1、 多线程有几种实现方法?同步有几种实现方法?

多线程有两种实现方法，分别是继承Thread 类与实现Runnable 接口。同步的实现方面有两种，分别是synchronized,wait 与notify 。
a. wait():使一个线程处于等待状态，并且释放所持有的对象的lock。
b. sleep():使一个正在运行的线程处于睡眠状态，是一个静态方法，调用此方法要捕捉InterruptedException 异常。
c. notify():唤醒一个处于等待状态的线程，注意的是在调用此方法的时候，并不能确切的唤醒某一个等待状态的线程，而是由JVM 确定唤醒哪个线程，而且不是按优先级。
d. allnotity():唤醒所有处入等待状态的线程，注意并不是给所有唤醒线程一个对象的锁，而是让它们竞争。

2、 JDBC 中的 PreparedStatement 相比 Statement 的好处？

预编译语句java.sql.PreparedStatement ,扩展自Statement,不但具有Statement的所有能力而且具有更强大的功能。不同的是，PreparedStatement 是在创建语句对象的同时给出要执行的sql 语句。这样，sql 语句就会被系统进行预编译，执行的速度会有所增加，
尤其是在执行大语句的时候，效果更加理想

3、 Java 中实现多态的机制是什么?

重写，重载
方法的重写Overriding 和重载Overloading 是Java 多态性的不同表现。重写Overriding 是父类与子类之间多态性的一种表现，重载Overloading 是一个类中多态性的一种表现。如果在子类中定义某方法与其父类有相同的名称和参数，我们说该方法被重写(Overriding)。子类的对象使用这个方法时，将调用子类中的定义，对它而言，父类中的定义如同被“屏蔽”了。果在一个类中定义了多个同名的方法，它们或有不同的参数个数或有不同的参数类型，则称为方法的重载(Overloading)。Overloaded 的方法是可以改变返回值的类型。

4、 说出 ArrayList,Vector, LinkedList 的存储性能和特性

ArrayList 和 Vector 都是使用数组方式存储数据，此数组元素数大于实际存储的数据以便增加和插入元素，它们都允许直接按序号索引元素，但是插入元素要涉及数组元素移动等内存操作，所以索引数据快而插入数据慢，Vector 由于使用了 synchronized 方法（线程安全），通常性能上较 ArrayList 差，而 LinkedList 使用双向链表实现存储，按序号索引数据需要进行前向或后向遍历，但是插入数据时只需要记录本项的前后项即可，所以插入速度较快。

5、 Collection 和 Collections 的区别。

Collection 是集合类的上级接口，继承与他的接口主要有 Set 和 List。Collections 是针对集合类的一个帮助类，他提供一系列静态方法实现对各种集合的搜索、排序、线程安全化等操作。

6、 HashMap 和 Hashtable 的区别。

HashMap 是 Hashtable 的轻量级实现（非线程安全的实现），他们都完成了 Map 接口，主要区别在于 HashMap 允许空（null）键值（key）,由于非线程安全，效率上可能高于 Hashtable。
HashMap 允许将 null 作为一个 entry 的 key 或者 value，而 Hashtable 不允许。
HashMap 把 Hashtable 的 contains 方法去掉了，改成 containsvalue 和 containsKey。因为 contains 方法容易让人引起误解。 Hashtable 继承自 Dictionary 类，而 HashMap 是 Java1.2 引进的 Map interface 的一个实现。
最大的不同是，Hashtable 的方法是 Synchronize 的，而 HashMap 不是，在多个线程访问 Hashtable 时，不需要自己为它的方法实现同步，而 HashMap 就必须为之提供外同步。 Hashtable 和 HashMap 采用的 hash/rehash 算法都大概一样，所以性能不会有很大的差异。

7、 线程的基本概念、线程的基本状态以及状态之间的关系

一个程序中可以有多条执行线索同时执行，一个线程就是程序中的一条执行线索，每个线程上都关联有要执行的代码，即可以有多段程序代码同时运行，每个程序至少都有一个线程，即main 方法执行的那个线程。如果只是一个cpu，它怎么能够同时执行多段程序呢？这是从宏观上来看的，cpu 一会执行a 线索，一会执行b 线索，切换时间很快，给人的感觉是a,b 在同时执行，好比大家在同一个办公室上网，只有一条链接到外部网线，其实，这条网线一会为a 传数据，一会为b 传数据，由于切换时间很短暂，所以，大家感觉都在同时上网。状态：就绪，运行，synchronize 阻塞，wait 和sleep 挂起，结束。wait 必须在synchronized内部调用。调用线程的start 方法后线程进入就绪状态，线程调度系统将就绪状态的线程转为运行状态，遇到synchronized 语句时，由运行状态转为阻塞，当synchronized 获得锁后，由阻塞转为运行，在这种情况可以调用wait 方法转为挂起状态，当线程关联的代码执行完后，线程变为结束状态。

8、 abstract class 和 interface 有什么区别?

