

## Java.SE.03.IO Streams

### 3.1 (required)

Для задания **Java.SE.01.Object-oriented programming in Java**, сохранить объекты приложения в одном или нескольких файлах с применением механизма сериализации. Объекты могут содержать поля, помеченные как **static**, а также **transient**. Для изменения информации и извлечения информации в файле создать специальный класс-коннектор с необходимыми для выполнения этих задач методами.

при реализации интерфейса `Serializable`  
у каждого класса должно быть поле  
**private static final long** *serialVersionUID*  
для чего оно нужно? что произойдет, если этого поля нет?

правильно пользуйтесь `ObjectInputStream` при чтении неизвестного числа объектов из потока `readObject()` вернет `null` только если в поток ранее был `null` записан.  
по достижении конца файла вбрасывается исключение `java.io.EOFException`  
обработайте правильно эту ситуацию  
пример:  
<http://www.javadb.com/reading-objects-from-file-using-objectinputstream>

### 3.2 (additionally)

Написать собственный декоратор ввода (расширить класс `BufferedReader`, дополнив метод `readLine()` ), который заменяет табуляции и последовательности пробелов одним пробелом.

Для задания **Java.SE.02.Information Handling**, получить текст из текстового файла используя собственный декоратор. Результат выполнения задания также сохранять в файл.

при использовании потока чтения/записи в обоих заданиях желательно придерживаться сценария

чтение:	запись:
<pre>try{     создание+открытие     try{         чтение/запись     }catch(...){         обработка искл     }finally{         закрытие     } }catch(...){     обработка искл }</pre>	<pre>создание+открытие try{     чтение/запись }catch(...){     обработка искл }finally{     закрытие }</pre>

Или

чтение:	запись:
<pre>try(создание+открытие){     чтение/запись }catch(...){     обработка искл }</pre>	<pre>try(создание+открытие){     чтение/запись }catch(...){     обработка искл }</pre>