**基本概念**

* 操作系统中 heap 和 stack 的区别

专业回答:

1:空间分配的区别:

1、栈（操作系统）：由操作系统自动分配释放 ，存放函数的参数值，局部变量的值等。其操作方式类似于数据结构中的栈；   
　　2、堆（操作系统）： 一般由程序员分配释放， 若程序员不释放，程序结束时可能由OS回收，分配方式倒是类似于链表。

2:缓存方式的区别:

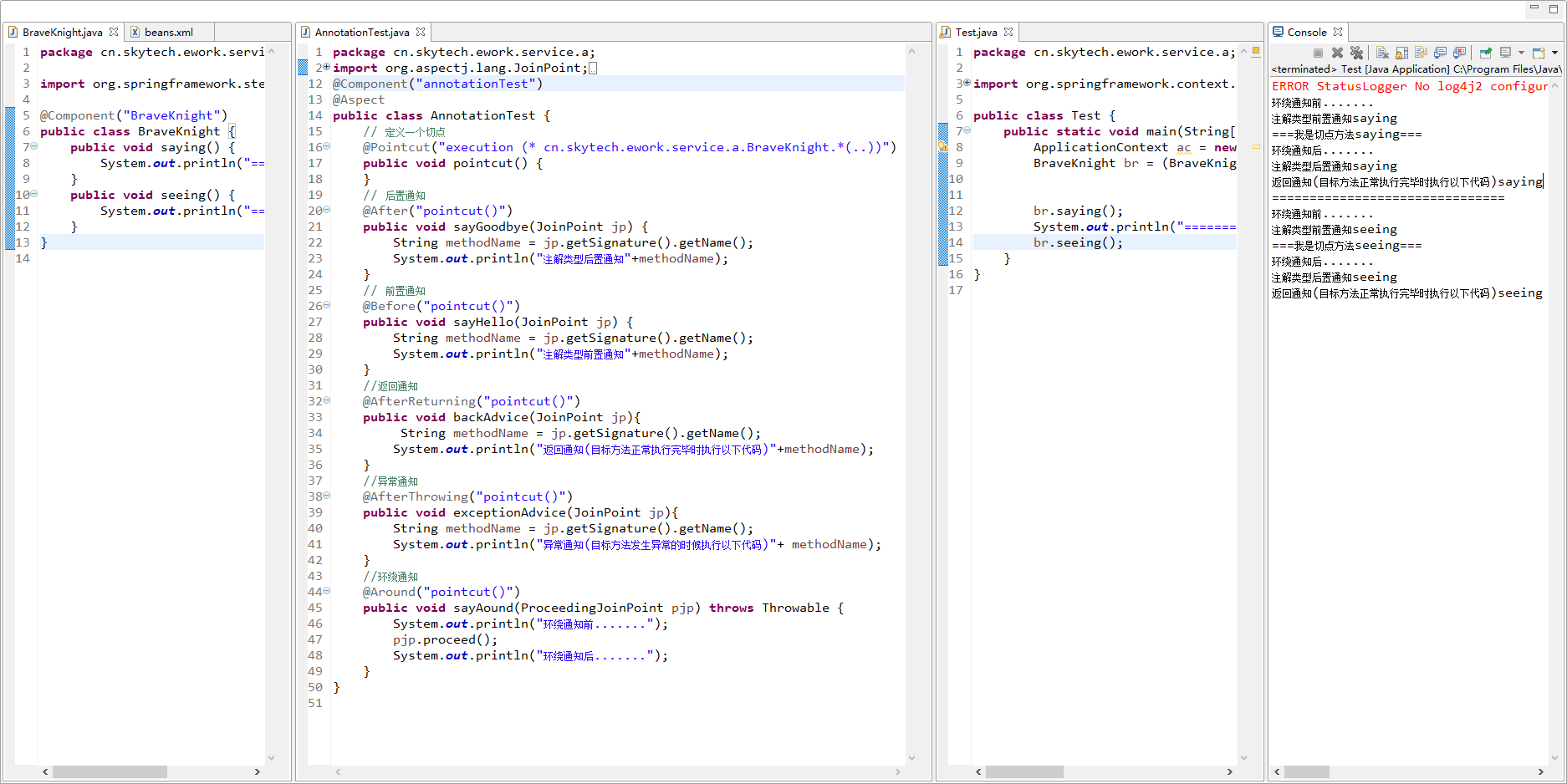
1、栈使用的是一级缓存， 他们通常都是被调用时处于存储空间中，调用完毕立即释放；   
　　2、堆是存放在二级缓存中，生命周期由虚拟机的垃圾回收算法来决定（并不是一旦成为孤儿对象就能被回收）。所以调用这些对象的速度要相对来得低一些。

3:数据结构的区别:

堆（数据结构）：堆可以被看成是一棵树，如：堆排序；   
　　栈（数据结构）：一种先进后出的数据结构。

堆:存放对象等大数据类型,容量大,先进先出

栈:存放基本数据类型,容量小,后进先出



* 什么是基于注解的切面实现

同一个类顺序：around之前通知》before通知》around之后通知》after通知》afterReturn通知

* 什么是 对象/关系 映射集成模块

对象关系映射，简单讲就是Java对象 Object 和关系型数据库 Relationship 之间的映射Mapping， 即ORM。

* 什么是 Java 的反射机制

反射机制是在运行状态中，对于任意一个类，都能够知道这个类的所有属性和方法；对于任意一个对象，都能够调用它的任意一个方法和属性；这种动态获取的信息以及动态调用对象的方法的功能称为java语言的反射机制。

<https://blog.csdn.net/u011514810/article/details/59712594>

* 什么是 ACID

Atomic（原子性）  
Consistency（一致性）  
Isolation（隔离性）  
Durability（持久性）

Atomic（原子性）：指整个数据库事务是不可分割的工作单位。只有使据库中所有的操作执行成功，才算整个事务成功；事务中任何一个SQL语句执行失败，那么已经执行成功的SQL语句也必须撤销，数据库状态应该退回到执行事务前的状态。

Consistency（一致性）：指数据库事务不能破坏关系数据的完整性以及业务逻辑上的一致性。例如对银行转帐事务，不管事务成功还是失败，应该保证事务结束后ACCOUNTS表中Tom和Jack的存款总额为2000元。

Isolation（隔离性）：指的是在并发环境中，当不同的事务同时操纵相同的数据时，每个事务都有各自的完整数据空间。

Durability（持久性）：指的是只要事务成功结束，它对数据库所做的更新就必须永久保存下来。即使发生系统崩溃，重新启动数据库系统后，数据库还能恢复到事务成功结束时的状态。

* BS与CS的联系与区别
* Cookie 和 Session的区别

      cookie数据保存在客户端，session数据保存在服务器端；

        cookie保存的数据类型只能是字符串，session可以保存除了资源以外的数据类型；

        cookie的大小有4K限制，session没有限制；

        cookie可以被浏览器禁用，session则不能；

        cookie不够安全，可以被拷贝进行cookie欺骗，考虑到安全性，重要信息应使用session

        session在访问量增多时，会增加服务器的负担，从服务器性能方面考虑，非重要信息若需要保留应使用cookie

        cookie不设置有效期的话，默认是浏览器关闭即失效，session服务器默认有效期1440秒，24分钟

* fail-fast 与 fail-safe 机制有什么区别
* get 和 post请求的区别

GET是用来从服务器上获取数据，而Post是用来向服务器上传递数据。

Get将表单中的数据按照variable=value的形式，添加到action所指的URL后面，并且两者使用“？”连接，而各个变量之间使用“&”连接；Post是将表单中的数据放在form的数据体中，按照变量和值相对应的方式，传递到action所指向URL；

