一、jsp九大隐含对象是什么

1、request对象 请求

request 对象是 javax.servlet.httpServletRequest类型的对象。 该对象代表了客户端的请求信息，主要用于接受通过HTTP协议传送到服务器的数据。（包括头信息、系统信息、请求方式以及请求参数等）。request对象的作用域为一次请求。

2、response对象 响应

response 代表的是对客户端的响应，主要是将JSP容器处理过的对象传回到客户端。response对象也具有作用域，它只在JSP页面内有效。

3、session对象 阶段

session 对象是由服务器自动创建的与用户请求相关的对象。服务器为每个用户都生成一个session对象，用于保存该用户的信息，跟踪用户的操作状态。session对象内部使用Map类来保存数据，因此保存数据的格式为 “Key/value”。 session对象的value可以使复杂的对象类型，而不仅仅局限于字符串类型。

4、application对象 应用

 application 对象可将信息保存在服务器中，直到服务器关闭，否则application对象中保存的信息会在整个应用中都有效。与session对象相比，application对象生命周期更长，类似于系统的“全局变量”。

5、out 对象 外面

out 对象用于在Web浏览器内输出信息，并且管理应用服务器上的输出缓冲区。在使用 out 对象输出数据时，可以对数据缓冲区进行操作，及时清除缓冲区中的残余数据，为其他的输出让出缓冲空间。待数据输出完毕后，要及时关闭输出流。

6、pageContext 对象 页面上下文

pageContext 对象的作用是取得任何范围的参数，通过它可以获取 JSP页面的out、request、reponse、session、application 等对象。pageContext对象的创建和初始化都是由容器来完成的，在JSP页面中可以直接使用 pageContext对象。

7、config 对象 配置

config 对象的主要作用是取得服务器的配置信息。通过 pageConext对象的 getServletConfig() 方法可以获取一个config对象。当一个Servlet 初始化时，容器把某些信息通过 config对象传递给这个 Servlet。 开发者可以在web.xml 文件中为应用程序环境中的Servlet程序和JSP页面提供初始化参数。

8、page 对象 页面

page 对象代表JSP本身，只有在JSP页面内才是合法的。 page隐含对象本质上包含当前 Servlet接口引用的变量，类似于Java编程中的 this 指针。

9、exception 对象 例外

exception 对象的作用是显示异常信息，只有在包含 isErrorPage="true" 的页面中才可以被使用，在一般的JSP页面中使用该对象将无法编译JSP文件。excepation对象和Java的所有对象一样，都具有系统提供的继承结构。exception 对象几乎定义了所有异常情况。在Java程序中，可以使用try/catch关键字来处理异常情况； 如果在JSP页面中出现没有捕获到的异常，就会生成 exception 对象，并把 exception 对象传送到在page指令中设定的错误页面中，然后在错误页面中处理相应的 exception 对象。

二、简述Servlet生命周期简述

（1）加载和实例化

当Servlet容器启动或客户端发送一个请求时，Servlet容器会查找内存中是否存在该Servlet实例，若存在，则直接读取该实例响应请求；如果不存在，就创建一个Servlet实例。

（2） 初始化

实例化后，Servlet容器将调用Servlet的init()方法进行初始化（一些准备工作或资源预加载工作）。

（3）服务

初始化后，Servlet处于能响应请求的就绪状态。当接收到客户端请求时，调用service()的方法处理客户端请求，HttpServlet的service()方法会根据不同的请求 转调不同的doXxx()方法。

（4）销毁

当Servlet容器关闭时，Servlet实例也随时销毁。其间，Servlet容器会调用Servlet 的destroy()方法去判断该Servlet是否应当被释放（或回收资源）。

三、cookie和session的区别是什么

1、存储位置不同

cookie的数据信息存放在客户端浏览器上。

session的数据信息存放在服务器上。

2、存储容量不同

单个cookie保存的数据<=4KB，一个站点最多保存20个Cookie。

对于session来说并没有上限，但出于对服务器端的性能考虑，session内不要存放过多的东西，并且设置session删除机制。

3、存储方式不同

cookie中只能保管ASCII字符串，并需要通过编码方式存储为Unicode字符或者二进制数据。

session中能够存储任何类型的数据，包括且不限于string，integer，list，map等。

4、隐私策略不同

cookie对客户端是可见的，别有用心的人可以分析存放在本地的cookie并进行cookie欺骗，所以它是不安全的。

session存储在服务器上，对客户端是透明对，不存在敏感信息泄漏的风险。

5、有效期上不同

开发可以通过设置cookie的属性，达到使cookie长期有效的效果。

session依赖于名为JSESSIONID的cookie，而cookie JSESSIONID的过期时间默认为-1，只需关闭窗口该session就会失效，因而session不能达到长期有效的效果。

