15.请说明在一个软件开发项目中，如果合同里开发要求界定不够清晰，导致客户不断在项目进行过程中提出需求变更，你的前任项目经理对比一直妥协，并作了大量变更开发，你接手该项目的时候，用户的继续这样要求你，项目验收项目已经滞后，你作为项目负责人将怎么面对？同时提出你的建议？

## 国富商通面试题

尹传辉总结：

- 1.J2EE是什么？有那些标准？ J2EE 分层。
- 2.JAVA 服务器的核心是什么？
- 3.写一段 JDBC 连接数据库的代码， CONNECTION 最后要关闭。
- 4.数据库表之间有那些关系？写出两表之间的等值连接代码？
- 5.只有一个 JSP页面，要求查询数据库的表，并且把结果显示在 JSP页面上，要用到 STRUTS 和 HIBERNATE 。
- 6.写一个 STRUTS 和 HIBERNATE 都需要生成什么文件，都起什么作用？
- 7.如何把数据库中的表生成一个 XML 文件？
- 8.说一下 STRUTS 的工作原理。
- 9.HIBERNATE 需要多复习一下，面试中问的比较多
- 10.SVN 版本控制问题。

尹俊总结：

- 1 介绍一下 j2ee 技术
- 2 struts 是如何实现 web层开发的
- 3 ajax 的主要作用，及其特点
- 4 struts 是如何处理 \*.do 的
- 5 hibernate 是如何实现对持久层管理的

其它的都是由这几个主要问题引出来的，不一定问到什么。

王光明总结：

没问其他知识，全部都是三大框架以及展开的知识  
给一个用户查询功能，将查询结果在同一个页面现实并将查询结果排序。  
要求用到三大框架 SSH，口述三大框架在实现这个功能时候所取的作用， 工作流程以及相关配置信息，如果实现这个功能并口述主要的代码。  
主要知识点有 SQL 查询语句， HQL，框架的配置信息，如何相互配合使用。问题比较深刻。

吴琼总结：

首先是了解一下个人信息，然后问工作经验和项目经验。（顺便提一下，项目经验很重要，几乎所有的面试都会问项目经验。）

接着问了一道大题：说说使用 ssh 框架的好处。在我回答完了之后他接着追问 Hibernate 中的实体类的配置文件以及以部门和员工为例怎么配置一对多的映射。然后又问了一下在项目中怎么样调用 Hibernate。（答案就是先拿 sessionFactory，再拿 session 后面的。。。。。）

最后问了一下 HQL 语句。

## 国富通信息技术有限公司

地址：北京市经济技术开发区荣华中路 11 号 中国国际电子商务大厦 5 层 电话：67800300

网址：<http://www.cofortune.com.cn/>

路线：坐地铁 5 号线到刘家窑下，坐公交 652 到万源街西口（亦庄管委会）下车即是

1. 简答 jsp 和 servlet 的联系和区别

2. web service 名词解释：

3. SOAP,UDDI,WSDL 解释

4. EJB 规范规定 EJB 中禁用的操作有哪些？（答出三道四点即可的满分）

答案：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。

5. Oracle 分页查询语句，假设表名为 table，查出 21 到 40 页的数据

6. 在 weblogic 中如何制定内存的大小，如何进行 ssl 的配置与客户端的认证配置？

7. 说明 servlet 的生命周期

8. java 中 RMI 和 CORBA 的区别

答案： 1、定义接口：

rmi 自己定义接口（ interface ）

corba 生成 idl 代码，然后使用 idlj -fall name.idl 生成接口和几个类文件

2、启动服务：

rmi 启动的是 rmiregistry (port) 默认 1099

corba 启动的是 tnameserv

3、实现的继承类：

rmi extends UnicastRemoteObject



corba extends 运行 idlj 时 生成的 \_NameImplBase

#### 4、实现类的 rebind

rmi 可以直接 rebind 如：Naming.rebind("rmi://localhost/meeting",meetingserver);

corba 需要先调用 init ( ) 如：

```
ORB orb=ORB.init(avgs,null);
```

```
MOTDImpl impl=new MOTDImpl(motdFile);
```

```
orb.connect(impl);
```

```
org.omg.CORBA.Object objRef=
```

```
orb.resolve_initial_references("NameService");
```

```
NamingContext ncRef=NamingContextHelper.narrow(objRef);
```

```
NameComponent nc=new NameComponent(motdService,"");
```

```
NameComponent[] path=new NameComponent[]{nc};
```

```
ncRef.rebind(path,impl);
```

这里 corba 就要麻烦很多了

#### 5、客户端调用 corba 的调用跟实现类的绑定差不多如：

```
ORB orb=ORB.init(avgs,null);
```

```
org.omg.CORBA.Object objRef=
```

```
orb.resolve_initial_references("NameService");
```

```
NamingContext ncRef=NamingContextHelper.narrow(objRef);
```

```
NameComponent nc=new NameComponent("MessageOfTheDay","");
```

```
NameComponent path[]=new NameComponent[]{nc};
```

```
org.omg.CORBA.Object motdObj=ncRef.resolve(path);
```

```
MOTD motdRef=MOTDHelper.narrow(motdObj);
```

```
System.out.println(motdRef.getMOTD());
```

rmi 只要 lookup 就可以了

总的来说其实 rmi 和 corba 都差不多，都是桩和框架，两者相互竞争，但是在 java 中都可以相互调用。这归功于 rmi\_iiop.

## 海辉集团 Sql 数据库笔试题

1. 触发器的作用
2. 什么是存储过程？用什么来调用？
3. 索引的作用？他的优缺点是什么？
4. 什么是内存泄露？
5. 维护数据库的完整性和一致性，你喜欢用存储过程还是喜欢自写业务逻辑你？为什么？
6. 什么是事务？什么是锁？
7. 什么叫视图？游标是什么？
8. 为管理业务信息建立 3 个表：  
S(S#,SN,SD,SA)S#,SN,SD,SA 分别代表学号，学员姓名，所属单位，学院年龄

C(C#,CN) C#,CN 分别代表课程编号，课程名称

SC(S#,G#,G) S#,G#,G分别代表学号，所选课程编号，成绩

- (1) 使用标准 sql 嵌套语句查询所选课程名称为“ 税收基础 ”的学员姓名和学员学号
- (2) 使用标准 sql 嵌套语句查询选修课程编号为“ C2 ”的学员姓名和所属单位
- (3) 使用标准 sql 嵌套语句查询选修不课程编号为“ C5 ”的学员姓名和所属单位
- (4) 查询选修了课程的学员人数
- (5) 查询选修课程超过 5 们的学员学号和所属单位

## 中国航空综合技术研究所笔试题目

1. 简单介绍一下 XMLHTTP,WEBSERVICE的特点和作用。
2. 接口和抽象类的区别
3. 简述面向对象的概念以及他的特征
4. 介绍页面传值的几种方法及其优缺点
5. 软件开发及其作用
6. 简述一下反射
7. 介绍一下对 sqlserver , oracle 和 mysql 的熟练程度。
8. 简单介绍一下设计模式
9. 都有什么持久层技术，你最擅长的是那一种？
10. 什么是依赖注入、反转控制和 AOP;
11. struts1.2 和 struts2.0 的区别
12. 简单介绍一下你使用 tomcat 。 Websphere 和 weblogic 的心得
13. 简述一下工作流开发的使用经验
14. 简单介绍一下 J2EE 体系架构下的软件快速开发平台的认识
15. 描述一下你所用到的 SOA
16. 描述一下已在开发和实施项目过程中所掌握的业务。
17. 简述一下 eclipse 插件开发心得 。

## 好帮手在线笔试题

一． 5 道选择题 读程序写运行结果（ 863 题库类似）

例如 对 try catch 语句的掌握

finally 语句

二． 5 道程序改错题

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。

### 三．6 道填空题

1. 面向对象的特征：
2. MVC 的三个核心核心部件：
3. XML 文档的解析方法：
4. 写出在实际开发中经常运用到哪些设计模式，写出三个：
5. 一道关于 UML 的图形举例
6. JSP 的 9 个内置对象，写出其中的四个

### 四．5 个问答题

1. 作用域 public private protected，及其不写时候的区别？
2. JSP 中静态 include 和动态 include 的区别？
3. 类变量和局部变量的区别？
4. 一道 JavaScript 程序题，要求能够弹出对话框提示当前多选的是那个对话框
5. 应用服务器有哪些？

六 . 2 道程序题

1.SQL 语句，两个表一个用户表，一个用户所对应的订单表  
两个表以申请单号相互关联

写出 4 个 SQL 语句  
问题是对两个表的同时操作，增删改查的操作

2.用循环语句打印

```
* *
** *
*** *
**** *
***** *
***** *

```

北京华路时代信息技术有限公司

—— JAVA 测试题目 ——

Section A:JavaScript HTML 试题：

1，使用 javascript 脚本函数 init 初始化表单（ formCustomize ）中的一个单选下拉列表（ language ）， language 是下拉菜单的内容，以及设置缺省被选择项目（值）。

```
//javascript 代码
var lanages = {};
languages[cn] = {value : Chinese }; text :
languages[en] = {value : English };, text :
languages[fr] = {value : French };, text :
/**
Select 是下拉菜单的文档对象
defaultValue 是缺省被选择值
*/
function setup(select,defaultValue){
 //清空下拉菜单内容
 for(var i = select .options .lengths -1 ; i>=0 ; i--)
 //删除下拉菜单内容

 for(var i in languages)
 //获取下拉菜单项目内容数据

 // 建立下拉菜单项文档对象

 // 将下拉菜单项文档对象加入下拉菜单里面

}
for(var i = 0 ;i<select.options.length){
```

```

 //设置下拉菜单缺省值
 if(defaultValue == select.options[i].value){

 break;
 }
 }
}
//初始化设置下拉菜单 language
function init(){
 //获取下拉菜单 language 文档对象
 var select = _____
 //下拉菜单的缺省值
 var defaultValue = Chinese ;
 setup(select,defaultValue);
}
//HTML 代码
<body onload= init() >
<form name= formCustomize >
A.

B.

C.

D.

E.
 var option = document.createElement(OPTION);
 option.value=value;
 option.text =text;
F.
 var text =languages[i].text;
 var value=languages[i].value;
G.
 var text =i.text;
 var value= i.value;
H.

I.

J.

K.

L.
```

document.getElementById( language );

M.

document.forms[ formCustomize ].elements[ language ];

2.能够判断 data 的文本内容是否是有效数值的脚本语句是下面那一个？

var data = ;...

- A. Math.isNaN(data);
- B. isNaN(data);
- C. data !=NaN;
- D. Math.isNumber(data);
- E. IsNumber(data).