声明方法的存在而不去实现它的类叫抽象类。不能创建抽象类的实例；然而可以创建安一个变量，其类型是一个抽象类，并让他指向具体子类的一个实例。不能有抽象构造函数或抽象静态方法。
接口是抽象类的变体，接口中所有方法都是抽象的。多继承性可通过实现这样的

接口而获得。接口只可以定义 static final 成员变量

9、 String、StringBuffer、StringBuilder 的区别

String 是不可变的对象，每次对 String 类型进行改变的时候其实是产生了一个新的 String 对象，然后指针指向新的 String 对象；
StringBuffer 是线程安全的可变字符序列，需要同步，则使用。
StringBuilder 线程不安全，速度更快，单线程使用。
（String 是一个类，但却是不可变的，所以 String 创建的算是一个字符串常量，StringBuffer 和 StringBuilder 都是可变的。所以每次修改 String 对象的值都是新建一个对象再指向这个对象。而使用 StringBuffer 则是对 StringBuffer 对象本身进行操作。所以在字符串 j 经常改变的情况下，使用 StringBuffer 要快得多。）

10、 常见的 runtime exception 有哪些

ClassCastExcetion，NullPointerException，NumberFormatException，OutOfMemoryException，ArrayIndexOfBoundsException，ArithmeticException, ArrayStoreException, BufferOverflowException, BufferUnderflowException, CannotRedoException, CannotUndoException, ClassCastException, CMMException, ConcurrentModificationException, DOMException, EmptyStackException, IllegalArgumentException, IllegalMonitorStateException, IllegalPathStateException, IllegalStateException, ImagingOpException, IndexOutOfBoundsException, MissingResourceException, NegativeArraySizeException, NoSuchElementException, NullPointerException, ProfileDataException, ProviderException, RasterFormatException, SecurityException, SystemException, UndeclaredThrowableException, UnmodifiableSetException, UnsupportedOperationException
非运行时异常：IO异常、SQL异常、NoSuchMethod异常

11、 heap 和 stack 有什么区别

栈是一种线形集合，其添加和删除元素的操作应在同一段完成。栈按照后进先出的方式进行处理。堆是栈的一个组成元素

12、 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。【基础】

答：如果数据将在线程间共享。例如正在写的数据以后可能被另一个线程读到，或者正在读的数据可能已经被另一个线程写过了，那么这些数据就是共享数据，必须进行同步存取。当应用程序在对象上调用了一个需要花费很长时间来执

行的方法，并且不希望让程序等待方法的返回时，就应该使用异步编程，在很多情况下采用异步途径往往更有效率。

13、 java 中有几种方法可以实现一个线程？用什么关键字修饰同步方法?stop()和 suspend()方法为何不推荐使用？【中等难度】

答：有两种实现方法，分别是继承 Thread 类与实现 Runnable 接口；用 synchronized 关键字修饰同步方法；反对使用 stop()，是因为它不安全。它会解除由线程获取的所有锁定，而且如果对象处于一种不连贯状态，那么其他线程能在那种状态下检查和修改它们。结果很难检查出真正的问题所在；suspend()方法容易发生死锁。调用 suspend()的时候，目标线程会停下来，但却仍然持有在这之前获得的锁定。此时，其他任何线程都不能访问锁定的资源，除非被“挂起”的线程恢复运行。对任何线程来说，如果它们想恢复目标线程，同时又试图使用任何一个锁定的资源，就会造成死锁。故不应该使用 suspend()，而应在自己的 Thread 类中置入一个标志，指出线程应该活动还是挂起。若标志指出线程应该挂起，便用 wait()命其进入等待状态。若标志指出线程应当恢复，则用一个 notify()重新启动线程。

14、说说你所熟悉或听说过的 j2ee 中的几种常用模式?及对设计模式的一些看法。【中等难度】

答：Session Facade Pattern：使用 SessionBean 访问 EntityBean；

Message Facade Pattern：实现异步调用；

EJB Command Pattern：使用 Command JavaBeans 取代 SessionBean，实现轻量级访问；

Data Transfer Object Factory：通过 DTO Factory 简化 EntityBean 数据提供特性；

Generic Attribute Access：通过 AttibuteAccess 接口简化 EntityBean 数据提供特性；

Business Interface：通过远程（本地）接口和 Bean 类实现相同接口规范业务逻辑一致性；

EJB 架构的设计好坏将直接影响系统的性能、可扩展性、可维护性、组件可重用性及开发效率。项目越复杂，项目队伍越庞大则越能体现良好设计的重要性。

15、比如 100 用户同时来访，要采取什么技术解决？【基础】

答：可采用连接池。

① 装载数据库驱动程序；

② 通过 jdbc 建立数据库连接；

③ 访问数据库，执行 sql 语句；

④ 断开数据库连接。

16、forward 和 redirect 的区别？【基础】

答：forward 是容器中控制权的转向，是服务器请求资源，服务器直接访问目标地址的 URL，把那个 URL 的响应内容读取过来，然后把这些内容再发给浏览器，浏览器根本不知道服务器发送的内容是从哪儿来的，所以它的地址栏中还是原来的地址。redirect 就是服务端根据逻辑，发送一个状态码，告诉浏览器重新去请求那个地址，一般来说浏览器会用刚才请求的所有参数重新请求，所以 session，request 参数都可以获取，并且从浏览器的地址栏中可以看到跳转后的链接地址。前者更加高效，在前者可以满足需要时，尽量使用 forward()方法，并且，这样也有助于隐藏实际的链接；在有些情况下，比如，需要跳转到一个其它服务器上的资源，则必须使用 sendRedirect()方法。