对于get方式，服务器端用Request.QueryString获取变量的值，对于post方式，服务器端用Request.Form获取提交的数据。

Get是不安全的，在客户端，Get方式在通过URL提交数据，数据在URL中可以看到；POST方式，数据放置在HTML HEADER内提交。

GET方式提交的数据最多只能有1024字节，这主要是因为URL长度限制，而POST则没有此限制，可以传输大量的数据。

Get限制Form表单的数据集的值必须为ASCII字符，而Post支持整个字符集

Get是Form的默认方法。

Get产生一个数据包，Post产生两个TCP数据包。

* Interface 与 abstract 类的区别

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。

* IOC的优点是什么

1、定义：控制反转（Inversion of Control，英文缩写为IoC）把创建对象的权利交给框架,是框架的重要特征，并非面向对象编程的专用术语。它包括依赖注入（Dependency Injection，简称DI）和依赖查找（Dependency Lookup）。

2、IOC最大的好处是：

因为把对象生成放在了XML里定义，所以当我们需要换一个实现子类将会变成很简单（一般这样的对象都是实现于某种接口的），只要修改XML就可以了，这样我们甚至可以实现对象的热插拔（有点像USB接口和SCSI硬盘了）。

3、IOC最大的缺点是：

（1）生成一个对象的步骤变复杂了（事实上操作上还是挺简单的），对于不习惯这种方式的人，会觉得有些别扭和不直观。

（2）对象生成因为是使用反射编程，在效率上有些损耗。但相对于IoC提高的维护性和灵活性来说，这点损耗是微不足道的，除非某对象的生成对效率要求特别高。

（3）缺少IDE重构操作的支持，如果在Eclipse要对类改名，那么你还需要去XML文件里手工去改了，这似乎是所有XML方式的缺陷所在。

* IO 和 NIO的区别，NIO优点
* Java 8 / Java 7 为我们提供了什么新功能
* 什么是竞态条件？ 举个例子说明。
* 当两个线程竞争同一资源时，如果对资源的访问顺序敏感，就称存在竞态条件。
* 导致竞态条件发生的代码区称作临界区。
* 在临界区中使用适当的同步就可以避免竞态条件。
* 临界区实现方法有两种，一种是用synchronized，一种是用Lock显式锁实现。
* JRE、JDK、JVM 及 JIT 之间有什么不同
* MVC的各个部分都有那些技术来实现?如何实现?
* "Model" 代表的是应用的业务逻辑（通过JavaBean，EJB 组件实现），
* "View" 是应用的表示面（由 JSP 页面产生），
* "Controller"是提供应用的处理过程控制（一般是一个 Servlet），通过这种设计模型把应用逻辑，处理过程和显示逻辑分成不同的组件实现。这些组件可以进行交互和重用
* RPC 通信和 RMI 区别
* 什么是 Web Service（Web服务）
* JSWDL开发包的介绍。JAXP、JAXM的解释。SOAP、UDDI,WSDL解释。
* WEB容器主要有哪些功能? 并请列出一些常见的WEB容器名字。

Web容器给处于其中的应用程序组件（JSP，SERVLET）提供一个环境，使JSP,SERVLET直接跟容器中的[环境变量](https://www.baidu.com/s?wd=%E7%8E%AF%E5%A2%83%E5%8F%98%E9%87%8F&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao)交互，不必关注其它系统问题。主要有[WEB服务器](https://www.baidu.com/s?wd=WEB%E6%9C%8D%E5%8A%A1%E5%99%A8&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao)来实现。例如：TOMCAT,WEBLOGIC,WEBSPHERE等。该容器提供的接口严格遵守J2EE规范中的WEB APPLICATION 标准。我们把遵守以上标准的[WEB服务器](https://www.baidu.com/s?wd=WEB%E6%9C%8D%E5%8A%A1%E5%99%A8&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao)就叫做J2EE中的WEB容器。

* 一个”.java”源文件中是否可以包含多个类（不是内部类）？有什么限制
* 简单说说你了解的类加载器。是否实现过类加载器

<https://blog.csdn.net/gjanyanlig/article/details/6818655/>

* 解释一下什么叫AOP（面向切面编程）
* 请简述 Servlet 的生命周期及其相关的方法

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。

* 请简述一下 Ajax 的原理及实现步骤
* 简单描述Struts的主要功能
* 什么是 N 层架构
* 什么是CORBA？用途是什么
* 什么是Java虚拟机？为什么Java被称作是“平台无关的编程语言”
* 什么是正则表达式？用途是什么？哪个包使用正则表达式来实现模式匹配
* 什么是懒加载（Lazy Loading）

懒加载---即为延迟加载,顾名思义在需要的时候才加载,这样做效率会比较低,但是占用内存低,iOS设备内存资源有限,如果程序启动使用一次性加载的方式可能会耗尽内存,这时可以使用懒加载,先判断是否有,没有再去创建

懒加载的好处:不必将创建对象的代码全部写在viewDidLoad方法中，代码的可读性更强代码之间的独立性强,低耦合,节省了内存资源

通过重写get方法,实现懒加载

 苹果官方也推荐使用懒加载,在添加到window上时

- (UIView \*)view {  
     if (!view) {

   [self loadView];

   [self  viewDidLoad];  
  }  
}

* 什么是尾递归，为什么需要尾递归
* 什么是控制反转（Inversion of Control）与依赖注入（Dependency Injection）

**关键字**

**finalize**

* 什么是finalize()方法

当垃圾回收器将要释放无用对象的内存时，先调用该对象的finalize()方法。如果在程序终止之前垃圾回收器始终没有执行垃圾回收操作，那么垃圾回收器将始终不会调用无用对象的finalize()方法。在Java的Object祖先类中提供了protected类型的finalize()方法，因此任何Java类都可以覆盖finalize()方法，在这个方法中进行释放对象所占的相关资源的操作。

Java虚拟机的垃圾回收操作对程序完全是透明的，因此程序无法预料某个无用对象的finalize()方法何时被调用。另外，除非垃圾回收器认为程序需要额外的内存，否则它不会试图释放无用对象占用的内存。换句话说，以下情况是完全可能的：一个程序只占用了少量内存，没有造成严重的内存需求，于是垃圾回收器没有释放那些无用对象占用的内存，因此这些对象的finalize()方法还没有被调用，程序就终止了。

程序即使显式调用System.gc()或Runtime.gc()方法，也不能保证垃圾回收操作一定执行，因此不能保证无用对象的finalize()方法一定被调用。

1. finalize()方法什么时候被调用
2. 析构函数(finalization)的目的是什么
3. final 和 finalize 的区别
4. final

* final关键字有哪些用法

概念：由字面可以了解，final有最终态，无法改变的意思。  
使用目的：为了阻止改变与提高效率。  
高效原因：Java内嵌机制，final方法会在编译的过程中利用内嵌机制进行inline优化。  
inline优化是指：在编译的时候直接调用方法代码替换，也就是内嵌，而不是在运行时调用方法。  
inline需要在编译的时候就知道最后要用哪个方法。  
显然，非final是不行的。  
非final方法可能在子类中被重写，由于可能出现多态的情况，编译器在编译阶段  
并不能确定将来调用方法的对象的真正类型，也就无法确定到底调用哪个方法。）  
  
修饰对象：  
1、非抽象类，由于被final修饰的类是不能被继承的，而抽象类必须被继承才有意义。  
2、非抽象方法，由于被final修饰的方法是不能被重写的，而抽象方法必须被重写才有意义。  
3、变量或常量。  
  
