# Java 工程师面试题

1、**servlet执行流程**

客户端发出http请求，web服务器将请求转发到servlet容器，servlet容器解析url并根据web.xml找到相对应的servlet，并将request、response对象传递给找到的servlet，servlet根据request就可以知道是谁发出的请求，请求信息及其他信息，当servlet处理完业务逻辑后会将信息放入到response并响应到客户端。

2、**springMVC的执行流程**

springMVC是由dispatchservlet为核心的分层控制框架。首先客户端发出一个请求web服务器解析请求url并去匹配dispatchservlet的映射url，如果匹配上就将这个请求放入到dispatchservlet，dispatchservlet根据mapping映射配置去寻找相对应的handel，然后把处理权交给找到的handel，handel封装了处理业务逻辑的代码，当handel处理完后会返回一个逻辑视图modelandview给dispatchservlet，此时的modelandview是一个逻辑视图不是一个正式视图，所以dispatchservlet会通过viewresource视图资源去解析modelandview，然后将解析后的参数放到view中返回到客户端并展现。

3、JSP请求是如何被处理的？

浏览器首先要请求一个以.jsp扩展名结尾的页面，发起JSP请求，然后，Web服务器读取这个请求，使用JSP编译器把JSP页面转化成一个Servlet类。需要注意的是，只有当第一次请求页面或者是JSP文件发生改变的时候JSP文件才会被编译，然后服务器调用servlet类，处理浏览器的请求。一旦请求执行结束，servlet会把响应发送给客户端。这里看下如何在JSP中获取请求参数。

4、**Java设计模式思想（单列模式，工厂模式，策略模式，共23种设计模式）**

a) 单例模式：单例模式核心只需要new一个实例对象的模式，比如数据库连接，在线人数等，一些网站上看到的在线人数统计就是通过单例模式实现的，把一个计时器存放在数据库或者内存中，当有人登陆的时候取出来加一再放回去，有人退出登陆的时候取出来减一再放回去，但是当有两个人同时登陆的时候，会同时取出计数器，同时加一，同时放回去，这样的话数据就会错误，所以需要一个全局变量的对象给全部人使用，只需要new出一个实例对象，这就是单例模式的应用，并且单例模式节省资源，因为它控制了实例对象的个数，并有利于gc回收。

b) 策略模式：就是将几个类中公共的方法提取到一个新的类中，从而使扩展更容易，保证代码的可移植性，可维护性强。比如有个需求是写鸭子对象，鸭子有叫，飞，外形这三种方法，如果每个鸭子类都写这三个方法会出现代码的冗余，这时候我们可以把鸭子中的叫，飞，外形这三个方法提取出来，放到鸭父类中，让每个鸭子都继承这个鸭父类，重写这三个方法，这样封装的代码可移植性强，当用户提出新的需求比如鸭子会游泳，那么对于我们oo程序员来讲就非常简单了我们只需要在鸭父类中加一个游泳的方法，让会游泳的鸭子重写游泳方法就可以了。

c) 工厂模式：简单的工厂模式主要是统一提供实例对象的引用，通过工厂模式接口获取实例对象的引用。比如一个登陆功能，后端有三个类，controller类，interface类，实现接口的实现类。当客户端发出一个请求，当请求传到controller类中时，controller获取接口的引用对象，而实现接口的实现类中封装好了登陆的业务逻辑代码。当你需要加一个注册需求的时候只需要在接口类中加一个注册方法，实现类中实现方法，controller获取接口的引用对象即可，不需要改动原来的代码，这种做法是的可拓展性强。

5、HTTP响应的结构是怎么样的？

HTTP响应由三个部分组成：

状态码(Status Code)：描述了响应的状态。可以用来检查是否成功的完成了请求。请求失败的情况下，状态码可用来找出失败的原因。如果Servlet没有返回状态码，默认会返回成功的状态码HttpServletResponse.SC\_OK。

HTTP头部(HTTP Header)：它们包含了更多关于响应的信息。比如：头部可以指定认为响应过期的过期日期，或者是指定用来给用户安全的传输实体内容的编码格式。如何在Serlet中检索HTTP的头部看这里。