6、服务器压力不同

cookie保管在客户端，不占用服务器资源。对于并发用户十分多的网站，cookie是很好的选择。

session是保管在服务器端的，每个用户都会产生一个session。假如并发访问的用户十分多，会产生十分多的session，耗费大量的内存。

7、浏览器支持不同

假如客户端浏览器不支持cookie：

cookie是需要客户端浏览器支持的，假如客户端禁用了cookie，或者不支持cookie，则会话跟踪会失效。关于WAP上的应用，常规的cookie就派不上用场了。

运用session需要使用URL地址重写的方式。一切用到session程序的URL都要进行URL地址重写，否则session会话跟踪还会失效。

假如客户端支持cookie：

cookie既能够设为本浏览器窗口以及子窗口内有效，也能够设为一切窗口内有效。

session只能在本窗口以及子窗口内有效。

8、跨域支持上不同

cookie支持跨域名访问。

session不支持跨域名访问。

四、get和post请求的区别是什么

GET在浏览器回退时是无害的，而POST会再次提交请求。

GET产生的URL地址可以被Bookmark，而POST不可以。

GET请求会被浏览器主动cache，而POST不会，除非手动设置。

GET请求只能进行url编码，而POST支持多种编码方式。

GET请求参数会被完整保留在浏览器历史记录里，而POST中的参数不会被保留。

GET请求在URL中传送的参数是有长度限制的，而POST么有。

对参数的数据类型，GET只接受ASCII字符，而POST没有限制。

GET比POST更不安全，因为参数直接暴露在URL上，所以不能用来传递敏感信息。

GET参数通过URL传递，POST放在Request body中。

五、线程的状态有哪几种

1. 新建状态(New): 线程对象被创建后，就进入了新建状态。例如，Thread thread = new Thread()。

2. 就绪状态(Runnable): 也被称为“可执行状态”。线程对象被创建后，其它线程调用了该对象的start()方法，从而来启动该线程。例如，thread.start()。处于就绪状态的线程，随时可能被CPU调度执行。

3. 运行状态(Running): 线程获取CPU权限进行执行。需要注意的是，线程只能从就绪状态进入到运行状态。

4. 阻塞状态(Blocked): 阻塞状态是线程因为某种原因放弃CPU使用权，暂时停止运行。直到线程进入就绪状态，才有机会转到运行状态。阻塞的情况分三种：

(01) 等待阻塞 -- 通过调用线程的wait()方法，让线程等待某工作的完成。

(02) 同步阻塞 -- 线程在获取synchronized同步锁失败(因为锁被其它线程所占用)，它会进入同步阻塞状态。

(03) 其他阻塞 -- 通过调用线程的sleep()或join()或发出了I/O请求时，线程会进入到阻塞状态。当sleep()状态超时、join()等待线程终止或者超时、或者I/O处理完毕时，线程重新转入就绪状态。

1. 死亡状态(Dead): 线程执行完了或者因异常退出了run()方法，该线程结束生命周期。

六、进程线程的区别：

地址空间：同一进程的线程共享本进程的地址空间，而进程之间则是独立的地址空间。

资源拥有：同一进程内的线程共享本进程的资源如内存、I/O、cpu等，但是进程之间的资源是独立的。

　　　　　一个进程崩溃后，在保护模式下不会对其他进程产生影响，但是一个线程崩溃整个进程都死掉。所以多进程要比多线程健壮。

　　　　　进程切换时，消耗的资源大，效率高。所以涉及到频繁的切换时，使用线程要好于进程。同样如果要求同时进行并且又要共享某些变量的并发操作，只能用线程不能用进程

执行过程：每个独立的进程程有一个程序运行的入口、顺序执行序列和程序入口。但是线程不能独立执行，必须依存在应用程序中，由应用程序提供多个线程执行控制。

线程是处理器调度的基本单位，但是进程不是。

两者均可并发执行。

七、HashMap和Hashtable的区别

HashMap和Hashtable都实现了Map接口，但决定用哪一个之前先要弄清楚它们之间的分别。主要的区别有：线程安全性，同步(synchronization)，以及速度。

