1. 什么是 Java 虚拟机？为什么 Java 被称作是“平台无关的编程语言”？

Jvm 虚拟机，java 编译文件运行重要核心，将 class 文件编译成机器特定的核心。而且其内部包含了很多 java 重要的类和工具，例如回收机制、内存管理、安全机制等。之所以称它是虚拟的，是因为它拥有不依赖底层操作系统以及硬件的接口。也因为它独立于操作系统和硬件，实现了一次编写代码各个平台都可以运行，这就是与平台无关的原因。

2. JDK 和 JRE 的区别是什么？

Jdk指java开发工具包，是java开发最重要的部分，提供了很多类，以及编译器、调试器、运行等工具。而jre是运行环境，将java文件编译成二进制文件。同时，jdk内部包含了jre,可以说jdk是jre的超集

3. ”static”关键字是什么意思？Java 中是否可以覆盖(override)一个 private 或者是 static 的方法？

Static表示静态变量或者静态方法，首先覆盖发生在继承当中，子类不能继承父类的private方法，所以不能覆盖。而static修饰的为静态方法，是在编译时绑定，而方法覆盖属于运行时动态绑定的，所以不可以覆盖。

4. 是否可以在 static 环境中访问非 static 变量？

不可以，但是非静态方法可以访问静态变量或者方法。

5. Java 支持的数据类型有哪些？什么是自动拆装箱？

八大基本数据类型：int,byte,
short,long,float,double,char,boolean,

自动装箱：例如当int赋值给对应的包装类时，会自动包装成Integer类型。

反之即为拆箱。

6. Java 中的方法覆盖(Overriding)和方法重载(Overloading)是什么意思?

方法重写发生在继承当中，子类为满足本身的需求将父类的方法进行重写，重写部分为方法体部分，方法名和父类一样。当子类调用该方法时，调用的是重写后的方法。

方法重载发生同一个类中，方法签名相同，只有参数列表不同，包括顺序不同，数量不同，类型不同；

7. Java 中，什么是构造函数？什么是构造函数重载？什么是复制构造函数?

创建对象时，本质就是调用其构造函数，其方法名和类型修饰，没有返回值类型。当我们自己创建一个构造函数时，创建对象时，就会调用该构造方法，如果没有创建构造函数，调用时，编译器会在低沉帮我们创建出来。构造函数重载指类内部,构造方法参数列表不同，包括顺序不同，数量不同，类型不同。复制构造函数是c语言当中的，java没有该操作。

8. Java 支持多继承么？

Java类只支撑单继承，但是在接口中，一个子接口可以继承多个接口；

9. 接口和抽象类的区别是什么？

抽象类是被abstract修饰的类，不能被实例化，只支持单继承。其类内部可以定义普通方法或者抽象方法，子类继承抽象类必须实现父类的所有抽象类，除非该类也是抽象类；

而接口，内部全部是抽象方法或者常量，不能有普通方法和实例变量；一个类实现一个接口时，必须实现接口的所有方法；接口支持多继承，一个子接口可以继承多个接口。

10. 什么是值传递和引用传递?

基本数据之间是指传递，彼此之间传的是值的副本，所以改变副本时，不影响原本的值；而java中对象类型之间的传递为引用传递，其本质为传递是内存地址的副本，但副本发生修改时，会导致原本的对象类型也发生改变；

11. 进程和线程的区别是什么?

进程表示运行的应用软件，而线程是指一个进程内部的执行序列；一个进程可以有多个线程；

12. 创建线程有几种不同的方式? 你喜欢哪一种? 为什么?

- 继承Thread类，实现Runable接口；更喜欢后者，可以实现多继承，使用线程池也比较方便。

- 继承Thread类后，无法再继承其他类，但是实现Runnable接口，还可以继承其他类。

13. 概括的解释下线程的几种可用状态。

- 新建、就绪、运行、阻塞

1. 新建(new)：新创建了一个线程对象。

2. 可运行(runnable)：线程对象创建后，其他线程(比如 main 线程)调用了该对象 的 start() 方法。该状态的线程位于可运行线程池中，等待被线程调度选中，获 取 cpu 的使用权 。 3. 运行(running)：可运行状态(runnable)的线程获得了 cpu 时间片 （ timeslice ） ，执行 程序代码。

