

JAVA面试题

一. 面向对象的特征有哪些方面

1.抽象:

抽象就是忽略一个主题中与当前目标无关的那些方面，以便更充分地注意与当前目标有关的方面。抽象并不打算了解全部问题，而只是选择其中的一部分，暂时不用部分细节。抽象包括两个方面，一是过程抽象，二是数据抽象。

2.继承:

继承是一种联结类的层次模型，并且允许和鼓励类的重用，它提供了一种明确表述共性的方法。对象的一个新类可以从现有的类中派生，这个过程称为类继承。新类继承了原始类的特性，新类称为原始类的派生类（子类），而原始类称为新类的基类（父类）。派生类可以从它的基类那里继承方法和实例变量，并且类可以修改或增加新的方法使之更适合特殊的需要。

3.封装:

封装是把过程和数据包围起来，对数据的访问只能通过已定义的界面。面向对象计算始于这个基本概念，即现实世界可以被描绘成一系列完全自治、封装的对象，这些对象通过一个受保护的接口访问其他对象。

4.多态性:

多态性是指允许不同类的对象对同一消息作出响应。多态性包括参数化多态性和包含多态性。多态性语言具有灵活、抽象、行为共享、代码共享的优势，很好的解决了应用程序函数同名问题。

二. String是最基本的数据类型吗?

基本数据类型包括byte、int、char、long、float、double、boolean和short。

java.lang.String类是final类型的，因此不可以继承这个类、不能修改这个类。为了提高效率节省空间，我们应该用StringBuffer类

三. int 和 Integer 有什么区别

Java 提供两种不同的类型：引用类型和原始类型（或内置类型）。Int是java的原始数据类型，Integer是java为int提供的封装类。Java为每个原始类型提供了封装类。

原始类型封装类：booleanBoolean charCharacter byteByte shortShort

intInteger longLong floatFloat doubleDouble

引用类型和原始类型的行为完全不同，并且它们具有不同的语义。引用类型和原始类型具有不同的特征和用法，它们包括：大小和速度问题，这种类型以哪种类型的数据结构存储，当引用类型和原始类型用作某个类的实例数据时所指定的缺省值。对象引用实例变量的缺省值为 null，而原始类型实例变量的缺省值与它们的类型有关。

四. String 和StringBuffer的区别

JAVA平台提供了两个类：String和StringBuffer，它们可以储存和操作字符串，即包含多个字符的字符数据。这个String类提供了数值不可改变的字符串。而这个StringBuffer类提供的字符串进行修改。当你知道字符数据要改变的时候你就可以使用StringBuffer。典型地，你可以使用StringBuffers来动态构造字符数据。

五. 运行时异常与一般异常有何异同?

异常表示程序运行过程中可能出现的非正常状态，运行时异常表示虚拟机的通常操作中可能遇到的异常，是一种常见运行错误。java编译器要求方法必须声明抛出可能发生的非运行时异常，但是并不要求必须声明抛出未被捕获的运行时异常。

六. 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。

七. 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。

八. &和&&的区别

&和&&都可作为逻辑运算符“与”使用，但是&&是“短路与”，运算时先判断符号前面的表达式的值，如果能够确定整个表达式的值，则不进行符号后面的表达式的运算。另外，&也可作为位运算符使用。

九. 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。

十. final, finally, finalize的区别。

final 用于声明属性，方法和类，分别表示属性不可变，方法不可覆盖，类不可继承。

finally是异常处理语句结构的一部分，表示总是执行。

finalize是Object类的一个方法，在垃圾收集器执行的时候会调用被回收对象的此方法，可以覆盖此方法提供垃圾收集时的其他资源回收，例如关闭文件等**

十一. sleep() 和 wait() 有什么区别？

sleep是线程类（Thread）的方法，导致此线程暂停执行指定时间，给执行机会给其他线程，但是监控状态依然保持，到时后会自动恢复。调用sleep不会释放对象锁。

wait是Object类的方法，对此对象调用wait方法导致本线程放弃对象锁，进入等待此对象的等待锁定池，只有针对此对象发出notify方法（或notifyAll）后本线程才进入对象锁定池准备获得对象锁进入运行状态。

十二. Overload和Override的区别。Overloaded的方法是否可以改变返回值的类型?