答案 : \_\_\_\_\_

3.下面那一个是 HTML 代码，能够完成文件的上传 ？

- A. <form captype= upload/filedata />
- B. <form enctype= upload/file />
- C. <form captype= multipart/file />
- D. <form enctype = multipart/file />
- E. <form enctype = multipart/file />
- F. <form captype= multipart/file />

答案 : \_\_\_\_\_

Section B:Java 程序设计

4.在空白处填写相应的代码完成数据类型的转换。

String codeText = 2.735

Double codeNum=\_\_\_\_\_

- A.(double)codeText;
- B.Double.valueOf(codeText);
- C.Double.parseDouble(codeText);
- D.new Double(codeText).doubleValue();

5.将下面的字符串从 ShiftJIS 编码转换成 UTF8 的编码。

String txt = 阿斯蒂，丰富的 ;

String utf=\_\_\_\_\_

- A. new String(txt, -JShift , UTF);
- B. new String(txt, -8UTF);
- C. new String(txt.getBytes( -JIS Shift , UTF);
- D. txt.setEncoding( -8UTF);

6.判断下面那些行的语句是正确的 ？

Float a= ... ;

Double b= ... ;

Number c=null;

- 1. c=b;
- 2. b=(Double)a;
- 3. a=(Float)b;

正确的是 : \_\_\_\_\_

错误的是 : \_\_\_\_\_

Section C:理论 (可选)



7,写出所了解的若干设计模式

A. 创建模式 :\_\_\_\_\_

B. 结构模式 :\_\_\_\_\_

C. 行为模式 :\_\_\_\_\_

8.J2EE 由那些服务，应用程序接口和协议组成？

9.事务隔离级别由那些？

10.JTA 事务比 JDBC 事务由那些优势？

11.使用 JDBC API 进行事务应用实现，请在空白处填写相应的代码。

//获取数据源

DataSource ds= ... ;

Connection connection=ds.getConnection();

//设定事务提交模式为手动提交

\_\_\_\_\_

//保存数据

stmt = connection.prepareStatement( UPDATE MOVIES.. );

stmt.setString(1, Superman Returns );

stmt.executeUpdate();

//事务提交

\_\_\_\_\_

// ...

A. stmt.setAutoCommit(false);

B. stmt.commit();

C. connection.setAutoCommit(false);

D. connection.commit();

E. Transaction tx=connection.getTransaction();

F. tx.commit();

答案 : \_\_\_\_\_

12.使用 JTA 的事务应用实现

// ...

InitialContext ctx= new InitialContext();

//获取 UserTransaction

UserTransaction tx= \_\_\_\_\_

//开始事务

\_\_\_\_\_

//获取数据源

DataSource ds= ... ;

Connection connection ds.getConnection();

//保存数据

stmt = connection.prepareStatement( UPDATE MOVIES.. );

stmt.setString(1, Transformer );

stmt.executeUpdate();

//事务提交

\_\_\_\_\_

// ...

- A. connection.getTransaction();
- B. new UserTransaction();
- C. ctx.lookup(        java:comp/UserTransaction        );
- D. connection.setAutoCommit(false);
- E. connection.commit();
- F. stmt.setAutoCommit(false);
- G. stmt.commit();
- H. tx.begin();
- I. tx.commit();

答案 : \_\_\_\_\_

13. 获得来自浏览器提交过来的数据，填写空白处的代码。

HttpServletRequest req=        ... .;

//获取从浏览器提交过来参数名字为 “        name ” 的数据

String name=\_\_\_\_\_

String[] names=\_\_\_\_\_

//获取从浏览器提交过来多个参数名称为 “        multi\_name ” 的数据

- A. req.getParameter(        name    );
- B. req.getAttribute(        name    );
- C. (String[])req.getAttribute(        multi\_name    );
- D. req.getProperty(        name    );
- E. req.getParameterValues(        multi\_name    );
- F. (String[])req.getProperty(        ulti\_name    );

答案 : \_\_\_\_\_

14.实现下面那一个接口能够得到        HttpSession 的生命周期变化的消息？

- A.HttpSessionBindingEvent;
- B.HttpSessionEventListener;
- C.HttpSessionBindingListener;
- D.HttpSessionListener;
- E.EventListener;

答案 : \_\_\_\_\_

#### Section D:Struts 程序设计

15.在 JSP 中使用 Struts、JSTL 等标签编写（初始化）一个下拉菜单，所需要的数据放在 request 中的下拉菜单的内容存名称为 “        countries ”，放在 request 中的被选择的值名称为 “ country ”。

答案 : \_\_\_\_\_

注：用 LableValue 存放下拉菜单项的数据，并存放在 “        countries ” 集合中， LableValue 主要方法如下。

```
public class LabelValue{
 private String label;
 private String value;
 public LabelValue(String label,String value){
 this.label=label;
```

```

 this.label=value;
 }
 public String getLabel(){return label;}
 public String getVlaue(){return value;}
};

public class ExamSelectAction extends org.apache.struts.action.Action{
 public ActionForward perform(
 ActionMapping mapping,
 ActionForm actionform,
 HttpServletRequest request,
 HttpServletResponse response){
 List countries =new Vector();
 countries.add(new LabelValue(C中国A));
 countries.add(new LabelValue(FR法国E));

 request.setAttribute(countries ,countries);
 request.setAttribute(country , CHINA);

 };
}

```

16.struts 框架中，关于 formbean，下列说法正确的是：

- A．Formbean 是一种数据 bean，主要用来封装表单提交上来的数据，并把这些数据传递给 Action
- B．在 Formbean 中可以对页面上传来的参数进行一下格式上的验证，这种验证是一种客户端的验证
- C.用户每次提交表单，都产生一个新的 Formbean 实例
- D.动态 Fombean 不能进行验证操作

答案：\_\_\_\_\_

17.Struts 框架中，在一个 Action 得配置信息中，name 属性指的是

- A．当前 action 实例的名字
- B．当前 action 所在类的名字
- C．该 action 中调用的 Frombean 的实例名字
- D．该 action 中调用的 Frombean 的类所在包名字

答案：\_\_\_\_\_

18.下列那些说法是正确的？

- A．struts 中无法完成上传功能
- B．struts 框架基于 mvc 模式
- C．struts 框架容易引起流程复杂，结构不清晰等问题
- D．struts 可以有效的降低项目的类文件数目

答案：\_\_\_\_\_

19.根据下面内容，要向 sunway 服务器发出的请求，请选择空白处的内容，能够完成 AccessoriesAction 中得 download 方法调用实现。

//向 sunway 服务器发出的请求

http://sunway/cms/accessories? \_\_\_\_\_&filename=....

Struts-config-cms.xml

```
<action path= /accessories
 Type= orh.apache.struts.actions.AccessoriesAction
 Name= AccessoriesForm
 Scope= request
 Input= /cms/accessories.jsp

/>
```

AcessoriesAction.java

Import org.apache.struts.action.\*

Public class AccessoriesAction extends \_\_\_\_{

Public ActionForward upload(ActionMapping mapping,ActionForm form,HttpServletRequest request,HttpServletResponse response)throws Exception{

.....

}

Public ActionForward download(ActionMapping mapping,ActionForm form,HttpServletRequest request,HttpServletResponse response)throws Exception{

.....

}

}

A . method=download;

B . command=download;

C . parameter= method ;

D . parameter= command ;

E.Action;

F.DispatchAion;

G.LookupingAction;

答案： \_\_\_\_\_

20.对于以下代码。 LoginAction , 要求把请求转发给 login.jsp , 在 LoginAction 的 execute ( ) 方法中可以使用那些代码能够实现 ( 多选 ) ?

LoginAction.java

Public loginAction extends org.apache.struts.action.Action{

Public ActionForward execute(ActionMapping mapping,ActionForm form,HttpServletRequest request,HttpServletResponse response)throws Exception{

.....

}

};

Struts-config.xml

```
<action path= /login type= com.vgc.loginAction name= loginform scope= request
```

```

validate= true input= /login.jsp >
 <forward name= login path= /login.jsp />
 < forward name= sucess path= /index.jsp />
</action>

```

- A . return(new ActionForward(mapping.getInput()));
- B . return(mapping.findForward( login ));
- C . return(mapping.findForward( index ));
- D . return(mapping.findForward( login.jsp ));
- E. return(mapping.findForward( index.jsp ));

答案： \_\_\_\_\_

21.spring 框架得有哪些模块（或组件）组成？

22.在下面的 spring 配置文件（路径为“fse.xml”）填写出相应得配置内容，其中 defaultRemindDAOImpl 的 getEmployees 方法只读，而方法 markRemind 则不是。

注：这里没用使用到 hibernate

```

<bean id= datasource class= org.springframework.jndi.jndiObjectFactoryBean >
... .
</bean>

```

```

<bean id= transactionManager
 Class= org.springframework.jdbc.datasource.DataSourceTransactionManager >

</bean>
.....

```

```

<bean id= remindBean
 Class= org.springframework..transaction.interceptor.TransactionProxyFactoryBean >

 <property name= transactionAttributes >
 <pros>
 <pro key= getemployees _____</pro>
 <pro key= markRemind _____</pro>
 </pros>
 </property>
</bean>

```

- A . <property name= target ref= remindDaoImpl />
- B . <property name= transactionManager ref= transactionManager />
- C . <property name= dataSource ref= dataSource />
- D . PROPAGATION\_REQUIRED
- E.PROPAGATION\_REQUIRED,readonly
- F.PROPAGATION\_REQUIRED,writeOnly

答案： \_\_\_\_\_

23.根据上面的 spring 配置文件内容，请写出相应的调用 “ remindBean ”的方法“ getEmployees ”。

24.使用 hibernate 完成数据持久化保存到数据库中的方法有哪些？

- A . org.hibernate.Session.save( ... );
- B . org.hibernate.Session.load( ... )
- C . org.hibernate.Session.store( ... )
- D . org.hibernate.Session.update( ... )
- E. org.hibernate.Session.saveOrUpdate( ... )
- F. org.hibernate.Session.refresh( ... )
- G. org.hibernate.Session.flush( ... )

答案： \_\_\_\_\_

25.使用 Hibernate 实现查询的分页，可以用下面哪个类和其相应的方法？

- A . Criteria
- B . Query
- C . scroll
- D . ScrollableResults
- E.setRowNumber
- F.setFirstResult
- G.setMaxResults
- H.setFetchSize

答案： \_\_\_\_\_

26.下面哪句话对 Session beans是真的，而对于 EntityBeans 却不是？

- A . Bean 被用做模型在 MVC 设计模式中使用
- B . 能够持久化保存状态到数据库中的表一行上
- C . Bean 能够使用数据访问对象容易的迁移到 CMP
- D . 能够保持一个会话状态为一个具体客户

答案： \_\_\_\_\_

27.关于无状态会话 Bean 错误的是：

- A . 不保存客户的状态信息
- B . 包括主接口，远程接口， bean 类和部署描述文件
- C . 部署时， EJB 容器会为这个 Bean 分配几个实例到组件池中，并有组件池动态管理组件数量
- D . 生命状态有四种，存在，不存在，钝化，激活

答案： \_\_\_\_\_

28. 关于状态会话 Bean 错误的是：

- A . 在客户端程序引用期间维护 Bean 中所有实例数据的状态值
- B . 多个客户不会同时共享同一个有状态 SessionBean实例
- C . 钝化过程就是容器调用 EJBremove 删除多余 EJB 实例的过程
- D . 有状态会话 Bean 得生命状态包括不存在，准备好和钝化