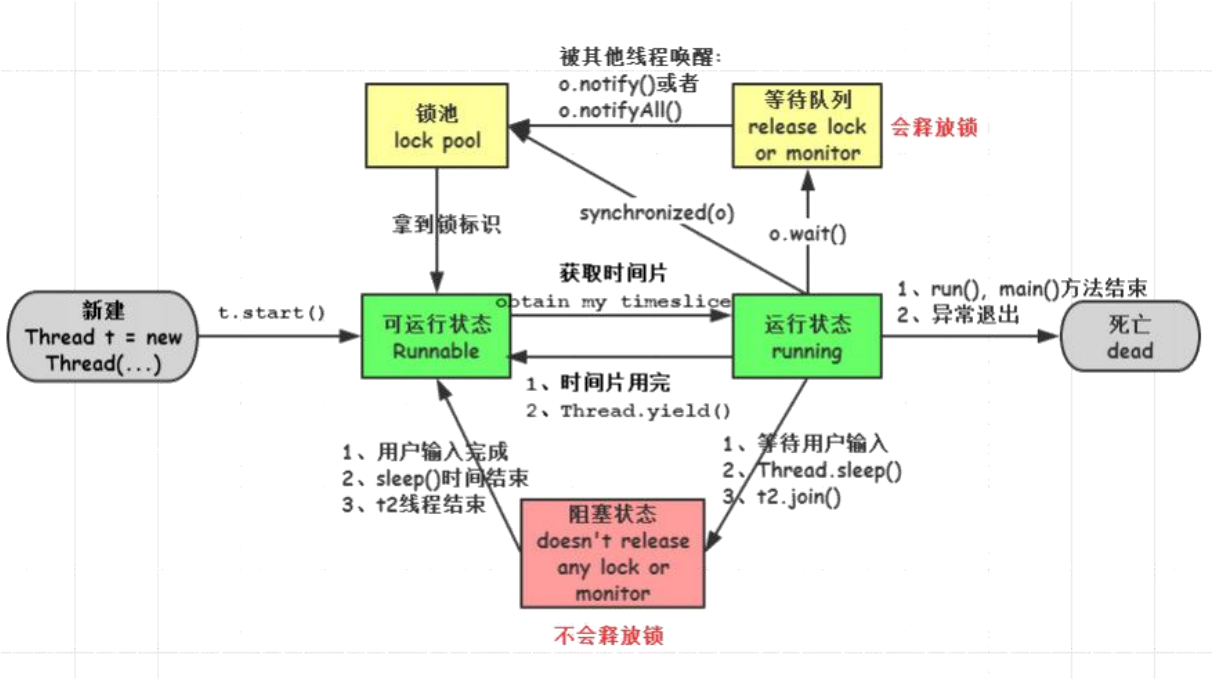
4. 阻塞(block)：阻塞状态是指线程因为某种原因放弃了 cpu 使用权，也即让出了 cpu timeslice ，暂时停止运行。直到线程进入可运行(runnable)状态，才有 机会再次获得 cpu timeslice 转到运行(running)状态。阻塞的情况分三种：

(一). 等待阻塞：运行(running)的线程执行 o.wait()方法， JVM 会把该线程放 入等待队列(waitting queue)中。

(二). 同步阻塞：运行(running)的线程在获取对象的同步锁时，若该同步锁 被别的线程占 用，则 JVM 会把该线程放入锁池(lock pool)中。

(三). 其他阻塞: 运行(running)的线程执行 Thread.sleep(long ms)或 t.join()方法，或者发出了 I/O 请求时， JVM 会把该线程置为阻塞状态。

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，线程重新转入可运行(runnable)状态。 5. 死亡(dead)：线程 run()、 main() 方法执行结束，或者因异常退出了 run()方法，则该 线程结束生命周期。死亡的线程不可再次复生。



14. 同步方法和同步代码块的区别是什么？

✧ 同步方法是在整个方法上用synchronized声明，而同步代码块，只同步会发生线性安全的代码，相比较同步方法，性能更高。

15. 在监视器(Monitor)内部，是如何做线程同步的？程序应该做哪种级别的同步？

✧ 没有接触

16. 什么是死锁(deadlock)?

✧ 复习。。。。

✧ 两个线程或两个以上线程都在等待对方执行完毕才能继续往下执行的时候就发生了死锁。结 果就是这些线程都陷入了无限的等待中。

17. 如何确保 N 个线程可以访问 N 个资源同时又不导致死锁?

