1、垃圾回收机制

JVM内存模型，每个区域存储什么数据，每个区域的作用，问题入口从String a,new String(“”)开始

这里再说一下String类型的知识点：

String是不可变的，体现在：String类内部通过char数组来保存字符串，而这个char数组被声明为final。之所以这样做是为了考虑线程安全的同时降低消耗(与普通的并发机制相比)，这在设计模式中叫做“不变模式”。

请看这两个语句：String x = "abc"; String y = new String("abcd");

内存分配情况：

Java栈 常量池

x ------> "abc"

"abcd"

Java堆

y ------> "abcd"

String a = "a1";

String b = "a" + 1; //"a"是字符串字面量

System.out.println(a == b); //true 两个字符串字面量相连，得到的新的字符串仍然是字符串字面量

String a = "ab";

String bb = "b";

String b = "a" + bb;

System.out.println(a == b); //false 当字符串字面量与String类型变量连接时，得到的新字符串不再保留在常量池中，而是在堆中新建一个对象来存放

String a = "ab";

final String bb = "b";

String b = "a" + bb;

System.out.println(a == b); //true 字符串字面量与String类型常量相连得到的新字符串依然保留在常量池中。

通过函数return返回字符串字面量与String a = new String("ab")相同;

public String getBB(){

return "b";

}

当调用intern方法时，如果常量池中已经包含一个等于此String对象的字符串(用equals()方法确定)，则返回常量池中的字符串，否则将此字符串添加到池中，并返回此字符串对象的引用。

2、java中的关键字作用

private protected public static final 等

3、Object类里面有哪几种方法，作用

protected Object clone()创建并返回此对象的一个副本(clone()方法是浅拷贝，需要重写clone()方法来实现深拷贝，需要重写的类要实现克隆Cloneable接口)

boolean equals()

protected void finalize()

Class<?> getClass()

int hashCode()返回此对象的hash码值

void notify()唤醒在次对象监视器上等待的单个线程

void notifyAll()唤醒在此对象监视器上等待的所有线程

toString()

void wait()在其他线程调用此对象的notify()或者notifyAll()方法前，导致当前线程等待

void wait(long timeout)

void wait(long timeout, int nanos)

4、equals 和 hashCode方法，重写equals的原则()

原始的equals()方法用来比较两个对象的地址值

hashCode()用来获得对象地址的hash码值

重写equals必须重写hashCode()，自反性，对称性、传递性、一致性

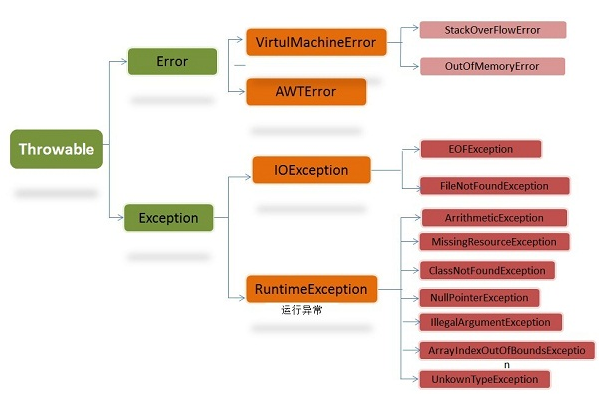
5、向上转型

6、Java引用类型(强引用，软引用，弱引用，虚引用)

7、线程相关的，主要是volitate，synchorized，wait()，notify()，notifyAll()，join()

8、Java异常机制

Java异常类层次结构图：



Error是程序无法处理的错误，表示运行应用程序中较严重的问题。表示代码运行时JVM中出现的问题。例如，Java虚拟机运行错误，如OutOfMemoryError。这些异常发生时一般JVM会选择线程终止。

Exception类有一个重要的子类RuntimeException。例如空指针异常、除数为零或数组下标越界，都属于运行时异常。

通常Java的异常分为可查的异常和不可查异常。不可查异常包括运行时异常和错误。其他的异常就是可查异常。

9、反射的用途

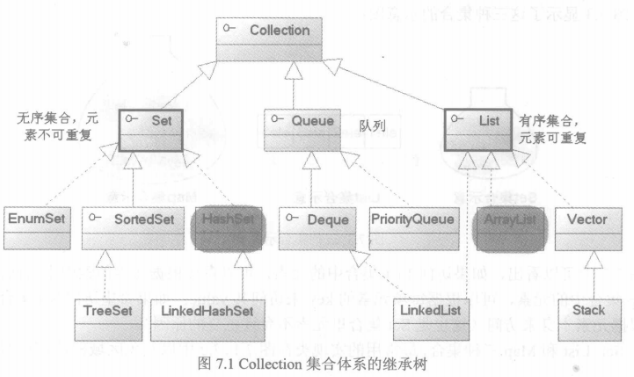
反射就是增加程序的灵活性，避免将程序写死到代码中，(例如，class.forName("com.lizan.Person").newInstance())一般常用在框架中。