答案： \_\_\_\_\_

答案： \_\_\_\_\_

29、填写调用一个 SessionBean ( JNDI 为 GhibliiStudio ) 的方法 “ getMovies ” 的步骤（写出步骤的说明或者代码都可以）。

Context ctx=new InitialContext();

---

---

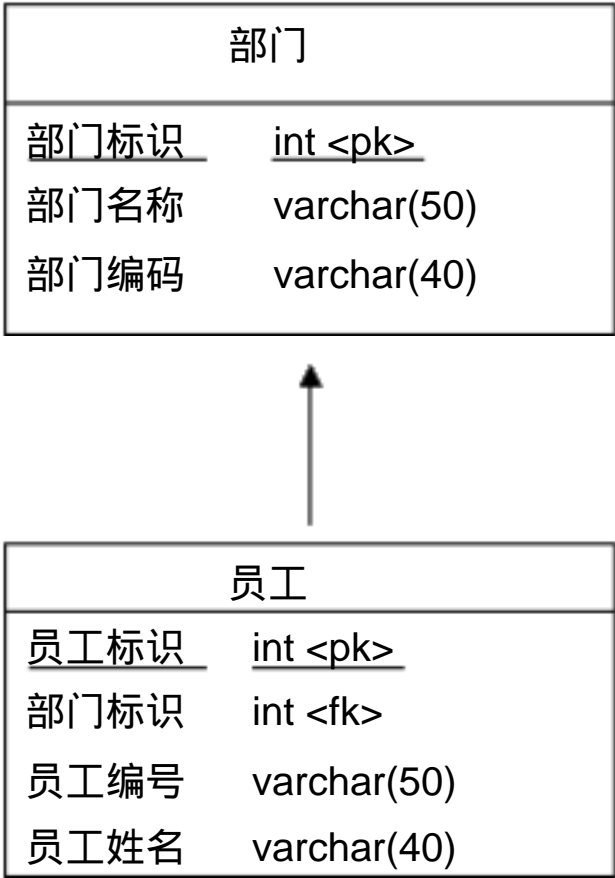
30, Web Services 标准和规范有那些？请从下面的答案中选择。

- A, ARP、SNMP、TCP/IP、UDP;
- B, COBRA、RMI、UDDI 、WS;
- C, FTP、HTTP、SNMP、SOAP;
- D, SOAP、WDSL 、UDDI 、WS-Security 。

答案：\_\_\_\_\_

Session G:数据库程序设计

这里有三张表，请根据相应的要求，编写 SQL 语句。下面是部门表 department 和员工表 staff 物理模型。



表： department

Name	Code	Data Type	Primary	Is Null
部门标识	department_id	int	TRUE	TRUE
部门名称	department_name	Varchar2(50)	FALSE	TRUE

表： staff

Name	Code	Data_Type	Primary	Foreign Table	Foreign Column
员工标识	staff_id	int	TRUE		
部门标识	department_id	int	FALSE	department	department_id
13/14					
员工编号	staff_no	Varchar(60)	FALSE		
员工姓名	staff_name	varchar ( 60 )	FALSE		

表： staff\_backup

Name	Code	Data_Type	Primary	Foreign Table	Foreign Column
员工标识	staff_id	int	TRUE		
部门标识	department_id	int	FALSE	department	department_id
员工编号	staff_no	Varchar(60)	FALSE		
员工姓名	staff_name	Varchar(60)	FALSE		
备份时间	Back_date	Datetime	FALSE		

- a) 在表 department 和 staff 对应的 Hibernate 的配置文件中，只写出两张表的一对多和多对一关系的配置就可以；
- 注：表 department 对应类为 Department，staff 对应类为 Staff。
- b) 写一 SQL 语句统计出每个部门的员工人数，统计结果数据要有部门名称和人数；
- c) 将在 staff 表中存在并且也在 staff\_backup 表中存在的员工记录从 staff 表中删除。
- d) 用一条插入语句将 staff 表中部门名称为“dept\_1”的员工记录全部插入到 staff\_backup 表中，同时将“备份时间”设置为数据库当前时间；
- e) 用一条查询语句检索出 staff 表中部门名称为 dept\_1 的前 10 条记录；

Session H 口述题：

- 1，Hibernate 的 session 有两个这样的方法“update”和“save”，他们区别在那里？
- 2，在前面的数据库试题中有两个表“员工”“部门”，当取到一个员工信息后，关闭 Hibernate 的 Session 然后从员工信息中取部门信息。

问题是：

如何保证在关闭 hibernat 的 session 之后，仍然能保证从员工信息中取部门信息？

## 华瑞实创信息技术有限公司笔试题

1

```
Public Class Test (){
 {
 System.out.println(b);
 }
 static{
 System.out.println(a);
 }
 Test(){
 System.out.println(c);
 }

 public String getOut(){
 try{
```



```

 return 1 ;
 }catch(Exception e){
 Return 2
 }finally{
 Return 3 ;
 }
}

public static void main(String[] args){
 System.out.println(getOut());
}
}

```

问题是 1 怎么修改程序让他正确，用尽可能多的方法，并写出执行结果

2 只修改 main 函数，打出 abc。能打出 bac 呢？为什么

如果只修改 main 函数不能打出 abc，也不打出 bac，因为程序无法编译过去，也就无法运。

2

写一个单例的例子

3

String str= WeeciolGmmcc

分别用 for, switch3, while 种循环写出程序 ,使程序能打印出 WelcomeGmcic ,一个打印语句只能打印出一个字符

4

有员工表 ,字段 id ,name,managerid ,还有个啥字段，记不清楚了，没用到，然后查姓名，领导的姓名，如果没领导的话则显示无

Id	name	managerid
1	张三	2
2	李四	4
3	王五	4
4	赵六	

5

写一个你最熟悉的 JAVA 技术，详细说明，不要用大量的书面语，要用自己的语言

6

一个益智题， 2 个人，一个说实话， 一个说谎话，有一个岔路口， 2 个人分别站在 2 个路口上，让你问 2 个人一样的一个问题，来判断出哪条路通往说谎国，哪个通往实话国  
这是通往实话国吗？或者这是通往说谎国吗？

这是必做题，还有一份选做，是有关 **CSS HTML** 和 **javascript** 的。没记住，但不难。

这个笔试题只给一个小时的时间，过了时间就会被淘汰。

## 汇科协同笔试题

### 1 培训的几大框架

2 三大框架整合 做过的实际开发项目

3 页面开发技术，模块技术 Struts Tiles

4 数据库编程 哪些数据库比较熟悉 重点是多层关联，多表查询

5 Hibernate 的关联配置 一对一，一对多，多对多

接触 Java 多久了，当时为什么选择培训 J2SE 面向对象，接口技术，编程基础

6 设计模式

7 MVC开发 用过 Spring 的 MVC 集成 Struts1 )

## 火山动力笔试题

### 基础题

- 1.你是否有过与其他团队成员发生冲突的经历？你是如何解决的？
- 2.你 uzi 喜欢的工作是什么？为什么？请谈谈你在选择工作时都考虑哪些因素？如何看待待遇和工作条件？
- 3.你如何看待超市和周末休息日加班？

### 第二部分

- 1.谈谈你对 java 显示发展情况的了解。你认为业界今后的发展如何？
- 2.springmvc 和 struts 有什么区别？
- 3.什么是 OOP？OOP 相对于面向过程编程有哪些优点？
- 4.请分别写 java5 个常用类，接口，包和异常
- 5.请问抽象类和接口有什么区别
- 6.请描述以下 ajax 在 web 服务器中的优点和缺点
- 7.try{} 中有一个 return 语句，那么紧跟在这个 try 后的 finally 中的 code 会不会执行，在 return 前还是后
- 8.两个对象值相同 ( x.equals( y ) ==true )，但是 hashCode不同，这句话对么？
- 9.请减速 forward 和 redirect 的区别
- 10.请减速 string 和 stringbuffer 的区别

## 极地银河（北京）信息科技有限公司笔试题

- 1、java 中的基本类型和高级类型
- 2、编程：将 a.txt 文件写入到 b.txt 文件中
- 3、异常处理机制
- 4、phone 下有 type = text  
编写一程序，如果输入的字符为空或大于 12 个字符，给予相应的提示
- 5、struts 实现了 mvc 中的什么？jsp、javabean、servlet、ejb 属于 mvc 中的什么？

这些是我昨天的面试题，都是学过的。

最后一题是写一 sql 语句

6. 最后一题是有四个表根据用户名查询到他拥有的权限和所有能查到的信息。

## 建龙钢铁集团复试总结

2008-07-29(完结 + 体检)

面试总结：

说明 :都在聊天记录中，所以考过来记录这样看起来比较有意境，呵呵。

菲姐 (291808436) 22:32:59

我是王菲菲，好像没人在了呢

菲姐 (291808436) 22:33:32

我上线我上线，没人和我说话呢

菲姐 (291808436) 22:34:21

我有可能去不上建龙

菲姐 (291808436) 22:35:25

今天体检没通过，心电图异常，以前照心电图没事，累得

菲姐 (291808436) 22:36:09

老总已经签字，还有 n 多手续要办，再说没有你们想像的那样简单

菲姐 (291808436) 22:36:41

心电图是在慈铭医院做得，跟建龙没关系

菲姐 (291808436) 22:37:17

但是也要我能通过才行啊

菲姐 (291808436) 22:37:52

原来的公司已经被我回绝了，掌上明珠

菲姐 (291808436) 22:38:36

所以，可能还要重新找工作，最近休息，心脏累出事了

菲姐 (291808436) 22:38:57

请假？我还没入职呢

菲姐 (291808436) 22:39:49

主动脉瓣膜返流，总之要注意休息，不能受刺激

菲姐 (291808436) 22:40:00

是的就在蓝点的对面

张海涛 (454000485) 22:40:41

不影响找工作把

菲姐 (291808436) 22:42:19

体检时是仪器出结果，心电图医生没法改，没办法我现让我家寄钱，又做了心脏彩超，想证明没病，其实不是心脏病，只是不健康了要注意，但是不知建龙的要求是否严格

菲姐 (291808436) 22:43:33

但是建龙和慈铭体检有业务关系，我已经说是入职了，这个工作对我很重要，千辛万苦得来的，要养家，可是没办法

菲姐 (291808436) 22:43:53

结果是寄到公司的，不过我的手

菲姐 (291808436) 22:45:13

总之,我真地尽力,所有能想的能做的能忍的都做了,如果进不了我也不后悔,大不了重新找

菲姐 (291808436) 22:45:37

好多人哦

菲姐 (291808436) 22:46:41

你们不知道我是怎么过的面试,九死一生,4轮技术面试后我死的心都有

菲姐 (291808436) 22:48:08

没事,是我的跑不了,不是我的也强求不来,我尽力了,尽全力了,无怨无悔了

菲姐 (291808436) 22:48:39

我现在在等消息

菲姐 (291808436) 22:48:59

其实我只是实习生

菲姐 (291808436) 22:50:15

使用三个月,要每天加班,周末有作业,要把某书的某章运行,熟悉,要有作业要实现

菲姐 (291808436) 22:51:28

从我过了到现在都没有开心过,因为不知道能不能过试用期,因为我的水平差的不是一点半点

菲姐 (291808436) 22:53:31

第四轮面试是一个做了8年程序员的台湾人,对我的态度就一个字:鄙视!