- //List<线程> lists = collections.synchronizedList(<List<.线程> ( ));

使用多线程的时候，一种非常简单的避免死锁的方式就是：指定获取锁的顺序，并强制线程按照指定的顺序获取锁。因此，如果所有的线程都是以同样的顺序加锁和释放锁，就不会出 现死锁了。

18. hashCode()和 equals()方法的重要性体现在什么地方?

- //当hashcode相同时，equals才有可能为true，反之，equals为true时，hashcode必定相等。

java中hashmap()使用hashcode和equals方法来确定键值对的索引;当根据键获得值时，也会使用这两个方法，如果没有正确的实现这两个方法，两个不同键可能会有相同hash值，可能会被集合认为是两个相等;这两个方法也可以用来发现重复元素,所以这两个方法对hashmap的正确性和精确性起到至关重要的作用。

19. Java集合类框架的基本接口有哪些?

Map和collection，其中collection包含了set和list两个子接口

20. 为什么集合类没有实现Cloneable和Serializable接口?

克隆(cloning)或者是序列化(serialization)的语义和含义是跟具体的

实现相关的。因此，应该 由集合类的具体实现来决定如何被克隆或者是序列化。

21. 什么是迭代器(Iterator)?

✧ Iterator提供了很多对集合元素迭代的方法，每一个集合类里面都有实现迭代器实例的方法，迭代器可以删除集合元素，但不能用集合的remove方法，应该用迭代器的remove（）方法；

22. Iterator和ListIterator的区别是什么？

✧ 下面列出了他们的区别：

✧ Iterator可用来遍历Set和List集合，但是ListIterator只能用来遍历List。

✧ Iterator对集合只能是前向遍历，ListIterator既可以前向也可以后向。

✧ ListIterator实现了Iterator接口，并包含其他的功能，比如：增加元素，替换元素，获取前一个和后一个元素的索引，等等。

23. 快速失败(fail--fast)和安全失败(fail--safe)的区别是什么？

✧ Iterator的安全失败是基于对底层集合做拷贝，因此，它不受源集合上修改的影响。java.util 包下面的所有的集合类都是快速失败的，而java.util.concurrent包下面的所有的类都是安全失败的。快速失败的迭代器会抛出ConcurrentModificationException异常，而安全失败的 迭代器永远不会抛出这样的异常。

24. Java中的HashMap的工作原理是什么？

✧ Java中的HashMap是以键值对(key--value)的形式存储元素的。HashMap需要一个hash 函数，它使用hashCode()和equals()方法来向集合/从集合添加和检索元素。当调用put() 方法的时候，HashMap会计算key的hash值，然后把键值对存储在集合中合适的索引上。 如果key已经存在了，value会被更新成新值。HashMap的一些重要的特性是它的容量 (capacity)，负载因子(load factor)和扩容极限(threshold resizing)。

25. HashMap 和 Hashtable 有什么区别?

✧ HashMap键值允许null值存在，属于非线性安全，而hashtable是线性安全的。Hashtable不允许键或者值是null。

26. 数组(Array)和列表(ArrayList)有什么区别？什么时候应该使用 Array 而不是 ArrayList?

✧ Array可以包含基本类型和对象类型，ArrayList只能包含对象类型。

✧ Array大小是固定的，ArrayList的大小是动态变化的。

✧ ArrayList提供了更多的方法和特性，比如：addAll()，removeAll()，iterator()等等。

✧ 对于基本类型数据，集合使用自动装箱来减少编码工作量。但是，当处理固定大小的基本数 据类型的时候，这种方式相对比较慢。

27. ArrayList 和 LinkedList 有什么区别?

ArrayList和LinkedList都实现了List接口，他们有以下的不同点：

✧ ArrayList是基于索引的数据接口，它的底层是数组。它可以以O(1)时间复杂度对元素进行随机访问。与此对应，LinkedList是以元素列表的形式存储它的数据，每一个元素都和它的 前一个和后一个元素链接在一起，在这种情况下，查找某个元素的时间复杂度是O(n)。

✧ 相对于ArrayList，LinkedList的插入，添加，删除操作速度更快，因为当元素被添加到集合 任意位置的时候，不需要像数组那样重新计算大小或者是更新索引。LinkedList比ArrayList更占内存，因为LinkedList为每一个节点存储了两个引用，一个指 向前一个元素，一个指向下一个元素。

28. 如何权衡是使用无序的数组还是有序的数组?

