

Java高级工程师面试总结

面试总结

Java基础

开源框架

缓存和NoSQL

分布式

数据结构和算法

网络编程

数据库

安全

设计模式

多线程

面试总结

最近两周面试了几家公司Java高级工程师的职位，主要有宜信、网信金融、阿里高德、口袋购物目前有部分公司已经面试通过，两家在等消息今天趁热把常见面试内容总结一下

Java基础

Hashtable和HashMap的区别

1. hashMap去掉了HashTable 的contains方法，但是加上了containsValue () 和 containsKey () 方法
2. hashtable同步的，而HashMap是非同步的，效率上比hashtable要高
3. hashMap允许空键值，而hashtable不允许

HashCode

1. hashCode的存在主要是用于查找的快捷性，如Hashtable，HashMap等，hashCode是用来在散列存储结构中确定对象的存储地址的；
2. 如果两个对象相同，就是适用于equals(java.lang.Object) 方法，那么这两个对象的hashCode一定要相同；
3. 如果对象的equals方法被重写，那么对象的hashCode也尽量重写，并且产生hashCode使用的对象，一定要和equals方法中使用的一致，否则就会违反上面提到的第2点；
4. 两个对象的hashCode相同，并不一定表示两个对象就相同，也就是不一定适用于equals(java.lang.Object) 方法，只能够说明这两个对象在散列存储结构中，如Hashtable，他们“存放在同一个篮子里”。

抽象类与接口的区别

抽象类里面可以有非抽象方法但接口里只能有抽象方法

一个类能继承多个接口

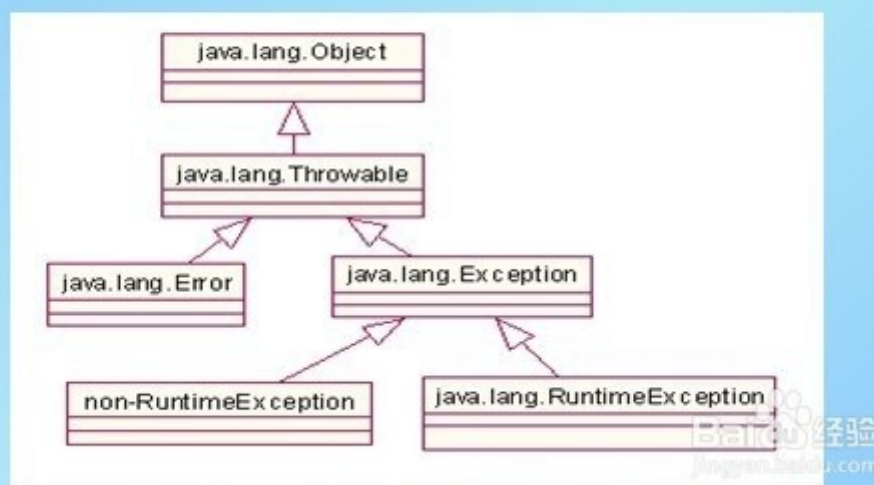
static final关键字的使用和区别

final类不能被继承，final变量为常量

static是全局静态，被static修饰的成员变量和成员方法独立于该类的任何对象

异常分类和处理机制

**Exception异常又分为RuntimeException和non-
RuntimeException两大类异常。**



JDK版本区别

StringBuilder内部实现机制

StringBuilder大小不够进行扩充容量，每次不需要new，而一直进行append操作。最终需要字符串的时候，再toString，效率高上百倍

反射机制的使用

程序在运行的时候能够获取自身的信息

匿名内部类的使用

泛型的概念和使用

泛型类是引用类型，是堆对象，主要是引入了类型参数这个概念

弱引用和虚引用的概念和使用方式

开源框架

SpringMVC和Struts2的区别

struts2是类级别的拦截，springmvc是方法级别的拦截
SpringMVC的入口是servlet，而Struts2是filter
SpringMVC集成了Ajax，只需注解@ResponseBody
Spring MVC和Spring是无缝的
SpringMVC开发效率和性能高于Struts2
SpringMVC可以认为已经100%零配置

SpringMVC实现原理

[SpringMVC运行原理.md](#)