菲姐 (291808436) 22:54:30

果老师曾说过模拟面试可以把人问哭,我领教了,但是我都忘了怎么哭!

菲姐 (291808436) 22:55:30

第一轮人事面试,我把建龙公司的简介背下来了,得到评语:加入意念强烈

菲姐 (291808436) 23:00:38

部长面试:基本没有会的,他问三个框架解决什么问题,为什么要用.我一直忙面试,都没时间看书,只是前一天

晚上1点睡,早上5点起现看的,能知道怎么用不错了

菲姐 (291808436) 23:01:01

他说:不要告诉我怎么用,我问为什么

菲姐 (291808436) 23:01:12

我想了半天也没打上来

菲姐 (291808436) 23:02:53

又问了N多,我都尽量答,但自己都不是很满意

菲姐 (291808436) 23:04:07

当我马上觉得没戏的时候,他不问了,他问的这段时间长,但还不是最难过的.不问了,开始介绍公司和软件部门了

菲姐 (291808436) 23:04:21

我心想,可能这一关又过了

菲姐 (291808436) 23:05:22

我以为可以见最后老总了,没想到还有一轮技术面试,这才是真正的高手,而在这一轮,我被问得连自尊都没有了

菲姐 (291808436) 23:07:11

这个人问,请你说一下jdk1.5分页用的api,我说您是说有类专门用于分页?他说你不知道吗?!我说对不起我做过分页,但不是您说的,我不知道

菲姐 (291808436) 23:09:10

他问:说一下jdbc应用步骤,我答:connection,statement,然后处理resultset,这是唯一他问我答得像样的问题,马

上问:你说一下jdbc的集合api

菲姐 (291808436) 23:11:08

我答 :您是说 resultset 结果集吗 ?不是 ,还有很多 set,你不知道吗 !! 很诧异 .....

我说 :对不起我只知道 resultset,他开始皱眉 ,我的自尊心啊

菲姐 (291808436) 23:12:48

他可能觉得再问下去我也不会什么了 ,看简历 ,惊愕 ,你没有计算机经验 !! 不是计算机专业 !! 开始摇头 .项目只有一个星期 ,是项目吗 ?

菲姐 (291808436) 23:19:00

我的简历是实话实说的 ,我说这两个项目是蓝点以前接触的 ,只是练习用得 .他已经无语 .

菲姐 (291808436) 23:19:38

终于 ,他说 :好吧给我一个理由 ,只要你能说服我 .

菲姐 (291808436) 23:21:36

我说无论如何一定要做这一行 ,之类 ,我都忘了怎么说得了 ,总之就是要编程

菲姐 (291808436) 23:22:46

他说 :我会给你机会 ,因为你之前有一个有两年工作经验的人 ,答得结果居然和你一样 (菜),所以我给你机会 .

菲姐 (291808436) 23:23:22

我说 :那我前面这位没有借口了 ,而我有

菲姐 (291808436) 23:24:56

他说 ,他是台湾人 ,其实不是计算机专业也没有关系 ,他也是在台湾的一个类似蓝点的学校学出来的 (我现在还记得他说这句话时的腔调 ,和台湾电视剧一模一样 )

菲姐 (291808436) 23:26:28

但是他很努力 ,他和同学比赛 ,看谁先做出题来 ,有事是他 ,有时是另一个人 ,而最后全班只有他们两个人编程 ,其他全都放弃

菲姐 (291808436) 23:28:26

他问 :如果说周末要给你一本书 ,把某章或某几章代码全实现 ,并会有作业 ,要把作业实现 ,你愿意吗 ,每周末都加班 ,也可能每天都加班 ,我想都没想 ,愿意

菲姐 (291808436) 23:29:09

他问 :你真的能吃的了这个苦吗 ?

我说 :我能 !(当时我已经不能说不了 )

菲姐 (291808436) 23:29:41

他说 :好 ,我的面试结束 ,你再做一下 .

我想 :过

菲姐 (291808436) 23:32:23

说明一下 ,这时过了 1 点 ,因为没打算成功 ,准备完事去医院体检 ,因为掌上明珠也要体检 ,所以一点水没喝还超饿 ,我的饭量大家相信也是知道的 ,口干舌燥 ,最后一轮 ,老总面试 !最重要的是 ,他一边吃饭一边面试 ,我饿 !!

菲姐 (291808436) 23:33:24

其实老总的面试最轻松 ,胖胖一老头 ,也不是太老 ,很和蔼 ,很好玩 ,一点压力没有 ,而且是校友 (虽然这点没用 )

菲姐 (291808436) 23:34:39

问我以前的经历 ,我说做过机械公司的统计 ,然后喜欢漂亮衣服 ,就走到一个服装店说我想当服务员 ,老总说 :你很有个性 !

菲姐 (291808436) 23:35:41

老总问 ,为什么做 java,销售也很好 ,我说我是学理的 ,要有一门实实在在的技术 ,而且我觉得我能行

菲姐 (291808436) 23:37:11

老总问 ,有什么规划 ,我说三年内技术扎实一些 ,然后走管理的道路带领团队 ,我是瞎说的 ,因为国内没有一辈子的程序员 ,只能走这条路 ,可惜我错了 ,这家公司不是

菲姐 (291808436) 23:38:22

老总开始测试我是否有管理才能 ,问了很多问题 ,不是技术 ,是反应 ,因为当时又困又饿没太记住 ,总之不是太好,答得不太满意

菲姐 (291808436) 23:38:49

我是东北大学毕业 ,果老师是 ,这位老总也是

菲姐 (291808436) 23:39:23

这家公司技术部门很专业 ,可以有一辈子程序员

菲姐 (291808436) 23:40:27

老总说 :这样吧 ,我和部长再商量一下 (就是第二轮技术面试的人 ),我以为 ok,没戏了 ,但是 ,他没有让我走 ,我的戏剧性开始出现了

菲姐 (291808436) 23:41:30

有人敲门 ,提示他主持一个会议 ,他真的超忙 ,没有让我走 ,说你在这里等我一下 ,我说好 ,结果等了两个小时

菲姐 (291808436) 23:44:00

在这两个小时里 ,有人给了我一杯水 ,我看着还不能喝 ,空调冷也不能动 ,困的快睡着了 ,给我的玫琳凯督导发短信 ,因为进了建龙就没有任何休息时间了 ,给袁征发了个短信 ,说最后一关需要支持 .然后 ,最后的时刻到来了,老总开完会了 ,之后这段最好玩

菲姐 (291808436) 23:45:26

这两个小时我已经想放弃了 ,觉得自己可能是真的入错了行 ,真的不知道能不能坚持下来 ,不如就去掌上明珠算了 ,周末还可以卖卖化妆品 .我决定退缩

菲姐 (291808436) 23:48:18

老总和部长都进了屋 ,准备签字 ,老总已经签了 ,但是他停了一下 ,其实他要写一段话 ,后来我才知道 ,但是我想放弃 ,于是我说 :要不您和部长再商量商量吧 ,我想过了 ,我虽然执着 ,但是我不强求

那一刻 ,老总和部长都楞住了 ,一秒中 ,可能他们没见过面试者在最后关头说这句话

菲姐 (291808436) 23:49:03

最最戏剧性的时刻出现 ,我的话他们没想到 ,老总的话我更没想到 ,我本是要退缩的

菲姐 (291808436) 23:50:50

老总说 :啊,强求 ,不强求啊 ,你要是强求我才不会签字呢 !你以为只有你有个性啊 ,我也很有啊 !!

说完 ,狂写了几句不知道什么话 ,说有事和方部长说吧 ,然后走了 !

菲姐 (291808436) 23:58:19

就这样我就过了 ,部长把我交给了人事部阴经理 ,女性 ,之前已被我的执着打动过 ,还问我怎么面试这么久啊 ,是久 ,从 9 点到 3 点 ,我饿的说不出话了 .然后说我是实习生 ,但也有工资 ,3420,试用期扣不扣我忘了 ,没听 .试用期三个月 ,签三年合同 (新劳动法规定 ,试用期一个月就要签 1 年合同 ,三个月签 3 年 ,就是为了试用期三个月而已,因为我不确定我会不会辞职 ),分公司会上交 1%的利润给总部 200 个人 ,所以会发 20 个月的工资 ,但多出的部分会有三分之一在年底发 (怕人跑 ,因为只要辞职至少损失三个月工资 ,这点我领教过 ),回去体检 ,等通知 ,因为手续还是比较麻烦地 .然后让我走了 ,今天体检 ,心电图不合格 ,崩溃中 .....

菲姐 (291808436) 23:58:27

完毕

菲姐 (291808436) 00:01:59

我的面试血泪史 ,也有很戏剧性的地方 ,但是确实没有开心过 ,因为其实我真地不知道能不能挺过三个月 ,我的力气已经用尽了 ,最近太拼心脏也出问题 ,之前没有事的 ,不知道这对我来说意味着什么 ,总之 ,啥也不知道了 ,掌上明珠今天我去一次说不去了 ,如果建龙不行 ,这几天我歇了 ,下周再找 .

菲姐 (291808436) 00:02:27

我说了很久啊 ,大家该睡了 ,真能唠叨 ,呵呵

菲姐 (291808436) 00:04:42

心电图不合格 ,不是因为饿或没休息好 ,后来我不死心又做的心脏彩超 ,也说明开始有问题 ,还不是心脏病 ,不能过度劳累 ,就素这样了

菲姐 (291808436) 00:05:58

而且最后我真地觉得是求来的 ,而不是我很强技术合格过的 ,总觉得不爽 ,是超郁闷

菲姐 (291808436) 00:07:43

其实你们只知道结果 ,而不知真相 ,所以会羡慕我

菲姐 (291808436) 00:08:50

只要你们坚持 ,像我一样 ,那就无论如何也能找到工作

菲姐 (291808436) 00:09:18

如果你们和我一样坚持 ,一定找到比这更好的工作 !

菲姐 (291808436) 00:10:42

我说完了 ,其实有一点不想进了 ,太难了 ,对我来说

菲姐 (291808436) 00:11:41

而且有朝一日我过了试用期 ,一定辞职 ,到时老总一定要说 :你很有个性 !

菲姐 (291808436) 00:16:32

总结 :找工作是一次成长的经历 ,经过这么多关 ,成长了一些 ,至少坚强 ,执着 ,执着到我都不到的地方 ,现在成败对我来说已经不重要了 ,就算不行 ,也未尝不是坏事 ,因为还会有其他的活法 .比如 ,周五下午轩辕互动复试 ,我又要去见老外了 ,呵呵 ,只要努力一定会成功 ,你们没有找到工作一定是什么没有做到 ,尽全力 ,就够了 ,一定找的到好工作 !

