**设计模式**

1．答：

在连接次数少的情况下，String的低效率表现并不是很突出，但是一旦连接次数多的时候，性能影响是很大的，String进行20万次字符串的连接，大约需要45s时间，而StringBuffer只需要10毫秒，相差接近5000倍以上。原因是Java中的String类不可变的。string类型具有不可变性，对string字符串的操作（如拼接、Trim()等）都会在内存中产生一个新的字符串对象，在对字符串进行频繁修改的情况下，如在For循环中进行操作等，那么将会频繁的创建新的字符串对象，造成系统的不必要开销。

2.答：

把对象转换为字节序列的过程称为对象的序列化；把字节序列恢复为对象的过程称为对象的反序列化。在很多应用中，需要对某些对象进行序列化，让它们离开内存空间，入住物理硬盘，以便长期保存。比如最常见的是Web服务器中的Session对象，当有 10万用户并发访问，就有可能出现10万个Session对象，内存可能吃不消，于是Web容器就会把一些seesion先序列化到硬盘中，等要用了，再把保存在硬盘中的对象还原到内存中。那么为什么要序列化呢？我们知道，当两个进程进行远程通信时，可以相互发送各种类型的数据，包括文本、图片、音频、视频等， 而这些数据都会以二进制序列的形式在网络上传送。那么当两个Java进程进行通信时，能否实现进程间的对象传送呢？答案是可以的。如何做到呢？这就需要Java序列化与反序列化了。换句话说，一方面，发送方需要把这个Java对象转换为字节序列，然后在网络上传送；另一方面，接收方需要从字节序列中恢复出Java对象。