10、HashMap实现原理(最常用的：数组+链表)，查找数据的时间复杂度为o(1)

11、List有哪些子类，各有什么区别

这里就引申为Java集合。

Collection集合体系继承树：



Collection是最基本的集合接口，Java不提供直接继承自Collection的类，Java API都是提供的继承自Collection的“子接口”如List和Set。

不论Collection的实际类型如何，它都支持一个iterator()的方法，该方法返回一个迭代子，使用该迭代子即可逐一访问集合中的每一个元素。

NIO相关，缓冲区、通道、selector。。。(不熟，面了这么多，挂在这里。其实主要是表现在同步阻塞和异步，传输方式不同。标准IO无法实现非阻塞模式、文件锁、读选择、分散聚集等)

内存泄露，举个例子

OOM是怎么出现的，有哪几块JVM区域会产生OOM，如何解决

Java里面的观察者模式实现

单例实现(我一般用enum写，不容易被挑毛病)

用Java模拟一个栈，并能够做到扩容，并且能有同步锁。（用数组实现）

Java泛型机制，泛型机制的优点，以及类型变量

Java集合框架源码

synchronized关键字原理

concurrent包

preparedstatement底层如何实现？

类之间有哪些关系？

Spring

GC太频繁如何处理？

设计模式

如何实现接口分离？

什么是虚拟内存？

动态代理的源码要看

进程和线程的区别

进程间通信机制

共享内存原理

TCP滑动窗口机制

电脑上访问一个网页的过程是什么样的？

拥塞避免/慢启动/快重传/快速恢复(所有公司都要问)

回退N帧协议

数据库范式

什么样的数据库设计是符合要求的(从范式的角度)

理论上说达到第三范式是符合要求的但是一般生产环境下为了数据查询方便，数据会有一定的冗余，也就是说一般达到第二范式即可。

第一范式：字段不可分

第二范式：非主属性必须完全函数依赖于码，即如果码的子集能唯一确定一个非主属性的话，那该表就不符合第二范式。

第三范式：消除了第二范式中的传递函数依赖

MySQL中的索引

连接池的原理是什么

连接池中的连接是长连接还是短连接？为什么？

连接池中的连接是基于什么协议的连接？为什么？

怎么建立TCP连接，怎么断开TCP连接？

为什么要三次握手，为什么要四次挥手？

剑指offer上的算法要理解并能自己写出来

项目中遇到的最大的困难是什么？

反转链表、冒泡排序、生产者消费者

========================J2SE基础=======================

作者：Xoper.ducky

链接：https://www.nowcoder.com/discuss/3043?type=2&order=0&pos=25&page=7

来源：牛客网

1.

九种基本数据类型的大小，以及他们的封装类。

2.

Switch能否用string做参数？

3.

equals与==的区别。

4.

Object有哪些公用方法？

5.

Java的四种引用，强弱软虚，用到的场景。

6.

Hashcode的作用。

7.

ArrayList、LinkedList、Vector的区别。

8.

String、StringBuffer与StringBuilder的区别。

9.

Map、Set、List、Queue、Stack的特点与用法。

10.

HashMap和HashTable的区别。

11.

HashMap和ConcurrentHashMap的区别，HashMap的底层源码。

12.

TreeMap、HashMap、LindedHashMap的区别。

13.

Collection包结构，与Collections的区别。

14.

try?catch?finally，try里有return，finally还执行么？

15.

Excption与Error包结构。OOM你遇到过哪些情况，SOF你遇到过哪些情况。

16.

Java面向对象的三个特征与含义。

17.

Override和Overload的含义去区别。

18.

Interface与abstract类的区别。

19.

Static?class?与non?static?class的区别。

20.

java多态的实现原理。

21.

实现多线程的两种方法：Thread与Runable。

22.

线程同步的方法：sychronized、lock、reentrantLock等。

23.

锁的等级：方法锁、对象锁、类锁。

24.

写出生产者消费者模式。

25.

ThreadLocal的设计理念与作用。

26.

ThreadPool用法与优势。

27.

Concurrent包里的其他东西：ArrayBlockingQueue、CountDownLatch等等。

28.

wait()和sleep()的区别。

29.

foreach与正常for循环效率对比。

30.

Java?IO与NIO。

31.

反射的作用于原理。

32.

泛型常用特点，List<String>能否转为List<Object>。

33.

解析XML的几种方式的原理与特点：DOM、SAX、PULL。

34.

Java与C++对比。

35.

Java1.7与1.8新特性。

36.

设计模式：单例、工厂、适配器、责任链、观察者等等。

37.

