

数据流只能实现对基本数据类型和字符串类型的读写，并不能读取对象(字符串除外)。要对对象进行读写：

ObjectInputStream/ObjectOutputStream。

ObjectInputStream/ObjectOutputStream是以“对象”为数据源：

1. 对象流不仅可以读写对象，还可以读写基本数据类型。
2. 使用对象流读写对象时，该对象必须序列化与反序列化。
3. 系统提供的类(如Date等)已经实现了序列化接口，自定义类必须手动实现序列化接口。

ObjectInputStream/ObjectOutputStream的使用

```
import java.io.BufferedInputStream; import java.io.BufferedOutputStream;
import java.io.File; import java.io.FileInputStream; import java.io.FileOutputStream;
import java.io.IOException; import java.io.InputStream; import java.io.ObjectInput
import java.io.ObjectOutputStream; import java.io.OutputStream; import java.util.I
public class TestObjectStream {
    public static void main(String[] args) throws IOException, ClassNotFoundException {
        write();      read();      }    /**使用对象输出流将数据写入文件*/
    public static void write() {      // 创建Object输出流，并包装缓冲流，增加缓冲功能
        OutputStream os = null;      BufferedOutputStream bos = null;
        ObjectOutputStream oos = null;      try {
            os = new FileOutputStream(new File("d:/bjsxt.txt"));
            bos = new BufferedOutputStream(os);
            oos = new ObjectOutputStream(bos);      // 使用Object输出流
            //对象流也可以对基本数据类型进行读写操作      oos.writeInt(12);
            oos.writeDouble(3.14);      oos.writeChar('A');
            oos.writeBoolean(true);      oos.writeUTF("北京尚学堂");
            //对象流能够对对象数据类型进行读写操作      //Date是系统提供的类，已经实现了序列化接口
            //如果是自定义类，则需要自己实现序列化接口      oos.writeObject(new Date());
        } catch (IOException e) {      e.printStackTrace();      } finally {
            //关闭输出流      if(oos != null) {      try {
                oos.close();      } catch (IOException e) {
                e.printStackTrace();      }
            }
            if(bos != null) {      try {      bos.close();
            } catch (IOException e) {      e.printStackTrace();
            }      }
            if(os != null) {      try {
                os.close();      } catch (IOException e) {
                e.printStackTrace();      }
            }
        }
    }
    /**使用对象输入流将数据读入程序*/    public static void read() {      // 创建ObjectInput
        InputStream is = null;      BufferedInputStream bis = null;
        ObjectInputStream ois = null;      try {
```

```
is = new FileInputStream(new File("d:/bjsxt.txt"));
bis = new BufferedInputStream(is);
ois = new ObjectInputStream(bis);           // 使用Object输入流按照写
System.out.println(ois.readInt());
System.out.println(ois.readDouble());
System.out.println(ois.readChar());
System.out.println(ois.readBoolean());
System.out.println(ois.readUTF());
System.out.println(ois.readObject().toString());
} catch (ClassNotFoundException e) {          e.printStackTrace();
} catch (IOException e) {                      e.printStackTrace();          } final
// 关闭Object输入流                          if(ois != null){                  try {
    ois.close();                              } catch (IOException e) {
    e.printStackTrace();                      }
if(bis != null){                              try {                                bis.close()
    } catch (IOException e) {                  e.printStackTrace()
    }                                          }
    is.close();                              } catch (IOException e) {
    e.printStackTrace();                      }
}
```