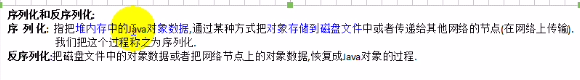
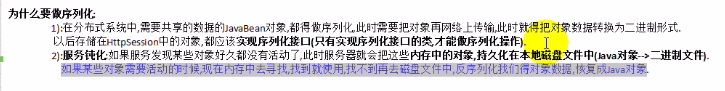
# Day21

## 01序列化和反序列化概述



序列化是java对象到可存储或可网络传输对象之间的转化，相反则是反序列化。

实质就是java对象到二进制形式之间的转换。



需要做序列化的对象要实现序列化接口：java.lang.Serializable



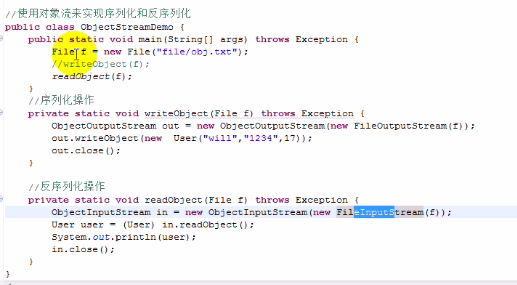
不包含抽象方法的接口是标志接口，用来限制类型（底层在使用的时候即在进行序列化的操作的时候，会通过instanceof判断是否实现了此接口；就如同对象clone时需要实现Cloneable，Object.clone调用的时候会判断是否实现了Cloneable接口一样）

## 02io对象流

**使用对象流来实现序列化和反序列化操作**

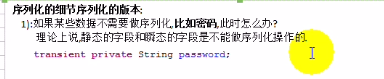
ObjectInputstream：反序列化

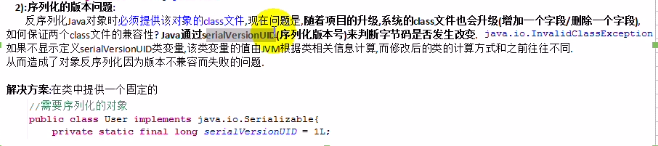
ObjectOutstream：序列化



反序列化一定要存在字节码对象，这样才能将object转换为指定的对象

## 03对象操作流和serialversionuid





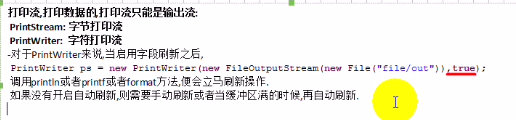
在类中指定一个固定的serilVersionUID时，如果序列化后的对象的类发生了改变(如增加/删除了字段或者方法等)，则能够保证序列化的对象仍然能够被反序列化。

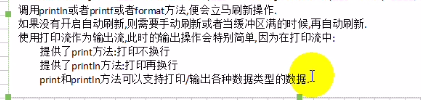
Private static final long serialVersionUID=1L;

## 04打印流-printwrite

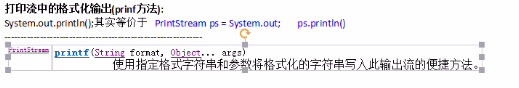
打印字节

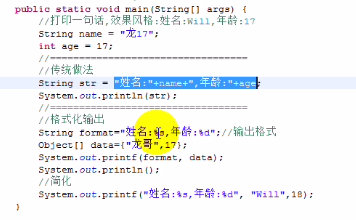
打印字符





## 05打印流的格式化输出-printf





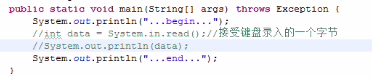
## 06标准io概述和操作

标准输入：通过键盘输入数据给程序

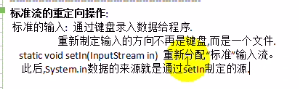
标准输出：通过屏幕显示程序数据

Inputstream =system.in

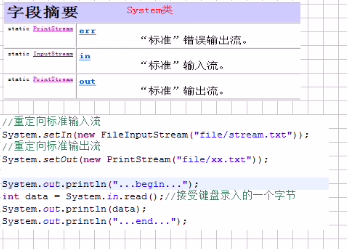
Outputstream=system.out



标准流的重定向：

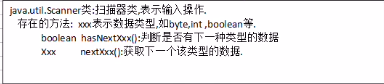






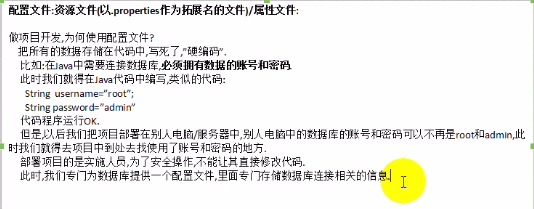
## 07扫描器类

Scanner类：扫描器类，表示输入操作

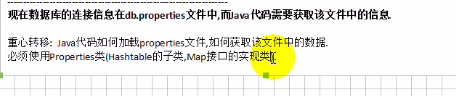




## 08properties类加载资源文件





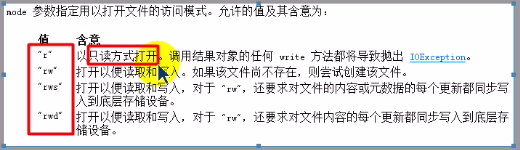


## 09数据流



## 10随机访问文件

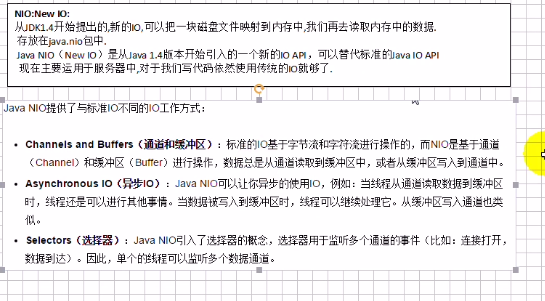
RandomAccessFile：指的是在文件的任何位置读取和写入数据



## 12nio概述

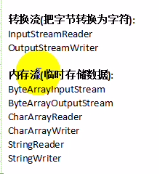
Nio:new io

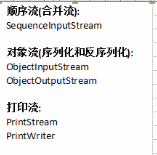
从jdk1.4开始提出的，他可以将磁盘的文件映射到内存中，我们再去读取内存中的数据

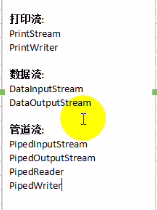


## 13io流的回顾









基本流：



## 14今日小结

