1.Spring 中 AOP 的应用场景、Aop 原理、好处？

答：AOP 用来封装横切关注点，具体可以在下面的场景中使用:

Authentication 权限、Caching 缓存、Context passing 内容传递、Error handling 错误处理 Lazy loading 懒加载、Debugging 调试、logging, tracing, profiling and monitoring 记录跟踪优化校准、Performance optimization 性能优化、Persistence 持久化、Resource pooling 资源池、Synchronization 同步、Transactions 事务

原理：AOP 是面向切面编程，是通过代理的方式为程序添加统一功能，集中解决一些公共问题。

好处：1.各个步骤之间的良好隔离性

2.源代码无关性

2.Spring 中 IOC 的作用与原理？对象创建的过程。

答：当某个角色需要另外一个角色协助的时候，在传统的程序设计过程中，通常由调用者来创建被调用者的实例。但在 spring 中创建被调用者的工作不再由调用者来完成，因此称为控制反转。创建被调用者的工作由 spring 来完成，然后注入调用者 。

IOC 本质上是一个容器，已 MAP 对 IOC 简单举例，服务器加载配置文件，由 xml 文档解析工具读取 bean 的 ID，获取 class，使用反射创建对象，以 K-V 的形式存入 MAP，K 是 ID，V 是反射创建的对象。获取对象可以调用 context.getBean（K）的方式。

3.介绍 spring 框架

它是一个 full-stack 框架，提供了从表现层到业务层再到持久层的一套完整的解决方案。我们在项目中可以只使用 spring 一个框架，它就可以提供表现层的 mvc 框架，持久层的 Dao 框架。它的两大核心 IoC 和 AOP 更是为我们程序解耦和代码简洁易维护提供了支持。

4.Spring 常见创建对象的注解？

答：@Component@Controller@ Service@ Repository

5.Spring 中用到的设计模式

答：简单工厂、工厂方法、单例模式、适配器、包装器、代理、观察者、策略、模板方法

详细介绍:http://www.cnblogs.com/yuefan/p/3763898.html

6.Spring 的优点？

答：1.降低了组件之间的耦合性，实现了软件各层之间的解耦

2.可以使用容易提供的众多服务，如事务管理，消息服务等

3.容器提供单例模式支持

4.容器提供了 AOP 技术，利用它很容易实现如权限拦截，运行期监控等功能

5.容器提供了众多的辅助类，能加快应用的开发

6.spring 对于主流的应用框架提供了集成支持，如 hibernate，JPA，Struts 等

7.spring 属于低侵入式设计，代码的污染极低

8.独立于各种应用服务器

9.spring 的 DI 机制降低了业务对象替换的复杂性

10.Spring 的高度开放性，并不强制应用完全依赖于 Spring，开发者可以自由选择 spring 的部分或全部

7.Spring Bean 的作用域之间有什么区别？

Spring 容器中的 bean 可以分为 5 个范围。所有范围的名称都是自说明的，但是为了避免混淆，还是让我们来解释一下：

singleton：这种 bean 范围是默认的，这种范围确保不管接受到多少个请求，每个容器中只有一个 bean 的实例，单例的模式由 bean factory 自身来维护。

prototype：原形范围与单例范围相反，为每一个 bean 请求提供一个实例。

request：在请求 bean 范围内会每一个来自客户端的网络请求创建一个实例，在请求完成以后，bean 会失效并被垃圾回收器回收。

Session：与请求范围类似，确保每个 session 中有一个 bean 的实例，在 session 过期后，bean 会随之失效。

global-session：global-session 和 Portlet 应用相关。当你的应用部署在 Portlet 容器中工作时，它包含很多 portlet。如果你想要声明让所有的 portlet 共用全局的存储变量的话，那么这全局变量需要存储在 global-session 中。

全局作用域与 Servlet 中的 session 作用域效果相同。

8.Spring 管理事务有几种方式？

答：有两种方式：

1、编程式事务，在代码中硬编码。(不推荐使用)