✧ 有序数组最大的好处在于查找的时间复杂度是O(log n)，而无序数组是O(n)。有序数组的缺点是插入操作的时间复杂度是O(n)，因为值大的元素需要往后移动来给新元素腾位置。 相反，无序数组的插入时间复杂度是常量O(1)。

29. HashSet 和 TreeSet 有什么区别?

✧ HashSet是由一个hash表来实现的，因此，它的元素是无序的。add()，remove()，contains()方法的时间复杂度是O(1)。另一方面，TreeSet是由一个树形的结构来实现的，它里面的元素是有序的。因此，add()，remove()，contains()方法的时间复杂度是O(logn)。

30. Java 中垃圾回收有什么目的？什么时候进行垃圾回收?

✧ 目的是识别并且丢弃不再使用的对象，以此来释放和重新利用资源

31. System.gc()和 Runtime.gc()会做什么事情?

✧ 这两个方法用来提示JVM要进行垃圾回收。但是，立即开始还是延迟进行垃圾回收是取决于JVM的。

32. finalize()方法什么时候被调用？析构函数(finalization)的目的是什么?

✧ 垃圾回收器(garbage colector)决定回收某对象时，就会运行该对

象的finalize()方法 但是在Java中很不幸，如果内存总是充足的，那么垃圾回收可能永远不会进行，也就是说filalize() 可能永远不被执行，显然指望它做收尾工作是靠不住的。 那么finalize()究竟是做什么的呢？ 它最主要的用途是回收特殊渠道申请的内存。Java程序有垃圾回收器，所以一般情况下内

✧ 存问题不用程序员操心。但有一种JNI(Java Native Interface)调用non--Java程序（C或C++）， finalize()的工作就是回收这部分的内存。

33. 如果对象的引用被置为 null，垃圾收集器是否会立即释放对象占用的内存?

✧ 不会，在下一个垃圾回收周期内，是可以被回收

34. 在 Java 中，对象什么时候可以被垃圾回收?

当对象对当前使用这个对象的应用程序变得不可触及的时候，这个对象就可以被回收了。

35. Java 中 Exception 和 Error 有什么区别?

Exception是由程序本身引起的异常，可以被用户捕获，而error由硬件或者应用软件本身引起的。

36. throw 和 throws 有什么区别?

Throw是用在方法体中，将异常手动抛出；而throws在方法中声明，以下方法体可能会发生异常

37. 异常处理完成以后，Exception 对象会发生什么变化?

Exception对象会在下个垃圾回收周期内被清理

38. finally 代码块和 finalize()方法有什么区别?

Finally用在try catch语言的最后，不管是否捕获到异常，都要进行

的操作。finalize()方法是Object类的一个protected方法，它是在对象被垃圾回收之前由Java虚拟机来调用 的。

39. 什么是JDBC?

JDBC是允许用户在不同数据库之间做选择的一个抽象层。JDBC允许开发者用JAVA写数据库应用程序，而不需要关心底层特定数据库的细节。

40. 解释下驱动(Driver)在JDBC中的角色。

JDBC驱动提供了特定厂商对JDBC API接口类的实现，驱动必须要提供java.sql包下面这 些类的实现：Connection, Statement, PreparedStatement,CallableStatement, ResultSet 和Driver。

41. Class.forName()方法有什么作用?

初始化参数指定的类，并且返回此类对应的Class 对象

42. PreparedStatement比Statement有什么优势?

PreparedStatement是预编译的，提高了性能，同时防止sql漏洞注入，提高安全性。

43. 什么时候使用CallableStatement? 用来准备CallableStatement的方法是什么?

CallableStatement用来执行存储过程。存储过程是由数据库存储和提供的。存储过程可以接受输入参数，也可以有返回结果。非常鼓励使用存储过程，因为它提供了安全性和模块化。 准备一个CallableStatement的方法是：CallableStament.prepareCall();

44. 数据库连接池是什么意思?

