**用对象流输出多个对象，然后再读回来的方法？**

**学生类：**

**public** **class** Student **implements** Serializable{

**int** age;

String name;

**public** Student(){}

**public** Student(**int** age,String name){

**this**.age=age;

**this**.name=name;

}

}

**利用对象数组进行读写：**

**将Student对象放入对象数组中一起写出去**

**public** **void** writer() **throws** Exception{

File f=**new** File("src/a.txt");

FileOutputStream fos=**new** FileOutputStream(f);

ObjectOutputStream oos=**new** ObjectOutputStream(fos);

Student[] stu={**new** Student(21,"aa"),**new** Student(20,"bb")};

oos.writeObject(stu);

oos.flush();

oos.close();

}

**读回来的是Object，强转即可得Student数组**

**public** **void** reader() **throws** Exception{

File f=**new** File("src/a.txt");

FileInputStream fis=**new** FileInputStream(f);

ObjectInputStream ois=**new** ObjectInputStream(fis);

Student[] stu = (Student[])ois.readObject();

**for**(Student s:stu){

System.*out*.println(s.name);

}

}

**方法二：一个对象一个对象放在同一个流中写出去**

**public** **void** writer() **throws** Exception{

File f=**new** File("src/a.txt");

FileOutputStream fos=**new** FileOutputStream(f);

ObjectOutputStream oos=**new** ObjectOutputStream(fos);

oos.writeObject(**new** Student(21,"aa"));

oos.writeObject(**new** Student(20,"bb"));

oos.writeObject(**null**); **//对象写完后需加写一个null 作为结束标志**

oos.flush();

oos.close();

}

**public** **void** reader() **throws** Exception{

File f=**new** File("src/a.txt");

FileInputStream fis=**new** FileInputStream(f);

ObjectInputStream ois=**new** ObjectInputStream(fis);

Object temp = **null**;

**while**((temp = ois.readObject())!=**null**){ **//然后此处通过null判断结束就没有问题，否则用此方法读时 会报EOFException**

System.*out*.println(((Student)temp).name);

}

}

**或者用available判断是否达到了文件末尾**：

**public** **void** writer() **throws** Exception{

File f=**new** File("src/a.txt");

FileOutputStream fos=**new** FileOutputStream(f);

ObjectOutputStream oos=**new** ObjectOutputStream(fos);

oos.writeObject(**new** Student(21,"aa"));

oos.writeObject(**new** Student(20,"bb"));

**//此处没有多写null**

oos.flush();

oos.close();

}

**public** **void** reader() **throws** Exception{

File f=**new** File("src/a.txt");

FileInputStream fis=**new** FileInputStream(f);

ObjectInputStream ois=**new** ObjectInputStream(fis);

**while**(fis.available()>0){

Object temp = ois.readObject();

System.*out*.println(((Student)temp).name);

}

}

**备注：不能对象流重复将对象输出到一个文件中：**

写出的方法：

**public** **void** writer() **throws** Exception{

File f=**new** File("src/a.txt");

FileOutputStream fos=new FileOutputStream(f,true);**// 不要对一个文件进行对象流的append操作 要改成new FileOutputStream(f,false)**

ObjectOutputStream oos=new ObjectOutputStream(fos);

oos.writeObject(**new** Student(21,"aa"));

oos.flush();

oos.close();

}

读取的方法：

**进行append操作的话（writer方法执行两次，意图通过将多个对象写入同一个文件中）存入文件中的二进制会乱掉，读取时会报StreamCorruptedException异常**

**java.io.StreamCorruptedException API中解释**：当从对象流中读取的控制信息与内部一致性检查相冲突时，抛出此异常 （**因为没有插入结束标志造成的**）

public void reader() throws Exception{

File f=**new** File("src/a.txt");

FileInputStream fis=**new** FileInputStream(f);

ObjectInputStream ois=**new** ObjectInputStream(fis);

**while**(fis.available()>0){

Object temp = ois.readObject();

System.*out*.println(((Student)temp).name);

}

}