JNI的使用。

============================JVM=============================

作者：Xoper.ducky

链接：https://www.nowcoder.com/discuss/3043?type=2&order=0&pos=25&page=7

来源：牛客网

1.

内存模型以及分区，需要详细到每个区放什么。

2.

堆里面的分区：Eden，survival?from?to，老年代，各自的特点。

3.

对象创建方法，对象的内存分配，对象的访问定位。

4.

GC的两种判定方法：引用计数与引用链。

5.

GC的三种收集方法：标记清除、标记整理、复制算法的原理与特点，分别用在什么地方，如果让你优化收集方法，有什么思路？

6.

GC收集器有哪些？CMS收集器与G1收集器的特点。

7.

Minor?GC与Full?GC分别在什么时候发生？

8.

几种常用的内存调试工具：jmap、jstack、jconsole。

9.

类加载的五个过程：加载、验证、准备、解析、初始化。

10.

双亲委派模型：Bootstrap?ClassLoader、Extension?ClassLoader、ApplicationClassLoader。

11.

分派：静态分派与动态分派。

=========================操作系统=============================

作者：Xoper.ducky

链接：https://www.nowcoder.com/discuss/3043?type=2&order=0&pos=25&page=7

来源：牛客网

1.

进程和线程的区别。

2.

死锁的必要条件，怎么处理死锁。

3.

Window内存管理方式：段存储，页存储，段页存储。

4.

进程的几种状态。

5.

IPC几种通信方式。

6.

什么是虚拟内存。

7.

虚拟地址、逻辑地址、线性地址、物理地址的区别。

=========================TCP/IP===============================

作者：Xoper.ducky

链接：https://www.nowcoder.com/discuss/3043?type=2&order=0&pos=25&page=7

来源：牛客网

1.

OSI与TCP/IP各层的结构与功能，都有哪些协议。

2.

TCP与UDP的区别。

3.

TCP报文结构。

4.

TCP的三次握手与四次挥手过程，各个状态名称与含义，TIMEWAIT的作用。

5.

TCP拥塞控制。

6.

TCP滑动窗口与回退N针协议。

7.

Http的报文结构。

8.

Http的状态码含义。

9.

Http?request的几种类型。

10.

Http1.1和Http1.0的区别

11.

Http怎么处理长连接。

12.

Cookie与Session的作用于原理。

13.

电脑上访问一个网页，整个过程是怎么样的：DNS、HTTP、TCP、OSPF、IP、ARP。

14.

Ping的整个过程。ICMP报文是什么。

15.

C/S模式下使用socket通信，几个关键函数。

16.

IP地址分类。

17.

路由器与交换机区别。

网络其实大体分为两块，一个TCP协议，一个HTTP协议，只要把这两块以及相关协议搞清楚，一般问题不大。

=====================================数据结构与算法===================================

作者：Xoper.ducky

链接：https://www.nowcoder.com/discuss/3043?type=2&order=0&pos=25&page=7

来源：牛客网

1.

链表与数组。

2.

队列和栈，出栈与入栈。

3.

链表的删除、插入、反向。

4.

字符串操作。

5.

Hash表的hash函数，冲突解决方法有哪些。

6.

各种排序：冒泡、选择、插入、希尔、归并、快排、堆排、桶排、基数的原理、平均时间复杂度、最坏时间复杂度、空间复杂度、是否稳定。

7.

快排的partition函数与归并的Merge函数。

8.

对冒泡与快排的改进。

9.

二分查找，与变种二分查找。

10.

二叉树、B+树、AVL树、红黑树、哈夫曼树。

11.

二叉树的前中后续遍历：递归与非递归写法，层序遍历算法。

12.

图的BFS与DFS算法，最小生成树prim算法与最短路径Dijkstra算法。

13.

KMP算法。

14.

排列组合问题。

15.

动态规划、贪心算法、分治算法。（一般不会问到）

16.

大数据处理：类似10亿条数据找出最大的1000个数.........等等

算法面试80题

=========================有什么要问的？==============================

作者：Xoper.ducky

链接：https://www.nowcoder.com/discuss/3043?type=2&order=0&pos=25&page=7

来源：牛客网

1.

贵公司一向以XXX著称，能不能说明一下公司这方面的特点？

2.

贵公司XXX业务发展很好，这是公司发展的重点么？

3.

对技术和业务怎么看？

4.

贵公司一般的团队是多大，几个人负责一个产品或者业务？

5.

贵公司的开发中是否会使用到一些最新技术？

6.

对新人有没有什么培训，会不会安排导师？

7.

对Full?Stack怎么看？您觉得我们这些程序员有必要向这方面发展吗？

8.

你觉得我有哪些需要提高的地方？

9.

是否看好人工智能呢？

10.

您现在在研究什么技术？您比较喜欢哪一方面？