Tema 6 - Actividades

1. Escribe una clase PruebaFicheros.java en un paquete aparte que emplearemos para realizar pruebas con ficheros. Crea un método pruebaFOS() que utilice la clase FileOutputStream para crear en el directorio `/usuario/FileTest` el fichero ***`pruebaFOS.bin`*** que contenga la palabra “hola”.

2. Sobrecarga el método pruebaFOS() para hacer que se cree el fichero con un nombre recibido por parámetros.

3. Sobrecarga nuevamente el método pruebaFOS() para admita un parámetro con el directorio en el que crear el fichero, y el nombre de dicho fichero.

4. Escribe un método pruebaFIS() que lea el fichero que hayas escrito en el último ejercicio mediante la clase FileInputStream. Deberás obtener el path del fichero por parámetros.

5. Realiza el ejercicio 3, pero empleando un BufferedOutputStream. ¿Encuentras alguna diferencia perceptible en el fichero?

6. Escribe un método pruebaBIS() que lea el fichero que hayas escrito en el último ejercicio mediante la clase BufferedInputStream. Deberás obtener el path del fichero por parámetros.

7. Lee el fichero generado en el ejercicio 5 con los métodos pruebaFIS() y pruebaBIS(). ¿Se produce algún tipo de error o conflicto? ¿A qué se debe esto?

8. Escribe un método pruebaDOS() que escriba el siguiente contenido en un fichero utilizando la clase DataOutputStream:

```
Escribiendo
Escribiendo un byte
Escribiendo un numero especial: 3.14
1 2 3 4
```

Deberás crear el fichero en el path y con el nombre especificados por parámetros.

9. Intenta leer el fichero del ejercicio 8 con pruebaFIS() y pruebaBIS(). ¿Qué ocurre? ¿A qué se debe?

10. Implementa el método pruebaDIS() que lea un fichero con la clase DataInputStream(). Deberá leer el path del fichero por parámetros. ¿Qué observas?

11. Escribe un método pruebaByteOS() que utilice la clase ByteArrayOutputStream para generar un fichero que contenga la palabra JAVA en mayúsculas.

12. Implementa el método pruebaByteIS() que utilice ByteArrayInputStream para leer desde un array de bytes (buffer) obtenido por parámetros).

13. Implementa el método pruebaPrint() que utilice PrintStream para escribir la palabra DAM en un determinado destino dependiendo de un parámetro recibido: bien en un fichero, o bien por pantalla.

14. Escribe un método pruebaSequence() que lea los ficheros generados en el ejercicio 11 y el ejercicio 8, combinando un ByteArrayInputStream y un DataInputStream.

Ficheros de texto

15. Escribe un método testFW() que escriba el siguiente fichero en el path y bajo el nombre obtenido por parámetros, utilizando para ello la clase FileWriter.

```
Examen de Programación - 1DAM
6 de marzo de 2024
```

16. Sobrecarga el método anterior para que vierta en el fichero el contenido de un String recibido por parámetros.

17. Utiliza la clase FileReader en un método testFR() que lea los ficheros generados en los ejercicios anteriores. Debe recibir por parámetros el path de los ficheros.

18. Escribe un método testBW que, mediante la clase BufferedWriter, genere el siguiente fichero en el path y bajo el nombre obtenido por parámetros.

```
Instrumental necesario:

-Equipo portátil.
-Cable de red RJ-45.
-Unidad de almacenamiento externa USB.
```

19. Emplea la clase BufferedReader en un método testBR para leer el fichero generado en el ejercicio anterior de la misma forma que los anteriores lectores. Utiliza opciones para posibilitar también una lectura línea a línea.

20. Sobrecarga nuevamente el método anterior para que devuelva un array de String que contenga en cada casilla un elemento del instrumental necesario.

21. Escribe un método testCAW() que emplee la clase CharArrayWriter para escribir en el fichero especificado por parámetros la fecha actual.

22. Utiliza la clase BufferedReader para convertir la lectura del fichero generado en el ejercicio anterior en un array de caracteres. Pasa dicho array como parámetro a un objeto de la clase CharArrayReader para leerlo y mostrarlo por pantalla.