注意：  
1、final不能修饰构造方法。  
2、父类的private方法是不能被子类重写的，因为private方法默认是final的。

-----------------------------------------------------------------------------------------------------------------------------------------------------------------

1. final关键字可以用于成员变量、本地变量、方法以及类。  
2. final成员变量必须在声明的时候初始化或者在构造器中初始化，否则就会报编译错误。  
3. 你不能够对final变量再次赋值。  
4. 本地变量必须在声明时赋值。  
5. 在匿名类中所有变量都必须是final变量。  
6. final方法不能被重写。  
7. final类不能被继承。  
8. final关键字不同于finally关键字，后者用于异常处理。  
9. final关键字容易与finalize()方法搞混，后者是在Object类中定义的方法，是在垃圾回收之前被JVM调用的方法。  
10. 接口中声明的所有变量本身是final的。  
11. final和abstract这两个关键字是反相关的，final类就不可能是abstract的。  
12. final方法在编译阶段绑定，称为静态绑定(static binding)。  
13. 没有在声明时初始化final变量的称为空白final变量(blank final variable)，它们必须在构造器中初始化，或者调用this()初始化。不这么做的话，编译器会报错“final变量(变量名)需要进行初始化”。  
14. 将类、方法、变量声明为final能够提高性能，这样JVM就有机会进行估计，然后优化。  
15. 按照Java代码惯例，final变量就是常量，而且通常常量名要大写：

1. final 与 static 关键字可以用于哪里？它们的作用是什么
2. final, finally, finalize的区别
3. final、finalize 和 finally 的不同之处？

在《Java编程思想》P86页有这样一段话：  
　　“static方法就是没有this的方法。在static方法内部不能调用非静态方法，反过来是可以的。而且可以在没有创建任何对象的前提下，仅仅通过类本身来调用static方法。这实际上正是static方法的主要用途。”  
  
这段话虽然只是说明了static方法的特殊之处，但是可以看出static关键字的基本作用，简而言之，一句话来描述就是：  
　　方便在没有创建对象的情况下来进行调用（方法/变量）。  
  
　　很显然，被static关键字修饰的方法或者变量不需要依赖于对象来进行访问，只要类被加载了，就可以通过类名去进行访问。  
　　static可以用来修饰类的成员方法、类的成员变量，另外可以编写static代码块来优化程序性能。

* 能否在运行时向 static final 类型的赋值

1. 使用final关键字修饰一个变量时，是引用不能变，还是引用的对象不能变
2. 一个类被声明为final类型，表示了什么意思
3. throws, throw, try, catch, finally分别代表什么意义

**Java 有几种修饰符？分别用来修饰什么**

**volatile**

**简而言之，volatile变量自身具有下列特性：  
• 可见性。对一个volatile变量的读，总是能看到（任意线程）对这个volatile变量最后的写入。  
• 原子性：对任意单个volatile变量的读/写具有原子性，但类似于volatile++这种复合操作不具有原子性。**

* volatile 修饰符的有过什么实践
* volatile 变量是什么？volatile 变量和 atomic 变量有什么不同
* volatile 类型变量提供什么保证？能使得一个非原子操作变成原子操作吗
* 能创建 volatile 数组吗？

**transient变量有什么特点**

**super什么时候使用**

**public static void 写成 static public void会怎样**

**说明一下public static void main(String args[])这段声明里每个关键字的作用**

**请说出作用域public, private, protected, 以及不写时的区别**

**sizeof 是Java 的关键字吗**

**static**

* static class 与 non static class的区别

|  |  |
| --- | --- |
| static class | non static class |
| 1、用static修饰的是内部类，此时这个内部类变为静态内部类；对测试有用；  2、内部静态类不需要有指向外部类的引用；  3、静态类只能访问外部类的静态成员，不能访问外部类的非静态成员； | 1、非静态内部类需要持有对外部类的引用；  2、非静态内部类能够访问外部类的静态和非静态成员；  3、一个非静态内部类不能脱离外部类实体被创建；  4、一个非静态内部类可以访问外部类的数据和方法； |

* static 关键字是什么意思？Java中是否可以覆盖(override)一个private或者是static的方法
* 静态类型有什么特点
* main() 方法为什么必须是静态的？能不能声明 main() 方法为非静态
* 是否可以从一个静态（static）方法内部发出对非静态（non-static）方法的调用
* 静态变量在什么时候加载？编译期还是运行期？静态代码块加载的时机呢
* 成员方法是否可以访问静态变量？为什么静态方法不能访问成员变量

原因：静态变量和静态方法在类被加载的时候就分配了内存空间，当非静态的方法调用他们的时候，他们已经有了内存空间，当然是可以调用的咯！

**switch**

* switch 语句中的表达式可以是什么类型数据
* 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
* while 循环和 do 循环有什么不同

**操作符**

* &操作符和&&操作符有什么区别?
* a = a + b 与 a += b 的区别？
* 逻辑操作符 (&,|,^)与条件操作符(&&,||)的区别
* 3\*0.1 == 0.3 将会返回什么？true 还是 false？
* float f=3.4; 是否正确？
* short s1 = 1; s1 = s1 + 1;有什么错?

**数据结构**

**基础类型(Primitives)**

* 基础类型(Primitives)与封装类型(Wrappers)的区别在哪里

一、传递方式不同

封装类是引用类型。

基本类型（原始数据类型）在传递参数时都是按值传递，而封装类型是按引用传递的(其实“引用也是按值传递的”，传递的是对象的地址)。由于包装类型都是不可变量，因此没有提供改变它值的方法，增加了对“按引用传递”的理解难度。

int是基本类型，直接存放数值；Integer是类，产生对象时用一个引用指向这个对象。

二、封装类可以有方法和属性

封装类可以有方法和属性，利用这些方法和属性来处理数据，如Integer.parseInt(Strings)。基本数据类型都是final修饰的，不能继承扩展新的类、新的方法。

三、默认值不同

基本类型跟封装类型的默认值是不一样的。如int i,i的预设为0；Integer j，j的预设为null,因为封装类产生的是对象，对象默认值为null。

四、存储位置

基本类型在内存中是存储在栈中，引用类型的引用（值的地址）存储在栈中，而实际的对象（值）是存在堆中。

虽然基本类型在栈上分配内存效率高，但是在堆栈上分配内存可能有内存泄漏的问题。

基本数据类型的好处就是速度快（不涉及到对象的构造和回收），封装类的目的主要是更好的处理数据之间的转换。

JDK5.0开始可以自动封包了，基本数据类型可以自动封装成封装类。 比如集合List，往里添加对象Object，需要将数字封装成封装类型对象，再存到List中。

1. List list=new ArreyList();
2. list.add(new Integer(1));

在JDK5.0 以后可以自动封包，简写成

1. List list=new ArrayList();
2. list.add(1);

* 简述九种基本数据类型的大小，以及他们的封装类
* int 和 Integer 哪个会占用更多的内存？ int 和 Integer 有什么区别？parseInt()函数在什么时候使用到
* float和double的默认值是多少
* 如何去小数四舍五入保留小数点后两位
* char 型变量中能不能存贮一个中文汉字，为什么

**类型转换**

* 怎样将 bytes 转换为 long 类型
* 怎么将 byte 转换为 String
* 如何将数值型字符转换为数字
* 我们能将 int 强制转换为 byte 类型的变量吗？如果该值大于 byte 类型的范围，将会出现什么现象
* 能在不进行强制转换的情况下将一个 double 值赋值给 long 类型的变量吗
* 类型向下转换是什么