Spring IOC和AOP的概念以及实现方式

IOC:依赖注入，和AOP:面向切面编程，这两个是Spring的灵魂。

Spring事务的管理

编程式、声明式

Hibernate与MyBatis的比较

Mybatis优势

MyBatis可以进行更为细致的SQL优化，可以减少查询字段。MyBatis容易掌握，而Hibernate门槛较高。

Hibernate优势

Hibernate的DAO层开发比MyBatis简单，Mybatis需要维护SQL和结果映射。

Hibernate对对象的维护和缓存要比MyBatis好，对增删改查的对象的维护要方便。

Hibernate数据库移植性很好，MyBatis的数据库移植性不好，不同的数据库需要写不同SQL。

Hibernate有更好的二级缓存机制，可以使用第三方缓存。MyBatis本身提供的缓存机制不佳。

Hibernate延迟加载的机制# JVM虚拟机

GC算法有哪些

垃圾回收器有哪些

如何调优JVM

缓存和NoSQL

缓存的使用场景

缓存命中率的计算

Memcache与Redis的比较

如何实现Redis的分片

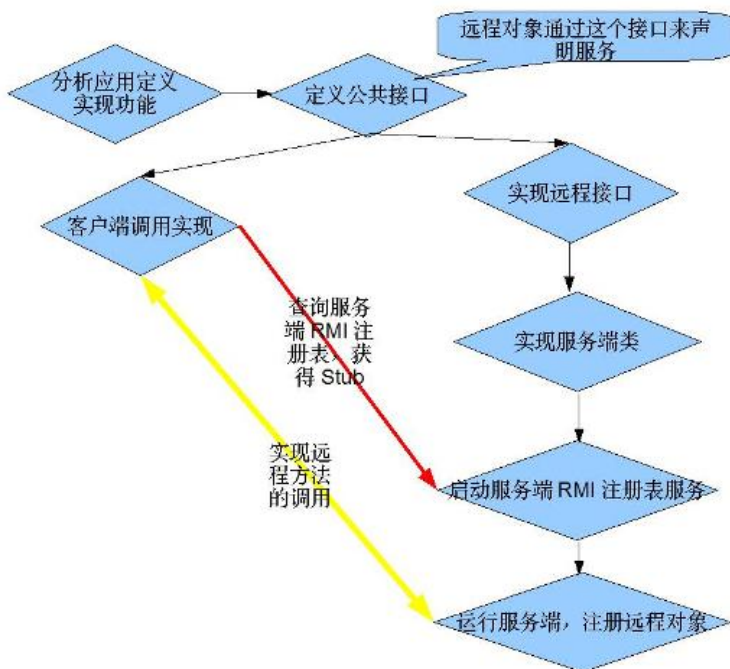
分布式

zookeeper的用途

dubbo的用途以及优点

dubbo的实现原理

RMI的实现原理



数据结构和算法

- 单向链表的逆序排列
- 双向链表的操作
- 1亿个整数的倒序输出
- 找出给定字符串中最长回文（回文：abcdcba，两端对称）

网络编程

- Get和Post的区别
- Https协议的实现
- 长连接的管理
- Socket的基本方法

数据库

- inner join和left join的区别

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

inner join(等值连接) 只返回两个表中联结字段相等的行

- 复杂SQL语句
- 数据库优化方式
- 数据库拆分方式
- 如何保证不同数据结构的数据一致性

安全

- 什么是XSS攻击，具体如何实现？
- 开放问题：如何保障系统安全？

设计模式

写出一个设计模式的类图

设计模式

设计模式的意义是什么

写个单例模式的代码

多线程

- 如何避免Quartz重复启动任务

`<property name="concurrent" value="false" />` 指定最终封装出的任务是否有状态

通过concurrent属性指定任务的类型，默认情况下封装为无状态的任务，如果希望目标封装为有状态的任务，仅需要将concurrent设置为false就可以了。Spring通过名为concurrent的属性指定任务的类型，能够更直接地描述到任务执行的方式（有状态的任务不能并发执行，无状态的任务可并发执行）

- 线程池满了如何处理额外的请求
- 同一个对象的连个同步方法能否被两个线程同时调用

答：不能，因为一个对象已经同步了实例方法，线程获取了对象的对象锁。所以只有执行完该方法释放对象锁后才能执行其它同步方法。

- 什么是死锁？

答：死锁就是两个或两个以上的线程被无限的阻塞，线程之间相互等待所需资源。这种情况可能发生在当两个线程尝试获取其它资源的锁，而每个线程又陷入无限等待其它资源锁的释放，除非一个用户进程被终止。

- sleep()、suspend()和wait()之间有什么区别？

答：thread.sleep()使当前线程在指定的时间处于“非运行”（not runnable）状态。线程一直持有对象的监视器。比如一个线程当前在一个同步块或同步方法中，其它线程不能进入该块或方法中。如果另一线程调用了interrupt()方法，它将唤醒那个“睡眠的”线程。

`t.suspend()`是过时的方法，使用`suspend()`导致线程进入停滞状态，该线程会一直持有对象的监视器，`suspend()`容易引起死锁问题。

`object.wait()`使当前线程处于“不可运行”状态，和`sleep()`不同的是`wait`是`object`的方法而不是`thread`。调用`object.wait()`时，线程先要获取这个对象的对象锁，当前线程必须在锁对象保持同步，把当前线程添加到等待队列中，随后另一线程可以同步同一个对象锁来调用`object.notify()`，这样将唤醒原来等待中的线程，然后释放该锁。基本上`wait()/notify()`与`sleep()/interrupt()`类似，只是前者需要获取对象锁。

- 如何让HashMap线程安全

1. 通过`Collections.synchronizedMap()`返回一个新的Map
2. 重新改写了HashMap,具体的可以查看
`java.util.concurrent.ConcurrentHashMap`

待更