2、声明式事务，在配置文件中配置（推荐使用）

声明式事务又分为两种：

a、基于 XML 的声明式事务

b、基于注解的声明式事务

9.spring 中自动装配的方式有哪些？

答：1、 No：即不启用自动装配。

2、 byName：通过属性的名字的方式查找 JavaBean 依赖的对象并为其注入。比如说类 Computer 有个属性 printer，指定其 autowire 属性为 byName 后，Spring IoC 容器会在配置文件中查找 id/name 属性为 printer 的 bean，然后使用 Seter 方法为其注入。

3、 byType：通过属性的类型查找 JavaBean 依赖的对象并为其注入。比如类 Computer 有个属性 printer，类型为 Printer，那么，指定其 autowire 属性为 byType 后，Spring IoC 容器会查找 Class 属性为 Printer 的 bean，使用 Seter 方法为其注入。

4、 constructor：通 byType 一样，也是通过类型查找依赖对象。与 byType 的区别在于它不是使用 Seter 方法注入，而是使用构造子注入。

5、 autodetect：在 byType 和 constructor 之间自动的选择注入方式。

6、 default：由上级标签<beans>的 default-autowire 属性确定。

10.spring 中的核心类有那些，各有什么作用？

答：BeanFactory：产生一个新的实例，可以实现单例模式

BeanWrapper：提供统一的 get 及 set 方法

ApplicationContext:提供框架的实现，包括 BeanFactory 的所有功能

11.Bean 的调用方式有哪些？

答：有三种方式可以得到 Bean 并进行调用：

1、使用 BeanWrapper

```
HelloWorld hw=new HelloWorld();
BeanWrapper bw=new BeanWrapperImpl(hw);
bw.setPropertyvalue(” msg”,” HelloWorld” );
system.out.println(bw.getPropertyCalue(” msg” ));
```

2、使用 BeanFactory

```
InputStream is=new FileInputStream(” config.xml” );
XmlBeanFactory factory=new XmlBeanFactory(is);
HelloWorld hw=(HelloWorld) factory.getBean(” HelloWorld” );
system.out.println(hw.getMsg());
```

3、使用 ApplicationConttext

ApplicationContext actx=new FleSystemXmlApplicationContext(”config.xml”);
HelloWorld hw=(HelloWorld) actx.getBean(”HelloWorld”);
System.out.println(hw.getMsg());

12.什么是 IOC，什么又是 DI，他们有什么区别？

答：依赖注入 DI 是一个程序设计模式和架构模型， 一些时候也称作控制反转，尽管在技术上来讲，依赖注入是一个 IOC 的特殊实现，依赖注入是指一个对象应用另外一个对象来提供一个特殊的能力，例如：把一个 数据库连接已参数的形式传到一个对象的结构方法里面而不是在那个对象内部自行创建一个连接。控制反转和依赖注入的基本思想就是把类的依赖从类内部转化到外 部以减少依赖

应用控制反转，对象在被创建的时候，由一个调控系统内所有对象的外界实体，将其所依赖的对象的引用，传递给它。也可以说，依赖被注入到对象中。所 以，控制反转是，关于一个对象如何获取他所依赖的对象的引用，这个责任的反转。

13.spring 有两种代理方式：

答: 若目标对象实现了若干接口，spring 使用 JDK 的 java.lang.reflect.Proxy 类代理。

优点：因为有接口，所以使系统更加松耦合

缺点：为每一个目标类创建接口

若目标对象没有实现任何接口，spring 使用 CGLIB 库生成目标对象的子类。

优点：因为代理类与目标类是继承关系，所以不需要有接口的存在。

缺点：因为没有使用接口，所以系统的耦合性没有使用 JDK 的动态代理好。

14.springMVC 的流程？

答：1.用户发送请求至前端控制器 DispatcherServlet

2.DispatcherServlet 收到请求调用 HandlerMapping 处理器映射器。

3.处理器映射器根据请求 url 找到具体的处理器，生成处理器对象及处理器拦截器(如果有则生成)一并返回给 DispatcherServlet。

4.DispatcherServlet 通过 HandlerAdapter 处理器适配器调用处理器

5.执行处理器(Controller，也叫后端控制器)。

6.Controller 执行完成返回 ModelAndView

7.HandlerAdapter 将 controller 执行结果 ModelAndView 返回给 DispatcherServlet

8.DispatcherServlet 将 ModelAndView 传给 ViewReslover 视图解析器

9.ViewReslover 解析后返回具体 View

10.DispatcherServlet 对 View 进行渲染视图（即将模型数据填充至视图中）。

11.DispatcherServlet 响应用户

15.Springmvc 的优点

答：1.它是基于组件技术的.全部的应用对象,无论控制器和视图,还是业务对象之类的都是 java 组件.并且和 Spring 提供的其他基础结构紧密集成.

