

开发面试

2016年11月17日 星期四 17:05

面试建议	步骤1：让对方先进行简短的自我介绍
	步骤2：让对方讲一讲毕业后的工作生涯，需要描述清楚“时间”、“公司”、 (这个阶段主要是观察对方的逻辑思路，还有让对方放松)
	步骤3：进行结构化面试，并打分
	步骤4：进行非结构化试题
	步骤5：薪资提问

非结构化试题

自我介绍	
描述最熟悉的项目	
描述最新学习的技术	
描述团队的组成和分工	
你认为在你的经历中，哪一段对你的成长帮助较大	

结构化试题

HashMap与Hashtable的区别	1.HashMap是线程不安全的，Hashta 2.HashMap允许有null的键或值，Ha 3.HashMap效率高，Hashtable效率低 4.HashMap继承于AbstractMap，Ha
StringBuffer与StringBuilder的区别	1.StringBuffer是线程安全的，Hashta 2.StringBuffer效率低，StringBuilde
Error与Exception的区别	1.Error是不可控制的，Exception是可 2.Error是系统引起，Exception是代码

“项目” 、 “职责”

able是线程安全的 ashtable不允许有null的键或值 低 ashtable继承于Dictionary	
able是线程不安全的 效率高	
可以控制的（try catch） 逻辑引起的	

接口与抽象类的区别	<ol style="list-style-type: none"> 1.接口是完全不实现的，抽象类是允许 2.接口不允许有构造器，抽象类是可以 3.接口的只能是public修饰，抽象类可 4.接口可以多继承接口，抽象类只能单 5.接口效率较低（需要寻找实现类）， 6.接口中的方法必须都实现，抽象类中
单例模式的实现方式	<ol style="list-style-type: none"> 1.懒汉式（第一次调用时才初始化） 2.饿汉式（先初始化后调用）
单例模式的实现方式优缺点	<p>优点：</p> <ol style="list-style-type: none"> 1.确保所有对象都访问唯一实例 2.节省系统的资源 <p>缺点：</p> <ol style="list-style-type: none"> 1.没有抽象层，不利于扩展 2.单例类职责过重，违背单一职责原则
Servlet是单实例还是多实例，单线程还是多线程	<ol style="list-style-type: none"> 1.单实例 2.多线程
Spring Ioc的好处	<ol style="list-style-type: none"> 1.实现了资源集中管理，方便维护和使 2.降低了资源之间的依赖，即低耦合 3.默认使用单例模式（饿汉），降低资
Spring Aop的应用场景	<ol style="list-style-type: none"> 1.Authentication 权限 2.Caching 缓存 3.Context passing 内容传递 4.Error handling 错误处理 5.Lazy loading 懒加载 6.Debugging 调试 7.logging, tracing, profiling and mo 8.Performance optimization 性能优 9.Persistence 持久化 10.Resource pooling 资源池 11.Synchronization 同步 12.Transactions 事务

部分实现的 有构造器 可以是public、protected修饰 继承抽象类 抽象类效率高 只需要实现部分	
使用	
资源消耗	
monitoring 记录跟踪 优化 校准 优化	

Spring中有哪几个事务隔离级别	<p>1.ISOLATION_DEFAULT 使用后端数据库默认的隔离级别</p> <p>2.ISOLATION_READ_UNCOMMITTED 允许读取尚未提交的更改。可能</p> <p>3.ISOLATION_READ_COMMITTED 允许从已经提交的并发事务读取</p> <p>4.ISOLATION_REPEATABLE_READ 对相同字段的多次读取的结果是</p> <p>5.ISOLATION_SERIALIZABLE 完全服从ACID的隔离级别，确保 全锁定当前事务所涉及的数据表</p>
Spring中有哪几种事务传播行为	<p>1.PROPAGATION_MANDATORY 表示该方法必须运行在一个事务</p> <p>2.PROPAGATION_NESTED 表示如果当前正有一个事务在进行 回滚。如果封装事务不存在，行</p> <p>3.PROPAGATION_NEVER 表示当前的方法不应该在一个事务</p> <p>4.PROPAGATION_NOT_SUPPORTED 表示该方法不应该在一个事务中</p> <p>5.PROPAGATION_SUPPORTS 表示当前方法不需要事务性上下</p> <p>6.PROPAGATION_REQUIRES_NEW 表示当前方法必须在它自己的事务 行期间被挂起。</p> <p>7.PROPAGATION_REQUIRES 表示当前方法必须在一个事务中 务。</p>
Oracle中如果使用执行计划	EXPLAIN PLAN FOR sql

<p>。可防止脏读、幻影读或不可重复读。</p> <p>。可防止脏读，但幻影读和不可重复读仍可能会发生。</p> <p>一致的，除非数据被当前事务本身改变。可防止脏读和不可重复读，但幻影读仍可能发生。</p> <p>保不发生脏读、不可重复读和幻影读。这在所有隔离级别中也是最慢的，因为它通常是通过完全锁来完成的。</p>	<p>1.数据</p> <p>2.允许</p> <p>3.避免</p> <p>4.避免</p> <p>5.避免</p> <p>脏读：</p> <p>不可重</p> <p>幻读：</p>
<p>中。如果当前没有事务正在发生，将抛出一个异常</p> <p>行中，则该方法应当运行在一个嵌套式事务中。被嵌套的事务可以独立于封装事务进行提交或回滚，就像PROPAGATION_REQUIRES一样。</p> <p>务中运行。如果一个事务正在进行，则会抛出一个异常。</p> <p>ED</p> <p>运行。如果一个现有事务正在进行中，它将在该方法的运行期间被挂起。</p> <p>文，但是如果有一个事务已经在运行的话，它也可以在这个事务里运行。</p> <p>V</p> <p>务里运行。一个新的事务将被启动，而且如果有一个现有事务在运行的话，则将在这个方法运行期间开始一个新事务。</p> <p>运行。如果一个现有事务正在进行中，该方法将在那个事务中运行，否则就要开始一个新事务。</p>	

数据库默认级别

读取未commit

脏读

不可重复读

奇幻读

读取未提交数据

复读：两次读取，数据内容不一致

两次读取，数据量不一致