**数组**

* 如何权衡是使用无序的数组还是有序的数组
* 怎么判断数组是 null 还是为空
* 怎么打印数组？ 怎样打印数组中的重复元素
* Array 和 ArrayList有什么区别？什么时候应该使用Array而不是ArrayList
* 数组和链表数据结构描述，各自的时间复杂度
* 数组有没有length()这个方法? String有没有length()这个方法

**队列**

* 队列和栈是什么，列出它们的区别
* BlockingQueue是什么
* 简述 ConcurrentLinkedQueue LinkedBlockingQueue 的用处和不同之处。

在Java多线程应用中，队列的使用率很高，多数生产消费模型的首选数据结构就是队列(先进先出)。Java提供的线程安全的Queue可以分为阻塞队列和非阻塞队列，其中阻塞队列的典型例子是BlockingQueue，非阻塞队列的典型例子是ConcurrentLinkedQueue，在实际应用中要根据实际需要选用阻塞队列或者非阻塞队列。

注：什么叫线程安全？这个首先要明确。线程安全就是说多线程访问同一代码，不会产生不确定的结果。

**并行和并发区别**

1、并行是指两者同时执行一件事，比如赛跑，两个人都在不停的往前跑；  
2、并发是指资源有限的情况下，两者交替轮流使用资源，比如一段路(单核CPU资源)同时只能过一个人，A走一段后，让给B，B用完继续给A ，交替使用，目的是提高效率

**LinkedBlockingQueue**  
**由于LinkedBlockingQueue实现是线程安全的，实现了先进先出等特性，是作为生产者消费者的首选**，LinkedBlockingQueue 可以指定容量，也可以不指定，不指定的话，默认最大是Integer.MAX\_VALUE，其中主要用到put和take方法，put方法在队列满的时候会阻塞直到有队列成员被消费，take方法在队列空的时候会阻塞，直到有队列成员被放进来。

ConcurrentLinkedQueue是Queue的一个安全实现．Queue中元素按FIFO原则进行排序．采用CAS操作，来保证元素的一致性。  
LinkedBlockingQueue是一个线程安全的阻塞队列，它实现了BlockingQueue接口，BlockingQueue接口继承自java.util.Queue接口，并在这个接口的基础上增加了take和put方法，这两个方法正是队列操作的阻塞版本。

**ArrayList、Vector、LinkedList的存储性能和特性**

**String**

**StringBuffer**

* ByteBuffer 与 StringBuffer有什么区别

**HashMap**

* HashMap的工作原理是什么
* 内部的数据结构是什么
* HashMap 的 table的容量如何确定？loadFactor 是什么？ 该容量如何变化？这种变化会带来什么问题？
* HashMap 实现的数据结构是什么？如何实现
* HashMap 和 HashTable、ConcurrentHashMap 的区别
* HashMap的遍历方式及效率
* HashMap、LinkedMap、TreeMap的区别
* 如何决定选用HashMap还是TreeMap
* 如果HashMap的大小超过了负载因子(load factor)定义的容量，怎么办
* HashMap 是线程安全的吗？并发下使用的 Map 是什么，它们内部原理分别是什么，比如存储方式、 hashcode、扩容、 默认容量等

**HashSet**

* HashSet和TreeSet有什么区别
* HashSet 内部是如何工作的
* WeakHashMap 是怎么工作的？

**Set**

* Set 里的元素是不能重复的，那么用什么方法来区分重复与否呢？是用 == 还是 equals()？ 它们有何区别?
* TreeMap：TreeMap 是采用什么树实现的？TreeMap、HashMap、LindedHashMap的区别。TreeMap和TreeSet在排序时如何比较元素？Collections工具类中的sort()方法如何比较元素？
* TreeSet：一个已经构建好的 TreeSet，怎么完成倒排序。
* EnumSet 是什么

**Hash算法**

* Hashcode 的作用
* 简述一致性 Hash 算法
* 有没有可能 两个不相等的对象有相同的 hashcode？当两个对象 hashcode 相同怎么办？如何获取值对象
* 为什么在重写 equals 方法的时候需要重写 hashCode 方法？equals与 hashCode 的异同点在哪里
* a.hashCode() 有什么用？与 a.equals(b) 有什么关系
* hashCode() 和 equals() 方法的重要性体现在什么地方
* Object：Object有哪些公用方法？Object类hashcode,equals 设计原则？ sun为什么这么设计？Object类的概述
* 如何在父类中为子类自动完成所有的 hashcode 和 equals 实现？这么做有何优劣。
* 可以在 hashcode() 中使用随机数字吗？

**LinkedHashMap**

* LinkedHashMap 和 PriorityQueue 的区别是什么

**List**

* List, Set, Map三个接口，存取元素时各有什么特点
* List, Set, Map 是否继承自 Collection 接口
* 遍历一个 List 有哪些不同的方式
* LinkedList

1. LinkedList 是单向链表还是双向链表
2. LinkedList 与 ArrayList 有什么区别
3. 描述下 Java 中集合（Collections），接口（Interfaces），实现（Implementations）的概念。LinkedList 与 ArrayList 的区别是什么？
4. 插入数据时，ArrayList, LinkedList, Vector谁速度较快？

* ArrayList

1. ArrayList 和 HashMap 的默认大小是多数
2. ArrayList 和 LinkedList 的区别，什么时候用 ArrayList？
3. ArrayList 和 Set 的区别？
4. ArrayList, LinkedList, Vector的区别
5. ArrayList是如何实现的，ArrayList 和 LinkedList 的区别
6. ArrayList如何实现扩容
7. Array 和 ArrayList 有何区别？什么时候更适合用Array
8. 说出ArraList,Vector, LinkedList的存储性能和特性

**Map**

* Map, Set, List, Queue, Stack
* Map 接口提供了哪些不同的集合视图
* 为什么 Map 接口不继承 Collection 接口

**Collections**

* 介绍Java中的Collection FrameWork。集合类框架的基本接口有哪些
* Collections类是什么？Collection 和 Collections的区别？Collection、Map的实现
* 集合类框架的最佳实践有哪些
* 为什么 Collection 不从 Cloneable 和 Serializable 接口继承
* 说出几点 Java 中使用 Collections 的最佳实践？
* Collections 中 遗留类 (HashTable、Vector) 和 现有类的区别

**什么是 B+树，B-树，列出实际的使用场景。**

**接口**

* Comparator 与 Comparable 接口是干什么的？列出它们的区别

**对象**

**拷贝(clone)**

* 如何实现对象克隆
* 深拷贝和浅拷贝区别
* 深拷贝和浅拷贝如何实现激活机制
* 写clone()方法时，通常都有一行代码，是什么

**比较**

* 在比较对象时，”==” 运算符和 equals 运算有何区别
* 如果要重写一个对象的equals方法，还要考虑什么
* 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

**构造器**

* 构造器链是什么
* 创建对象时构造器的调用顺序

**不可变对象**

* 什么是不可变象（immutable object）
* 为什么 Java 中的 String 是不可变的（Immutable）
* 如何构建不可变的类结构？关键点在哪里
* 能创建一个包含可变对象的不可变对象吗

**如何对一组对象进行排序**

**方法**

* 构造器（constructor）是否可被重写（override）
* 方法可以同时即是 static 又是 synchronized 的吗
* abstract 的 method是否可同时是 static，是否可同时是 native，是否可同时是synchronized
* Java支持哪种参数传递类型
* 一个对象被当作参数传递到一个方法，是值传递还是引用传递
* 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
* 我们能否重载main()方法
* 如果main方法被声明为private会怎样