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。如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。

十三. error和exception有什么区别?

error 表示恢复不是不可能但很困难的情况下的一种严重问题。比如说内存溢出。不可能指望程序能处理这样的情况。

exception 表示一种设计或实现问题。也就是说, 它表示如果程序运行正常, 从不会发生的情况。

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。

如果数据将在线程间共享。例如正在写的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—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。

十四. abstract class和interface有什么区别?

声明方法的存在而不去实现它的类被叫做抽象类 (abstract class), 它用于要创建一个体现某些基本行为的类, 并为该类声明方法, 但不能在该类中实现该类的情况。不能创建abstract 类的实例。然而可以创建一个变量, 其类型是一个抽象类, 并让它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。Abstract 类的子类为它们父类中的所有抽象方法提供实现, 否则它们也是抽象类。取而代之, 在子类中实现该方法。知道其行为的其它类可以在类中实现这些方法。

接口 (interface) 是抽象类的变体。在接口中, 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。接口中的所有方法都是抽象的, 没有一个有程序体。接口只可以定义static final成员变量。接口的实现与子类相似, 除了该实现类不能从接口定义中继承行为。当类实现特殊接口时, 它定义 (即将程序体给予) 所有这种接口的方法。然后, 它可以在实现了该接口的类的任何对象上调用接口的方法。由于有抽象类, 它允许使用接口名作为引用变量的类型。通常的动态联编将生效。引用可以转换到接口类型或从接口类型转换, instanceof 运算符可以用来决定某对象的类是否实现了接口。

十五. Static Nested Class 和 Inner Class的不同。

Static Nested Class是被声明为静态 (static) 的内部类, 它可以不依赖于外部类实例被实例化。而通常的内部类需要在外部类实例化后才能实例化。

十六. GC是什么? 为什么要有GC?

GC是垃圾收集的意思 (Garbage Collection), 内存处理是编程人员容易出现问题的地方, 忘记或者错误的内存回收会导致程序或系统的不稳定甚至崩溃, Java提供的GC功能可以自动监测对象是否超过作用域从而达到自动回收内存的目的, Java语言没有提供释放已分配内存的显示操作方法。

十七. Java有没有goto?

java中的保留字, 现在没有在java中使用。

十八. 启动一个线程是用run()还是start()?

启动一个线程是调用start()方法，使线程所代表的虚拟处理机处于可运行状态，这意味着它可以由JVM调度并执行。这并不意味着线程就会立即运行。run()方法可以产生必须退出的标志来停止一个线程。

十九. 给我一个你最常见到的runtime exception。

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

二十.接口是否可继承接口? 抽象类是否可实现(implements)接口? 抽象类是否可继承实体类(concrete class)?

接口可以继承接口。抽象类可以实现(implements)接口，抽象类可继承实体类

二十一.List, Set, Map是否继承自Collection接口?

List, Set是, Map

二十二.abstract的method是否可同时是static, 是否可同时是native, 是否可同时是synchronized?

都不能

二十三.数组有没有length()这个方法? String有没有length()这个方法?

数组没有length()这个方法，有length的属性。String有length()这个方法。

二十四.构造器Constructor是否可被override?

构造器Constructor不能被继承，因此不能重写Overriding，但可以被重载Overloading。

二十五.是否可以继承String类?

String类是final类故不可以继承。

二十六.swtich是否能作用在byte上, 是否能作用在long上, 是否能作用在String上?

switch (expr1) 中, expr1是一个整数表达式。因此传递给 switch 和 case 语句的参数应该是 int、short、char 或者 byte。long, string 都不能作用于swtich。

二十七.你所知道的集合类都有哪些? 主要方法?

最常用的集合类是 List 和 Map。List 的具体实现包括 ArrayList 和 Vector，它们是可变大小的列表，比较适合构建、存储和操作任何类型对象的元素列表。List 适用于按数值索引访问元素的情形。

Map 提供了一个更通用的元素存储方法。Map 集合类用于存储元素对（称作“键”和“值”），其中每个键映射到一个值。

二十八.char型变量中能不能存贮一个中文汉字?为什么?