答：由于创建连接和释放连接都有很大的开销（尤其是数据库服务器不在本地时，每次建立连接都需要进行 TCP 的三次握手，再加

上网络延迟，造成的开销是不可忽视的），为了提升系统访问数据库的性能，可以事先创建若干连接置于连接池中，需要时直接从连接池获取，使用结束时归还连接池而不必关闭连接，从而避免频繁创建和释放连接所造成的开销，这是典型的用空间换取时间的策略（浪费了空间存储连接，但节省了创建和释放连接的时间）。池化技术在 Java 中是很常见的，在使用线程时创建线程池的道理与此相同。据库连接池常用的有： C3P0、DBCP 等。

45. 什么是分布式垃圾回收(DGC)？它是如何工作的？

DGC叫做分布式垃圾回收。RMI使用DGC来做自动垃圾回收。因为RMI包含了跨虚拟机的远程对象的引用，垃圾回收是很困难的。DGC使用引用计数算法来给远程对象提供自动 内存管理。

46. 解释下 Serialization 和 Deserialization。

Java提供了一种叫做对象序列化的机制，他把对象表示成一连串的字节，里面包含了对象的数据，对象的类型信息，对象内部的数据的类型信息等等。因此，序列化可以看成是为了把对象存储在磁盘上或者是从磁盘上读出来并重建对象而把对象扁平化的一种方式。反序列化是把对象从扁平状态转化成活动对象的相反的步骤。

47. 什么是 Servlet?

Servlet是用来处理客户端请求并产生动态网页内容的Java类。Servlet主要是用来处理或者是存储HTML表单提交的数据，产生动态内容，在无状态的HTTP协议下管理状态信息。

48. 说一下 Servlet 的体系结构。

所有的Servlet都必须要实现的核心的接口是javax.servlet.Servlet。每一个Servlet都必须 要直接或者是间接实现这个接口，或者是继承javax.servlet.GenericServlet或者javax.servlet.http.HTTPServlet。最后，Servlet使用多线程可以并行的为多个请求服务。

49. GenericServlet 和 HttpServlet 有什么区别?

GenericServlet是一个通用的协议无关的Servlet，它实现了Servlet和ServletConfig接口。 继承自GenericServlet的Servlet应该要覆盖service()方法。最后，为了开发一个能用在网页上服务于使用HTTP协议请求的Servlet，你的Servlet必须要继承自HttpServlet。

50. 解释下 Servlet 的生命周期。

对每一个客户端的请求，Servlet引擎载入Servlet，调用它的init()方法，完成Servlet的初始化。然后，Servlet对象通过为每一个请求单独调用service()方法来处理所有随后来自客 户端的请求，最后，调用Servlet的destroy()方法把Servlet删除掉。

51. doGet()方法和 doPost()方法有什么区别?

Doget:(1)重定向跳转使用 URL 重写的方式进行值的传递,值显示在浏览器地址栏中(2) 重定向适合传递不敏感的数据以及简单的字符串数字等基本数据类型 (3) 重定向跳转后浏览器的地址显示的是跳转目标的 URL（地址栏发生了改变） (4) 重定向不会引起表单的重复提交 (5) 重定向本质上是服务器产生 302 响应，在消息报文中增加 location 键值 对，客户端获得 location 的值并向服务器发起第二次请求。 (6) 实现方式调用 request 的 sendRedirect()方法,不可以通过 setAttribute()设置值

请求转发： (1) 请求转发使用HttpServletRequest对象的setAttribute方法进行值传递，值不会显示在地址栏中(2) 请求转发跳转传值方式适合传递敏感数据以及对象以及对象、数组、集合 等类型的数据 (3) 请求转发跳转后地址栏不会显示跳转的目标的 URL(地址栏不会发生改变) (4) 请求转发跳转会引起表单的重复提交 (5) WEB-INF 下的jsp 页面不能被客户端直接访问，只能通过服务器跳转访 问，因此访问页面只能通过请求转发跳转访问，不能通过重定向（本质是 客户端请求） (6) 实现方式是调用 request 的 getRequestDispatch.forward()方法

52. 什么是 Web 应用程序?

Web应用程序是对Web或者是应用服务器的动态扩展。有两种类型的Web应用：面向表现的和面向服务的。面向表现的Web应用程序会产生包含了很多种标记语言和动态内容的 交互的web页面作为对请求的响应。而面向服务的Web应用实现了Web服务的端点(endpoint)。一般来说，一个Web应用可以看成是一组安装在服务器URL名称空间的特定 子集下面的Servlet的集合。

53. 什么是服务端包含(Server Side Include)?