**GC**

**概念**

* GC是什么？为什么要有GC
* 什么时候会导致垃圾回收
* GC是怎么样运行的
* 新老以及永久区是什么
* GC 有几种方式？怎么配置
* 什么时候一个对象会被GC？ 如何判断一个对象是否存活
* System.gc() Runtime.gc()会做什么事情？ 能保证 GC 执行吗
* 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？
* Minor GC 、Major GC、Young GC 与 Full GC分别在什么时候发生
* 垃圾回收算法的实现原理
* 如果对象的引用被置为null，垃圾收集器是否会立即释放对象占用的内存？
* 垃圾回收的最佳做法是什么

**GC收集器有哪些**

* 垃圾回收器的基本原理是什么？
* 串行(serial)收集器和吞吐量(throughput)收集器的区别是什么
* Serial 与 Parallel GC之间的不同之处
* CMS 收集器 与 G1 收集器的特点与区别
* CMS垃圾回收器的工作过程
* JVM 中一次完整的 GC 流程是怎样的？ 对象如何晋升到老年代
* 吞吐量优先和响应优先的垃圾收集器选择

**GC策略**

* 举个实际的场景，选择一个GC策略
* JVM的永久代中会发生垃圾回收吗

**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

**JVM**

**参数**

* 说说你知道的几种主要的jvm 参数
* -XX:+UseCompressedOops 有什么作用

**类加载器(ClassLoader)**

* Java 类加载器都有哪些
* JVM如何加载字节码文件

**内存管理**

* JVM内存分哪几个区，每个区的作用是什么
* 一个对象从创建到销毁都是怎么在这些部分里存活和转移的
* 解释内存中的栈(stack)、堆(heap)和方法区(method area)的用法
* JVM中哪个参数是用来控制线程的栈堆栈小
* 简述内存分配与回收策略
* 简述重排序，内存屏障，happen-before，主内存，工作内存
* Java中存在内存泄漏问题吗？请举例说明
* 简述 Java 中软引用（SoftReferenc）、弱引用（WeakReference）和虚引用
* 内存映射缓存区是什么

1. jstack，jstat，jmap，jconsole怎么用
2. 32 位 JVM 和 64 位 JVM 的最大堆内存分别是多数？32 位和 64 位的 JVM，int 类型变量的长度是多数？
3. 怎样通过 Java 程序来判断 JVM 是 32 位 还是 64 位
4. JVM自身会维护缓存吗？是不是在堆中进行对象分配，操作系统的堆还是JVM自己管理堆
5. 什么情况下会发生栈内存溢出
6. 双亲委派模型是什么

**多线程**

**基本概念**

* 什么是线程
* 多线程的优点
* 多线程的几种实现方式

1. 用 Runnable 还是 Thread

* 什么是线程安全

1. Vector, SimpleDateFormat 是线程安全类吗
2. 什么 Java 原型不是线程安全的
3. 哪些集合类是线程安全的

* 多线程中的忙循环是什么
* 如何创建一个线程
* 编写多线程程序有几种实现方式
* 什么是线程局部变量
* 线程和进程有什么区别？进程间如何通讯，线程间如何通讯
* 什么是多线程环境下的伪共享（false sharing）
* 同步和异步有何异同，在什么情况下分别使用他们？举例说明

**Current**

* ConcurrentHashMap 和 Hashtable的区别
* ArrayBlockingQueue, CountDownLatch的用法
* ConcurrentHashMap的并发度是什么

**CyclicBarrier 和 CountDownLatch有什么不同？各自的内部原理和用法是什么**

**Semaphore的用法**

**Thread**

* 启动一个线程是调用 run() 还是 start() 方法？start() 和 run() 方法有什么区别
* 调用start()方法时会执行run()方法，为什么不能直接调用run()方法
* sleep() 方法和对象的 wait() 方法都可以让线程暂停执行，它们有什么区别
* yield方法有什么作用？sleep() 方法和 yield() 方法有什么区别
* Java 中如何停止一个线程
* stop() 和 suspend() 方法为何不推荐使用
* 如何在两个线程间共享数据
* 如何强制启动一个线程
* 如何让正在运行的线程暂停一段时间
* 什么是线程组，为什么在Java中不推荐使用
* 你是如何调用 wait（方法的）？使用 if 块还是循环？为什么

**生命周期**

* 有哪些不同的线程生命周期
* 线程状态，BLOCKED 和 WAITING 有什么区别
* 画一个线程的生命周期状态图

**ThreadLocal 用途是什么，原理是什么，用的时候要注意什么**

**ThreadPool**

* 线程池是什么？为什么要使用它
* 如何创建一个Java线程池
* ThreadPool用法与优势
* 提交任务时，线程池队列已满时会发会生什么
* newCache 和 newFixed 有什么区别？简述原理。构造函数的各个参数的含义是什么，比如 coreSize, maxsize 等
* 线程池的实现策略
* 线程池的关闭方式有几种，各自的区别是什么
* 线程池中submit() 和 execute()方法有什么区别？

**线程调度**

* Java中用到的线程调度算法是什么
* 什么是多线程中的上下文切换
* 你对线程优先级的理解是什么
* 什么是线程调度器 (Thread Scheduler) 和时间分片 (Time Slicing)

**线程同步**

* 请说出你所知的线程同步的方法
* synchronized 的原理是什么
* synchronized 和 ReentrantLock 有什么不同
* 什么场景下可以使用 volatile 替换 synchronized
* 有T1，T2，T3三个线程，怎么确保它们按顺序执行？怎样保证T2在T1执行完后执行，T3在T2执行完后执行
* 同步块内的线程抛出异常会发生什么
* 当一个线程进入一个对象的 synchronized 方法A 之后，其它线程是否可进入此对象的 synchronized 方法B
* 使用 synchronized 修饰静态方法和非静态方法有什么区别
* 如何从给定集合那里创建一个 synchronized 的集合

**锁**

* Java Concurrency API 中 的 Lock 接口是什么？对比同步它有什么优势
* Lock 与 Synchronized 的区别？Lock 接口比 synchronized 块的优势是什么
* ReadWriteLock是什么？
* 锁机制有什么用
* 什么是乐观锁（Optimistic Locking）？如何实现乐观锁？如何避免ABA问题
* 解释以下名词：重排序，自旋锁，偏向锁，轻量级锁，可重入锁，公平锁，非公平锁，乐观锁，悲观锁
* 什么时候应该使用可重入锁
* 简述锁的等级方法锁、对象锁、类锁
* Java中活锁和死锁有什么区别？
* 什么是死锁(Deadlock)？导致线程死锁的原因？如何确保 N 个线程可以访问 N 个资源同时又不导致死锁
* 死锁与活锁的区别，死锁与饥饿的区别
* 怎么检测一个线程是否拥有锁
* 如何实现分布式锁
* 有哪些无锁数据结构，他们实现的原理是什么
* 读写锁可以用于什么应用场景

1. Executors类是什么？ Executor和Executors的区别
2. 什么是Java线程转储(Thread Dump)，如何得到它
3. 如何在Java中获取线程堆栈
4. 说出 3 条在 Java 中使用线程的最佳实践
5. 在线程中你怎么处理不可捕捉异常
6. 实际项目中使用多线程举例。你在多线程环境中遇到的常见的问题是什么？你是怎么解决它的
7. 请说出与线程同步以及线程调度相关的方法
8. 程序中有3个 socket，需要多少个线程来处理
9. 假如有一个第三方接口，有很多个线程去调用获取数据，现在规定每秒钟最多有 10 个线程同时调用它，如何做到
10. 如何在 Windows 和 Linux 上查找哪个线程使用的 CPU 时间最长
11. 如何确保 main() 方法所在的线程是 Java 程序最后结束的线程
12. 非常多个线程（可能是不同机器），相互之间需要等待协调才能完成某种工作，问怎么设计这种协调方案
13. 你需要实现一个高效的缓存，它允许多个用户读，但只允许一个用户写，以此来保持它的完整性，你会怎样去实现它