17、说出数据连接池的工作机制是什么？【基础】

答：J2EE 服务器启动时会建立一定数量的池连接，并一直维持不少于此数目的池连接。客户端程序需要连接时，池驱动程序会返回一个未使用的池连接并将其表记为忙。如果当前没有空闲连接，池驱动程序就新建一定数量的连接，新建连接的数量有配置参数决定。当使用的池连接调用完成后，池驱动程序将此连接表记为空闲，其他调用就可以使用这个连接。

18、jsp 有哪些动作?作用分别是什么？【基础】

答：JSP 共有以下 6 种基本动作：

jsp:include：在页面被请求的时候引入一个文件；

jsp:useBean：寻找或者实例化一个 JavaBean。；

jsp:setProperty：设置 JavaBean 的属性。；

jsp:getProperty：输出某个 JavaBean 的属性；

jsp:forward：把请求转到一个新的页面；

jsp:plugin：根据浏览器类型为 Java 插件生成 OBJECT 或 EMBED 标记。

19、jsp 有哪些内置对象?作用分别是什么？【基础】

答：JSP 共有以下 9 种基本内置组件（可与 ASP 的 6 种内部组件相对应）：

request：用户端请求，此请求会包含来自 GET/POST 请求的参数；

response：网页传回用户端的回应；

pageContext：网页的属性是在这里管理；

session：与请求有关的会话期；

application：servlet 正在执行的内容；

out：用来传送回应的输出；

config：servlet 的构架部件；

page：JSP 网页本身；

exception：针对错误网页，未捕捉的例外。

20、get 和 post 的区别？【基础】

答：Form 中的 get 和 post 方法，在数据传输过程中分别对应了 HTTP 协议中的 GET 和 POST 方法。二者主要区别如下：

1）Get 是用来从服务器上获得数据，而 Post 是用来向服务器上传递数据；

2）Get 将表单中数据按照 variable=value 的形式，添加到 action 所指向的 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；

3）Get 是不安全的，因为在传输过程，数据被放在请求的 URL 中；Post 的所有操作对用户来说都是不可见的；

4）Get 传输的数据量小，这主要是因为受 URL 长度限制；而 Post 可以传输大量的数据，所以在上传文件只能使用 Post；

5）Get 限制 Form 表单的数据集必须为 ASCII 字符，而 Post 支持整个 ISO10646 字符集；

6）Get 是 Form 的默认方法

21、JSP 和 Servlet 有哪些相同点和不同点，他们之间的联系是什么？【基础】

答：JSP 是 Servlet 技术的扩展，本质上是 Servlet 的简易方式，更强调应用的外表表达。JSP 编译后是"类 servlet"。Servlet 和 JSP 最主要的不同点在于，Servlet 的应用逻辑是在 Java 文件中，并且完全从表示层中的 HTML 里分离开来。而 JSP 的情况是 Java 和 HTML 可以组合成一个扩展名为.jsp 的文件。JSP 侧重于视图，Servlet 主要用于控制逻辑。

22、jsp 的四种范围？【基础】

答：a.page 是代表与一个页面相关的对象和属性。一个页面由一个编译好的 Java servlet 类（可以带有

任何的 indude 指令，但是没有 indude 动作）表示。这既包括 servlet 又包括被编译成 servlet 的 JSP 页面 b.request 是代表与 Web 客户机发出的一个请求相关的对象和属性。一个请求可能跨越多个页面，涉及多个 Web 组件（由于 forward 指令和 include 动作的关系）c.session 是代表与用于某个 Web 客户机的一个用户体验相关的对象和属性。一个 Web 会话可以也经常会跨越多个客户机请求 d.application 是代表与整个 Web 应用程序相关的对象和属性。这实质上是跨越整个 Web 应用程序，包括多个页面、请求和会话的一个全局作用域。

23、详细描述 MVC。【基础】

答：基于 Java 的 Web 应用系统采用 MVC 架构模式，即 model（模型）、view（视图）、control（控制）分离设计；这是目前 WEB 应用服务系统的主流设计方向。