2.不依赖于 Servlet API(目标虽是如此,但是在实现的时候确实是依赖于 Servlet 的)

3. 可以任意使用各种视图技术,而不仅仅局限于 JSP

4 . 支持各种请求资源的映射策略

5 .它应是易于扩展的

16.Struts2 的流程？

答：1.客户端初始化一个指向 Servlet 容器（例如 Tomcat）的请求

2.这个请求经过一系列的过滤器（Filter）（这些过滤器中有一个叫做 ActionContextCleanUp 的可选过滤器，这个过滤器对于 Struts2 和其他框架的集成很有帮助，例如：SiteMesh Plugin）

3.接着 StrutsPrepareAndExecuteFilter 被调用，StrutsPrepareAndExecuteFilter 询问 ActionMapper 来解析和判断该次请求是否需要由 struts2 框架来处理.

4.如果 ActionMapper 判断需要 struts2 来处理请求，StrutsPrepareAndExecuteFilter 会把请求的处理交给 ActionProxy

5 .ActionProxy 通过 Configuration Manager 加载框架的配置文件，找到需要调用的 Action 以及拦截器配置信息

6 .ActionProxy 创建一个 ActionInvocation 的实例。

7 .ActionInvocation 实例使用命名模式来调用，在调用 Action 的过程前后，涉及到相关拦截器（Intercepter）的调用。

8.一旦 Action 执行完毕，ActionInvocation 负责根据 struts.xml 中的配置找到对应的返回结果配置。根据配置找到对应的 Result 处理类来处理结果集.大多数情况输出会交由模版语言(JSP,FreeMarker)完成输出内容拼装

17.struts2 中有哪些常用结果类型？

答：1) dispatcher ： Action 转发给 JSP

2) chain ： Action 转发到另一个 Action （同一次请求）

3） redirect ： Action 重定向到 JSP

4） redirectAction ： Action 重定向到另一个 Action

stream：下载用的（文件上传和下载时再议）

plainText：以纯文本的形式展现内容

18.struts2 的 Action 有几种编写方式？

答：第一种 创建类，这个类不继承任何类，不实现任何的接口

第二种 创建类，实现接口 Action 接口

第三种 创建类，继承类 ActionSupport 类,ActionSupport 类是 Action 接口的实现类

19.使用 struts2 如何实现多文件上传？

答：1 第一步 上传表单页面，满足三个要求，提交到 action 里面,要求：多个文件上传项 name 属性值必须要一样

2 创建 action，在 action 实现多文件的上传,在 action 中使用数组形式得到多个文件的息

```
private File[] uploadImages;//得到上传的文件
private String[] uploadImagesContentType;//得到文件的类型
private String[] uploadImagesFileName;//得到文件的名称
```

3 遍历数组，得到每一个文件的信息，一个一个上传到服务器中

```
if(uploadImages!=null&&uploadImages.length>0){
    for(int i=0;i<uploadImages.length;i++){
        File destFile = new File(realpath,uploadImageFileNames[i]);
        FileUtils.copyFile(uploadImages[i], destFile);
    }
}
```

20.springMVC 与 Struts2 的区别？

答: 1.springmvc 的入口是一个 servlet 即前端控制器，而 struts2 入口是一个 filter 过虑器。

2.springmvc 是基于方法开发(一个 url 对应一个方法)，请求参数传递到方法的形参，可以设计为单例或多例(建议单例)，struts2 是基于类开发，传递参数是通过类的属性，只能设计为多例。

3.Struts 采用值栈存储请求和响应的数据，通过 OGNL 存取数据， springmvc 通过参数

解析器是将 request 请求内容解析，并给方法形参赋值，将数据和视图封装成 ModelAndView 对象，最后又将 ModelAndView 中的模型数据通过 reques 域传输到页面。Jsp 视图解析器默认使用 jstl。