**异常**

**基本概念**

* Error 和 Exception有什么区别

1. UnsupportedOperationException是什么
2. NullPointerException 和 ArrayIndexOutOfBoundException 之间有什么相同之处

* 什么是受检查的异常，什么是运行时异常
* 运行时异常与一般异常有何异同
* 简述一个你最常见到的runtime exception(运行时异常)

**finally**

* finally关键词在异常处理中如何使用

1. 如果执行finally代码块之前方法返回了结果，或者JVM退出了，finally块中的代码还会执行吗
2. try里有return，finally还执行么？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
3. 在什么情况下，finally语句不会执行

**throw 和 throws 有什么区别？**

**OOM你遇到过哪些情况？你是怎么搞定的？**

**SOF你遇到过哪些情况？**

**既然我们可以用RuntimeException来处理错误，那么你认为为什么Java中还存在检查型异常**

**当自己创建异常类的时候应该注意什么**

**导致空指针异常的原因**

**异常处理 handle or declare 原则应该如何理解**

**怎么利用 JUnit 来测试一个方法的异常**

**catch块里别不写代码有什么问题**

**你曾经自定义实现过异常吗？怎么写的**

**什么是 异常链**

**在try块中可以抛出异常吗**

**JDBC**

* 通过 JDBC 连接数据库有哪几种方式
* 阐述 JDBC 操作数据库的基本步骤
* JDBC 中如何进行事务处理
* 什么是 JdbcTemplate
* 什么是 DAO 模块
* 使用 JDBC 操作数据库时，如何提升读取数据的性能？如何提升更新数据的性能
* 列出 5 个应该遵循的 JDBC 最佳实践

**IO**

**File**

* File类型中定义了什么方法来创建一级目录
* File类型中定义了什么方法来判断一个文件是否存在

**流**

* 为了提高读写性能，可以采用什么流
* Java中有几种类型的流
* JDK 为每种类型的流提供了一些抽象类以供继承，分别是哪些类
* 对文本文件操作用什么I/O流
* 对各种基本数据类型和String类型的读写，采用什么流
* 能指定字符编码的 I/O 流类型是什么

**序列化**

* 什么是序列化？如何实现 Java 序列化及注意事项
* Serializable 与 Externalizable 的区别

**Socket**

* socket 选项 TCP NO DELAY 是指什么
* Socket 工作在 TCP/IP 协议栈是哪一层
* TCP、UDP 区别及 Java 实现方式

**说几点 IO 的最佳实践**

**直接缓冲区与非直接缓冲器有什么区别？**

**怎么读写 ByteBuffer？ByteBuffer 中的字节序是什么**

**当用System.in.read(buffer)从键盘输入一行n个字符后，存储在缓冲区buffer中的字节数是多少**

**如何使用扫描器类（Scanner Class）令牌化**

**面向对象编程（OOP）**

* 解释下多态性（polymorphism），封装性（encapsulation），内聚（cohesion）以及耦合（coupling）
* 多态的实现原理
* 封装、继承和多态是什么
* 对象封装的原则是什么?
* 类

1. 获得一个类的类对象有哪些方式
2. 重载（Overload）和重写（Override）的区别。重载的方法能否根据返回类型进行区分？
3. 说出几条 Java 中方法重载的最佳实践

* 抽象类

1. 抽象类和接口的区别
2. 抽象类中是否可以有静态的main方法
3. 抽象类是否可实现(implements)接口
4. 抽象类是否可继承具体类(concrete class)

* 匿名类（Anonymous Inner Class）

1. 匿名内部类是否可以继承其它类？是否可以实现接口

* 内部类

1. 内部类分为几种
2. 内部类可以引用它的包含类（外部类）的成员吗
3. 请说一下 Java 中为什么要引入内部类？还有匿名内部类

* 继承

1. 继承（Inheritance）与聚合（Aggregation）的区别在哪里
2. 继承和组合之间有什么不同
3. 为什么类只能单继承，接口可以多继承
4. 存在两个类，B 继承 A，C 继承 B，能将 B 转换为 C 么？如 C = (C) B
5. 如果类 a 继承类 b，实现接口c，而类 b 和接口 c 中定义了同名变量，请问会出现什么问题

* 接口

1. 接口是什么
2. 接口是否可继承接口
3. 为什么要使用接口而不是直接使用具体类？接口有什么优点

**泛型**

* 泛型的存在是用来解决什么问题
* 泛型的常用特点
* List能否转为List

**工具类**

**日历**

* Calendar Class的用途
* 如何在Java中获取日历类的实例
* 解释一些日历类中的重要方法
* GregorianCalendar 类是什么
* SimpleTimeZone 类是什么
* Locale类是什么
* 如何格式化日期对象
* 如何添加小时(hour)到一个日期对象(Date Objects)
* 如何将字符串 YYYYMMDD 转换为日期

**Math**

* Math.round()什么作用？Math.round(11.5) 等于多少？Math.round(-11.5)等于多少？

**XML**

* XML文档定义有几种形式？它们之间有何本质区别？解析XML文档有哪几种方式？DOM 和 SAX 解析器有什么不同？
* Java解析XML的方式
* 用 jdom 解析 xml 文件时如何解决中文问题？如何解析
* 你在项目中用到了 XML 技术的哪些方面？如何实现

**动态代理**

* 描述动态代理的几种实现方式，分别说出相应的优缺点

**设计模式**

* 什么是设计模式（Design Patterns）？你用过哪种设计模式？用在什么场合
* 你知道哪些商业级设计模式？
* 哪些设计模式可以增加系统的可扩展性
* 单例模式

1. 除了单例模式，你在生产环境中还用过什么设计模式？
2. 写 Singleton 单例模式
3. 单例模式的双检锁是什么
4. 如何创建线程安全的 Singleton
5. 什么是类的单例模式
6. 写出三种单例模式实现

* 适配器模式

1. 适配器模式是什么？什么时候使用
2. 适配器模式和代理模式之前有什么不同
3. 适配器模式和装饰器模式有什么区别

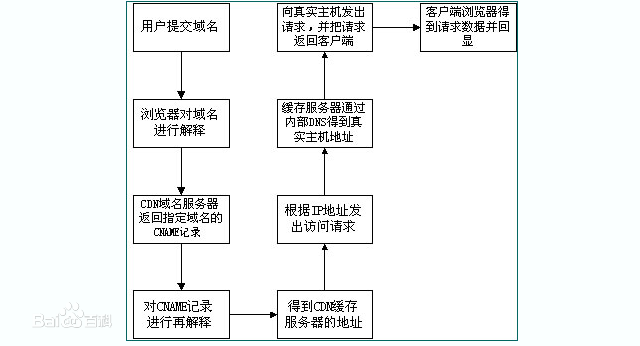
* 什么时候使用享元模式
* 什么时候使用组合模式
* 什么时候使用访问者模式
* 什么是模板方法模式
* 请给出1个符合开闭原则的设计模式的例子

