**项目一组面试题集合**

1. **常用的设计模式有哪几种？**

Java中的23种设计模式：  
Factory（工厂模式）， Builder（建造模式）， Factory Method（工厂方法模式），  
Prototype（原始模型模式），Singleton（单例模式）， Facade（门面模式），  
Adapter（适配器模式）， Bridge（桥梁模式）， Composite（合成模式），  
Decorator（装饰模式）， Flyweight（享元模式）， Proxy（代理模式），  
Command（命令模式）， Interpreter（解释器模式）， Visitor（访问者模式），  
Iterator（迭代子模式）， Mediator（调停者模式）， Memento（备忘录模式），  
Observer（观察者模式）， State（状态模式）， Strategy（策略模式），  
Template Method（模板方法模式）， Chain Of Responsibleity（责任链模式）

1. **单例模式如何实现，它的作用是什么？**

a.单例模式是设计模式中最简单的形式之一，这一模式的目的是使得类的一个对象成为系统中的唯一实例。

b.能避免实例重复创建；

应用于避免存在多个实例引起程序逻辑错误的场合；

较节约内存

1. **hibernate 与mybatis有什么区别？**
2. hibernate完全封装了SQL，不再需要我们自己去编写SQL语句，对表的操作封装的更加彻底，可以让我们更加集中精力去处理业务流程。

mybatis则需要我们自己编写SQL语句，可以说是一个半自动化的持久化工具。

1. hibernate对数据库的切换，非常方便

mybatis因为需要自己编写SQL，而对于不同的数据库，SQL是有一些细节的差异的，所以要切换数据库相对麻烦。

1. hibernate学习曲线更陡峭

mybatis学习相对较为容易

1. hibernate因为不用编写SQL，所以对于优化要求更高。Hibernate优化方式，需要通过延迟加载，第三方缓存，更加合理的映射关系等。

mybatis因为直接编写SQL，所以优化可以更加地集中在SQL上面。

1. hibernate需要根据映射去生成SQL语句，所以会导致执行过多冗杂的SQL，同时会查询出过多无用的数据。

mybatis来说，不会执行无用的SQL，也最大限度的避免了查询出过多无用数据的问题。

1. hibernate对于基本的增，删，改，查操作时非常简洁，但是对于复杂关系的查询，或者大数据量查询，会存在相当的复杂度。

mybatis对于基本的增，删，改，查操作相对繁琐，但是对于复杂关系查询，和大数据量查询操作，更加容易去完成一些优化。

1. hibernate可以控制对象的不同状态来同步数据库

mybatis没有对象状态的做法

1. **你所了解的加密方式？**

网络专线加密：笼统来说，网络专线就是为某个机构拉一条独立的网线，也就是一个独立的局域网，例如军事，银行等，让用户的数据传输变得可靠可信，专线的优点就是安全性好，QoS可以得到保证。不过，专线租用价格也相对比较高，而且管理也需要专业人员。

秘钥加密方式：在私用[密钥](http://baike.baidu.com/view/934.htm)机制中，信息采用发送方和接收方保存的私有的密钥进行加密。这种系统假定双方已经通过一些人工方法交换了[密钥](http://baike.baidu.com/view/934.htm)，并且采用的密钥交换方式并不危及安全性。

公共[密钥](http://baike.baidu.com/view/934.htm)机制为每个用户产生两个相关的密钥。一个由用户私下保存（[私钥](http://baike.baidu.com/view/493846.htm)），另一个放于公共区（[公钥](http://baike.baidu.com/view/355291.htm)）。如果某人想给你发送消息，他（她）用你的[公开密钥](http://baike.baidu.com/view/1145160.htm)对信息加密。当收到信息后，你可以用私存的[密钥](http://baike.baidu.com/view/934.htm)对信息解密。