Model：即处理业务逻辑的模块，每一种处理一个模块；

View：负责页面显示，显示 MODEL 处理结果给用户，主要实现数据到页面转换过程；

Control：负责每个请求的分发，把 FORM 数据传递给 MODEL 处理，把处理结果的数据传递给 VIEW 显示。

26. BS 与 CS 的联系与区别？

C/S 是 Client/Server 的缩写，是客户机与服务器结构的应用程序

c/s: 必须电脑本地安装的程序，CS 的开发成本高

B/S 是 Brower/Server 的缩写，是浏览器和服务器结构的应用程序 b/s: 用浏览器打开的系统，BS 的开发成本低

24、简述 HttpSession 的作用、使用方法，可用代码说明。（3 分钟）【基础】

答：HttpSession 中可以跟踪并储存用户信息，把值设置到属性中，有 2 个方法：setAttribute(),getAttrribute()；

例如：在一个方法中用 session.setAttribute(“student”,student);在 session 中设置一个属性名为 student,值为一个名为 student 的对象。而后可在同一 session 范围内用 getAttribute(“student”)取出该属性，得到 student 对象。

25、介绍在 JSP 中如何使用 JavaBeans？【基础】

答：在 JSP 中使用 JavaBean 常用的动作有：

1）<jsp:useBean />：用来创建和查找 bean 对象；

2）<jsp:setProperty/>：用来设置 bean 的属性，即调用其 setXxx()方法；

3）<jsp:getProperty/>：用来获得 bean 的属性，即调用其 getXxx()方法。

26、JSP 和 Servlet 中的请求转发分别如何实现？【基础】

答：JSP 中的请求转发可利用 forward 动作实现：<jsp:forward />；Servlet 中实现请求转发的方式为：getServletContext().getRequestDispatcher(path).forward(req,res)。

27、Web.Xml 的作用？【基础】

答：用于配置 web 应用的信息；如 listener、filter 及 servlet 的配置信息等。

28、写出熟悉的 JSTL 标签。【基础】

答：<c:if>、<c:choose>、<c:when>、<c:otherwise>、<c:forEach>、<c:set>。

29、JSP 标签的作用？如何定义？【中等难度】

答：作用：分离 jsp 页面的内容和逻辑；

业务逻辑开发者可以创建自定义标签；

封装业务逻辑；

可重用并且易维护；

易于手工修改、易于工具维护；

提供简洁的语法；

定义:

写标签处理器；

写 tld 文件；

讲标签处理器和 tld 文件放到同一个包里面；

把 jsp 页面和标签库配置部署在一起。

30、 什么是懒加载 懒加载用什么技术实现,如何解决 session 关闭导致的懒加载问题 解决的方案有缺点吗、

懒加载就是延迟加载,采用代理技术实现，采用 openSessionInViewFilter,openSessionInViewFilter 其实上就是一个过滤器
页面打开时开启 session，页面访问结束时关闭 session
当访问用户过多，而且网速过慢的时候,会挤爆系统
事务扩大了 加锁导致资源等待时间过长
session 范围变大 如果加载页面的时间过长 如网速比较慢 session 内存时间过长 导致系统性能下载
数据库连接不能及时释放

31、 Spring 工作原理

内部最核心的就是IOC 了，动态注入，让一个对象的创建不用new 了，可以自动的生产，这其实就是利用java 里的反射,反射其实就是在运行时动态的去创建、调用对象，Spring就是在运行时，跟xml Spring 的配置文件来动态的创建对象，和调用对象里的方法的还有一个核心就是AOP 这个就是面向切面编程，可以为某一类对象进行监督和控制（也就是在调用这类对象的具体方法的前后去调用你指定的模块）从而达到对一个模块扩充的功能。这些都是

通过配置类达到的

Spring 目的：

就是让对象与对象（模块与模块）之间的关系没有通过代码来关联，都是通过配置类说明管理的（Spring 根据这些配置内部通过反射去动态的组装对象）要记住：Spring 是一个容器，凡是在容器里的对象才会有Spring 所提供的这些服务和功能

32、 ActionContext、ServletContext、pageContext 的区别 ？

① ActionContext Struts2 的 API：是当前的 Action 的上下文环境
② ServletContext 和 PageContext 是 Servlet 的 API

33、 拦截器的生命周期与工作过程 ？

每个拦截器都是需要实现 Interceptor 接口

> init()：在拦截器被创建后立即被调用，它在拦截器的生命周期内只被调用一次. 可以在该方法中对相关资源进行必要的初始化；

> intercept(ActionInvocation invocation)：每拦截一个动作请求，该方法就会被调用一次；

> destroy：该方法将在拦截器被销毁之前被调用，它在拦截器的生命周期内也只被调用一次；

34、 解释一下 IOC,以及 spring 的举例

IOC 称为控制反转，也叫依赖注入，ioc 是Spring 的核心组件，它通过配置文件，将需要创建的对象以池的方式管理，将实例注入到需要的对象中区，是对象依赖于注入而不依赖于实现，解决了各个组件的耦合度，使得项目在后期的维护和扩展上非常方便。