能够定义成为一个中文的, 因为java中以unicode编码, 一个char占16个字节, 所以放一个中文是没问题的

二十九.多线程有几种实现方法, 都是什么?同步有几种实现方法, 都是什么?

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

三十.JAVA语言如何进行异常处理, 关键字: throws, throw, try, catch, finally分别代表什么意义? 在try块中可以抛出异常吗?

Java通过面向对象的方法进行异常处理, 把各种不同的异常进行分类, 并提供了良好的接口。在Java中, 每个异常都是一个对象, 它是Throwable类或其它子类的实例。当一个方法出现异常后便抛出一个异常对象, 该对象中包含有异常信息, 调用这个方法可以捕获到这个异常并进行处理。Java的异常处理是通过5个关键词来实现的: try、catch、throw、throws和finally。一般情况下是用try来执行一段程序, 如果出现异常, 系统会抛出 (throws) 一个异常, 这时候你可以通过它的类型来捕捉 (catch) 它, 或最后 (finally) 由缺省处理器来处理。

用try来指定一块预防所有“异常”的程序。紧跟在try程序后面, 应包含一个catch子句来指定你想要捕捉的“异常”的类型。

throw语句用来明确地抛出一个“异常”。throws用来标明一个成员函数可能抛出的各种“异常”。

Finally为确保一段代码不管发生什么“异常”都被执行一段代码。

可以在一个成员函数调用的外面写一个try语句, 在这个成员函数内部写另一个try语句保护其他代码。每当遇到一个try语句, “异常”的框架就放到堆栈上面, 直到所有的try语句都完成。如果下一级的try语句没有对某种“异常”进行处理, 堆栈就会展开, 直到遇到有处理这种“异常”的try语句。

三十一.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?

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。所以不应该使用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。

三十二.java中有几种类型的流? JDK为每种类型的流提供了一些抽象类以供继承, 请说出他们分别是哪些类?

字节流, 字符流。字节流继承于InputStream \ OutputStream, 字符流继承于InputStreamReader \ OutputStreamWriter。在java.io包中还有许多其他的流, 主要是为了提高性能和使用方便。

三十三.java中实现多态的机制是什么?

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。

三十四.垃圾回收器的基本原理是什么？垃圾回收器可以马上回收内存吗？有什么办法主动通知虚拟机进行垃圾回收？

对于GC来说，当程序员创建对象时，GC就开始监控这个对象的地址、大小以及使用情况。通常，GC采用有向图的方式记录和管理堆(heap)中的所有对象。通过这种方式确定哪些对象是“可达的”，哪些对象是“不可达的”。当GC确定一些对象为“不可达”时，GC就有责任回收这些内存空间。可以。程序员可以手动执行System.gc()，通知GC运行，但是Java语言规范并不保证GC一定会执行。

三十五.什么是java序列化，如何实现java序列化？

序列化就是一种用来处理对象流的机制，所谓对象流也就是将对象的内容进行流化。可以对流化后的对象进行读写操作，也可将流化后的对象传输于网络之间。序列化是为了解决在对对象流进行读写操作时所引发的问题。

序列化的实现：将需要被序列化的类实现Serializable接口，该接口没有需要实现的方法，implements Serializable只是为了标注该对象是可被序列化的，然后使用一个输出流(如：FileOutputStream)来构造一个ObjectOutputStream(对象流)对象，接着，使用ObjectOutputStream对象的writeObject(Object obj)方法就可以将参数为obj的对象写出(即保存其状态)，要恢复的话则用输入流。

三十六.jdbc调用数据库的步骤？

用JDBC来实现访问数据库记录可以采用下面的几个步骤：

- 1、通过驱动器管理器获取连接接口。
- 2、获得Statement或它的子类。
- 3、限制Statement中的参数。
- 4、执行Statement。
- 5、查看返回的行数是否超出范围。
- 6、关闭Statement。
- 7、处理其它的Statement
- 8、关闭连接接口。