**开放问题**

* 用一句话概括 Web 编程的特点
* Google是如何在一秒内把搜索结果返回给用户
* 哪种依赖注入方式你建议使用，构造器注入，还是 Setter方法注入
* 树（二叉或其他）形成许多普通数据结构的基础。请描述一些这样的数据结构以及何时可以使用它们
* 某一项功能如何设计
* 线上系统突然变得异常缓慢，你如何查找问题
* 什么样的项目不适合用框架
* 新浪微博是如何实现把微博推给订阅者
* 简要介绍下从浏览器输入 URL 开始到获取到请求界面之后 Java Web 应用中发生了什么
* 请你谈谈SSH整合
* 高并发下，如何做到安全的修改同一行数据
* 12306网站的订票系统如何实现，如何保证不会票不被超卖
* 网站性能优化如何优化的
* 聊了下曾经参与设计的服务器架构
* 请思考一个方案，实现分布式环境下的 countDownLatch
* 请思考一个方案，设计一个可以控制缓存总体大小的自动适应的本地缓存
* 在你的职业生涯中，算得上最困难的技术挑战是什么
* 如何写一篇设计文档，目录是什么
* 大写的O是什么？举几个例子
* 编程中自己都怎么考虑一些设计原则的，比如开闭原则，以及在工作中的应用
* 解释一下网络应用的模式及其特点
* 设计一个在线文档系统，文档可以被编辑，如何防止多人同时对同一份文档进行编辑更新
* 说出数据连接池的工作机制是什么
* 怎么获取一个文件中单词出现的最高频率
* 描述一下你最常用的编程风格
* 如果有机会重新设计你们的产品，你会怎么做
* 如何搭建一个高可用系统
* 如何启动时不需输入用户名与密码
* 如何在基于Java的Web项目中实现文件上传和下载
* 如何实现一个秒杀系统，保证只有几位用户能买到某件商品。
* 如何实现负载均衡，有哪些算法可以实现
* 如何设计一个购物车？想想淘宝的购物车如何实现的
* 如何设计一套高并发支付方案，架构如何设计
* 如何设计建立和保持 100w 的长连接
* 如何避免浏览器缓存。
* 如何防止缓存雪崩
* 如果AB两个系统互相依赖，如何解除依
* 如果有人恶意创建非法连接，怎么解决
* 如果有几十亿的白名单，每天白天需要高并发查询，晚上需要更新一次，如何设计这个功能
* 如果系统要使用超大整数（超过long长度范围），请你设计一个数据结构来存储这种超大型数字以及设计一种算法来实现超大整数加法运算）
* 如果要设计一个图形系统，请你设计基本的图形元件(Point,Line,Rectangle,Triangle)的简单实现
* 如果让你实现一个并发安全的链表，你会怎么做
* 应用服务器与WEB 服务器的区别？应用服务器怎么监控性能，各种方式的区别？你使用过的应用服务器优化技术有哪些
* 大型网站在架构上应当考虑哪些问题
* 有没有处理过线上问题？出现内存泄露，CPU利用率标高，应用无响应时如何处理的
* 最近看什么书，印象最深刻的是什么
* 描述下常用的重构技巧
* 你使用什么版本管理工具？分支（Branch）与标签（Tag）之间的区别在哪里
* 你有了解过存在哪些反模式（Anti-Patterns）吗
* 你用过的网站前端优化的技术有哪些
* 如何分析Thread dump
* 你如何理解AOP中的连接点（Joinpoint）、切点（Pointcut）、增强（Advice）、引介（Introduction）、织入（Weaving）、切面（Aspect）这些概念
* 你是如何处理内存泄露或者栈溢出问题的
* 你们线上应用的 JVM 参数有哪些
* 怎么提升系统的QPS和吞吐量

**知识面**

* 解释什么是 MESI 协议(缓存一致性)
* 谈谈 reactor 模型
* Java 9 带来了怎样的新功能
* Java 与 C++ 对比，C++ 或 Java 中的异常处理机制的简单原理和应用
* 简单讲讲 Tomcat 结构，以及其类加载器流程
* 虚拟内存是什么
* 阐述下 SOLID 原则
* 请简要讲一下你对测试驱动开发（TDD）的认识
* CDN实现原理
* 在描述CDN的实现原理，让我们先看传统的未加缓存服务的访问过程，以便了解CDN缓存访问方式与未加缓存访问方式的差别：
* 用户提交域名→浏览器对域名进行解释→得到目的主机的IP地址→根据IP地址访问发出请求→得到请求数据并回复
* 由上可见，用户访问未使用CDN缓存网站的过程为:
* 1)、用户向浏览器提供要访问的域名；
* 2)、浏览器调用域名解析函数库对域名进行解析，以得到此域名对应的IP地址；
* 3)、浏览器使用所得到的IP地址，向域名的服务主机发出数据访问请求；
* 4)、浏览器根据域名主机返回的数据显示网页的内容。
* 通过以上四个步骤，浏览器完成从用户处接收用户要访问的域名到从域名服务主机处获取数据的整个过程。CDN网络是在用户和服务器之间增加Cache层，如何将用户的请求引导到Cache上获得源服务器的数据，主要是通过接管DNS实现，下面让我们看看访问使用CDN缓存后的网站的过程：



通过上图，我们可以了解到，使用了CDN缓存后的网站的访问过程变为：

1)、用户向浏览器提供要访问的域名；

2)、浏览器调用域名解析库对域名进行解析，由于CDN对域名解析过程进行了调整，所以解析函数库一般得到的是该域名对应的CNAME记录，为了得到实际IP地址，浏览器需要再次对获得的CNAME域名进行解析以得到实际的IP地址；在此过程中，使用的全局负载均衡DNS解析，如根据地理位置信息解析对应的IP地址，使得用户能就近访问。

3)、此次解析得到CDN缓存服务器的IP地址，浏览器在得到实际的IP地址以后，向缓存服务器发出访问请求；

4)、缓存服务器根据浏览器提供的要访问的域名，通过Cache内部专用DNS解析得到此域名的实际IP地址，再由缓存服务器向此实际IP地址提交访问请求；

5)、缓存服务器从实际IP地址得得到内容以后，一方面在本地进行保存，以备以后使用，另一方面把获取的数据返回给客户端，完成数据服务过程；

6)、客户端得到由缓存服务器返回的数据以后显示出来并完成整个浏览的数据请求过程。

通过以上的分析我们可以得到，为了实现既要对普通用户透明(即加入缓存以后用户客户端无需进行任何设置，直接使用被加速网站原有的域名即可访问，又要在为指定的网站提供加速服务的同时降低对ICP的影响，只要修改整个访问过程中的域名解析部分，以实现透明的加速服务，下面是CDN网络实现的具体操作过程。

1)、作为ICP，只需要把域名解释权交给CDN运营商，其他方面不需要进行任何的修改；操作时，ICP修改自己域名的解析记录，一般用cname方式指向CDN网络Cache服务器的地址。

2)、作为CDN运营商，首先需要为ICP的域名提供公开的解析，为了实现sortlist，一般是把ICP的域名解释结果指向一个CNAME记录；

3)、当需要进行sortlist时，CDN运营商可以利用DNS对CNAME指向的域名解析过程进行特殊处理，使DNS服务器在接收到客户端请求时可以根据客户端的IP地址，返回相同域名的不同IP地址；

4)、由于从cname获得的IP地址，并且带有hostname信息，请求到达Cache之后，Cache必须知道源服务器的IP地址，所以在CDN运营商内部维护一个内部DNS服务器，用于解释用户所访问的域名的真实IP地址；