HashMap几乎可以等价于Hashtable，除了HashMap是非synchronized的，并可以接受null(HashMap可以接受为null的键值(key)和值(value)，而Hashtable则不行)。

HashMap是非synchronized，而Hashtable是synchronized，这意味着Hashtable是线程安全的，多个线程可以共享一个Hashtable；而如果没有正确的同步的话，多个线程是不能共享HashMap的。Java 5提供了ConcurrentHashMap，它是HashTable的替代，比HashTable的扩展性更好。

另一个区别是HashMap的迭代器(Iterator)是fail-fast迭代器，而Hashtable的enumerator迭代器不是fail-fast的。所以当有其它线程改变了HashMap的结构（增加或者移除元素），将会抛出ConcurrentModificationException，但迭代器本身的remove()方法移除元素则不会抛出ConcurrentModificationException异常。但这并不是一个一定发生的行为，要看JVM。这条同样也是Enumeration和Iterator的区别。

由于Hashtable是线程安全的也是synchronized，所以在单线程环境下它比HashMap要慢。如果你不需要同步，只需要单一线程，那么使用HashMap性能要好过Hashtable。

HashMap不能保证随着时间的推移Map中的元素次序是不变的。

八、list集合和set集合的区别？map集合的区别？

List按对象进入的顺序保存对象，不做排序或编辑操作。Set对每个对象只接受一次，并使用自己内部的排序方法(通常，你只关心某个元素是否属于Set,而不关心它的顺序--否则应该使用List)。Map同样对每个元素保存一份，但这是基于"键"的，Map也有内置的排序，因而不关心元素添加的顺序。如果添加元素的顺序对你很重要，应该使用 LinkedHashSet或者LinkedHashMap.

九、String和StringBuilder、StringBuffer的区别？

Java平台提供了两种类型的字符串：String和StringBuffer/StringBuilder，它们可以储存和操作字符串。其中String是只读字符串，也就意味着String引用的字符串内容是不能被改变的。而StringBuffer/StringBuilder类表示的字符串对象可以直接进行修改。StringBuilder是Java 5中引入的，它和StringBuffer的方法完全相同，区别在于它是在单线程环境下使用的，因为它的所有方面都没有被synchronized修饰（非同步），因此它的效率也比StringBuffer要高。

十、jdk1.8的新特性有哪些？

1. Lambda表达式
2. 函数式接口
3. \*方法引用和构造器调用
4. Stream API
5. 接口中的默认方法和静态方法
6. 新时间日期API

十一、异常的顶级父类是什么？讲出常见的5种异常类型？

Throwable **1.NullPointerException: 空指针异常。**

1. **ClassCastException: 类型强制转换异常。**
2. **ArrayIndexOutOfBoundsException: 数组下标越界异常**
3. **ArithmeticException:算术运算异常。**
4. **NumberFormatException: 数字格式异常。**
5. **&和&&的区别?**

&运算符有两种用法：(1)按位与；(2)逻辑与。&&运算符是短路与运算。逻辑与跟短路与的差别是非常巨大的，虽然二者都要求运算符左右两端的布尔值都是true整个表达式的值才是true。&&之所以称为短路运算是因为，如果&&左边的表达式的值是false，右边的表达式会被直接短路掉，不会进行运算。很多时候我们可能都需要用&&而不是&，例如在验证用户登录时判定用户名不是null而且不是空字符串，应当写为：username != null &&!username.equals("")，二者的顺序不能交换，更不能用&运算符，因为第一个条件如果不成立，根本不能进行字符串的equals比较，否则会产生NullPointerException异常。注意：逻辑或运算符（|）和短路或运算符（||）的差别也是如此。

1. **==和equals的区别？**

==对于基本类型来说是值比较，对于引用类型来说是比较的引用；  
equals默认情况下是引用比较，只是很多类重写了equals方法，比如String、Integer等把它改成了值比较，所以一般情况下equals比较的是值是否相等。

