

Objetivos:

- Repasar y profundizar en tratamiento de flujos: ficheros

Contenido:

Jerarquía de clase de Java para Flujos en Java

Jerarquía de clase para flujos de entrada:

Flujos con bytes	Flujos con caracteres
InputStream	Reader
ByteArrayInputStream	BufferedReader
FileInputStream	LineNumberReader
FilterInputStream	CharArrayReader
BufferedInputStream	FilterReader
DataInputStream	PushBackReader
LineNumberInputStream	InputStreamReader
PushBackInputStream	FileReader
ObjectInputStream	PipedReader
PipedInputStream	StringReader
SequenceInputStream	
StringBufferInputStream	

Jerarquía de clase para flujos de salida:

Flujos con bytes	Flujos con caracteres
OutputStream	Writer
ByteArrayOutputStream	BufferedWriter
FileOutputStream	CharArrayWriter
FilterOutputStream	FilterWriter
BufferedOutputStream	OutputStreamWriter
DataOutputStream	FileWriter
PrintStream	PipedWriter
ObjectOutputStream	PrintWriter
PipedOutputStream	StringWriter

Ejercicio 1: Ejemplos básicos de Flujos (30 m)

Se proporciona en el bitbucket de la asignatura tres ficheros Java.

1. EjemploFicheroBinario donde se salva y carga información en formato binario. A su vez, dependiendo del patrón decorator usado en la instanciación, se podrá cargar o salvar Byte a byte (FileInputStream) o mediante Buffered new BufferedInputStream(new FileInputStream()).
2. EjemploFicheroTextual donde se salva y carga información en formato alfanumérico. A su vez, dependiendo del patrón decorator usado en la instanciación, se podrá cargar o salvar Byte a byte (FileReader) o mediante Buffered new BufferedReader(new InputStreamReader(new FileInputStream()))).
3. EjemploFicheroObjeto donde se salva y carga información en formato objeto (mediante serialización).

Ejercicio 2: Flujos en Desguace (30 m)

Dada la práctica de la sesión 7 con su menu oportuno, se la incorporará las opciones de guardar y salvar la información del desguace usando: ficheros y serialización.

Por una parte se salvarán las piezas del desguace en un fichero textual.

1. **Piezas.txt:** que contiene la información del repositorio de piezas y que se organiza como sigue. Cada línea del fichero tiene la información de una pieza con los campos separados por comas. De este modo sería: <id>,<nombre>,<contador> → Por ejemplo:

1, Faro laguna 2004, 2
2, Faro laguna 2006, 3
3, Faro Audi 6 2000, 1

Por otra parte se salvarán el resto de la información en un fichero binario.

2. **Datos.dat:** En este caso se va a serializar el desguace entero, excepto el catálogo de piezas. Para indicar que un atributo no debe salvarse se usa **transient**. En el resto de clase se debe indicar que implements serializable).

La aplicación deberá crear un menú que permita:

1. Cargar y salvar las piezas (código en Desguace)
2. Cargar y salvar todo el desguace (código en el main)

Nota: Se deberá controlar cualquier excepción que se produzca en la entrada/salida de los ficheros y decidir en consecuencia las acciones a realizar.