JSP

一．浏览器jsp，html之间的关系

- 1.JSP与Java Servlet一样，是在服务器端执行的，通常返回该客户端的就是一个HTML文本，因此客户端只要有浏览器就能浏览
- 2.在大多数Browser/Server结构的Web应用中，浏览器直接通过HTML或者JSP的形式与用户交互，响应用户的请求
- 3.JSP在服务器上执行，并将执行结果输出到客户端浏览器，我们可以说基本上与浏览器无关

二．自定义标签要继承哪个类

这个类可以继承TagSupport或者BodyTagSupport，两者的差别是前者适用于没有主体的标签，而后者适用于有主体的标签。如果选择继承TagSupport，可以实现doStartTag和doEndTag两个方法实现Tag的功能，如果选择继承BodyTagSupport，可以实现doAfterBody这个方法。

三. 过滤器Filter的作用及配置

过滤器的作用:

过滤器是一个对象，可以传输请求或修改响应。它可以在请求到达Servlet/JSP之前对其进行预处理，而且能够在响应离开Servlet/JSP之后对其进行后处理。所以如果你有几个Servlet/JSP需要执行同样的数据转换或页面处理的话，你就可以写一个过滤器类，然后在部署描述文件（web.xml）中把该过滤器与对应的Servlet/JSP联系起来。你可以一个过滤器以作用于一个或一组servlet，零个或多个过滤器能过滤一个或多个servlet。一个过滤器实现java.servlet.Filter接口并定义它的三个方法：

1. void init(FilterConfig config) throws ServletException:在过滤器执行service前被调用，以设置过滤器的配置对象。
- 2 void destroy();在过滤器执行service后被调用。
- 3 Void doFilter(ServletRequest req, ServletResponse res, FilterChain chain) throws IOException, ServletException;

四. forward, 与redirect 的区别?有哪些方式实现

forward是把另一个页面加载到本页面，不改变浏览器的路径，redirect是跳转到另一个页面，会改变浏览器的路径

重定向:response.sendRedirect(重定向的路径")

转发:request.getRequestDispatcher(转发路径").forward(request, response);

五. jsp内置对象和作用?

有九个内置对象：request、response、out、session、application、pageContext、config、page、exception

作用如下：

(1) HttpServletRequest类的Request对象

作用：代表请求对象，主要用于接受客户端通过HTTP协议连接传输到服务器端的数据。

(2) HttpServletResponse类的Response对象

作用：代表响应对象，主要用于向客户端发送数据

(3) JspWriter类的out对象

作用：主要用于向客户端输出数据; Out的基类是JspWriter

(4) HttpSession类的session对象

作用：主要用于来分别保存每个用户信息，与请求关联的会话；会话状态维持是Web应用开发者必须面对的问题。

(5) ServletContext类的application对象

作用：主要用于保存用户信息，代码片段的运行环境；它是一个共享的内置对象，即一个容器中的多个用户共享一个application对象，故其保存的信息被所有用户所共享。

(6) PageContext类的PageContext对象

作用：管理网页属性，为JSP页面包装页面的上下文，管理对属于JSP中特殊可见部分中已命名对象的访问，它的创建和初始化都是由容器来完成的。

(7) ServletConfig类的Config对象

作用：代码片段配置对象，表示Servlet的配置。

(8) Object类的Page（相当于this）对象

作用：处理JSP网页，是Object类的一个实例，指的是JSP实现类的实例，即它也是JSP本身，只有在JSP页面范围之内才是合法的。

(9)Exception

作用：处理JSP文件执行时发生的错误和异常

六. 如果用JSP开发一个聊天程序，不用数据库存储聊天纪录，请问聊天记录最好存储在（）中。

Application

七. doPost, doGet的区别？

- 1.当你直接访问一个servlet时，调用的是doGet方法。
- 2.如果你的html里面规定了method访问哪个方法，则调用该方法。
- 3.get和post提交的数据量是不一样的.get好像最多只能在url后跟8K，post没这个限制