**2008 年 7 月 23 日：建龙钢铁集团**

公司网站 :<http://www.ejianlong.com/jlWeb/index.do?action=enter>

## 1.前序遍历 , 中序遍历 , 后序遍历

答：前序遍历 :根 >>左树 >>右树 ;

中序遍历 :左树 >>根 >>右树 ;

后序遍历 :左树 >>右树 >>根 ; 也就是说以根遍历的位置取名 .

来源 : <http://zhidao.baidu.com/question/28435209.html>;

来源 : <http://zhidao.baidu.com/question/55024808.html?fr=qrl&fr2=query>;

## 2.冒泡排序 ----BubbleSort:

### 基本概念

冒泡排序的基本概念是：依次比较相邻的两个数，将大数放在前面，小数放在后面。即首先比较第 1 个和第 2 个数，将大数放前，小数放后。然后比较第 2 个数和第 3 个数，将大数放前，小数放后，如此继续，直至比较最后两个数，将大数放前，小数放后，此时第一趟结束，在最后的数必是所有数中的最小数。重复以上过程，仍从第一对数开始比较（因为可能由于第 2 个数和第 3 个数的交换，使得第 1 个数不再大于第 2 个数），将大数放前，小数放后，一直比较到最小数前的一对相邻数，将大数放前，小数放后，第二趟结束，在倒数第二个数中得到一个新的最小数。如此下去，直至最终完成排序。



由于在排序过程中总是大数往前放，小数往后放，相当于气泡往上升，所以称作冒泡排序。

用二重循环实现，外循环变量设为  $i$ ，内循环变量设为  $j$ 。外循环重复 9 次，内循环依次重复 9, 8, ..., 1 次。每次进行比较的两个元素都是与内循环  $j$  有关的，它们可以分别用  $a[j]$  和  $a[j+1]$  标识， $i$  的值依次为 1, 2, ..., 9，对于每一个  $i$ ,  $j$  的值依次为 1, 2, ..., 10- $i$ 。

```
Dim a(1 To 10) As Integer, i As Integer, t As Integer, j As Integer
Randomize
For i = 1 To 10
 a(i) = Int(100 * Rnd)
 Print a(i);
Next i

For j = 2 To 10
 For i = 10 To j Step -1
 If a(i) < a(i - 1) Then
 t = a(i)
 a(i) = a(i - 1)
 a(i - 1) = t
 End If
 Next i
Next j

Print
Print "排序结果是:"
For i = 1 To 10
 Print a(i);
Next i

Print
```

产生10个随机整数

一趟排序

交换

来源：<http://baike.baidu.com/view/254413.htm>

图片：<http://imgsrc.baidu.com/baike/pic/item/9319cf09e47fa691d1581b91.jpg>

快速排序：(项目经理给过，今天上网查，说得很清楚，但是没看懂，也许用得到)

算法过程 [\[编辑本段\]](#)

设要排序的数组是  $A[1] \dots A[N]$ ，首先任意选取一个数据（通常选用第一个数据）作为关键数据，然后将所有比它小的数都放到它前面，所有比它大的数都放到它后面，这个过程称为一趟快速排序。一趟快速排序的算法是：

- 1) 设置两个变量  $I$ 、 $J$ ，排序开始的时候  $I:=1$ ， $J:=N$ ；
- 2) 以第一个数组元素作为关键数据，赋值给  $X$ ，即  $X:=A[1]$ ；
- 3) 从  $J$  开始向前搜索，即由后开始向前搜索（ $J:=J-1$ ），找到第一个小于  $X$  的值，两者交换；
- 4) 从  $I$  开始向后搜索，即由前开始向后搜索（ $I:=I+1$ ），找到第一个大于  $X$  的值，两者交换；
- 5) 重复第 3、4 步，直到  $I=J$ ；

例如：待排序的数组  $A$  的值分别是：（初始关键数据  $X:=49$ ）

$A[1] A[2] A[3] A[4] A[5] A[6] A[7]$ ：

49 38 65 97 76 13 27

进行第一次交换后：27 38 65 97 76 13 49

（按照算法的第三步从后面开始找）

进行第二次交换后：27 38 49 97 76 13 65

（按照算法的第四步从前面开始找  $>X$  的值， $65>49$ ，两者交换，此时  $I:=3$ ）

进行第三次交换后：27 38 13 97 76 49 65

（按照算法的第五步将又一次执行算法的第三步从后开始找

进行第四次交换后：27 38 13 49 76 97 65



( 按照算法的第四步从前面开始找大于 X 的值 , 97>49,两者交换 , 此时 J:=4 )

此时再执行第三步的时候就发现 I=J , 从而结束一趟快速排序 , 那么经过一趟快速排序之后的结果是 :

27 38 13 49 76 97 65 , 即所以大于 49的数全部在 49的后面 , 所以小于 49的数全部在 49的前面。

快速排序就是递归调用此过程 —— 在以 49为中点分割这个数据序列 , 分别对前面一部分和后面一部分进行类似的快速排序 , 从而完成全部数据序列的快速排序 , 最后把此数据序列变成一个有序的序列 , 根据这种思想对于上述数组 A 的快速排序的全过程如图 6所示 :

初始状态 {49 38 65 97 76 13 27}

进行一次快速排序之后划分为 {27 38 13} 49 {76 97 65}

分别对前后两部分进行快速排序 {27 38 13} 经第三步和第四步交换后变成 {13 27 38} 完成排序

{76 97 65} 经第三步和第四步交换后变成 {65 76 97} 完成排序

Jave中的Qsort [ [编辑本段](#)]

```
public class QuickSort {

 public static void sort(int[] number) {
 sort(number, 0, number.length-1);
 }

 private static void sort(int[] number,
 int left, int right) {
 if(left < right) {
 int s = number[(left+right)/2];
 int i = left - 1;
 int j = right + 1;
 while(true) {
 // 向右找
 while(number[++i] < s) ;
 // 向左找
 while(number[--j] > s) ;
 if(i >= j)
 break;
 swap(number, i, j);
 }
 sort(number, left, i-1); // 对左边进行递归
 sort(number, j+1, right); // 对右边进行递归
 }
 }

 private static void swap(int[] number, int i, int j) {
 int t;
 t = number[i];
 number[i] = number[j];
 number[j] = t;
 }
}
```

来源 : <http://baike.baidu.com/view/19016.htm#2>

3.子查询中的 in 和 exists什么时候使用 ？

答：关于 EXISTS 与 IN 的区别：

EXISTS 检查是否有结果，判断是否有记录，返回的是一个布尔型（ TRUE/FALSE ）。

IN 是对结果值进行比较，判断一个字段是否存在于几个值的范围中，所以 EXISTS 比 IN 快。

主要区别是：

- exists 主要用于片面的，有满足一个条件的即可，
- in主要用于具体的集合操作，有多少满足条件。
- exists 是判断是否存在这样的记录，
- In是判断某个字段是否在指定的某个范围内。
- exists 快一些吧。
- in 适合内外表都很大的情况，exists 适合外表结果集很小的情况。

来源：<http://hi.baidu.com/xinhuijun/blog/item/af6081187e0aabb54aedbc8f.html>

EXISTS 与 IN 的区别：

IN: 确定给定的值是否与子查询或列表中的值相匹配。

EXISTS: 指定一个子查询，检测行的存在。

下面从具体的语句来看：

select \* from 表 A where exists(select \* from 表B where 表B.id= 表A.id)

这句相当于：select \* from 表 A where id in (select id from 表 B)

对于表 A的每一条数据，都执行 select \* from 表 B where 表 B.id= 表 A.id 的存在性判断，如果表 B中存在表 A当前行相同的 id,则 exists为真,该行显示，否则不显示

来源：<http://www.cnitblog.com/youyou/archive/2007/09/07/33132.html>

11.数据库查询（题不记得了）

12.数据库查询（题不记得了）

13.TCP 三次握手

答：TCP 三次握手

TCP 连接是通过三次握手进行初始化的。三次握手的目的是同步连接双方的序列号和确认号并交换 TCP 窗口大小信息。以下步骤概述了通常情况下客户端计算机联系服务器计算机的过程：

1. 客户端向服务器发送一个 SYN 置位的 TCP 报文，其中包含连接的初始序列号 x 和一个窗口大小（表示客户端上用来存储从服务器发送来的传入段的缓冲区的大小）。
2. 服务器收到客户端发送过来的 SYN 报文后，向客户端发送一个 SYN 和 ACK 都置位的 TCP 报文，其中包含它选择的初始序列号 y、对客户端的序列号的确认 x+1 和一个窗口大小（表示服务器上用来存储从客户端发送来的传入段的缓冲区的大小）。
3. 客户端接收到服务器端返回的 SYN+ACK 报文后，向服务器端返回一个确认号 y+1 和序号 x+1 的 ACK 报文，一个标准的 TCP 连接完成。

TCP 使用类似的握手过程来结束连接。这可确保两个主机均能完成传输并确保所有的数据均得以接收

TCP Client	Flags	TCP Server
1 Send SYN (seq=x)	----SYN-->	SYN Received
2 SYN/ACK Received	<---SYN/ACK----	Send SYN (seq=y) , ACK (x+1)

3 Send ACK (y+1)	----ACK--->	ACK Received , <u>Connection</u> Established
w: ISN (Initial Sequence Number) of the <u>Client</u>		
x: ISN of the Server		

来源：[http://hi.baidu.com/it\\_hawk/blog/item/d053ab346830783e5ab5f54e.html](http://hi.baidu.com/it_hawk/blog/item/d053ab346830783e5ab5f54e.html)

#### 14.OOP,三种什么什么（英文没看懂）

答：面向对象编程（Object Oriented Programming，OOP，面向对象程序设计）是一种 计算机编程架构。

OOP 的一条基本原则是 计算机程序 是由单个能够起到子 程序 作用的单元或 对象 组合而成。OOP 达到了 软件工程 的三个主要目标：重用性、灵活性和扩展性。为了实现整体运算，每个对象都能够接收信息、处理数据和向其它对象发送信息。OOP 主要有以下的概念和 组件：

组件 - 数据和功能一起在运行着的计算机程序中形成的单元，组件在 OOP 计算机程序中是模块和结构化的基础。

抽象性 - 程序有能力忽略正在处理中信息的某些方面，即对信息主要方面关注的能力。

封装 - 也叫做信息封装：确保组件不会以不可预期的方式改变其它组件的内部状态；只有在那些提供了内部状态改变方法的组件中，才可以访问其内部状态。每类组件都提供了一个与其它组件联系的接口，并规定了其它组件进行调用的方法。

多态性 - 组件的引用和类集会涉及到其它许多不同 类型 的组件，而且引用组件所产生的结果得依据实际调用的类型。

继承性 - 允许在现存的组件基础上创建子类组件，这统一并增强了多态性和封装性。典型地来说就是用类来对组件进行分组，而且还可以定义新类为现存的类的扩展，这样就可以将类组织成树形或网状结构，这体现了动作的通用性。

由于抽象性、封装性、重用性以及便于使用等方面的原因，以组件为基础的编程在脚本语言中已经变得特别流行。Python 和 Ruby 是最近才出现的语言，在开发时完全采用了 OOP 的思想，而流行的 Perl 脚本语言从版本 5 开始也慢慢地加入了新的 面向对象 的功能组件。用组件代替现实 "上的实体成为 JavaScript（ECMAScript）得以流行的原因，有论证表明对组件进行适当的组合就可以在英特网上代替 HTML 和 XML 的文档对象模型（DOM）。

来源：<http://www.itisedu.com/phrase/200604240956125.html>

15. 四个人站在桥的一侧黑天过桥，一个手电筒照明，桥上一次只能过一个人或两个人，并且手电筒要有人返回给原来的一侧，就是说人来回过桥都要有人拿手电筒。只有 17 分钟四个人都要过去。每个人的速度不同，两个人走过桥，时间和速度以慢的人为准。四个人单独过桥的时间分别为：  
man1：1 分钟，man2：2 分钟，man3：5 分钟，man4：10 分钟，  
请问你能想到的最快的过桥方式？

例如：man1 和 man4 一起过桥，要 10 分钟，如果 man4 返回送手电筒，又要 10 分钟，总计 20 分钟。但是失败了。

#### 16.UML 翻译题（没想起来，原来是建模语言）

17.描述一下你觉得做得最好的项目（没时间答）

总结： 复习有很多漏洞， 但是通过面试复习这些知识点， 下次出现同样的问题不要错！(告诉自己)。 总之，用我的玫琳凯督导的话说我， 死活都要做 IT!!