21.如何防止表单重复提交？表单重复提交的原因？

答：令牌机制。

原因：1.服务器处理服务后，转发页面，客户端点击刷新（重定向）

2.客户端网络过慢，按钮连续点击。

22.hibernate 原理？

答： hibernate，通过对 jdbc 进行封装，对 java 类和 关系数据库进行 mapping，实现了对关系数据库的面向对象方式的操作，改变了传统的 jdbc + sql 操作数据的方式，从而使开发人员可以话更多精力进行对象方面的开发

1.读取并解析配置文件

2.读取并解析映射信息，创建 SessionFactory

3.打开 Sesssion

4.创建事务 Transation

5.持久化操作

6.提交事务

7.关闭 Session

8.关闭 SesstionFactory

23.hibernate 的优点？

答：1.对 JDBC 访问数据库的代码做了封装，大大简化了数据访问层繁琐的重复性代码。

2.Hibernate 是一个基于 JDBC 的主流持久化框架，是一个优秀的 ORM 实现。他很大程度的简化 DAO 层的编码工作

3.hibernate 的性能非常好，因为它是个轻量级框架。映射的灵活性很出色。它支持各种关系数据库，从一对一到多对多的各种复杂关系。

24.什么是 Hibernate 延迟加载?

答：延迟加载机制是为了避免一些无谓的性能开销而提出来的，所谓延迟加载就是当在真正需要数据的时候，才真正执行数据加载操作。在 Hibernate 中提供了对实体对象的延迟加载以及对集合的延迟加载，另外在 Hibernate3 中还提供了对属性的延迟加载。

25.Hibernate 中类之间的关联关系有几种?

答：many-to-one、one-to-many、many-to-many、 one-to-one

26.说下 Hibernate 的缓存机制

答：A：hibernate 一级缓存

（1）hibernate 支持两个级别的缓存，默认只支持一级缓存；

（2）每个 Session 内部自带一个一级缓存；

（3）某个 Session 被关闭时，其对应的一级缓存自动清除；

B：hibernate 二级缓存

(1) 二级缓存独立于 session，默认不开启；

27.Hibernate 的查询方式

答：本地 SQL 查询、Criteria、Hql

28.如何优化 Hibernate?

答：1.使用双向一对多关联，不使用单向一对多

2.灵活使用单向一对多关联

3.不用一对一，用多对一取代

4.配置对象缓存，不使用集合缓存

5.一对多集合使用 Bag,多对多集合使用 Set

6．继承类使用显式多态

7．表字段要少，表关联不要怕多，有二级缓存撑腰

29.Hibernate 中一级缓存与二级缓存

答：Hibernate 缓存包括两大类：Hibernate 一级缓存和 Hibernate 二级缓存

Hibernate 一级缓存又称为“Session 的缓存”，它是内置的，不能被卸载（不能被卸载的意思就是这种缓存不具有可选性，必须有的功能，不可以取消 session 缓存）。由于 Session 对象的生命周期通常对应一个数据库事务或者一个应用事务，因此它的缓存是事务范围的缓存。第一级缓存是必需的，不允许而且事实上也无法卸除。在第一级缓存中，持久化类的每个实例都具有唯一的 OID。

Hibernate 二级缓存又称为“SessionFactory 的缓存”，由于 SessionFactory 对象的生命周期和应用程序的整个过程对应，因此 Hibernate 二级缓存是进程范围或者集群范围的缓存，有可能出现并发问题，因此需要采用适当的并发访问策略，该策略为被缓存的数据提供了事务隔离级别。第二级缓存是可选的，是一个可配置的插件，在默认情况下，SessionFactory 不会启用这个插件。需要程序员手动开启。

30.Hibernate 中关于对象的三种状态？

答：瞬时态（没有主键，new 出来的）、持久态（session 开启，有主键）、隔离态（session 关闭，有主键）。

31.Hibernate 二级缓存中存放数据的原则？适合放到二级缓存中的数据？

答：1 很少被修改的数据

2 不是很重要的数据，允许出现偶尔并发的数据

3 不会被并发访问的数据

4 常量数据

不适合存放到第二级缓存的数据？

1 经常被修改的数据

2 绝对不允许出现并发访问的数据，如财务数据，绝对不允许出现并发

3 与其他应用共享的数据

32.Hibernate 查找对象如何应用缓存？

答：当 Hibernate 根据 ID 访问数据对象的时候，首先从 Session 一级缓存中查；查不到，如果配置了二级缓存，那么从二级缓存中查；如果都查不到，再查询数据库，把结果按照 ID 放入到缓存删除、更新、增加数据的时候，同时更新缓存。