主体(Body)：它包含了响应的内容。它可以包含HTML代码，图片，等等。主体是由传输在HTTP消息中紧跟在头部后面的数据字节组成的。

6、**对ajax的理解**

a) Ajax为异步请求，即局部刷新技术，在传统的页面中，用户需要点击按钮或者事件触发请求，到刷新页面，而异步技术为不需要点击即可触发事件，这样使得用户体验感增强，比如商城购物车的异步加载，当你点击商品时无需请求后台而直接动态修改参数。

9、**父类与子类之间的调用顺序（打印结果）**

a) 父类静态代码块

b) 子类静态代码块

c) 父类构造方法

d) 子类构造方法

e) 子类普通方法

f) 重写父类的方法，则打印重写后的方法

10、**内部类与外部类的调用**

a) 内部类可以直接调用外部类包括private的成员变量，使用外部类引用的this.关键字调用即可

b) 而外部类调用内部类需要建立内部类对象

11、**多线程**

a)一个进程是一个独立的运行环境，可以看做是一个程序，而线程可以看做是进程的一个任务，比如QQ是一个进程，而一个QQ窗口是一个线程。

b)在多线程程序中，多线程并发可以提高程序的效率，cpu不会因为某个线程等待资源而进入空闲状态，它会把资源让给其他的线程。

c)用户线程就是我们开发程序是创建的线程，而守护线程为系统线程，如JVM虚拟中的GC

d)线程的优先级别：每一个线程都有优先级别，有限级别高的可以先获取CPU资源使该线程从就绪状态转为运行状态。也可以自定义线程的有限级别

e)死锁：至少两个以上线程争取两个以上cpu资源，避免死锁就避免使用嵌套锁，只需要在他们需要同步的地方加锁和避免无限等待

12、**AOP与IOC的概念（即spring的核心）**

a) IOC：Spring是开源框架，使用框架可以使我们减少工作量，提高工作效率并且它是分层结构，即相对应的层处理对应的业务逻辑，减少代码的耦合度。而spring的核心是IOC控制反转和AOP面向切面编程。IOC控制反转主要强调的是程序之间的关系是由容器控制的，容器控制对象，控制了对外部资源的获取。而反转即为，在传统的编程中都是由我们创建对象获取依赖对象，而在IOC中是容器帮我们创建对象并注入依赖对象，正是容器帮我们查找和注入对象，对象是被获取，所以叫反转。

b) AOP：面向切面编程，主要是管理系统层的业务，比如日志，权限，事物等。AOP是将封装好的对象剖开，找出其中对多个对象产生影响的公共行为，并将其封装为一个可重用的模块，这个模块被命名为切面（aspect），切面将那些与业务逻辑无关，却被业务模块共同调用的逻辑提取并封装起来，减少了系统中的重复代码，降低了模块间的耦合度，同时提高了系统的可维护性。

13、**hibernate的核心思想**

a) Hibernate的核心思想是ROM对象关系映射机制。它是将表与表之间的操作映射成对象与对象之间的操作。也就是从数据库中提取的信息会自动按照你设置的映射要求封装成特定的对象。所以hibernate就是通过将数据表实体类的映射，使得对对象的修改对应数据行的修改。

14、**Arraylist与linkedlist的区别**

a) 都是实现list接口的列表，arraylist是基于数组的数据结构，linkedlist是基于链表的数据结构，当获取特定元素时，ArrayList效率比较快，它通过数组下标即可获取，而linkedlist则需要移动指针。当存储元素与删除元素时linkedlist效率较快，只需要将指针移动指定位置增加或者删除即可，而arraylist需要移动数据。