1. **0si七层模型分别是什么？**

**物理层、数据链路层、传输层、网络层、会话层、表示层、应用层**

1. **数据库字段约束类型有哪些？**

主键约束（Primary Key）,外键约束（Foreign Key）,非空约束（Not Null）,唯一性约束（Unique）,默认约束（Default）。

**十六、简单说一重写和重载的区别**

重载（overload）：发生在同一个类中，方法名相同，参数不同，参数包括：参数的个数，参数类型，参数的类型顺序。

重写（override）：发生在继承过程中方法名相同、参数相同、返回值类型相同。

只要说重载：就是说两个方法的比较，绝对不是两个以上方法的比较。

**十七、Java语言的特点是什么**

1，**跨平台性:**跨平台性是指软件可以不受计算机硬件和操作系统的约束而在任意计算机环境下正常运行。 Java自带的虚拟机很好地实现了跨平台性。 Java源程序代码经过编译后生成二进制的字节码是与平台无关的，但是可被Java虚拟机识别的一种机器码指令，Java虚拟机提供了一个字节码到底层硬件平台及操作系统的屏障，使得Java语言具备跨平台性。

2，**简单**:Java语言是一种相当简洁的“面向对象”程序设计语言。Java语言省略了C++语言中所有的难以理解、容易混淆的特性，例如头文件、指针、结构、单元、运算符重载、虚拟基础类等。它更加严谨、简洁。

3，**面向对象:**面向对象技术使得应用程序的开发变得简单易用，节省代码。Java是一种面向对象的语言，也继承了面向对象的诸多好处，如代码扩展、代码复用等。

**4，安全性:Java**编译时要进行Java语言和语义的检查，保证每个变量对应一个相应的值，编译后生成Java类。运行时Java类需要类加载器载入，并经由字节码校验器校验之后才可以运行。 Java类在网络上使用时，对它的权限进行了设置，保证了被访问用户的安全性。

5，**多线程:**是一种轻量级进程，是现代程序设计中必不可少的一种特性。多线程处理能力使得程序能够具有更好的交互性、实时性。

**十八、讲讲你使用过哪些前端技术**

HTML、CSS和JavaScript、jquery

**十九、面对对象的三大特征是，封装你是怎么理解的**

# **封装，继承，多态**

## ****封装：将对象的实现细节隐藏，然后定义一些公有方法来显示该对象的功能，要访问该类的代码和数据，必须通过接口控制。****

**二十、接口和抽象类的区别**

1. 抽象类有普通的方法，接口没有  
   2.抽象类可以有自己的成员属性和方法，接口只能有public 常量。  
   3.抽象类可有可无构造方法，接口没有构造方法  
   4.抽象类单根继承，接口多重继承

**二十一、线程中的sleep和wait（）方法的区别是什么**

1.sleep（）使当前线程进入停滞状态，让出cpu的使用以供其他线程的使用；shleep（）是Thread类的静态方法，在任何地方调用时，当休眠期满会自动苏醒，但线程不会立即执行；为此sleep（）不能改变对象的机锁，当在synchronized中调用时，其他线程仍不能访问该对象。

2.wait（）使该线程进入到和该对象相等的等待池中；wait（）是Object类的一个方法，必须在synchronized中调用，wait（）通过notify（）、notifyAll（）方法或者设置等待时间达到唤醒等待池中的线程；为此调用wait（）方法时，该线程会暂时失去对象锁，wait（long timeOut）超时后还需返还。

**二十二、Java中如何创建线程池?线程池的作用是什么**

newCachedThreadPool创建一个可缓存线程池，如果线程池长度超过处理需要，可灵活回收空闲线程，若无可回收，则新建线程。

newFixedThreadPool 创建一个定长线程池，可控制线程最大并发数，超出的线程会在队列中等待。

newScheduledThreadPool 创建一个定长线程池，支持定时及周期性任务执行。

newSingleThreadExecutor 创建一个单线程化的线程池，它只会用唯一的工作线程来执行任务，保证所有任务按照指定顺序(FIFO, LIFO, 优先级)执行。

1. 减少在创建和销毁线程上所花的时间以及系统资源的开销   
       2.如不使用线程池，有可能造成系统创建大量线程而导致消耗完系统内存以及”过度切换”。

