1. jdk jre区别

2. java 为什么跨平台

3. 对java面向对象的理解（对比一下C++）

4. 例举几个java中常用的类，例举几个常用的集合类（List array map： ArrayList， LinkedList，HashMap（HashMap实现，HashMap与HashTable区别，是不是任何类都可作为Map的key）； 什么时候会用Array，什么时候会用ArrayList）

5. java的多线程两种实现方式及区别（实现runnable接口或继承Thread类）

6. Java与erlang在处理并发方面的区别

7. Java 异常处理（try catch 或 throw）; exception 与 error 区别，他们都继承自Throwable类；

Exception：分为checked exception 和unchecked exception， 表示由程序员导致的错误，是在程序这一层被处理的；

Error： 是unchecked， 用来表示系统级或底层资源的错误，应该在系统级被处理；

例举几个Exception

8. 锁

synchronized 和 volatile关键字 （ReentrantLock可以提一下，不用细看）

什么时候用Sychronized，什么时候用volatile

Synchronized中wait notify notifyAll（wait和sleep的区别； notify和notifyAll的区别）

volatile的原理

9. Java的autoboxing 和unboxing

10. 线程池 Executor左右及举几个例子

11. Interface与abstract class区别；java不能多继承，但可以实现一个接口同时继承其他类；接口继承接口；

12. 速度：StringBuilder > StringBuffer > String

13. final， finally，finalize区别：

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

     finally是异常处理语句结构的一部分，表示总是执行；

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

14.内存泄露：无法释放已申请的内存（占着茅坑不拉屎）

     内存溢出：申请不到足够的内存（内存不够用了）

15.GC 机制及常用回收算法（大概了解几个名字就可以了，可以不用细看）

16. JavaIO： 字符流（Writer/Reader），字节流(InputStream/OutputStream)

17. 如何确保N个线程可以访问N个资源同时又不导致死锁？

指定获取锁的顺序，并强制线程按照指定的顺序获取锁

18. Fail-fast/ fail-safe

例如在遍历的过程中去增删元素；

安全失败是基于对底层集合做拷贝，因此，它不受源集合上修改的影响。java.util包下面的所有的集合类都是快速失败的，而java.util.concurrent包下面的所有的类都是安全失败的。快速失败的迭代器会抛出ConcurrentModificationException异常，而安全失败的迭代器永远不会抛出这样的异常

19. Big-O notation

大O符号描述了当数据结构里面的元素增加的时候，算法的规模或者是性能在最坏的场景下有多么好。

大O符号也可用来描述其他的行为，比如：内存消耗。因为集合类实际上是数据结构，我们一般使用大O符号基于时间，内存和性能来选择最好的实现。大O符号可以对大量数据的性能给出一个很好的说明

20. swtich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swtich

21.java 中常用的包例举几个：

lang, io, util, swing, net, applet, beans

22. 设计模式

单例模式：构造方法私有；

工厂模式：为创建对象定义一个接口，让子类决定初始化哪个类，工厂方法让一个类的初始化延迟至子类；

模板模式：公共操作放到一个抽象类里面；

23. 排序： 冒泡，快排

24. Java 测试框架：Junit， TestNG(爱立信BSP的测试Jcat case 用的TestNG )

25. Java Web相关的框架：Spring（IOC，AOP）（提一下名字就好，不用细看）

补充：

jvm内存分为 栈， 堆， 方法区， 程序计数器，本地方法栈

基本数据类型（如int double float等）和reference放到栈中；数组和实例数据放到堆中；加载的class信息放到方法区中；加载完一个方法后它的入口 本地变量等也放入栈中；