23. Implementa el método testPrint() que utilice PrintWriter para escribir la cadena “que harto estoy de ficheros” en un determinado destino dependiendo de un parámetro recibido: bien en un fichero, o bien por pantalla.

24. Utilizando un procesador de textos, guarda un fichero con el siguiente contenido.

```
I wake up to the sounds of the silence that allows
For my mind to run around with my ear up to the ground
I'm searching to behold the stories that are told
When my back is to the world that was smiling when I turned

Tell you you're the greatest
But once you turn they hate us

Oh, the misery
Everybody wants to be my enemy
Spare the sympathy
Everybody wants to be my enemy
(Look out for yourself)
My enemy (look, look, look, look)
(Look out for yourself)
But I'm ready
```

Emplea la clase LineNumberReader para leer y mostrar por pantalla de la línea 9 en adelante.

25. Interpreta que es lo que hace el siguiente método:

```
public static String stringWriter(String path){
	char[] buffer = new char[1024];
	Writer stringWriter = new StringWriter();
	try {
		FileReader fReader = new FileReader(path);
		BufferedReader bReader = new BufferedReader(fReader);
		int n;
		while ((n = bReader.read(buffer)) != -1) {
			stringWriter.write(buffer, 0, n);
		}
		bReader.close();
		stringWriter.close();
		fReader.close();
	}catch(IOException ioe){
		System.err.println(ioe.toString());
	}
	return stringWriter.toString();
}
```

26. Escribe un método que, haciendo uso de la clase StringReader, lea el output del método presente en el ejercicio anterior.

Clase File

27. ¿Qué métodos de la clase File hubieran sido útiles para la realización de los ejercicios anteriores? Haz un resumen apoyándote de la documentación de la clase.

28. Modifica los métodos de los ejercicios del 15 al 26 para que utilicen la clase File para la apertura del fichero, y realicen comprobaciones al respecto del mismo. Deberán comprobar que los ficheros que quieren abrir son ficheros como tal, y que además, Netbeans dispone de permisos de lectura/escritura para ellos.

Lectura avanzada de ficheros

Estos ejercicios utilizarán los ficheros de la carpeta “Ficheros” adjuntos a la entrega en Classroom. Asegúrate de realizar todas las comprobaciones pertinentes a la hora de leer/escribir ficheros.

29. Escribe un programa que lea el fichero espronceda.txt y muestre por pantalla todas las líneas que contienen la palabra “barco” o la palabra “pirata”.

30. Escribe un programa que lea el fichero espronceda.txt y escriba en otro fichero esp_mar.txt todas las líneas que contengan la palabra “mar”.

31. Implementa un método que lea el fichero datos.txt y muestre por pantalla los nombres de todos los usuarios que tengan 20 años o menos.

32. Implementa un método que lea el fichero datos.txt y muestre por pantalla los nombres y correos electrónicos de todos los usuarios cuyo apellido empiece por C.

33. Implementa un método que genere un script con inserciones para una base de datos MySQL de acuerdo al formato del fichero sql.txt, utilizando para ello los datos obtenidos del método escrito en el ejercicio 31.

34. Escribe un método que, a partir de los datos del fichero ciudades.csv, recopile en un array de String el ID de las primeras 100 ciudades que hayan elegido la opción Sí.

35. Diseña una clase Videojuego que permita almacenar los datos presentes en el archivo video_games.csv. Asegúrate de incluir los atributos correctamente, además de constructores por defecto, con parámetros y de copia. Añade también un método toString().

36. Crea un programa que, a partir del fichero video_games.csv y de la clase del ejercicio anterior, inserte en un array de 100 Videojuegos todos aquellos que cumplan las condiciones de ser de Nintendo DS y tener un máximo de 2 jugadores. Rellena un fichero NDS.txt con el contenido de dicho array.

37. Escribe un programa que tome el fichero fechas.csv y lea su contenido, ordenando sus entradas según la elección de cada persona.

38. Modifica el ejercicio anterior para que dentro de cada elección, las personas estén ordenadas por la fecha asignada de más antiguo a más reciente.