服务端包含(SSI)是一种简单的解释型服务端脚本语言，大多数时候仅用在Web上，用servlet标签嵌入进来。SSI最常用的场景把一个或多个文件包含到Web服务器的一个Web页面中。当浏览器访问Web页面的时候,Web服务器会用对应的servlet产生的文本来替换Web页面中的servlet标签。

54. 什么是 Servlet 链(Servlet Chaining)?

Servlet链是把一个Servlet的输出发送给另一个Servlet的方法。第二个Servlet的输出可以发送给第三个Servlet，依次类推。链条上最后一个Servlet负责把响应发送给客户端。

55. 如何知道是哪一个客户端的机器正在请求你的 Servlet?

ServletRequest类可以找出客户端机器的IP地址或者是主机名。getRemoteAddr()方法获取客户端主机的IP地址，getRemoteHost()可以获取主机名

56. 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。

57. 什么是 cookie？session 和 cookie 有什么区别？

cookie是Web服务器发送给浏览器的一块信息。浏览器会在本地文件中给每一个Web服务器存储cookie。以后浏览器在给特定的Web服务器发请求的时候，同时会发送所有为该服务器存储的cookie。下面列出了session和cookie的区别：

无论客户端浏览器做怎么样的设置，session都应该能正常工作。客户端可以选择禁用cookie， 但是，session仍然是能够工作的，因为客户端无法禁用服务端的session。在存储的数据量方面session和cookies也是不一样的。session能够存储任意的Java对象， cookie只能存储String类型的对象。

58. 浏览器和 Servlet 通信使用的是什么协议？

HTTP协议

59. 什么是 HTTP 隧道？

HTTP隧道是一种利用HTTP或者是HTTPS把多种网络协议封装起来进行通信的技术。因此，HTTP协议扮演了一个打通用于通信的网络协议的管道的包装器的角色。把其他协议的 请求掩盖成HTTP的请求就是HTTP隧道。

60. sendRedirect()和 forward()方法有什么区别？

sendRedirect()方法会创建一个新的请求，而forward()方法只是把请求转发到一个新的目标 上。重定向(redirect)以后，之前请求作用域范围以内的对象就失效了，因为会产生一个新的请求，而转发(forwarding)以后，之前请求作用域范围以内的对象还是能访问的。一般认为sendRedirect()比forward()要慢。

61. 什么是 URL 编码和 URL 解码？

URL编码是负责把URL里面的空格和其他的特殊字符替换成对应的十六进制表示，反之就 是解码。

62. 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。

63. 什么是 JSP 指令(Directive)? JSP 中有哪些不同类型的指令?

Directive是当JSP页面被编译成Servlet的时候，JSP引擎要处理的指令。Directive用来设置页面级别的指令，从外部文件插入数据，指定自定义的标签库。Directive是定义在 <%@ 和 %>之间的。下面列出了不同类型的Directive：

包含指令(Include directive)：用来包含文件和合并文件内容到当前的页面。

页面指令(Page directive)：用来定义JSP页面中特定的属性，比如错误页面和缓冲区。 Taglib指令： 用来声明页面中使用的自定义的标签库。

64. 什么是 JSP 动作(JSP action)?

JSP动作以XML语法的结构来控制Servlet引擎的行为。当JSP页面被请求的时候，JSP 动作会被执行。它们可以被动态的插入到文件中，重用JavaBean组件，转发用户到其他的 页面，或者是给Java插件产生HTML代码。下面列出了可用的动作：

jsp:include--当JSP页面被请求的时候包含一个文件。

jsp:useBean--找出或者是初始化Javabean。

jsp:setProperty--设置JavaBean的属性。

jsp:getProperty--获取JavaBean的属性。

jsp:forward--把请求转发到新的页面。

jsp:plugin--产生特定浏览器的代码。

65. 什么是表达式(Expression)?

JSP表达式是Web服务器把脚本语言表达式的值转化成一个String对象，插入到返回给客 户端的数据流中。表达式是在<%=和%>这两个标签之间定义的。

66. 隐含对象是什么意思？有哪些隐含对象？（内置对象）

JSP隐含对象是页面中的一些Java对象，JSP容器让这些Java对象可以为开发者所使用。开发者不用明确的声明就可以直接使用他们。JSP隐含对象也叫做预定义变量。下面列出 了JSP页面中的隐含对象:

application

pag

request

response

session

exception

out

config

pageContext

67. 面向对象软件开发的优点有哪些?