八. jsp乱码如何解决，几种解决方案？

- 1.JSP页面显示乱码
- 2.表单提交中文时出现乱码
- 3.数据库连接时出现乱码

九. 页面间对象传递的方法

request, session, application, cookie等

request.setAttribute(key, value)

session.setAttribute(key, value)

application.setAttribute(key, value)

我们在web应用开发过程中经常遇到输出某种编码的字符，如iso8859-1等，如何输出一个某种编码的字符串？

```
text = new String( text.getBytes("iso8859-1"), "GBK");
```

首先以"iso8859-1"解码为byte数组，再用"GBK"重构字符串

十. 简单介绍一下servlet

servlet容器：

负责处理客户请求、把请求传送给servlet并把结果返回给客户。不同程序的容器实际实现可能有所变化，但容器与servlet之间的接口是由servletAPI定义好的，这个接口定义了servlet容器在servlet上要调用的方法及传递给servlet的对象类。

servlet的生命周期：

servlet容器创建servlet的一个实例

容器调用该实例的init()方法

如果容器对该servlet有请求，则调用此实例的service()方法

容器在销毁本实例前调用它的destroy()方法

销毁并标记该实例以供作为垃圾收集

一旦请求了一个servlet，就没有办法阻止容器执行一个完整的生命周期。

容器在servlet首次被调用时创建它的一个实例，并保持该实例在内存中，让它对所有的请求进行处理。容器可以决定在什么时候把这个实例从内存中移走。在典型的模型中，容器为每个servlet创建一个单独的实例，容器并不会每接到一个请求就创建一个新线程，而是使用一个线程池来动态的将线程分配给到来的请求，但是这从servlet的观点来看，效果和为每个请求创建一个新线程的效果相同。

十一. BS与CS的联系与区别？

B/S模式是指在TCP/IP的支持下，以HTTP为传输协议，客户端通过Browser访问Web服务器以及与之相连的后台数据库的技术及体系结构。它由浏览器、Web服务器、应用服务器和数据库服务器组成。客户端的浏览器通过URL访问Web服务器，Web服务器请求数据库服务器，并将获得的结果以HTML形式返回客户端浏览器。

c/s在系统机构上和B/S相似，不过需要在客户端安装一个客户端软件，由这个软件对服务器的数据进行读写，就像我们常用的qq，就是这种模式。

十二. JSP与SERVLET的区别。

JSP先编译成SERVLET然后再编译成CLASS文件

JSP——SERVLET——JAVA文件——CLASS

JSP主要做视图层，servlet主要做控制层

JSP中动态INCLUDE与静态INCLUDE的区别？

动态INCLUDE用jsp:include动作实现 它总是会检查所含文件中的变化，适合用于包含动态页面，并且可以带参数。

静态INCLUDE用include伪码实现，定不会检查所含文件的变化，适用于包含静态页面

十三. JSP的内置对象及方法。

9大内置对象：

request, response, pagecontext, session, application, out, config, page, exception

request—HttpServletRequest的子类

response—HttpServletResponse的子类

session—HttpSession 的子类

pagecontext—PageContext的子类

application—ServletContext 的子类

out—JspWriter 的子类

config—ServletConfig 的子类

page—Object 的子类

exception—Throwab 的子类

十四. 四种会话跟踪技术

会话跟踪是为了跟踪用户于service之间的多次交互.

1.cookie 也就是我们常见的在C盘/文件夹下有一个COOKIE目录

2.session 比Cookie安全.不可见.但占用srevice资源

3.url重写.这是在用户的IE禁止Cookie的时候采用的方法。因为不管是Cookie还是 session 都要借助4.隐藏域

十五. <%...%>和<%!...%>的区别

<%...%>用于在JSP页面中嵌入Java脚本

<%!...%>用于在JSP页面中申明变量或方法，可以在该页面中的<%...%>脚本中调用，声明的变量相当于Servlet中的定义的成员变量。

十六. 你认为哪种分页效率最高? JSP , MYSQL, SQL25?