## 建龙钢铁面试总结

笔试题目不少，一部分是以前见过的题，我只记得几道以前没见过的题。

- 1, http 中的四个方法作用分别是什么
- 2, 具体记不清了，里面有一个静态变量和一个非静态变量，两次赋值（远程和本地），写出 RMI 调用后的输出结果
- 3, 有关 JPA 的题
- 4, 写代码实现 comparable 接口
- 5, 两种方法写代码实现单例模式
- 6, 调用方法时的参数传递方式问题

编程题：

自己编程产生一个无序整数数列， 排序，打印出排序后的数列及每个数字在原无序数列中的序号。

### 经典时空面试总结

笔试大多数题都比较简单，全是基础语法题。只有一道题以前没见过：

Java 中如何进行文件（及文件夹）的新建，移动，删除等？最好写出实现代码，如果不能，文字说明实现方案。

上机题：

- 1,html 页面中有一组单选框，选中一个后点击按钮弹出你选的是第几个按钮。
- 2, 将 html 页面中两个文本框中的内容对换。
- 3, 用 json 作为数据传递格式实现浏览器和服务器的异步交互。
- 4, 打印从 1 到 n 的所有排列。（n 为整数，运行时键盘输入）

## 北京捷通华声语音技术有限公司笔试题

这个公司的笔试分为两个部分，包括技术部分和测试部分：

第一部分：

1. 作用域 public,private,protected, 以及不写时的区别  
Public 是公共的作用域，对当前类，同一个包及子孙类，其他包都包括  
Protected 作用域为 当前包的范围 包括子送类  
Friendly 作用与 同包下 不包括子送类  
Private 只作用与当前类  
不写默认为 private 作用域

2 请写出每种基本类型的默认值？取值范围？分别给出一个定义的例子

3.安全线程

4 一个线程的英文选择题 （好像是选 e)记不好了

5.重写和重载的区别

答：方法的重写 Overriding 和重载 Overloading 是 java 多态的不同表现，重写 Overriding 是父类与子类之间多态的一种表现， Overloading 是一个类中多态的一种表现，如果在子类中

定义某种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，对他而言，父类中的定义如同被屏蔽了。如果一个类中定义了多个同名的方法， 他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 ）。Overload 的方法是可以改变返回值的类型。

6.编程 一个多态的（一个蝙蝠和动物和兽类和鸟类的 ）写个动物的接口 剩下的写三个类实现接口重写方法就行了

```
Class Animal
{
Public void
```

7.编程题 :自动洗扑克牌（给传进去一个数组 产生一个随即数 把数组内的  
249992885 2009-02-11 16:25:45  
顺序打乱 ）

8.编程 :一个类似迷宫的编程题 根据给的游戏图编程（手机游戏 有左转右转 前进 ）

9.脑筋题：有六个职位 3个人 每个人有两个职位 根据即句话判断每个人的职位

答案是 第一个人 :艺术家 律师

第二个人 :建筑师 教师

第三个人 :作家 医生 (答案记得大概是这样 要是当时实在想不出来就用这个嘿  
嘿)

10.数组有没有 length() 这个方法？String 有没有 length() 这个方法

11.位运算题

具体题想不起来了

类似于：

-5>>3=-1;

-5<<3=-40;

5&3=1;

5|3=7;

-5|3=-5;

~5=-6;

-5^3=-8;

-5>>>3=53687091

就记得这些了

还有个公司出的类似面试的题 就是问情况的（就两个需要注意的）1.里面一个让估算长江每年流入大海的水是多少吨 2.还有一个工作经验需要留一个证明人的电话

(考完才知道 他们没有时间限制 一上午坐完就行 )

第二部分：

一、 你如何选择自己的职业？

二、 在高薪、表彰和晋升之间，你认为哪种形式最有价值？

三、 你有什么业余爱好？

四、 你为什么来我们公司工作？

五、 我们为什么要录用你？

六、 我们和另一个公司都聘用你，你会如何选择？

- 七、 最基础的工作你也会愿意干么？
- 八、 你有哪些主要的优点？
- 九、 你有哪些主要的缺点？
- 十、 你以前的经验和我们现在的工作有哪些联系？
- 十一、 你对以后有什么打算
- 十二、 在学校你最不喜欢的课程是什么？为什么？
- 十三、 你认为你在学校属于好学生吗？
- 十四、 你是应届毕业生，缺乏经验，如何能胜任这项工作？
- 十五、 对这项工作，你有哪些可预见的困难？
- 十六、 你和其他求职者有什么不同？
- 十七、 你希望与什么样的上级共事？
- 十八、 你因为工作过失与同事产生了矛盾，你怎样去做？
- 十九、 你对公司有什么问题要问的？
- 二十、 你对工资有什么期望？

最新补充： （三月二十三）

第一部分加的新题，

```
String s1 = new Sting (hello);
String s2 = new String (hello)
```

```
System.out.println(s1==s2);
结果是什么？ 为什么是这样的结果？
```

- 1 你知道相对论吗？请简单介绍下。
- 2 假如把你放在一个孤岛上一年，让你选三本书和两张 CD ，你选什么？
- 3 你家水龙头坏了（水量很大） ，你如何处理？
- 4 长江每年流入海洋的水有多少？
- 5 你和你的朋友开餐馆，给 总经理，前台经理，厨房管理，广告宣传，融资，财务管理，及购买原料，你会选择做什么？
- 6 假如你被捷通录用，你想从事哪方面的工作，你擅长什么？
- 7 你在捷通工作了一年了 ，描述一下你对自己的职业生涯和职业规划。
- 8 你期望的主管应该怎么做？
- 9 建议一下公司应该为员工做哪方面的培训？
- 10 你能为企业做出什么贡献？
- 11 对员工的考核应该从哪些方面？
- 11 你原来要有工作经验，写下主管或同事的姓名，职位，联系方式；要是应届生，写老师的联系方式。
- 12 你期望的月薪是多少？

## 金鑫耀笔试题

- 1.面向对象的特点
- 2.java 的基础数据类型
- 3.jsp 的 9 大内置对象
- 4.网页中 <input type="text" id="text1"> 如何获得这个元素
- 5.线程中的关键字是什么？实现线程有哪两种方法
- 6.编程实现 4 个线程，2 个是对数字 j 加 1，另外 2 个是对数字 j 减 1.

## 经典时空面试总结

笔试大多数题都比较简单，全是基础语法题。只有一道题以前没见过：

Java 中如何进行文件（及文件夹）的新建，移动，删除等？最好写出实现代码，如果不能，文字说明实现方案。

上机题：

- 1,html 页面中有一组单选框，选中一个后点击按钮弹出你选的是第几个按钮。
- 2，将 html 页面中两个文本框中的内容对换。
- 3，用 json 作为数据传递格式实现浏览器和服务器的异步交互。
- 4，打印从 1 到 n 的所有排列。（n 为整数，运行时键盘输入 ）

## 九州传奇的上机题

用 java 和 sql server2000或 oracle 或 mysql 实现：

servlet，struts，webwork，struts2 实现均可

用户管理信息：

主界面要求有姓名，密码输入框和两个按钮 "确定"和"用户管理"，  
当点击第二个按钮时出来一个对话框包括：增加,删除,修改 3 个按钮

纽

(可批量删除，批量删除后，弹出对话框显示被删除的用户名字。

如：被批量删除的用户有 [张三，李四，等等 ]，

实现对应的数据库操作。

## 久其面试题（研发类）

- 1.常用的开源框架

2.Eclipse 开发中用到的插件

3.项目开发中，遇到问题如何解决，举例说明

4.MVC 控制层的作用

5.开发中用到的工具

6.谈谈数据库的内外连接

7.你用过哪些数据库

8.学习了哪些课程

9.举例说明快速学习的成果

10.在大学期间所获的奖励

11.谈谈数据结构

12.说说 struts 框架

在前台会先让填一张表， 正面是自己的基本信息， 反面是心理测试题， 就是让你选出符合描述自己的名词。

之后是做一套题， 内容涉及到操作系统、 数据结构、 计算机网络、 计算机组成原理、 数据库。题目类型分别是： 30 道单选、 10 道多项（多数有 5-7 个备选项）、填空题、 3 个程序设计题。先说程序设计题：

- 1、 给出打保龄球计算得分的规则，然后给出程序设计流程图（ N-S 图），让根据给出的规则将流程图补充完整。（网上好像有）
- 2、 补充程序题，让就一个问题给出其递归算法和相应的递推算法（ while ）

综合的回忆一下其他题目：

1 数据结构中的图，用广义表给出，要求从 A 点出发，给出广度优先搜索，深度优先搜索以及最优路径

2 关于测试的试题，有语句覆盖、路径覆盖、选择覆盖等，然后给出测试用例，让选出需要哪些用例才能实现以上覆盖

以上两个题就占了 30 各选择题当中的 10 个

3 问链路层的协议是什么

4 关于数据库的题很简单

5 填空题中有关于 uml 的概念题要懂得一些概念及相关作用



## 酷讯网笔试题

1. 将"C:/win/kuxun/\*.\*" 目录下所有的文件打印到控制台，用笔写
2. 面试的时候问了一些基础概念，比如什么是类、方法、函数。

## 联创题目

选择题：（总共 20 道）

- 1 关于类的描述
- 2 JAVA 关键字的相关题目
- 3 JAVA 的垃圾回收机制
- 4 一个 INT 型数据，让它的值翻倍
- 5 基本全是问程序的执行结果，想不起来了 （有关于赋值，异常等很多，部份 653 考试题中的选择题）