代码开发模块化，更易维护和修改。

代码复用。

增强代码的可靠性和灵活性。

增加代码的可理解性。

面向对象编程有很多重要的特性，比如：封装，继承，多态

68. 封装的定义和好处有哪些?

封装的基本原则：

1.将实例变量标记为私有的(private)

2.提供公有(public)的setter和getter方法，用来控制对实例变量的存取动作this关键字。

当方法中参数的变量名和实例变量的变量名相同时，则需要使用this.变量名来表示实例变量。

private修饰符：被private修饰符修饰的实例变量或方法，只能在该类的内部使用。

下面列出了使用封装的一些好处：

通过隐藏对象的属性来保护对象内部的状态。

提高了代码的可用性和可维护性，因为对象的行为可以被单独的改变或者是扩展。

禁止对象之间的不良交互提高模块化。

69. 多态的定义?

父类引用指向子类对象

接口指向实现类对象

父类引用指向子类对象时，该对象只能调用父类中定义有的方法，如果该方法在子类中得到了重

写，那么就调用重写后的方法，如果没有重写那么就调用父类中原定义的方法。该对象不能调用子类中特有的方法

高内聚，低耦合

当父类作为方法的参数时，可以传入父类对象及其子类对象

当父类作为方法的返回值时，可以返回父类对象及其子类对象

当父类作为方法的返回值时，并且方法返回的是具体的子类对象，在调用的时候可以使用父类对象接收（不能调用子类特有的方法）或者使用对应的子类对象接收（需要强制类型转换，可以使用子类特有的方法）

当强制转换的类型不匹配时，会抛出异常

70. 构造方法

构造方法是方法的一种特殊形式

构造方法唯一的调用方式在创建类的对象的时候，通过new关键字

构造方法必须和类名相同，并且大小写都得一样

构造方法不能有返回值，连void都不能有

构造方法是可以有参数列表

当一个类没有显示声明一个构造方法时，JVM在运行时会自动创建一个参数列表为空、方法体为空的构造方法。

当开发者声明了类的构造方法，那么JVM就不会在运行时自动创建构造方法

构造方法可以有参数，如果参数是类必须的，那么可以在构造方法的参数列表中出现；如果参数是可选的，那么可以考虑使用set方法

构造方法可以调用普通方法，普通方法不可以调用构造方法

构造方法之间的调用是通过this(参数列表)的形式调用

构造方法调用其他构造方法时必须是方法的第一行代码

构造方法调用其他构造方法时，不能将类的属性作为参数进行传递

71. 面向对象的理解

面向对象是为了实现代码复用

要想使用类中的方法或者属性，必须先声明类的对象，通过对象来调用类中的属性和方法

对象具有类中定义的属性和方法，通过.操作符来调用类的属性和方法。

类可以有多个对象

类是对象的蓝图，模版，类是引用数据类型

我们通过对象的属性来区分对象

72. 继承的作用的定义

继承的目的是实现代码的复用 ○

继承多层次继承 ○

只能有一个直接父类 ○

继承具有单根性和传递性 ○

子类也具有间接父类的属性和方法 ○

通过extends关键字实现继承 ○

子类可以调用父类的属性和方法，但是父类不能调用子类的属性和方法

在java中如果没有显式的使用extends关键字声明类的父类时，那么该类的父类为

在子类中通过super关键字调用父类的属性和方法,反之父类不能调用子类的属性和方法

创建子类对象时必先创建父类的对象，默认调用父类无参的构造方法

如果父类没有无参的构造方法，那么子类的构造方法中必须指定调用父类的哪个构造方法来创建父类的对象

通过super(参数列表)的形式在子类构造方法中调用父类的构造方法,同类中使用this(参数列表)调

用其他的构造方法，该代码都必须是构造方法的第一行代码

73. 抽象的定义？抽象和封装的不同点？

抽象是把想法从具体的实例中分离出来的步骤，因此，要根据他们的功能而不是实现细节来 创建类。Java支持创建只暴漏接口而不包含方法实现的抽象的类。这种抽象技术的主要目的是把类的行为和实现细节分离开。

抽象和封装是互补的概念。一方面，抽象关注对象的行为。另一方面，封装关注对象行为的

细节。一般是通过隐藏对象内部状态信息做到封装，因此，封装可以看成是用来提供抽象的 一种策略。