SQL25效率高:

因为mysql的分页还是先查出所有的记录，再从起始位置开始查出所需要的记录，而SQL25不会查所有记录，只查筛选后所需要的记录

十七. 简单说明下你对servlet的认识及其Servlet的基本架构、生命周期

HttpServlet类是一个抽象类，可以从该类派生出一个子类来实现一个HttpServlet，接受来自Web站点的请求（该请求来自访问该Web站点的客户浏览器），并将处理后的响应结果发回Web站点（Web站点再将响应结果发送给客户浏览器），在HttpServlet的子类中，必须至少重载下表中的其中一种方法。

方法名doGet

如果Servlet支持Http GET请求，用于Http GET请求

doPost

如果Servlet支持Http POST请求，用于Http POST请求

doPut

如果Servlet支持Http PUT请求，用于Http PUT请求

doDelete

如果Servlet支持Http DELETE请求，用于Http DELETE请求

init和destroy

如果需要管理Servlet生命周期内所持有资源，可以重载这两个方法

通常，不重载service方法，对于上表中的每一种HTTP请求，service方法通过分派它们到相应的Handler线程（doXXX方法）来处理这些标准的HTTP请求。

同样地，通常也不重载doOptions和doTrace方法，service方法通过分派它们到doTrace和doOptions来支持HTTP1.1 TRACE和OPTIONS。

Servlet通常运行在多线程的服务器中，因此，所编写的Servlet代码必须能够处理并行请求和对数据资源的同步访问。共享资源包括内存中的数据（例如：实例或类变量）和外部对象（例如：文件、数据库连接或网络连接）。

Protected void doGet(HttpServletRequest req, HttpServletResponse resp)

throws ServletException, IOException

从service方法授并处理HTTP GET请求。GET方法允许客户读取来自Web服务器的信息，客户通过传递一个带URL的查询字符串可以告诉服务器它需要什么信息。

重载支持GET请求的doGet方法还将自动支持HTTP HEAD请求，HEAD请求也是一个GET请求，它得到的返回响应中只有一个请求头（header）字段，而没有响应信息的内容。

如果重载方法，应该从该请求读数据，在响应中设置整个headers，访问PrintWriter或输出流对象，最后写响应数据。当设置headers时，应确保包含content type和encoding。如果使用PrintWriter对象返回响应，在存取PrintWriter对象之前必须设置content type。

Servlet引擎必须在写响应数据之前写headers，因为在写数据之后headers随时都可能被刷新。

十八. 简单描述下数据连接池的工作机制是什么？

数据库连接池在初始化时将创建一定数量的数据库连接放到连接池中，这些数据库连接的数量是由最小数据库连接数来设定的。无论这些数据库连接是否被使用，连接池都将一直保证至少拥有这么多的连接数量。连接池的最大数据库连接数量限定了这个连接池能占有的最大连接数，当应用程序向连接池请求的连接数超过最大连接数量时，这些请求将被加入到等待队列中。

数据库连接池的最小连接数和最大连接数的设置要考虑到下列几个因素：

1. 最小连接数是连接池一直保持的数据库连接，所以如果应用程序对数据库连接的使用量不大，将会有大量的数据库连接资源被浪费；
2. 最大连接数是连接池能申请的最大连接数，如果数据库连接请求超过此数，后面的数据库连接请求将被加入到等待队列中，这会影响之后的数据库操作。
3. 如果最小连接数与最大连接数相差太大，那么最先的连接请求将会获利，之后超过最小连接数量的连接请求等价于建立一个新的数据库连接。不过，这些大于最小连接数的数据库连接在使用完不会马上被释放，它将被放到连接池中等待重复使用或是空闲超时后被释放。

十九. jsp乱码如何解决，给出三种以上的对应解决方案，并给出对应的程序案例；

一、JSP页面显示乱码

```
<%@ page contentType="text/html; charset=gb2312"%>
```

二、表单提交中文时出现乱码

```
request.setCharacterEncoding("gb2312")对请求进行统一编码
```

三、数据库连接出现乱码

要涉及中文的地方全部是乱码，解决办法：在数据库的数据库URL中加上useUnicode=true&characterEncoding=GBK就OK了。

四、通过过滤器完成

五、在server.xml中的设置编码格式