33.Hibernate 拒接连接、服务器崩溃的原因？

答：1．db 没有打开

2．网络连接可能出了问题

3．连接配置错了

4．驱动的 driver，url 是否都写对了

5．LIB 下加入相应驱动，数据连接代码是否有误

6．数据库配置可能有问题

7．当前连接太多了，服务器都有访问人数限制的

8．服务器的相应端口没有开，即它不提供相应的服务

34.过滤器和拦截器的区别

答：过滤器：在目标资源之前进行的操作

过滤所有的内容，比如 action、servlet、jsp、html

拦截器：在目标资源之前进行的操作

不能拦截所有的内容，拦截 action，不能拦截 jsp，不能拦截 html

拦截器和过滤器之间有很多相同之处，但是两者之间存在根本的差别。其主要区别为以下几点：

1）拦截器是基于 JAVA 反射机制的，而过滤器是基于函数回调的。

2）拦截器不依赖于 Servlet 容器，而过滤器依赖于 Servlet 容器

3）拦截器只能对 Action 请求起作用，而过滤器可以对几乎所有的请求起作用。

4）拦截器可以访问 Action 上下文、值栈里的对象，而过滤器不能

5）在 Action 的生命周期中，拦截器可以多次被调用，而过滤器只能在容器初始化时被调用一次。

35.Mybatis 的优劣势？

答：1．入门简单，即学即用，提供了数据库查询的自动对象绑定功能，而且延续了很好的 SQL 使用经验，对于没有那么高的对象模型要求的项目来说，相当完美。

2．可以进行更为细致的 SQL 优化，可以减少查询字段。

3．缺点就是框架还是比较简陋，功能尚有缺失，虽然简化了数据绑定代码，但是整个底层数据库查询实际还是要自己写的，工作量也比较大，而且不太容易适应快速数据库修改。

4．二级缓存机制不佳

36.Mybatis 中一级缓存与二级缓存区别？

答：一级缓存是 SqlSession 级别的缓存。在操作数据库时需要构造 sqlSession 对象，在对象中有一个(内存区域)数据结构（HashMap）用于存储缓存数据。不同的 sqlSession 之间的缓存数据区域（HashMap）是互相不影响的。

一级缓存的作用域是同一个 SqlSession，在同一个 sqlSession 中两次执行相同的 sql 语句，第一次执行完毕会将数据库中查询的数据写到缓存（内存），第二次会从缓存中获取数据将不再从数据库查询，从而提高查询效率。当一个 sqlSession 结束后该 sqlSession 中的一级缓存也就不存在了。Mybatis 默认开启一级缓存。

二级缓存是 mapper 级别的缓存，多个 SqlSession 去操作同一个 Mapper 的 sql 语句，多个 SqlSession 去操作数据库得到数据会存在二级缓存区域，多个 SqlSession 可以共用二级缓存，二级缓存是跨 SqlSession 的。

二级缓存是多个 SqlSession 共享的，其作用域是 mapper 的同一个 namespace，不同的 sqlSession 两次执行相同 namespace 下的 sql 语句且向 sql 中传递参数也相同即最终执行相同的 sql 语句，第一次执行完毕会将数据库中查询的数据写到缓存（内存），第二次会从缓存中获取数据将不再从数据库查询，从而提高查询效率。Mybatis 默认没有开启二级缓存需要在 setting 全局参数中配置开启二级缓存。

37.Ssh 整合流程及注意事项

答：原理：http://blog.csdn.net/u014010769/article/details/44993533

注意事项：http://blog.csdn.net/github_32658299/article/details/53469124

38.Ssm 整合流程及注意事项？

答：流程：http://blog.csdn.net/zhshulin/article/details/37956105/

注意事项：http://blog.csdn.net/github_32658299/article/details/53957585

39.ssm 的优缺点及使用场景。

答：1.Mybatis 和 hibernate 不同，它不完全是一个 ORM 框架，因为 MyBatis 需要程序员自己编写 Sql 语句，不过 mybatis 可以通过 XML 或注解方式灵活配置要运行的 sql 语句，并将 java 对象和 sql 语句映射生成最终执行的 sql，最后将 sql 执行的结果再映射生成 java 对象。