**二十三、数据库配置文件常配置的参数有哪些?dbch是什么？为什么要用dbch？**

**datadir**：指定mysql的数据目录位置，用于存放mysql数据库文件、日志文件等。

　　配置示例：datadir=D:/wamp/mysqldata/Data

**default-character-set**：mysql服务器默认字符集设置。

　　配置示例：default-character-set=utf8

**skip-grant-tables**：当忘记mysql用户密码的时候，可以在mysql配置文件中配置该参数，跳过权限表验证，不需要密码即可登录mysql。

DBCP(DataBase connection pool),数据库连接池。是 apache 上的一个 java 连接池项目，也是 tomcat 使用的连接池组件。

由于建立数据库连接是一个非常耗时耗资源的行为，所以通过连接池预先同数据库建立一些连接，放在内存中，应用程序需要建立数据库连接时直接到连接池中申请一个就行，用完后再放回去。

**二十四、数据库中对重复的数据进行去重，如何实现？**

1、查出所有有重复记录的所有记录

2、查出有重复记录的各个记录组中id最大的记录

3、查出多余的记录，不查出id最小的记录

4、删除多余的重复记录，只保留id最小的记录

**二十五、如何优化数据库查询的效率？**

1. 存储引擎的选择，如果数据表需要食物处理，应该考虑Innodb，因为它完全符合ACID的特性，如果不需要食物处理，使用默认存储引擎MyISAM是比较明智的
2. 分表分库主从
3. 对查询进行优化，要尽量避免全表扫描，首先应考虑在shere以及orderby涉及的列上建立索引
4. 应尽量避免在where子句当中使用，不等于或者大于小于操作符，否则将导致引擎放弃使用索引而进行全表扫描
5. 应尽量避免在where子句当中使用null值判断，否则将导致引擎放弃使用索引而进行全表扫描
6. 应尽量避免在where子句当中使用or的判断，如果一个字段有索引，一个字段没有索引，将会导致引擎放弃使用索引而进行全表扫描
7. update语句如果只改一两个字段，不要update全部字段，否则频繁调用，会引起明显的性能消耗，同时带来的大量的日志
8. 对于多张数据量的表进行join，要先分页在做join，否则逻辑读写高，性能差

**二十九、mysql分页查询是怎么实现的**

limit 基本实现方式

建立主键或者唯一索引

基于数据再排序

**三十、mvc是什么？有什么作用？**

MVC是三个单词的缩写,分别为： 模型(Model),视图(View)和控制Controller)。MVC是一种框架型模式，它本身并不引入新的功能，只是用来指导我们改善应用程序的架构，使应用的模型和视图相分离，从而得到更好的开发和维护效率。

1：耦合性低

2：重用性高

3：部署快

4：可维护性高

5：有利软件工程化管理

**三十一、HTTP协议常见的异常状态码有哪些？是什么原因引起的？**

200 - 服务器成功返回网页，客户端请求已成功。   
302 - 对象临时移动。服务器目前从不同位置的网页响应请求，但请求者应继续使用原有位置来进行以后的请求。  
304 - 属于重定向。自上次请求后，请求的网页未修改过。服务器返回此响应时，不会返回网页内容。  
401 - 未授权。请求要求身份验证。 对于需要登录的网页，服务器可能返回此响应。  
404 - 未找到。服务器找不到请求的网页。

1xx（临时响应）  
表示临时响应并需要请求者继续执行操作的状态代码。  
2xx - 成功。表示服务器成功地接受了客户端请求。  
3xx - 重定向。表示要完成请求，需要进一步操作。客户端浏览器必须采取更多操作来实现请求。例如，浏览器可能不得不请求服务器上的不同的页面，或通过代理服务器重复该请求。  
4xx - 请求错误。这些状态代码表示请求可能出错，妨碍了服务器的处理。

5xx - 服务器错误。表示服务器在尝试处理请求时发生内部错误。 这些错误可能是服务器本身的错误，而不是请求出错。

**三十二、C重定向和转发的区别是什么？**

重定向：redirect

地址栏发生变化

重定向可以访问其他站点（服务器）的资源

重定向是两次请求。不能使用request对象来共享数据

转发：forward

转发地址栏路径不变

转发只能访问当前服务器下的资源

转发是一次请求，可以使用request对象共享数据