内容加密（数据本身进行加密）：MD5 spring Security, shiro

http://www.iteye.com/topic/1122076/

1. **spring 有什么作用?**
2. Spring能有效地组织你的中间层对象。
3. Spring能消除在许多工程中常见的对Singleton的过多使用。
4. Spring能消除各种各样自定义格式的属性文件的需要，使配置信息一元化。
5. Spring能够帮助我们真正意义上实现针对接口编程。
6. 在Spring应用中的大多数业务对象没有依赖于Spring。
7. 使用Spring构建的应用程序易于[单元测试](https://www.baidu.com/s?wd=%E5%8D%95%E5%85%83%E6%B5%8B%E8%AF%95&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YYuAcYmH99nWP9rj0srjFW0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnW0LrjRzPWDsnHbdrHRzrjDYr0)。
8. Spring支持JDBC和O/R Mapping产品(Hibernate)
9. MVC Web框架，提供一种清晰，无侵略性的MVC实现方式。
10. JNDI抽象层，便于改变实现细节，可以方便地在[远程服务](https://www.baidu.com/s?wd=%E8%BF%9C%E7%A8%8B%E6%9C%8D%E5%8A%A1&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YYuAcYmH99nWP9rj0srjFW0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnW0LrjRzPWDsnHbdrHRzrjDYr0)和本地服务间切换。
11. 简化访问数据库时的例外处理。
12. Spring能使用AOP提供声明性[事务管理](https://www.baidu.com/s?wd=%E4%BA%8B%E5%8A%A1%E7%AE%A1%E7%90%86&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YYuAcYmH99nWP9rj0srjFW0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnW0LrjRzPWDsnHbdrHRzrjDYr0)，可以不直接操作JTA也能够对事务进行管理。
13. 提供了JavaMail或其他邮件系统的支持。
14. **spring的工作流程（工作顺序）是什么样的？**
15. **SpringMVC的流程和原理？**

1.spring mvc将所有的请求都提交给DispatcherServlet,它会委托应用系统的其他模块负责对请求进行真正的处理工作。  
2.DispatcherServlet查询一个或多个HandlerMapping,找到处理请求的Controller.  
3.DispatcherServlet请请求提交到目标Controller  
4.Controller进行[业务逻辑](https://www.baidu.com/s?wd=%E4%B8%9A%E5%8A%A1%E9%80%BB%E8%BE%91&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YdPAwhmW9huHw9P1uWPHcL0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EPjDYPW01PH0s)处理后，会返回一个ModelAndView  
5.Dispathcher查询一个或多个ViewResolver视图解析器,找到ModelAndView对象指定的视图对象  
6.视图对象负责渲染返回给客户端。

1. **string 与stringbuffer有什么区别？**

Strin**g一旦赋值或实例化后就不可更改**，如果赋予新值将会重新开辟[内存地址](https://www.baidu.com/s?wd=%E5%86%85%E5%AD%98%E5%9C%B0%E5%9D%80&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YdPWn3P1IbuHNBmHmzuWfd0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EP1DYn1fsPjDkn1RYPHc3nHmd)进行存储。而StringBuffer类使用append和insert等方法改变字符串值时只是在原有对象存储的[内存地址](https://www.baidu.com/s?wd=%E5%86%85%E5%AD%98%E5%9C%B0%E5%9D%80&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YdPWn3P1IbuHNBmHmzuWfd0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EP1DYn1fsPjDkn1RYPHc3nHmd)上进行连续操作，减少了资源的开销。因此：当**需要进行频繁修改字符串的操作时先建立StringBuffer**[**类对象**](https://www.baidu.com/s?wd=%E7%B1%BB%E5%AF%B9%E8%B1%A1&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YdPWn3P1IbuHNBmHmzuWfd0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EP1DYn1fsPjDkn1RYPHc3nHmd)**进**行操作，将最后结果转化成[String类](https://www.baidu.com/s?wd=String%E7%B1%BB&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YdPWn3P1IbuHNBmHmzuWfd0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EP1DYn1fsPjDkn1RYPHc3nHmd)对象返回，这样效率会高很多。  
在操**作少量变动的数据时，使用String；在字符串内容或长度经常发生变化时，**使用StringBuffer