35、 谈谈你对 Spring AOP 思想的理解。

【参考答案】

AOP（Aspect-OrientedProgramming，面向方面编程），可以说是OOP（Object-OrientedPrograming，面向对象编程）的补充和完善。OOP 引入封装、继承和多态性等概念来建立一种对象层次结构，用以模拟公共行为的一个集合。当我们需要为分散的对象引入公共行为的时候，OOP 则显得无能为力。也就是说，OOP 允许你定义从上到下的关系，但并不适合定义从左到右的关系。例如日志功能，日志代码往往水平地散布在所有对象层次中，而与它所散布到的对象的核心功能毫无关系。对于其他类型的代码，如安全性、异常处理和透明的持续性也是如此。这种散布在各处的无关的代码被称为横切（cross-cutting）代码，在OOP 设计中，它导致了大量代码的重复，而不利于各个模块的重用。而AOP 技术则恰恰相反，它利用一种称为“横切”的技术，剖解开封装的对象内部，并将那些影响了多个类的公共行为封装到一个可重用模块，并将其名为“Aspect”，即方面。所谓“方面”，简单地说，就是将那些与业务无关，却为业务模块所共同调用的逻辑或责任封装起来，便于减少系统的重复代码，降低模块间的耦合度，并有利于未来的可操作性

和可维护性。AOP 代表的是一个横向的关系，如果说“对象”是一个空心的圆柱体，其中封装的是对象的属性和行为；那么面向方面编程的方法，就仿佛一把利刃，将这些空心圆柱体剖开，以获得其内部的消息。而剖开的切面，也就是所谓的“方面”了。然后它又以巧夺天功的妙手将这些剖开的切面复原，不留痕迹。使用“横切”技术，AOP 把软件系统分为两个部分：核心关注点和横切关注点。业务处理的主要流程是核心关注点，与之关系不大的部分是横切关注点。横切关注点的一个特点是，他们经常发生在核心关注点的多处，而各处都基本相似。比如权限认证、日志、事务处理。Aop 的作用在于分离系统中的各种关注点，将核心关注点和横切关注点分离开来。实现AOP 的技术，主要分为两大类：一是采用动态代理技术，利用截取消息的方式，对该消息进行装饰，以取代原有对象行为的执行；二是采用静态织入的方式，引入特定的语法创建“方面”，从而使得编译器可以在编译期间织入有关“方面”的代码。

36、 Spring 有哪几种注入方式?

3 种方法。构造属入、属性注入、接口注入

37、 Spring MVC 工作机制及为什么要用?

1) 客户端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。

38、 springmvc 与 struts2 比较

1 在数据封装方面 ，spring3mvc 方法级别 struts2 类级别 spring3mvc 开发效率高于 struts2

2 spring3mvc 注解版基本上是零配置

3 springmvc 与 spring 是一家人，兼容性好

4 struts2 存在安全漏洞
可以通过 ongl 表达式 格式化硬盘
使用重定向定位到钓鱼网站

39、 spring mvc 注解

1 @Controller 标注为 spring 容器的中的 bean 做为 c

2　作用域注解　有五大作用域：原型　单例　request　session　全局 session

3　@RequestMapping　访问路径，可以用在类或者方法上 访问路径类/方法

4　@responsebody　返回字符串，一般用于返回 json 格式

5　@ModelAttribute　放入 request 作用域

6　@SessionAttributes　值能用在类上

7　@requestParam　主要用于数据封装时，页面的参数名与方法参数不一致时

8　@pathVarible　主要用于获取路径变量的值

40、 说说 spring dao

1 对 jdbc 进行了封装,提供了两大模板技术封装了 jdbc 步骤,数据源的实现,行映射器进行记录与对象的转换工作,

使用 daoSupport 方便获取模板

2 给模板类注入数据源，之后使用模板类的 api 进行数据操作

41、 mybatis 框架的优缺点?

优点: 易于上手和掌握

sql 写在 xml 里,便于统一管理和优化

解除 sql 与程序代码的耦合

提供对象关系映射标签,支持对象关系组建维护

提供 xml 标签,支持编写动态 sql

缺点: 可读性低,调试非常困难,非常受限,无法像 jdbc 那样在代码里根据逻辑实现复杂动态 sql 拼接

42、 缓存

1 使用

mybatis　在映射文件配置 Cache，实体类实现序列化

2 优点与缺点

1 减少应用程序对数据库的访问次数，提高效率

2 数据量大时导致内存溢出

3 容忍无效的数据（缓存里的数据有可能是过期的），经常修改的数据不适合放入缓存中

3 缓存中配置一些什么东西

<!--

defaultCache:默认的缓存配置信息,如果不加特殊说明,则所有对象按照此配置项处理

maxElementsInMemory:设置了缓存的上限,最多存储多少个记录对象

eternal:代表对象是否永不过期

timeToIdleSeconds:最大的发呆时间

timeToLiveSeconds:最大的存活时间

overflowToDisk:是否允许对象被写入到磁盘

43、 解释一下 IOC,以及 spring 的举例

IOC 称为控制反转，也叫依赖注入，ioc 是 Spring 的核心组件，它通过配置文件，将需要创建的对象以池的方式管理，将实例注入到需要的对象中区，是对象依赖于注入而不依赖于实现，解决了各个组件的耦合度，使得项目在后期的维护和扩展上非常方便。