2.Mybatis 学习门槛低，简单易学，程序员直接编写原生态 sql，可严格控制 sql 执行性能，灵活度高，非常适合对关系数据模型要求不高的软件开发，例如互联网软件、企业运营类软件等，因为这类软件需求变化频繁，一但需求变化要求成果输出迅速。但是灵活的前提是 mybatis 无法做到数据库无关性，如果需要实现支持多种数据库的软件则需要自定义多套 sql 映射文件，工作量大。

3.Hibernate 对象/关系映射能力强，数据库无关性好，对于关系模型要求高的软件（例如需求固定的定制化软件）如果用 hibernate 开发可以节省很多代码，提高效率。但是 Hibernate 的学习门槛高，要精通门槛更高，而且怎么设计 O/R 映射，在性能和对象模型之间如何权衡，以及怎样用好 Hibernate 需要具有很强的经验和能力才行。

总之，按照用户的需求在有限的资源环境下只要能做出维护性、扩展性良好的软件架构都是好架构，所以框架只有适合才是最好。

40.Hibernate 和 mybatis 的区别？

答：Hibernate:hibernate 是一个标准的 ORM 框架，不需要写 sql 语句，维护关系比较复杂，sql 语句自动生成，对 sql 语句优化，修改比较困难。

Hibernate 的优缺点：

优点：面向对象开发，不需要自己写 sql 语句。如果进行数据库迁移不需要修改 sql 语句，只需要修改一下方言。

缺点：hibernate 维护数据表关系比较复杂。完全是有 hibernate 来管理数据表的关系，对于我们来说完全是透明的，不易维护。

Hibernate 自动生成 sql 语句，生成 sql 语句比较复杂，比较难挑错。

Hibernate 由于是面向对象开发，不能开发比较复杂的业务。

应用场景：

适合需求变化较少的项目，比如 ERP，CRM 等等

Mybatis 框架对 jdbc 框架进行封装，屏蔽了 jdbc 的缺点，开发简单。

Mybatis 只需要程序员关注 sql 本身，不需要过多的关注业务。对 sql 的优化，修改比较容易

适应场景：适合需求变化多端的项目，比如：互联网项目

41.用 spring 怎样实现单例？

答：http://blog.csdn.net/cs408/article/details/48982085

42.Spring 事务传播机制和数据库隔离级别

答：http://www.cnblogs.com/sxl525blogs/p/3674834.html

43.mybatis 如何处理批量插入

答：http://chenzhou123520.iteye.com/blog/1583407/

44.mybatis 插入一条数据如何返回主键

答：需要在 mybatis 的 mapper.xml 中指定 keyProperty 属性，示例如下：

```
40    <insert id="insertUserInfo" useGeneratedKeys="true" keyProperty="userId" parameterType="com.example.User">
41    insert into User(userName,password) values(#{userName},#{password})
42    </insert>
```

45.hibernate 的五个核心类

答：1，Configuration 接口：配置 Hibernate，根据其启动 Hibernate，创建 SessionFactory 对象；

2，SessionFactory 接口：初始化 Hibernate，充当数据存储源的代理，创建 session 对象，SessionFactory 是线程安全的，意味着它的同一个实例可以被应用的多个线程共享，是重量级二级缓存；

3，session 接口：负责保存、更新、删除、加载和查询对象，是一个非线程安全的，避免多个线程共享一个 session，是轻量级，一级缓存。

4，Transaction 接口：管理事务。可以对事务进行提交和回滚；

5，Query 和 Criteria 接口：执行数据库的查询。

46.spring 和 hibernate 管理事务有啥区别

答：1、从编码上说，hibernate 的事务管理是硬编码，是写在程序之中的。这就造成了，如果需要调整，就要修改源码，重新编译。

2、从事务控制的位置来说：hibernate 是持久层框架，事务是控制在持久层的，这样就造成了越权操作。事务应放在业务层，而非持久层

3、从代码维护上来说：hibernate 控制事务，需要在每个需要事务支持的地方编写代码，后期维护不便。