Transient 关键字 private transient String password;（密码重要，不能用默认的序列化）

private void writeObject(ObjectOutputStream stream)throws IOException {

stream.defaultWriteObject(); //先按默认方式序列化

stream.writeObject(change(password.getBytes()));--🡪>>>自定义序列化

stream.writeInt(count);

}

private byte[] change(byte[] buff){

for(int i=0;i<buff.length;i++){

int b=0;

for(int j=0;j<8;j++){

int bit=(buff[i]>>j & 1)==0 ? 1:0;

b+=(1<<j)\*bit;

}

buff[i]=(byte)b;

}

return buff;

}

自定义序列化过程 ------>.>通过Java再带的序列化框架

（1）自定义一个方法叫writeobject（）；

（2）在上述方法内部先调用outputstream.defaultWriteObject()方法

（3）在定义自己的序列化过程

单例类的反序列化

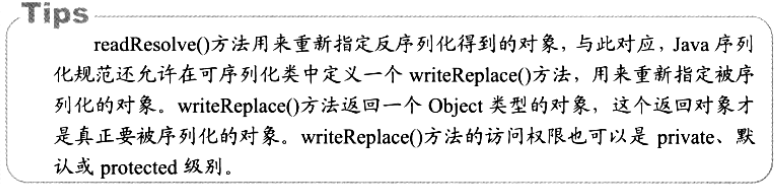
单例在整个程序运行过程中应该只有一个实例，而反序列化可以打破这个状态-------->>>在要反序列化的类中实现

private Object readResolve() throws ObjectStreamException{

return INSTANCE;

}

在反序列化的过程中，会调用这个方法。。。。所有实现单实例只要在该方法中返回原对象的实例即可



为了保证不同的序列化之间兼容或者不兼容，可以显示的制定序列化的serialVersionUID的值-------》在类中添加 private static final long serivalVersionUID=-1293819L

定制的序列化除了通过实现serizable接口外，还可以实现Externalizable接口，通过接口定义的writeExternal()和readExternal()方法实现序列化，不过通过实现此接口的可序列化类必须实现无参的构造方法