44、 Spring 工作原理

内部最核心的就是 IOC 了，动态注入，让一个对象的创建不用 new 了，可以自动的生产,这其实就是利用 java 里的反射,反射其实就是在运行时动态的去创建、调用对象，Spring 就是在运行时，跟 xml Spring 的配置文件来动态的创建对象，和调用对象里的方法的还有一个核心就是 AOP 这个就是面向切面编程，可以为某一类对象进行监督和控制（也就是在调用这类对象的具体方法的前后去调用你指定的模块）从而达到对一个模块扩充的功能。这些都是通过配置类达到的

Spring 目的：

就是让对象与对象（模块与模块）之间的关系没有通过代码来关联，都是通过配置类说明管理的(Spring 根据这些配置内部通过反射去动态的组装对象)要记住：Spring 是一个容器，凡是在容器里的对象才会有 Spring 所提供的这些服务和功能

45、 简述 spring 的事务传播行为和隔离级别

spring 的事务传播行为：

Spring 在 TransactionDefinition 接口中规定了 7 种类型的事务传播行为，它们规定了事务方法和事务方法发生嵌套调用时事务如何进行传播：

PROPAGATION_REQUIRED：如果当前没有事务，就新建一个事务，如果已经存在一个事务中，加入到这个事务中。这是最常见的选择。PROPAGATION_SUPPORTS：支持当前事务，如果当前没有事务，就以非事务方式执行。

PROPAGATION_MANDATORY：使用当前的事务，如果当前没有事务，就抛出异常。

PROPAGATION_REQUIRES_NEW：新建事务，如果当前存在事务，把当前事务挂起。

PROPAGATION_NOT_SUPPORTED：以非事务方式执行操作，如果当前存在事务，就把当前事务挂起。
PROPAGATION_NEVER：以非事务方式执行，如果当前存在事务，则抛出异常。
PROPAGATION_NESTED：如果当前存在事务，则在嵌套事务内执行。如果当前没有事务，则执行与 PROPAGATION_REQUIRED 类似的操作。

Spring 的隔离级别

1、Serializable：最严格的级别，事务串行执行，资源消耗最大；
2、REPEATABLE READ：保证了一个事务不会修改已经由另一个事务读取但未提交（回滚）的数据。避免了“脏读取”和“不可重复读取”的情况，但是带来了更多的性能损失。
3、READ COMMITTED:大多数主流数据库的默认事务等级，保证了一个事务不会读到另一个并行事务已修改但未提交的数据，避免了“脏读取”。该级别适用于大多数系统。
4、Read Uncommitted：保证了读取过程中不会读取到非法数据。

46、 谈谈你对 Spring 的理解。

1.Spring 实现了工厂模式的工厂类（在这里有必要解释清楚什么是工厂模式），这个类名为 BeanFactory （ 实际上是一个接口）， 在程序中通常 BeanFactory 的子类 ApplicationContext。Spring 相当于一个大的工厂类，在其配置文件中通过<bean>元素配置用于创建实例对象的类名和实例对象的属性。
2. Spring 提供了对 IOC 良好支持，IOC 是一种编程思想，是一种架构艺术，利用这种思想可以很好地实现模块之间的解耦。IOC 也称为 DI（Depency Injection）。
3. Spring 提供了对 AOP 技术的良好封装， AOP 称为面向切面编程，就是系统中有很多各不相干的类的方法，在这些众多方法中要加入某种系统功能的代码，例如，加入日志，加入权限判断，加入异常处理，这种应用称为 AOP。实现 AOP 功能采用的是代理技术，客户端程序不再调用目标，而调用代理类，代理类与目标类对外具有相同的方法声明，有两种方式可以实现相同的方法声明，一是实现相同的接口，二是作为目标的子类在，JDK 中采用 Proxy 类产生动态代理的方式为某个接口生成实现类，如果要为某个类生成子类，则可以用 CGLI B。在生成的代理类的方法中加入系统功能和调用目标类的相应方法，系统功能的代理以 Advice 对象进行提供，显然要创建出代理对象，至少需要目标类和 Advice 类。

47、 Spring 中 bean 的配置 scope 表示什么含义？可以有哪几种取值

scope 表示 Bean 的生命周期或者叫 Bean 的作用域。scope 的值有两个：
1、singleton，为单例属性，即 Spring IoC 容器只会创建该 bean 的唯一一个实例，

这也是默认的。
2、prototype 为原型属性，即每一次请求都会产生一个新的 bean 实例。

48、 软件开发的流程是怎样的

需求分析、概要设计、详细设计、编码、测试、交付、验收、维护

49、 spring 的优点

降低了组件之间的耦合性 ，实现了软件各层之间的解耦
可以使用容易提供的众多服务，如事务管理，消息服务等
容器提供单例模式支持
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
容器提供了众多的辅助类，能加快应用的开发
spring 对于主流的应用框架提供了集成支持，如 hibernate，JPA，Struts 等
spring 属于低侵入式设计，代码的污染极低
独立于各种应用服务器
spring 的 DI 机制降低了业务对象替换的复杂性
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

50、 、开发中都用到了那些设计模式?用在什么场合?

