

序列化是一种将java对象流化的技术，这样就可以进行读写、保存。

它可以有以下应用场景：

1. 远程调用（RMI）需要将对象或参数转化为字节流用于网络传输。利用对象的串行化实现保存应用程序的当前工作状态，下次再启动的时候将自动地恢复到上次执行的状态。

writeObject和readObject本身就是线程安全的，传输过程中是不允许被并发访问的。所以对对象能一个一个接连不断的传过来

Serializable接口只是一个标记接口，不包括任何方法。

序列化的实现：将需要被序列化的类实现Serializable接口，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。

声明为static和transient类型的成员数据不能被串行化。因为static代表类的状态， transient代表对象的临时数据；

对于父类的处理，如果父类没有实现串行化接口，则其必须有默认的构造函数（即没有参数的构造函数）。否则编译的时候就会报错。在反串行化的时候，默认构造函数会被调用。但是若把父类标记为可以串行化，则在反串行化的时候，其默认构造函数不会被调用。这是为什么呢？这是因为Java 对串行化的对象进行反串行化的时候，直接从流里获取其对象数据来生成一个对象实例，而不是通过其构造函数来完成。

若某个特定的类添加了writeObject、readObject方法，则会执行自己的，而不会执行默认的。还要另外一个技巧，如果想在writeObject中调用默认的方法，需使用defaultWriteObject方法。