**脏读：**（针对未提交数据）如果一个事务中对数据进行了更新，但事务还没有提交，另一个事务可以“看到”该事务没有提交的更新结果。

**幻读：**（针对其他事务提交前后，读取数据条数（新增或者删除）的对比）幻读是指同样一笔查询在整个事务过程中多次执行后，查询所得的结果集是不一样的。

**不可重复读：**（针对其他事务提交前后，读取数据本身的对比）不可重复读取是指同一个事务在整个事务过程中对同一笔数据进行读取，每次读取结果都不同。

Java中8种基本数据类型 byte、short、int、long、float、double、boolean、char。

其中整型有四个，分别为byte、short、int、long

pageContext和page都是jsp中的隐含对象，pageContext代表jsp页面的上下文关系，能够调用、存取其他隐含对象；page代表处理当前请求的时候，这个页面的实现类的实例。

**向上转型：**用父类引用执行子类对象，并可以用父类引用调用子类中重写了的同名方法。但是不能调用子类中新增的方法。

**动态绑定：**在执行期间（程序运行期间）判断所用对象的实际类型，根据其实际的类型调用其相应的方法（确定选择哪个多态形式被调用）。

**JAVA多线程实现的三种方式：**

1、继承Thread类，重写run方法

2、实现Runnable接口，重写run方法，实现Runnable接口的实现类的实例对象作为Thread构造函数的target（Thread t1 = new Thread(new MyThread());）

3、通过Callable和FutureTask创建线程

4、通过线程池创建线程a. newSingleThreadExecutor

创建一个单线程的线程池。这个线程池只有一个线程在工作，单线程串行执行所有任务。如果线程因为异常结束，那么会有一个新的线程来替代它。此线程池保证所有任务的执行顺序按照任务的提交顺序执行。

b.newFixedThreadPool

创建固定大小的线程池。每次提交一个任务就创建一个线程，线程池的大小一旦达到最大值就会保持不变，如果某个线程因为执行异常而结束，那么线程池会补充一个新线程。

c. newCachedThreadPool

创建一个可缓存的线程池。会回收部分空闲（60秒不执行任务）的线程，当任务数增加时，可以添加新线程来处理任务。此线程池不会对线程池大小做限制，线程池大小完全依赖于操作系统（或者说JVM）能够创建的最大线程大小。

d.newScheduledThreadPool

创建一个大小无限的线程池。此线程池支持定时以及周期性执行任务的需求。

**volitile关键字的作用，原理：**

1、保证了不同线程对这个变量进行操作时的可见性，即一个线程修改了某个缓存中变量的值后立即写入主存，并使缓存行无效，其他线程需要读取主存中的值。

2、禁止进行指令重排序。

如果没有对equals方法进行重写，则比较的是引用类型的变量所指向的对象的地址；

诸如String、Date等类对equals方法进行了重写的话，比较的是所指向的对象的内容。

++a在表达式中是先++，再取值。b--在表达式中是先取值，再--。所以表达式：++a\*b-- 相当于 (a+1)\*b

**连表结构：**物理存储单元上非连续、非顺序的存储结构，数据元素的逻辑顺序是通过链表中的指针链接次序实现的。

**双向链表：**每个数据结点中都有两个指针，分别指向直接后继和直接前驱。

**循环链表：**最后一个结点的[指针](https://baike.baidu.com/item/%E6%8C%87%E9%92%88/2878304)域指向[头结点](https://baike.baidu.com/item/%E5%A4%B4%E7%BB%93%E7%82%B9/9794729)，整个链表形成一个环。

**TCP的优点：**可靠，稳定 TCP的可靠体现在TCP在传递数据之前，会有三次握手来建立连接，而且在数据传递时，有确认、窗口、重传、拥塞控制机制，在数据传完后，还会断开连接用来节约系统资源。TCP的缺点：慢，效率低，占用系统资源高，易被攻击 TCP在传递数据之前，要先建连接，这会消耗时间，而且在数据传递时，确认机制、重传机制、拥塞控制机制等都会消耗大量的时间，而且要在每台设备上维护所有的传输连接，事实上，每个连接都会占用系统的CPU、内存等硬件资源。而且，因为TCP有确认机制、三次握手机制，这些也导致TCP容易被人利用，实现DOS、DDOS、CC等攻击。

**UDP的优点：**快，比TCP稍安全UDP没有TCP的握手、确认、窗口、重传、拥塞控制等机制，UDP是一个无状态的传输协议，所以它在传递数据时非常快。没有TCP的这些机制，UDP较TCP被攻击者利用的漏洞就要少一些。但UDP也是无法避免攻击的，比如：UDP Flood攻击…… UDP的缺点： 不可靠，不稳定 因为UDP没有TCP那些可靠的机制，在数据传递时，如果网络质量不好，就会很容易丢包。 基于上面的优缺点，那么： 什么时候应该使用TCP： 当对网络通讯质量有要求的时候，比如：整个数据要准确无误的传递给对方，这往往用于一些要求可靠的应用，比如HTTP、HTTPS、FTP等传输文件的协议，POP、SMTP等邮件传输的协议。 在日常生活中，常见使用TCP协议的应用如下：浏览器，用的HTTP FlashFXP，用的FTP Outlook，用的POP、SMTP Putty，用的Telnet、SSH QQ文件传输 ………… 什么时候应该使用UDP： 当对网络通讯质量要求不高的时候，要求网络通讯速度能尽量的快，这时就可以使用UDP。 比如，日常生活中，常见使用UDP协议的应用如下： QQ语音 QQ视频 TFTP ……

**同步：**串行访问方式，是多个线程同时访问同一资源，其它线程需要等待资源访问结束

**异步：**并行访问方式，线程访问资源时不必等待一个任务执行完，在等待时可以访问其他资源。

脚本语言是一种解释性的语言,例如[Python](https://baike.baidu.com/item/Python)、vbscript,javascript,installshield script,ActionScript等等,它不象c\c++等可以编译成二进制代码,以[可执行文件](https://baike.baidu.com/item/%E5%8F%AF%E6%89%A7%E8%A1%8C%E6%96%87%E4%BB%B6)的形式存在，脚本语言不需要编译，可以直接用，由解释器来负责解释。