15、**数据库优化**

a) 选择合适的字段，比如邮箱字段可以设为char（6），尽量把字段设置为notnull，这样查询的时候数据库就不需要比较null值

b) 使用关联查询（ left join on）查询代替子查询

c) 使用union联合查询手动创建临时表

d) 开启事物，当数据库执行多条语句出现错误时，事物会回滚，可以维护数据库的完整性

e) 使用外键，事物可以维护数据的完整性但是它却不能保证数据的关联性，使用外键可以保证数据的关联性

f) 使用索引，索引是提高数据库性能的常用方法，它可以令数据库服务器以比没有索引快的多的速度检索特定的行，特别是对于max，min，order by查询时，效果更明显

g) 优化的查询语句，绝大多数情况下，使用索引可以提高查询的速度，但如果sql语句使用不恰当的话，索引无法发挥它的特性。

16、T**omcat服务器优化（内存，并发连接数，缓存）**

a) 内存优化：主要是对Tomcat启动参数进行优化，我们可以在Tomcat启动脚本中修改它的最大内存数等等。

b) 线程数优化：Tomcat的并发连接参数，主要在Tomcat配置文件中server.xml中配置，比如修改最小空闲连接线程数，用于提高系统处理性能等等。

c) 优化缓存：打开压缩功能，修改参数，比如压缩的输出内容大小默认为2KB，可以适当的修改。

20、**HTTP协议**

a) 常用的请求方法有get、post

b) Get与post的区别：传送数据，get携带参数与访问地址传送，用户可以看见，这的话信息会不安全，导致信息泄露。而post则将字段与对应值封装在实体中传送，这个过程用户是不可见的。Get传递参数有限制，而post无限制。

17、http/https**协议**

18、Java集合类框架的基本接口有哪些

a) Collection集合接口，List、set实现Collection接口，arraylist、linkedlist，vector实现list接口，stack继承vector，Map接口，hashtable、hashmap实现map接口

19、**类加载的过程**

a) 遇到一个新的类时，首先会到方法区去找class文件，如果没有找到就会去硬盘中找class文件，找到后会返回，将class文件加载到方法区中，在类加载的时候，静态成员变量会被分配到方法区的静态区域，非静态成员变量分配到非静态区域，然后开始给静态成员变量初始化，赋默认值，赋完默认值后，会根据静态成员变量书写的位置赋显示值，然后执行静态代码。当所有的静态代码执行完，类加载才算完成。

20、**对象的创建**

a) 遇到一个新类时，会进行类的加载，定位到class文件

b) 对所有静态成员变量初始化，静态代码块也会执行，而且只在类加载的时候执行一次

c) New 对象时，jvm会在堆中分配一个足够大的存储空间

d) 存储空间清空，为所有的变量赋默认值，所有的对象引用赋值为null

e) 根据书写的位置给字段一些初始化操作

f) 调用构造器方法（没有继承）

21、**jvm的优化**

a) 设置参数，设置jvm的最大内存数

b) 垃圾回收器的选择

22、**高并发处理**

a) 了解一点高并发性问题，比如一W人抢一张票时，如何保证票在没买走的情况下所有人都能看见这张票，显然是不能用同步机制，因为synchronize是锁同步一次只能一个人进行。这时候可以用到锁机制，采用乐观锁可以解决这个问题。乐观锁的简单意思是在不锁定表的情况下，利用业务的控制来解决并发问题，这样即保证数据的可读性，又保证保存数据的排他性，保证性能的同时解决了并发带来的脏读数据问题。

23、**事物的理解**

a) 事物具有原子性，一致性，持久性，隔离性

b) 原子性：是指在一个事物中，要么全部执行成功，要么全部失败回滚。

c) 一致性：事物执行之前和执行之后都处于一致性状态

d) 持久性：事物多数据的操作是永久性

e) 隔离性：当一个事物正在对数据进行操作时，另一个事物不可以对数据进行操作，也就是多个并发事物之间相互隔离。

24、Struts工作流程

a) 客户端发出一个请求到servlet容器

b) 请求经过一些列过滤被filterdispatcher调用，filterdispatch通过actionMapper去找相对应的action。

c) Actionmapper找到对应的action返回给filterdispatch，dispatch把处理权交给actionproxy

d) Actionproxy通过配置文件找到对应的action类

e) Actionproxy创建一个actionIinvocation的实例处理业务逻辑

f) 一旦action处理完毕，actioninvocation负责根据stuts.xml的配置找到对应的返回结果。返回结果通常是jsp页面。