简答题：

- 1 struts中关于共享资源区作用域的问题，以及转发跟重定向的主要区别
- 2 struts中 formbean、 validator 的相关内容。（我觉得这个题如果出现似乎应该是在编程题中）
- 3 JSP中有三种独立的向页面添加代码段的技术，写出其中的两种
- 4 结合你所做项目中用到的一个框架， 谈谈它的特点， 利弊，主要结合业务描述(简述 **JAVA** 的垃圾回收机制优点和原理，并考虑 2 种回收机制。有可能在简答中考)

编程题

- 1 写一个 JSP中关于 error-page 404错误的配置，让其指向 error404.jsp的页面，或者描述其配置的原理
- 2 将一个 int 型数组写到一个 temp.dat文件中，再从 temp.dat 中读出，并在控制台上逆序打印数组元素
- 3 帮一个业务操作运营平台写一个用户登录页面

智力题

爱因斯坦所出的一道推理题（ 他说世界上有 98%的人答不出来。 ）  
求解：" 谁养鱼 "

- 1、在一条街上，有 5 座房子，喷了 5 种颜色。
- 2、每个房里住着不同国籍的人

3、每个人喝不同的饮料，抽不同品牌的香烟，养不同的宠物

问题是：谁养鱼？

提示：

1、英国人住红色房子

2、瑞典人养狗

3、丹麦人喝茶

4、绿色房子在白色房子左面

5、绿色房子主人喝咖啡

6、抽 Pall Mall 香烟的人养鸟

7、黄色房子主人抽 Dunhill 香烟

8、住在中间房子的人喝牛奶

9、挪威人住第一间房

10、抽 Blends 香烟的人住在养猫的人隔壁

11、养马的人住抽 Dunhill 香烟的人隔壁

12、抽 Blue Master 的人喝啤酒

13、德国人抽 Prince 香烟

14、挪威人住蓝色房子隔壁

15、抽 Blends 香烟的人有一

答案：德国人

联创集团北京项目组 地址：北京市大郊亭桥东 200 米电信大厦 8 层联创办公室

乘车路线：乘 1 号地铁大望路下车 B 口出，转乘 11 路公交车，东石门下车，对面既是

联系人： 马鹭 手机：13581830976

## 联创题目

选择题：（总共 20 道）

- 1 关于类的描述
- 2 JAVA 关键字的相关题目
- 3 JAVA 的垃圾回收机制
- 4 一个 INT 型数据，让它的值翻倍
- 5 基本全是问程序的执行结果，想不起来了 （有关于赋值，异常等很多，部份 653 考试题中的选择题）

简答题：（好像是选做三个）

- 1 struts 中关于共享资源区作用域的问题，以及转发跟重定向的主要区别  
转发跟重定向的主要区别在于： 1.请求转发的时候，请求中所带的参数是放在数据包的包体当中，而重定向的时候请求所带的参数是放在包头当中的。 2. 转发的时候可以传参传的是对象，而重定向的时候所带的参数只能是字符串。 3. 使用转发的时候地址栏无变化，而使用重定向的时候地址栏会发生变化。 4.请求范围不同，转发属于同一个请求范围而重定向属于两个不同的请求范围。
- 2 struts 中 formbean、 validator 的相关内容。（我觉得这个题如果出现似乎应该是在编程题中）
- 3 JSP 中有三种独立的向页面添加代码段的技术，写出其中的两种  
在<%%>中加入 Java 脚本； jstl 、 el 表达式。
- 4 结合你所做项目中用到的一个框架， 谈谈它的特点， 利弊， 主要结合业务描述  
Hibernate 框架：其主要优点在于： 1. 对 JDBC 访问数据库的代码做了封装，大大简化了数据访问层繁琐的重复性代码。 2. Hibernate 是一个基于 JDBC 的主流持久化框架，是一个优秀的 ORM 实现。他很大程度的简化 DAO 层的编码工作  
3. hibernate使用 Java反射机制，而不是字节码增强程序来实现透明性。 4. hibernate的性能非常好，因为它是个轻量级框架。 映射的灵活性很出色。 它支持各种关系数据库，从一对一到多对多的各种复杂关系。 5. 当 Hibernate在查询数据的时候，数据并没有存在与内存中，当程序真正对数据的操作时，对象才存在与内存中，就实现了延迟加载，他节省了服务器的内存开销，从而提高了服务器的性能。  
6.hibernate中加入了一、二级缓存， ehcache提升了服务器的性能。  
其主要缺点在于： 1.虽然使用 hibernate减少了大量重复代码的编写， 提升了程序以及服务器的性能但是对于初学者而言配置文件太多而且太繁杂， 不能很好的应用他的优势。 2.使用 hibernate的时候他的底层对于我们而言是透明的， 出现问题的时候调试不是很方便。

(简述 **JAVA** 的垃圾回收机制优点和原理，并考虑 2 种回收机制。有可能在简答中考)

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。

编程题

1 写一个 JSP中关于 error-page 404错误的配置，让其指向 error404.jsp的页面，或者描述其配置的原理

```
<error-page>
```

```
 <error-code>404</error-code>
```

```
 <location>error404.jsp</location>
```

```
</error_page>
```

2 将一个 int 型数组写到一个 temp.dat文件中，再从 temp.dat 中读出，并在控制台上逆序打印数组元素（包自己导）

```
public class Test {
 public static void main(String[] args) {
 int[] num = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 0 };
 PrintWriter os = null;
 try {
 os = new PrintWriter(new
FileOutputStream("d:/tmp.dat"));
 for (int b : num) {
 os.print(b);
 }
 os.flush();
 } catch (Exception e) {
 e.printStackTrace();
 } finally {
 try {
 os.close();
 } catch (Exception e) {
 e.printStackTrace();
 }
 }
 InputStream is = null;
 try {
 List list = new ArrayList();
 is = new FileInputStream("d:/tmp.dat");
 int i;
 while ((i = is.read()) > 0) {
 list.add(i);
 }
 for (int n = list.size(); n >= 0; n--) {
 System.out.println(n);
 }
 }
 }
}
```

### 3 帮一个业务操作运营平台写一个用户登录页面

这个仅仅是一个页面 ~没有什么！

乘车路线：乘 1 号地铁大望路下车 B 口出，转乘 11 路公交车，东石门下车，对面既是

用什么方法校验：正则， Ajax 等

共九道题，我只记住了六道

灵泰克 JAVA程序员应聘测试题

姓名：\_\_\_\_\_ 联系方式：\_\_\_\_\_

一、阅读下面的代码，回答：（1）写出这段代码所表示的数学表达试；（2）找出这段代码存在的问题。

```
double d=0;
for(int i=1;i<=10000;i=i+1){
 d=d+1/(i*(i+1)*(i+2));
 if(i==100)
 System.out.println("i=100 时："+d*4);
 else if(i==1000)
 System.out.println("i=1000 时："+d*4);
}
System.out.println("i=10000 时："+d*4);
```

二、编写一个 JSP 文件，要求用 JSTL 的循环标签将“ 1 , 2 , 3 , 4 ; 5 , 6 , 7 , 8 ”显示成如下效果，写出主要的代码。

第一列	第二列	第三列	第四列
1	2	3	4
5	6	7	8

三、写出符合 MYSQL 要求的查询语句： 查出某一个班级一次考试中成绩排前 10 名的学生。

四、某一培训学校要做一个网站，其中有一个招生报名功能模块，这件事情安排给你做，你描述一下工作如何开展，具体的实施步骤。

五、从软件设计人员的角度，你认为：“无他，唯手熟尔！”这句话蕴涵了什么哲理

北京灵泰克基础教育服务中心 地址：北京市丰台区西南四环花乡桥南 200 米，波峰商务楼 2 楼 网址：<http://www.lingtek.com>

乘车线路：四环沿线公交车（下车后向南 200 米路东） 657( 东湖 -丰北桥 )花乡桥 736( 灵秀山庄东 站 -玉泉路口南 )花乡桥 740( 十八里店村 -十八里店村 )花乡桥 996( 平西王府 -肖村桥西 )花乡桥特 3(燕莎桥南 -造甲村东 )花乡桥特 9(黄土岗村 -黄土岗村 )花乡桥运通 115(金庄 -泰河一街西口 )花乡桥 直达公交车 689( 花乡驾校 -惠新东桥西 )花乡桥南 937 专(南礼士路 -羊坊村 )花乡桥南

## JAVA程序员应聘测试题

姓名：\_\_\_\_\_ 联系方式：\_\_\_\_\_

一、阅读下面的代码，回答：（1）写出这段代码所表示的数学表达式；（2）找出这段代码存在的问题。

```
double d=0;
for(int i=1;i<=10000;i=i+1){
 d=d+1/(i*(i+1)*(i+2));
 if(i==100)
 System.out.println("i=100 时："+d*4);
 else if(i==1000)
 System.out.println("i=1000 时："+d*4);
}
System.out.println("i=10000 时："+d*4);
```

第一种方法：double d=0;

```
for(int i=1;i<=10000;i=i+1){
 d=d+(double)1/(i*(i+1)*(i+2));//---- 修改
 if(i==100)
 System.out.println("i=100 时："+d*4);
 else if(i==1000)
 System.out.println("i=1000 时："+d*4);
}
System.out.println("i=10000 时："+d*4);
System.out.println(d);
```

10000

第二种方法：问题一：d= 1/(i\*(i+1)\*(i+2))

i=1

问题二：此程序是为了对 1/(i\*(i+1)\*(i+2)) 求和，但是此表达式分母和分子都是整型，根据 java 的数据类型隐式转换规则，1/(i\*(i+1)\*(i+2)) 计算的结果的是 0，这样是达

不到程序的目的。所以应该把 for(int i=1;i<=10000;i=i+1) 这句中的 i 定义成 double

型，修改后应为 for(double i=1;i<=10000;i=i+1) 。

二、编写一个 JSP 文件，要求用 JSTL 的循环标签将 “ 1 , 2 , 3 , 4 ; 5 , 6 , 7 , 8 ” 显示成如下效果，写出主要的代码。

第一列      第二列      第三列      第四列

1	2	3	4
5	6	7	8

```

<table>

<tr>

 <th> 第一列 </th>

 <th> 第二列 </th>

 <th> 第三列 </th>

 <th> 第四列 </th>

</tr>

<tr>

<c:forEach var="i" begin="1" end="8" step="1">

 <td><c:out value="\${i}"/></td>

 <c:if test="\${i==4}">

</tr><tr>

 </c:if>

</c:forEach>

</tr>

</table>

```

三、写出符合 MYSQL 要求的查询语句： 查出某一个班级一次考试中成绩排前 10 名的学生。

```
selet top 10 * from 学生表 order by 成绩 desc
```