5)、在维护内部DNS服务器时，还需要维护一台授权服务器，控制哪些域名可以进行缓存，而哪些又不进行缓存，以免发生开放代理的情况。

* Maven 和 ANT 有什么区别
* UML中有哪些常用的图
* Linux

1. Linux 下 IO 模型有几种，各自的含义是什么。
2. Linux 系统下你关注过哪些内核参数，说说你知道的
3. Linux 下用一行命令查看文件的最后五行
4. 平时用到哪些 Linux 命令
5. 用一行命令输出正在运行的 Java 进程
6. 使用什么命令来确定是否有 Tomcat 实例运行在机器上

* 什么是 N+1 难题
* 什么是 paxos 算法
* 什么是 restful，讲讲你理解的 restful
* 什么是 zab 协议
* 什么是领域模型(domain model)？贫血模型(anaemic domain model) 和充血模型(rich domain model)有什么区别
* 什么是领域驱动开发（Domain Driven Development）
* 介绍一下了解的 Java 领域的 Web Service 框架
* Web Server、Web Container 与 Application Server 的区别是什么
* 微服务（MicroServices）与巨石型应用（Monolithic Applications）之间的区别在哪里
* 描述 Cookie 和 Session 的作用，区别和各自的应用范围，Session工作原理
* 你常用的持续集成（Continuous Integration）、静态代码分析（Static Code Analysis）工具有哪些
* 简述下数据库正则化（Normalizations）
* KISS,DRY,YAGNI 等原则是什么含义
* 分布式事务的原理，优缺点，如何使用分布式事务？
* 布式集群下如何做到唯一序列号
* 网络

1. HTTPS 的加密方式是什么，讲讲整个加密解密流程
2. HTTPS和HTTP的区别
3. HTTP连接池实现原理
4. HTTP集群方案
5. Nginx、lighttpd、Apache三大主流 Web服务器的区别

* 是否看过框架的一些代码
* 持久层设计要考虑的问题有哪些？你用过的持久层框架有哪些
* 数值提升是什么
* 你能解释一下里氏替换原则吗
* 你是如何测试一个应用的？知道哪些测试框架
* 传输层常见编程协议有哪些？并说出各自的特点

**编程题**

**计算加班费**

加班10小时以下加班费是时薪的1.5倍。加班10小时或以上，按4元/时算。提示：（一个月工作26天，一天正常工作8小时）

* 计算1000月薪，加班9小时的加班费
* 计算2500月薪，加班11小时的加班费
* 计算1000月薪，加班15小时的加班费

**卖东西**

一家商场有红苹果和青苹果出售。（红苹果5元/个，青苹果4元/个）。

* 模拟一个进货。红苹果跟青苹果各进200个。
* 模拟一个出售。红苹果跟青苹果各买出10个。每卖出一个苹果需要进行统计。

提示：一个苹果是一个单独的实体。

**日期提取**

有这样一个时间字符串：2008-8-8 20:08:08 ， 请编写能够匹配它的正则表达式，并编写Java代码将日期后面的时分秒提取出来，即：20:08:08

**线程**

* 8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。
* 用Java写一个多线程程序，如写四个线程，二个加1，二个对一个变量减一，输出
* wait-notify 写一段代码来解决生产者-消费者问题

**数字**

* 判断101-200之间有多少个素数，并输出所有素数
* 用最有效率的方法算出2乘以17等于多少
* 有 1 亿个数字，其中有 2 个是重复的，快速找到它，时间和空间要最优
* 2 亿个随机生成的无序整数,找出中间大小的值
* 10 亿个数字里里面找最小的 10 个
* 1到1亿的自然数，求所有数的拆分后的数字之和，如286 拆分成2、8、6，如1到11拆分后的数字之和 => 1 + … + 9 + 1 + 0 + 1 + 1
* 一个数如果恰好等于它的因子之和，这个数就称为 “完数 “。例如6=1＋2＋3.编程 找出1000以内的所有完数
* 一个数组中所有的元素都出现了三次，只有一个元素出现了一次找到这个元素
* 一球从100米高度自由落下，每次落地后反跳回原高度的一半；再落下，求它在 第10次落地时，共经过多少米？第10次反弹多高？
* 求100－1000内质数的和
* 求1到100的和的平均数
* 求s=a+a+aaa+aaaa+aa…a的值，其中a是一个数字。例如2+22+222+2222+22222(此时共有5个数相加)，几个数相加有键盘控制。 求出1到100的和
* 算出1到40的质数，放进数组里

1. 显示放组里的数
2. 找出第[5]个数
3. 删除第[9]个数，再显示删除后的第[9]个

* 有 3n+1 个数字，其中 3n 个中是重复的，只有 1 个是不重复的，怎么找出来。
* 有一组数1.1.2.3.5.8.13.21.34。写出程序随便输入一个数就能给出和前一组数字同规律的头5个数
* 计算指定数字的阶乘
* 开发 Fizz Buzz
* 给定一个包含 N 个整数的数组，找出丢失的整数
* 一个排好序的数组，找出两数之和为m的所有组合
* 将一个正整数分解质因数。例如：输入90,打印出90=2\*3\*3\*5。
* 打印出所有的 “水仙花数 “，所谓 “水仙花数 “是指一个三位数，其各位数字立方和等于该数本身。例如：153是一个 “水仙花数 “，因为153=1的三次方＋5的三次方＋3的三次方
* 原地交换两个变量的值
* 找出4字节整数的中位数
* 找到整数的平方根
* 实现斐波那契

**网络**

* 用Java Socket编程，读服务器几个字符，再写入本地显示

**反射**

* 反射机制提供了什么功能？
* 反射是如何实现的
* 哪里用到反射机制
* 反射中 Class.forName 和 ClassLoader 区别
* 反射创建类实例的三种方式是什么
* 如何通过反射调用对象的方法
* 如何通过反射获取和设置对象私有字段的值
* 反射机制的优缺点

**数据库**

* 写一段 JDBC 连Oracle的程序,并实现数据查询

**算法**

* 50个人围坐一圈，当数到三或者三的倍数出圈，问剩下的人是谁，原来的位置是多少
* 实现一个电梯模拟器用
* 写一个冒泡排序
* 写一个折半查找
* 随机产生20个不能重复的字符并排序
* 写一个函数，传入 2 个有序的整数数组，返回一个有序的整数数组
* 写一段代码在遍历 ArrayList 时移除一个元素
* 古典问题：有一对兔子，从出生后第3个月起每个月都生一对兔子，小兔子长到第四个月后每个月又生一对兔子，假如兔子都不死，问每个月的兔子总数为多少
* 约瑟芬环游戏

**正则**

* 请编写一段匹配IP地址的正则表达式
* 写出一个正则表达式来判断一个字符串是否是一个数字

**字符串**

* 写一个方法，入一个文件名和一个字符串，统计这个字符串在这个文件中出现的次数。
* 写一个程序找出所有字符串的组合，并检查它们是否是回文串
* 写一个字符串反转函数，输入abcde转换成edcba代码
* 小游戏，倒转句子中的单词
* 将GB2312编码的字符串转换为ISO-8859-1编码的字符串
* 请写一段代码来计算给定文本内字符“A”的个数。分别用迭代和递归两种方式
* 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”
* 给定 2 个包含单词列表（每行一个）的文件，编程列出交集
* 打印出一个字符串的所有排列
* 将一个键盘输入的数字转化成中文输出(例如：输入1234567，输出:一百二拾三万四千五百六拾七)
* 在Web应用开发过程中经常遇到输出某种编码的字符，如从 GBK 到 ISO8859-1等，如何输出一个某种编码的字符串

**日期**

* 计算两个日期之间的差距