

Ejercicios Java

Ficheros

Ejercicio 1 - Obtener información de ficheros y directorios	1
Ejercicio 2 - Leer de <code>FileInputStream</code> con y sin buffer	1
Ejercicio 3 - Lectura de fichero de texto	1
Ejercicio 4 - Escritura en fichero binario	1
Ejercicio 5 - Fichero binario a fichero de texto	2
Ejercicio 6 - Lectura de ficheros de texto teniendo en cuenta la codificación de caracteres	2
Ejercicio 7 - Lemario	2
Ejercicio 8 - Ampliación del Torneo de Pokémon con ficheros	2
Ejercicio 9 - Serialización	2
Ejercicio 10 -Ampliación de la agenda con ficheros	2

Ejercicio 1 - Obtener información de ficheros y directorios

Programa que reciba a través de un parámetro de línea de comando una ruta dentro del sistema de archivos y muestre en la consola la información siguiente:

- Si existe el fichero o directorio referenciado.
- Si se trata de un fichero o de un directorio.
- Nombre del fichero o directorio y permisos de acceso.
- Si se trata de un fichero, debe mostrar además el tamaño del mismo.

Ejercicio 2 - Leer de `FileInputStream` con y sin buffer

Programa que muestre la diferencia que existe entre utilizar o no utilizar buffer al leer el fichero de texto "El Quijote UTF8.txt" carácter a carácter con un `FileInputStream`.

Ejercicio 3 - Lectura de fichero de texto

Programa que lea un fichero de texto plano y muestre en la consola el número de caracteres, el número de palabras y el número de líneas que contiene, entendiendo por palabra cualquier secuencia de caracteres que no contenga espacios. La ruta al fichero se puede especificar en un parámetro de línea de comando. Si no se especifica de esa forma, se pedirá por teclado.

Ejercicio 4 - Escritura en fichero binario

Crea una segunda versión del programa anterior en la que, en lugar de mostrar los datos en la consola, los almacene en un fichero binario (ruta, número de caracteres, número de palabras y número de líneas). El fichero binario podrá contener los datos de varios ficheros de texto, por tanto, cada vez que se ejecute este programa los nuevos datos se agregarán a los ya existentes.

Ejercicio 5 - Fichero binario a fichero de texto

Crea un programa que lea del fichero binario generado en el ejercicio anterior y vuelque los datos a un fichero de texto. Cada registro leído del fichero binario (ruta, número de caracteres, número de palabras y número de líneas) se escribirá en una línea en el fichero de texto.

Ejercicio 6 - Lectura de ficheros de texto teniendo en cuenta la codificación de caracteres

Crea un programa que lea el fichero "El Quijote UTF8.txt" o el fichero "El Quijote Windows-1252.txt" (el usuario elegirá uno de los dos mediante un menú) y muestre su contenido en la consola. Asegúrate de que el contenido se muestra correctamente.

Ejercicio 7 - Lemario

Crea un programa que use un objeto de la clase Palabras, definida en el ejercicio 4 de colecciones, para contener las palabras almacenadas en el fichero "Lemario ES UTF-8.txt". A continuación, el programa almacenará en una colección todas aquellas palabras del fichero "El Quijote UTF-8.txt" que no estén en el leuario sin repetir ninguna. Después mostrarlas en orden alfabético.

Ejercicio 8 - Ampliación del Torneo de Pokémon con ficheros

Escribe un programa que almacene en un fichero de texto los datos necesarios para el desarrollo de un torneo Pokémon a partir de datos leídos por teclado con el formato que se especifica en el ejercicio 2 de la unidad 6.

Modifica el ejercicio 2 de la unidad 6 (Clases, Objetos y Excepciones), Torneo de Pokémon, para que ejecute el torneo leyendo los datos del fichero si recibe la ruta del fichero a través de un parámetro de línea de comando. De no ser así, ejecutará el torneo leyendo los datos de teclado.

Ejercicio 9 - Serialización

A partir de las clases *Publicación* definidas en el ejercicio 1 de la unidad 7, escribe un programa que almacene en un fichero datos de libros y revistas mediante serialización de objetos *Libro* y *Revista*.

Ejercicio 10 - Ampliación de la agenda con ficheros

Ampliar el ejercicio 6 de colecciones de la unidad 8 para que acepte dos comandos más:

- Comando para guardar la agenda en un fichero de texto:

```
> guardar:ruta
```

Los contactos se guardarán en orden alfabético en un fichero de texto, cada contacto en una línea con el formato

nombre-teléfono, teléfono,...

Si el fichero ya existe se mantendrán los contactos almacenados en el mismo que no estén en la agenda.

Si un contacto de la agenda ya existe en el fichero y los teléfonos no coinciden, el teléfono del fichero se sustituye con el que se encuentra en la agenda.

Se mostrará en la consola cualquier error relacionado con el acceso al archivo o la escritura en el mismo.

- Comando para cargar datos de la agenda desde un fichero de texto:

```
> cargar:ruta
```

Se leerán los contactos del fichero y se añadirán a la agenda.

Si un contacto en el fichero ya existe en la agenda y los teléfonos son distintos, el programa preguntará al usuario si desea actualizar el teléfono con el que ha leído del fichero o mantener el teléfono de la agenda.

Se mostrará en la consola cualquier error relacionado con el acceso al archivo o la lectura del mismo.