四、某一培训学校要做一个网站，其中有一个招生报名功能模块，这件事情安排给你做，你描述一下工作如何开展，具体的实施步骤。

首先需要了解的有：

- 1、网站的建站目的；（招生）
- 2、客户以及网站功能需求；（学习与招生的流程）
- 3、制定项目执行计划；
- 4、需求分析后根据实际情况设定功能属性（数据库表建立）；
- 5、根据计划完成编码、测试（其中包含反馈实际执行进度）；

五、从软件设计人员的角度，你认为：“无他，唯手熟尔！”这句话蕴涵了什么哲理

我觉得做软件设计首先思路要清晰，然后就是去熟悉业务，要做到不仅仅是思路成熟，还有



手熟，就像做一件事情，知道应该怎么做，并且能那样做，就是一定去执行。

北京灵泰克基础教育服务中心 地址：北京市丰台区西南四环花乡桥南 200 米，波峰商务楼 2 楼 网址：<http://www.lingtek.com>

乘车线路：四环沿线公交车（下车后向南 200 米路东） 657( 东湖 - 丰北桥 ) 花乡桥 736( 灵秀山庄东 - 玉泉路口南 ) 花乡桥 740( 十八里店村 - 十八里店村 ) 花乡桥 996( 平西王府 - 肖村桥西 ) 花乡桥特 3( 燕莎桥南 - 造甲村东 ) 花乡桥特 9( 黄土岗村 - 黄土岗村 ) 花乡桥运通 115( 金庄 - 泰河一街西口 ) 花乡桥 直达公交车 689( 花乡驾校 - 惠新东桥西 ) 花乡桥南 937 专 ( 南礼士路 - 羊坊村 ) 花乡桥南

## 领时科技笔试题

### 一、基础题

1. 谈谈 final, finally, finalize 的区别。
2. error 和 exception 有什么区别？
3. abstract class( 抽象类 ) 和 interface 有什么区别？
4. 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？
5. 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，在什么时候被执行，return 前还是后？
6. 谈谈 & 与 &&, equals() 与 == 之间的区别。
7. 谈谈 int 和 Integer 有什么区别。

### 二、分析题

1. 如果要向服务器上传或下载 100M 或更大的文件，你有什么好的处理办法？
2. 如果客户端正向服务器发送请求时，服务器突然崩溃，你认为可能出现问题的原因是什么？
3. 你觉得 Hibernate3.0 在处理复杂的多表之间的关联关系，有什么优势？它的略施是什么？除了 Hibernate 以外，你还接触过哪些 O/R 映射框架？

苗栗强面试题总结：

1. 你是怎么理解 oa, od, op 这三个概念的。他们之间的联系。
2. soap 是什么，请你列举出三种常用的例子。
3. 请你说说 java(j2ee) 三层架构体系。
4. 请你从 ( oracle, mysql, ms sql, db2 ) 任意选一个数据库回答它是如何进行数据库管理？
5. 请你如何构架 erp 企业网络系统一个总部和异地的网络拓扑图，会议视频系统，oa 系统。ERP 系统。把他们分布图画出来。
6. 请你说出两种常用的建模语言工具。
7. 还有一道就是数据库题，我觉得用全连接查询。
8. 请你 Struts 框架写一个登录程序。

## 民航中天笔试题

1. 抽象类的定义，在定义四个选项选择一个可以通过编译的抽象类。
2. A 继承 B，B 中有个 final 的构造方法，A 的构造方法与 B 不同，能编译通过吗？
3. A 继承 B，A、B 各有一个不同的构造方法（里面都有打印），通过 A 的构造方法 NEW A，问打印那个？
4. 写一个类的重载。
5. 使一个对象符合 java 虚拟机的垃圾回收条件，写两个方法。
6. 简单描述一下两种异常。
7. try catch finally 的嵌套，输出打印结果
8. 注意 StringBuffer 的删除 CAHR 插入 CHAR 的方法即可。
9. 将 1、1、2、3、4 的英文单词存放到一个 TreeSet 中，便利打印，写出结果
10. (SELECT COL1,COL2 FROM TABLENAME WHERE COL1='TEST')  
写一个方法实现上述语句所对应的功能。

整套题目，考的比较基础，有好多地方迷惑性比较强，需要小心。

## 铭万公司笔试题

### 一.IQ 题

1.  $62-63=1$  答案是 2 的 6 次方减 63

怎样移动一位数字使上面等式成立，只能移动数字。

2. 根据图形变化填写图形 答案是两个 8 连着写

二、写书 sql 语句 (有三个表 学生表 {学生 id 学生姓名} 课程表 {课程名 课程 id} 成绩表 {课程 id 学生 id 成绩表 id 成绩})

一 查所有学生的姓名 课程 成就 (等值连接用 where 判断)

二 查所有缺课的学生姓名 就是课程 id 和学生 id 在成绩表中没有的 (等值连接用 where 判断)

三、将数据库中表中重复的数据删除掉，只保留 ID 号最小的那条记录

delete from T

where (a,b,c,d) in (select a,b,c,d from T group by a,d,c,d having count(\*) > 1)

and rowid not in (select min(rowid) from T group by a,b,c,d having count(\*)>1)

四、语句查出 A、B 表中全部的数据，并且删除掉重复的数据

Select from a union select from b

### 五、java 基础题

1. final、finally、finaliz 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

2.& 和&& 的区别， ==和 equals 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）

equals 方法比较的是对象的值

==:比较的是对象值的内存地址，对基本数据类型来说 == 比较的也是值

3.静态类和内部类的区别

和普通的类一样，内部类也可以有静态的。不过和非静态内部类相比，区别就在于静态内部类没有了指向外部的引用。这实际上和 C++中的嵌套类很相像了，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这一点上，当然从设计的角度以及以它一些细节来讲还有区别。

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

除此之外，在任何非静态内部类中，都不能有静态数据，静态方法或者又一个静态内部类（内部类的嵌套可以不止一层）。不过静态内部类中却可以拥有这一切。

4. 匿名内部类是否可以 extends 其他类，是否可以实现接口

答：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

5 表 a 的外键 id 是表 b 的外键 id 要删除表 a 的 id 应该怎么做

先删外键的再删主键的

六、写出执行语句

第一个题是怎样调用父类代码

答案 super.(父类的方法名)

第二个题是内部类的执行语句

答案 先输出内部类的再输出外类的

七编程：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;
 }
}
```

```

public void SplitIt()
{
 int loopCount;

 loopCount=(SplitStr.length()%SplitByte==0)?(SplitStr.length()/SplitByte):(SplitStr.length()/Split
 Byte+1);

 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(" 我 ABC 汉 DEF",6);
 ss.SplitIt();
}
}

```

铭万信息技术有限公司          总部：北京市朝阳区安苑路          15 号铭万信息大厦          网址：  
<http://corp.mainone.com/>  
 乘车路线：公交：车站站名安苑路          409、 18 即可到达  
                  地铁：地铁 5 号线惠新西街北口站下，      A 出口安徽大厦南第一个红绿灯右拐      100  
                  米白色九层大楼即是

## 南北软件笔试题

孙文中总结：

- 1 有一个方法，传进一个字符串和拆分符，根据拆分符返回一个          String[] ，不能用 split()  
 (如传入 String s =      -eeppp 返回 [    abc    ,    eee    ,    ppp    ])  
 Public static String[] getArray(String s,char mark) {  
                  //写出具体实现  
 }  
 答案： public static void main(String[] args) {

```

 String s = "abc-eee-ppp";
 String[] strs = getArray(s,"-");
 for(String ss :strs){
 System.out.println(ss);
 }
 }

 public static String[] getArray(String s,String mark) {
 StringTokenizer st = new StringTokenizer(s,mark);
 int count = st.countTokens();
 String[] strs = new String[count];
 int i = 0;
 while(st.hasMoreElements()){
 strs[i] = (String)st.nextElement();
 i ++;
 }
 return strs;
 }
}

```

2 根据几个表，数据库 7 道题：

Employee 表：

ename 员工名字，主键

street 员工街道

city 员工家城市

Works 表：

ename 员工名字，外键，引用 Employee 的 name

cname 公司所在城市名，外键，引用 Company 的 cname

salary 员工的年薪

Company 表：

cname 公司所在城市名，主键

city 公司所在城市

Manager 表：

ename 员工名字，外键，引用 Employee

mname 经理名字

1 查询与经理同街道同城市的雇员的名字

2 First Bank 的雇员姓名

(都是基于多表的链接，还有子查询，内建视图和分组)

3

class A

class B extends A

class C extends B

class D extends A

class E extends D

有几个静态方法

```
Public static void f1(A a,B b) {System.out.println(1)};
```

```
Public static void f1(D d,C c) {System.out.println(2)};
```

```
Public static void f1(C c,B b) {System.out.println(3)};
```

(好像是这样，考得是多态和继承，基础 )

给几个方法，判断可否执行，可以写结果，不可以写原因 s

F1(new A(),new C()); (类似于这样的 )

F2(new A(),new C());

(画出继承图就行，参数要超类给子类的没问题，要子类给超类的不行 )

4 class A extends Thread implements Runnable {

```
 Public void run() {
```

```
 Sysout(1);
```

```
 }
```

```
}
```

有没有错误，有是编译期还是执行期，原因

5 一道内部类的题，内部类的声明方式，内部类里方法的声明方式，给了选项几个选择。

6 一道选择题，没见过的就是 Boolean boo = Boolean.FALSE; 是否正确，别的都简单。

7 基础

```
public static void f1(String s) {sysout(s + world)}
```

```
public static String f2(String s) {return s + world }
```

```
public static void main() {
```

```
 String s = hello ;
```

```
 f1(s);
```

```
 sysout(s);
```

```
 String s2 = hello ;
```

```
 f2(s2);
```

```
 sysout(s2);
```

```
}
```

有几个忘了，最后还有一道翻译，大概有一篇四级文章大小

做完还有一套类似于公务员行政能力考试的题， (从出的题看，估计考察一下对数字的敏感程度，数字推理，还有找不同什么的 )

规定用时两个半小时，数据库好像他们挺看重，多表，子查询，内建视图，分组都用到了，没有特别难的好像

## 韩岳峰总结：

1. 一个公司人员值班表 赵比孙晚一天 李比吴晚两天 钱比王早三天 郑在钱和孙之间  
有几个选项没记住 看选项谁对 就是吧这几个人排好就可以

2. 指出正确的表达式

3. 给一段程序看是否正确

4. 

```
class super{
 public float getnum(){return 3.0f}
}
public class sub extends super{
 //insert code here 处加入一个方法 错误的是
}
```

  
选项没记住

5. 找出程序中的错误的

6. 同上

7. 

```
public static void main(){
 static int arr[][] = new int [10][];
 int x=3;
 int y=1;
 if(x==y)
 sysout(1)
 else
 sysout(2)
}
```

  
问结果

8. 也是以个读结果题

第二题

1. int 和 intager 区别

2. string 和 stringB 区别

3. 线程的实现方式有几种 分别是什么 STOP()和 SUSPEND() 为何不推荐

4. XML 几种技术的区别

第三题就是数据库的有 4 到查询语句 和上次题基本一样

第四题 是 英文翻译 一段英文翻译 不好翻译