1.你用过哪些关系或非关系型NoSQL数据库,如何看待数据库和数据仓库?(12分)

 第1层面：mysql、 sqlserver、oracle、db2、Sybase、access、mongodb、redis-内存数据库等)等（初步用过） - 初级

 第2层面：不同数据库的版本特征(有一定层面的认识) - 中级

 第3层面：对数据库及数据仓库又深刻系统的认识，有分区分表及数据库集群的经验，如mysql cluster（集群）或mysql读写分离、oracle rac等、分区分表如sharding-jdbc等方式.. - 高级

2. **什么是ORM？**(12分)

 第1层面：说出用过类型orm框架 - hibernate等框架 - 初级

 第2层面：概念层面理解比较透，对象关系映射（Object-Relational Mapping，简称ORM），对象和数据库之间映射的元数据（在Java中可以用XML或者是注解）- 中&高级

3. 抽象类和接口有什么区别？(12分)

 a.接口可以多重继承 ，抽象类不可以

 b.接口定义方法，不给实现；而抽象类可以实现部分方法

 c.接口中基本数据类型的数据成员，都默认为static和final，抽象类则不是

 初级 - 说出一个及相关

 中级 - 基本都说完整

 高级 - 除了所完整之外，还说了一些个人见解

4.请说一下StringBuffer和StringBuilder的区别？(12分)

 StringBuffer线程安全的，效率低

 StringBuilder线程非安全的，效率高

 初中级 - 线程安全

 高级 - 除外，和说了性能及原理

5. 设计模式三大类别，你所知道的设计模式？ (12分)

 创建模式、结构模式、行为模式

 创建模式：Factory工厂模式、Prototype享元模式、Builder创建者模式、Singleton单例模式

 结构模式：Facade门面模式、Proxy代理模式、Adapter适配器模式 、Composite组合模式、Decorator装饰者模式、Bridge桥梁模式、Flyweight享元模式

 行为模式：Template模板模式、Memento 备忘录模式、Observer观察者模式、Chain of Responsibility责任链模式、Command命令模式、State状态模式、Strategy策略模式、Mediator中介者模式、Interpreter解释器模式、Visitor 访问者模式

初级 - 能够涉及一些，不够系统，或多或少了解过

中级 - 概念基本掌握，对于原理及取舍不够自信

高级 - 对使用设计模式的机理有透彻了解，完全自信

5.Spring框架给java ee企业级开发带来了什么？ (16分)

非侵入式：支持基于POJO的编程模式，不强制性的要求实现Spring框架中的接口或继承Spring框架中的类

IoC容器：通过简单的形式（XML配置或注释）管理对象以及对象之间的依赖关系 而不需要修改代码，程序员再也不需要自己编写工厂、单例

AOP（面向切面编程）：将所有的横切关注功能封装到切面（aspect）中，通过配置的方式将横切关注功能动态添加到目标代码上，进一步实现了业务逻辑和系统服务之间的分离。省去或简化了区写系统的业务代理类实现

MVC：Spring MVC更加主流概念实现（Restful等）比struts mvc更加的优秀

事务管理：Spring以宽广的胸怀接纳多种持久层技术，并且为其提供了声明式的事务管理，在不需要任何一行代码的情况下就能够完成事务管理

其他：对于Spring无论从基础构件、系统业务解耦或安全角度无不涉及，可见Spring一站式框架的强大

初级 - 使用Spring 的框架，对于Ioc和AOP比较概念化

中级 - 系统的用过Spring的基本构件，没有更加完善和系统的认识

高级 - 对Spring的框架有系统的认识，或对Spring的构架原理有深层次的实践经验

6.你使用过的应用服务器优化技术有哪些？(16分)

① **分布式缓存**：缓存的本质就是内存中的哈希表，如果设计一个优质的哈希函数，那么理论上哈希表读写的渐近时间复杂度为O(1)。缓存主要用来存放那些读写比 很高、变化很少的数据，这样应用程序读取数据时先到缓存中读取，如果没有或者数据已经失效再去访问数据库或文件系统，并根据拟定的规则将数据写入缓存。对 网站数据的访问也符合二八定律（Pareto分布，幂律分布），即80%的访问都集中在20%的数据上，如果能够将这20%的数据缓存起来，那么系统的性 能将得到显著的改善。当然，使用缓存需要解决以下几个问题：
- 频繁修改的数据；
- 数据不一致与脏读；
- 缓存雪崩（可以采用分布式缓存服务器集群加以解决，memcached是广泛采用的解决方案）；
- 缓存预热；
- 缓存穿透（恶意持续请求不存在的数据）。

② **异步操作**：可以使用消息队列将调用异步化，通过异步处理将短时间高并发产生的事件消息存储在消息队列中，从而起到削峰作用。电商网站在进行促销活动时，可 以将用户的订单请求存入消息队列，这样可以抵御大量的并发订单请求对系统和数据库的冲击。目前，绝大多数的电商网站即便不进行促销活动，订单系统都采用了 消息队列来处理。

③ **使用集群**。 有状态行为Session同步及设计实现原理，数据库集群、LVS或F5原理 - 常用负载的均衡原理

④ **代码优化**：
- 多线程：基于Java的Web开发基本上都通过多线程的方式响应用户的并发请求，使用多线程技术在编程上要解决线程安全问题，主要可以考虑以下几个方 面：A. 将对象设计为无状态对象（这和面向对象的编程观点是矛盾的，在面向对象的世界中被视为不良设计），这样就不会存在并发访问时对象状态不一致的问题。B. 在方法内部创建对象，这样对象由进入方法的线程创建，不会出现多个线程访问同一对象的问题。使用ThreadLocal将对象与线程绑定也是很好的做法， 这一点在前面已经探讨过了。C. 对资源进行并发访问时应当使用合理的锁机制。
- 非阻塞I/O： 使用单线程和非阻塞I/O是目前公认的比多线程的方式更能充分发挥服务器性能的应用模式，基于Node.js构建的服务器就采用了这样的方式。Java在 JDK 1.4中就引入了NIO（Non-blocking I/O）,在Servlet 3规范中又引入了异步Servlet的概念，这些都为在服务器端采用非阻塞I/O提供了必要的基础。
- 资源复用：资源复用主要有两种方式，一是单例，二是对象池，我们使用的数据库连接池、线程池都是对象池化技术，这是典型的用空间换取时间的策略，另一方面也实现对资源的复用，从而避免了不必要的创建和释放资源所带来的开销。

初级 - 简单设计几个概念，不系统且不深刻

中级 - 有一两个比较完整的实践经验

高级 - 能深入浅出阐述，对大多数技术都有深刻的体会

7. 在java中怎么实现多线程方式? (10分)

a.继承Thread类实现多线程

b.实现Runnable接口方式

c.ExecutorService、Callable、Future

初级 - 只有线程概念或有过一些线程的实例

中级 - 上面的每一种基本都用过，但对线程池底层概念及数据状态同步不够了解

高级 - 有完整线程池设计经验，对数据状态同步处理 能有抽象的认识

8.对JVM内存结构模型认识？(10分)

JVM的内存结构组成：方法区、堆、本地方法栈、PC寄存器及方法栈

初级 - 了解简单的内存结构概念

中级 - 能很好的区分堆栈的，包括引发内存溢出的场景

高级 - 对实现jvm的底层内存结构模型组成，有深刻的见解，能系统说出个所以然