简单工厂模式、单例模式、观察者模式、适配器模式等。

51、 Java 中访问数据库的步骤

导入驱动包、获得连接、获取语句对象、获得结果集、关闭结果集、语句对象、连接。

52、 Statement，PreparedStatement，CallableStatement 的功能、特点。

Statement，用于执行静态 SQL 语句并返回它所生成结果的对象；（只执行一次的语句用这个）

PreparedStatement，表示预编译的 SQL 语句的对象，执行前可以进行赋值操作；（反复使用的语句用这个，有效的防止 sql 注入）
CallableStatement 执行存储过程，预编译的，带参数的；

53、 Servlet 的生命周期

初始化、实例化、服务、销毁
服务器加载servlet、服务器创建servlet实例、调用servlet实例的init方法、收到请求、调用service方法、调用doXxx方法处理请求并将输出结果返回客户端、等待下一个请求或由服务器卸载、调用destroy方法后被卸载。

54、 重定向和转发的区别，对应的方法是什么？

重定向：redirect，告诉浏览器请求另一个地址，地址栏 url 改变
转发：forward，请求不中断，转发到另一个资源，请求另一个地址后再把返回类容返回给客户端，地址栏 url 不改变

55、 forward 和 redirect 的区别？

答：forward 是容器中控制权的转向，是服务器请求资源，服务器直接访问目标地址的 URL，把那个 URL 的响应内容读取过来，然后把这些内容再发给浏览器，浏览器根本不知道服务器发送的内容是从哪儿来的，所以它的地址栏中还是原来的地址。redirect 就是服务端根据逻辑,发送一个状态码,告诉浏览器重新去请求那个地址，一般来说浏览器会用刚才请求的所有参数重新请求，所以 session,request 参数都可以获取，并且从浏览器的地址栏中可以看到跳转后的链接地址。前者更加高效，在前者可以满足需要时，尽量使用 forward()方法，并且，这样也有助于隐藏实际的链接；在有些情况下，比如，需要跳转到一个其它服务器上的资源，则必须使用 sendRedirect()方法。

56、 什么情况下调用 doGet()和 doPost()？【基础】

答：Jsp 页面中的 form 标签里的 method 属性为 get 时调用 doGet()，为 post 时调用 doPost()。

57、 JSP 和 Servlet 有哪些相同点和不同点，他们之间的联系是什么？【基础】

答：JSP 是 Servlet 技术的扩展，本质上是 Servlet 的简易方式，更强调应用的外表表达。JSP 编译后是"类 servlet"。Servlet 和 JSP 最主要的不同点在于，Servlet 的应用逻辑是在 Java 文件中，并且完全从表示层中的 HTML 里分离开来。而 JSP 的情况是 Java 和 HTML 可以组合成一个扩展名为.jsp 的文件。JSP 侧重于视图，Servlet 主要用于控制逻辑。

58、 JSP 的常用指令

1) <%@include >用来在JSP 页面包含静态资源
2) <%@taglib >用来指定JSP 页面标签类型
3) <%@page >用来指定页面相关属性

59、 MVC 的各个部分都有那些技术来实现?如何实现?

MVC 是Model－View－Controller 的简写。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。

60、 说出数据连接池的工作机制是什么?

web 服务器启动时会建立一定数量的池连接，并一直维持不少于此数目的池连接。客户端程序需要连接时，池驱动程序会返回一个未使用的池连接并将其表记为忙。如果当前没有空闲连接，池驱动程序就新建一定数量的连接，新建连接的数量有配置参数决定。当使用的池连接调用完成后，池驱动程序将此连接表记为空闲，其他调用就可以使用这个连接。实现方式，返回的Connection 是原始Connection 的代理，代理Connection 的close 方法不是真正关连接，而是把它代理的Connection 对象还回到连接池中。

61、 Web 容器里面的对象存活周期？

当然由web 容器进行创建管理的对象主要有application，session，request，page 这四个级别的对象，而这4 种级别的对象，根据它们自身的特点来管理所持的对象，如：request中的对象的生命周期就是在请求范围内，Session 在是会话周期内，page 是在当前JSP Page内，Application 是在服务器启、停的周期内。

62、 SQL 有哪三种注入方式？SQL 安全

动态 SQL 拼装注入、SQL 溢出漏洞、获取管理员权限、

63、 事务四大属性

原子性、一致性、隔离性、持久性。

1. 怎样保持事务的一致性

Java 中为了控制事务的一致性，会使用插入回滚点、callback 方法，保证数据不被篡改

64、 join 与 left join 的区别：

inner join(等值连接) 只返回两个表中联结字段相等的行
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