1. **ajax的作用是什么？涉及哪些参数？如何提交的？ajax的工作流程？**

Ajax最为主要的作用在于：实现页面和web服务器之间数据的异步传输。

ajax的工作流程：建立XMLHttpRequest对象，并在XMLHttpRequest上设置一个回调函数，使用XMLHttpRequest对象的open方法与服务器建立连接，并向服务器发异步请求，当服务器数据返回时，就可以在回调函数中用javascript来操作DOM而更新页面。

1. **java中有哪些集合？你所知道的集合类都有哪些？主要方法？**

主要有LinkedList，ArrayList，Vector等。下面是详细：

Collection

├List

│├LinkedList

│├ArrayList

│└Vector

│ └Stack

└Set

Map

├Hashtable

├HashMap

└WeakHashMap  
最常用的集合类是 List 和 Map。 List 的具体实现包括 ArrayList 和 Vector，它们是可变大小的列表，比较适合构建、存储和操作任何类型对象的元素列表。 List 适用于按数值索引访问元素的情形。 Map 提供了一个更通用的元素存储方法。 Map 集合类用于存储元素对（称作“键”和“值”）其中每个键映射到一个值。

1. **list 和 map 有哪些区别？**

List接口 继承[collection接口](https://www.baidu.com/s?wd=collection%E6%8E%A5%E5%8F%A3&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1Y3PyR3mW6vnHmLuW79PWnk0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EPjRznjDLPjck)，Map是个顶级接口  
List此接口的用户可以对列表中每个元素的插入位置进行精确地控制。  
用户可以根据元素的整数 索引（在列表中的位置）访问元素，并搜索列表中的元素  
map将键映射到值的对象。一个映射不能包含重复的键；每个键最多只能映射到一个值  
简单的说就是list只能保存一列表;Map<k,v>能保存2列表

1. **jquery选择器有哪些？**

很多种，大概归纳为9种。

(1)基本 #id element .class \* selector1,selector2,selectorN

(2)层次选择器： ancestor descendant parent > child prev + next prev ~ siblings

(3)基本[过滤器](https://www.baidu.com/s?wd=%E8%BF%87%E6%BB%A4%E5%99%A8&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1Yknhuhry7Bn1c1ryRsryRk0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnHRLP1cLrjDYrjRkn1fkn10vn0)选择器 :first :last :not :even :odd :eq :gt :lt :header :animated (4)内容[过滤器](https://www.baidu.com/s?wd=%E8%BF%87%E6%BB%A4%E5%99%A8&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1Yknhuhry7Bn1c1ryRsryRk0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnHRLP1cLrjDYrjRkn1fkn10vn0)选择器 :contains :empty :has :parent

(5)可见性[过滤器](https://www.baidu.com/s?wd=%E8%BF%87%E6%BB%A4%E5%99%A8&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1Yknhuhry7Bn1c1ryRsryRk0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnHRLP1cLrjDYrjRkn1fkn10vn0)选择器 :hidden :visible

(6)属性过滤器选择器 [attribute] [attribute=value] [attribute!=value] [attribute^=value] [attribute$=value] [attribute\*=value] [attrSel1][attrSel2][attrSelN]

(7)子元素过滤器选择器 :nth-child :first-child :last-child :only-child

(8) 表单选择器：input :text :password :radio :checkbox :submit :image :reset

:button :file :hidden

(9)表单过滤器选择器 :enabled :disabled :checked :selected

1. **行锁是什么？**

为在数据库里，同一个数据可能有多个人来读取或更改，为了防止我更改的时候别人也同时更改，从而保持数据的一致性从而进行的锁定操作。

行级锁定的优点：  
·         当在许多线程中访问不同的行时只存在少量锁定冲突。  
·         回滚时只有少量的更改。  
·         可以长时间锁定单一的行。

行级锁定的缺点：  
·         比页级或表级锁定占用更多的内存。  
·         当在表的大部分中使用时，比页级或表级锁定速度慢，因为你必须获取更多的锁。  
·         如果你在大部分数据上经常进行GROUP BY操作或者必须经常扫描整个表，比其它锁定明显慢很多。  
·         用高级别锁定，通过支持不同的类型锁定，你也可以很容易地调节应用程序，因为其锁成本小于行级锁定。

三种锁的特性可大致归纳如下：  
1） 表级锁：开销小，加锁快；不会出现死锁；锁定粒度大，发生锁冲突的概率最高，并发度最低。  
2） 行级锁：开销大，加锁慢；会出现死锁；锁定粒度最小，发生锁冲突的概率最低，并发度也最高。  
3） 页面锁：开销和加锁时间界于表锁和行锁之间；会出现死锁；锁定粒度界于表锁和行锁之间，并发度一般。

**14. mysql数据库索引如何创建？索引如何实现查询优化？**

http://www.cnblogs.com/hustcat/archive/2009/10/28/1591648.html

**15. 简述servlet流程的一个例子;**

1) 客户端在浏览器的地址栏中输入一个请求的URL,按回车后就向服务器端发起一个http request(由浏览器生成)。

2) 服务器端的Web Server首先接受到请求，并将请求转交给容器，容器会根据请求的URL去调用客户端要访问的Servlet。

   3) 容器会根据web.xml中对Servlet的描述去查找要访问的Servlet，若找到，将此Servlet

装载进虚拟机并实例化(第一次访问)，然后调用Servlet实例中的service方法处理请求，并分配一个线程去执行。

注: 当第二次去访问同一个Servlet时，若容器判断到该Servlet已经被装载并实例化，

那么容器就不会再去创建一个新的Servlet实例，直接调用原来那个Servlet实例中的service方法

来处理请求。

    4) 若没有查找到，直接返回一个404的错误代码到客户端，表示访问的资源不存在。

**16. 当有很多数据的情况下，怎么样减轻查询工作或者更快的实现查询性能?**

可以通过创建索引的方式;

可以在查询sql语句中只写自己所需的字段;

在SQL语句where关键字后筛选掉最多数据的条件放最右;

可以通过创建索引的方式；

**17. sql语句的优化有哪些？**

1.创建表的时候。应尽量建立主键,根据主键查询数据；  
2.大数据表删除，用truncate table代替delete。  
3.合理使用索引，在[OLTP](https://www.baidu.com/s?wd=OLTP&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YdPW9-mWu-P1fsPv7bnH0L0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EPjbdPH0vnHbYnWD3n1T4nW0Y)应用中一张表的索引不要太多。组合索引的列顺序尽量与查询条件列顺序保持一致；对于数据操作频繁的表，索引需要定期重建，以减少失效的索引和碎片。  
4.查询尽量用确定的列名，少用\*号。  
5.尽量少嵌套子查询，这种查询会消耗大量的[CPU](https://www.baidu.com/s?wd=CPU&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YdPW9-mWu-P1fsPv7bnH0L0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EPjbdPH0vnHbYnWD3n1T4nW0Y)资源；对于有比较多or运算的查询，建议分成多个查询，用union all联结起来；  
6.多表查询的查询语句中，选择最有效率的表名顺序(基于规则的优化器中有效)。[Oracle](https://www.baidu.com/s?wd=Oracle&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YdPW9-mWu-P1fsPv7bnH0L0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EPjbdPH0vnHbYnWD3n1T4nW0Y)解析器对表解析从右到左，所以记录少的表放在右边。  
7.尽量多用commit语句提交事务，可以及时释放资源、解锁、释放日志空间、减少管理花费；在频繁的、性能要求比较高的  
8.数据操作中，尽量避免远程访问，如数据库链等，访问频繁的表可以常驻内存：alter table．．．cache；