65、 你觉得 jquery 中的 ajax 好用吗，为什么？

答: 好用的。因为 jQuery 提供了一些日常开发中夙瑶的快捷操作，例 load，ajax，

get，post 等等，所以使用 jQuery 开发 ajax 将变得极其简单，我们就可以集中精力在业务和用户的体验上，不需要去理会那些繁琐的 XMLHttpRequest 对象了。

66、 jquery 中$.get()提交和$.post()提交有区别吗？

1) $.get() 方法使用 GET 方法来进行异步请求的。$.post() 方法使用 POST 方法来进行异步请求的。

2) get 请求会将参数跟在 URL 后进行传递，而 POST 请求则是作为 HTTP 消息的实体内容发送给 Web 服务器的，这种传递是对用户不可见的。

3) get 方式传输的数据大小不能超过 2KB 而 POST 要大的多

4) GET 方式请求的数据会被浏览器缓存起来，因此有安全问题。

67、 jQuery 是如何处理缓存的？

【参考】要处理缓存就是禁用缓存.
1) 通过$.post() 方法来获取数据，那么默认就是禁用缓存的。
2) 通过$.get()方法来获取数据，可以通过设置时间戳来避免缓存。可以在URL后面加上+(+new Date)例$.get('ajax.xml?'+(+new Date),function () { //内容
});
3) 通过$.ajax 方法来获取数据，只要设置cache:false 即可。

68、 jQuery 能做什么？

【参考】
1) 获取页面的元素
2) 修改页面的外观
3) 改变页面大的内容
4) 响应用户的页面操作
5) 为页面添加动态效果

6）无需刷新页面，即可以从服务器获取信息
7）简化常见的javascript 任务

69、 css+div 的优势

Div+CSS 标准的优点:
1.大大缩减页面代码，提高页面浏览速度,缩减带宽成本;
2.结构清晰，容易被搜索引擎搜索到，天生优化了seo
3.缩短改版时间。只要简单的修改几个CSS 文件就可以重新设计一个有成百上千页面的站点。
4.强大的字体控制和排版能力。CSS 控制字体的能力比糟糕的FONT 标签好多了，有了CSS，我们不再需要用FONT 标签或者透明的1 px GIF 图片来控制标题，改变字体颜色，字体样式等等。
5.CSS 非常容易编写。你可以象写html 代码一样轻松地编写CSS。
6.提高易用性。使用CSS 可以结构化HTML
7.可以一次设计，随处发布。更好的控制页面布局

70、 xml 有哪些解析技术，有什么区别？

有DOM,SAX,STAX 等
DOM:处理大型文件时其性能下降的非常厉害。这个问题是由DOM 的树结构所造成的，这种结构占用的内存较多，而且DOM 必须在解析文件之前把整个文档装入内存,适合对XML 的随机访问SAX:不现于DOM,SAX 是事件驱动型的XML 解析方式。它顺序读取XML 文件，不需要一次全部装载整个文件。当遇到像文件开头，文档结束，或者标签开头与标签结束时，它会触发一个事件，用户通过在其回调事件中写入处理代码来处理XML 文件，适合对XML 的顺序访问

71、 ajax 的工作原理？

Ajax的工作原理相当于在用户和服务器之间加了—个中间层,使用户操作与服务器响应异步化。这样把以前的一些服务器负担的工作转嫁到客户端，利于客户端闲置的处理能力来处理，减轻服务器和带宽的负担，从而达到节约ISP的空间及带宽租用成本的目的。
结果就是类似于桌面应用程序的动态、快速响应、高交互性的体验。

72、 JSON 和 XML 的优缺点

1）在可读性方面，JSON 和XML 的数据可读性基本相同。JSON 和XML 的可读性可谓不相上下，一边是建议的语法，一边是规范的标签形式，很难分出胜负。
2）在可扩展性方面，XML 天生有很好的扩展性，JSON 当然也有，没有什么是XML

能扩展，JSON 不能的。
3) 在编码难度方面，XML 有丰富的编码工具，比如Dom4j、JDom 等，JSON 也有json.org提供的工具，但是JSON 的编码明显比XML 容易许多，即使不借助工具也能写出JSON的代码，可是要写好XML 就不太容易了。
4) 在解码难度方面，XML 的解析得考虑子节点父节点，让人头昏眼花，而JSON 的解析难度几乎为0。这一点XML 输的真是没话说。
5) 在流行度方面，XML 已经被业界广泛的使用，而JSON 才刚刚开始，但是在Ajax这个特定的领域，未来的发展一定是XML 让位于JSON。到时Ajax 应该变成Ajaj(AsynchronousJavascript and JSON)了。
6) JSON 和XML 同样拥有丰富的解析手段。
7) JSON 相对于XML 来讲，数据的体积小。
8) JSON 与JavaScript 的交互更加方便。
9) JSON 对数据的描述性比XML 较差。
10) JSON 的速度要远远快于XML。