**18. 什么是一级缓存和二级缓存？了解第三方缓存？**

[**hibernate**](http://lib.csdn.net/base/javaee)的缓存包括Session的缓存和SessionFactory的缓存，其中SessionFactory的缓存又可以分为两类：内置缓存和外置缓存。Session的缓存是内置的，不能被卸载，也被称为Hibernate的第一级缓存。SessionFactory的内置缓存和Session的缓存在实现方式上比较相似，前者是SessionFactory对象的一些集合属性包含的数据，后者是指Session的一些集合属性包含的数据。SessionFactory的内置缓存中存放了映射元数据和预定义SQL语句，映射元数据是映射文件中数据的拷贝，而预定义SQL语句是在Hibernate初始化阶段根据映射元数据推导出来，SessionFactory的内置缓存是只读的，应用程序不能修改缓存中的映射元数据和预定义SQL语句，因此SessionFactory不需要进行内置缓存与映射文件的同步。SessionFactory的外置缓存是一个可配置的插件。在默认情况下，SessionFactory不会启用这个插件。外置缓存的数据是[**数据库**](http://lib.csdn.net/base/mysql)数据的拷贝，外置缓存的介质可以是内存或者硬盘。SessionFactory的外置缓存也被称为Hibernate的第二级缓存

**19. 100万个数如何快速取出最大的十个？**

是采用堆排序。  
具体做法是：  
构建一个只有10个元素的min－heap，那么根结点就是这10个数中最小的数，然后开始遍历数组，如果遇到的数比min－heap的根结点还小，直接跳过，遇到比min－heap根结点大的数，就替代根结点，然后对这个min－heap进行维护（也就是排序，保证heap的特征）。那么遍历完数组后，这个min－heap的10个元素就是最大的10个数。

100万平均分成10份，每份中取出最大的10个，得到100个数，然后从这100个数里面取出最大的10个，就是我要的

**20. 有没有用过闭包，干嘛用的？**

闭包就是定义在一个函数内部的函数，它的主要作用一是外部读取函数内部的变量，另一个就是让这些变量的值始终保持在内存中

什么是闭包：外部函数里面，再定义个内部函数，内部函数里面的变量值在外部是不能使用的。为了让外部能够使用内部函数的值或方法，在外部定义变量来保存，这样在外部就可以使用了。。。

好处:js面向对象过程中使用：缓存;封装内部函数的属性和方法;方便对象的使用;

参考如下代码：

|  |
| --- |
| 1. **function** Person(){ 2. **var** name = "default"; 4. **return** { 5. getName : **function**(){ 6. **return** name; 7. }, 8. setName : **function**(newName){ 9. name = newName; 10. } 11. } 12. };  15. **var** john = Person(); 16. print(john.getName()); 17. john.setName("john"); 18. print(john.getName()); 20. **var** jack = Person(); 21. print(jack.getName()); 22. jack.setName("jack"); 23. print(jack.getName()); |

**21. 数据的左连接右连接？**

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

**22.游标不关闭会出现什么后果**

游标不关闭，游标打开的表会缓存在PGA里，并且连接到系统的每个用户连接都独占了一个PGA，直到用户断开连接或者关闭游标才会释放游标所占的内存。

如果连接上来的用户多了，都使用了这游标，切不关闭，这累加起来的PGA会把系统内存吃光，最终拖垮系统。

用户少的话往往感觉不到系统性能的下降。

**23. 用SVN时，有两个文件夹一样你怎么处理?**

在使用SVN时，可能出现同名的情况是在服务器中有一个文件夹，而在我提交的文件中也有个同名的文件夹，最好的解决办法就是更改我自己的文件夹名，因为服务器端的文件名是不可随意更改的。所O以在使用SVN前最好先更新，再操作，避免出现类似的问题。

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*个人回答题（没固定答案）\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**1.如果给你个项目你会如何操作（项目思路）？**

**2.如果在项目中遇到问题，你会如何解决，花多长时间解决？**

**3.抛开技术，你最擅长哪一块？**

**4.你对加班这块怎么看？**

**5.假如有个项目，项目经理把项目计划已经准备好，分发一个块功能让你来实现，限定在2天内，你会怎么做？多久能做好？**

**6. 学过那些前端的框架？**

**7.自我介绍——性格——爱好——项目介绍**