

Cuaderno de ejercicios

Tema 1 - Ficheros y conectores

Sección 1

1. Realiza un programa que reciba como parámetro de entrada un directorio y lo muestre por pantalla.
2. Ampliar el programa anterior para que muestre todas las características de interés del directorio, tomando como referencia la información que proporciona la clase File.
3. Introducir una comprobación en el programa anterior para determinar si el directorio existe.
4. Realizar un programa que dado un directorio, compruebe si existe y devuelva un mensaje de confirmación si existe o una alerta en caso contrario.
5. Realiza un programa que reciba como parámetros de entrada un directorio y una extensión de fichero (por ejemplo .txt) y devuelva por pantalla todos los ficheros del directorio que cumplan el criterio.
6. Modifica el programa anterior para que tenga en cuenta que si no se le pasa ninguna extensión como parámetro muestre todo el contenido del directorio.
7. Modifica el programa anterior para que admita como parámetros de entrada un número cualquiera de extensiones, devolviendo después por pantalla todos los ficheros del directorio que tengan alguna de las extensiones indicadas.
8. Desarrolla un programa que dado un fichero, realice una copia del mismo (en el mismo directorio y cambiándole el nombre) y lo borre después. Muestra una traza por pantalla de las acciones para ver que se realizan.

Sección 2

9. Escribe un programa que reciba como parámetro de entrada la ruta de un fichero, lea su contenido y lo muestre por pantalla carácter a carácter.
10. Introduce una modificación en el programa anterior para que admita otro parámetro de entrada adicional que permita especificar la velocidad a la que se muestren los caracteres.
11. Realiza otro programa que muestre un número determinado de caracteres por pantalla (por ejemplo 100), espere a que el usuario presione alguna tecla, muestre otro bloque de caracteres, vuelva a esperar, y así sucesivamente hasta mostrar todo el contenido.
12. Crea un programa que dado un fichero de texto, lea y muestre su contenido línea a línea.
13. Modifica el programa anterior para que acepte como parámetros de entrada un número que indique la velocidad a la que se muestran las líneas.
14. Crea otro programa a partir del anterior que en vez de mostrar el contenido por consola lo escriba en otro fichero del mismo directorio.

15. Realiza un programa que permita recibir por teclado una serie de strings por parte del usuario y los vaya escribiendo en un fichero de texto. Como condición de finalización, el usuario deberá escribir un string que sea "exit".
16. Modifica el programa anterior para que el nombre del fichero incluya la fecha y la hora de creación.

Sección 3

17. Abre un editor de texto (Notepad++ o similar) y crea un fichero XML con una estructura típica que represente un objeto y sus principales características. Asegúrate de incluir varios objetos en el fichero.
18. Realiza un programa que dado el fichero creado en el ejercicio anterior lo muestre por pantalla línea a línea.
19. Crea un programa que implemente un parser para gestionar el fichero XML y devuelva por pantalla el número de nodos (objetos) que haya en el fichero.
20. Amplía el programa anterior para que además recorra los nodos uno a uno y muestre por pantalla sus atributos.
21. Introduce en el programa anterior un método que implemente la clase objeto que has elegido para el XML. Puede ser un objeto Java común (POJO, Plain Old Java Object) con constructor, setters y getters.
22. Para probar que el objeto funciona correctamente, realizamos otra modificación que implemente que a medida que se lean los nodos del XML se vayan creando objetos y guardándolos en una lista.
23. Crea otra funcionalidad que permita a un usuario introducir objetos nuevos en la lista. Para ello se le deben pedir los valores de los atributos, posteriormente crear un objeto con dichos atributos y, finalmente, añadir el objeto a la lista.
24. Como última funcionalidad, se pide que se guarde la lista completa de objetos en un nuevo fichero XML. Se debe comprobar que el formato del fichero resultante se corresponda al esperado para un fichero XML (*indent* o sangría adecuados).

Sección 4

25. Crea una base de datos MySQL (con una sola tabla de momento) tomando como referencia la lista de objetos que has creado en el bloque de ejercicios anterior. Puedes utilizar phpMyAdmin, MySQL Workbench o cualquier otro programa que permita administrar el SGBD. Ajusta los tipos de datos a los atributos del objeto, es decir, utiliza el tipo de dato que más se adecúe al atributo. Recuerda utilizar como clave primaria un id que sea autoincremental.
26. Realiza un programa que importe la librería necesaria para poder realizar una conexión a una base de datos MySQL, realice la conexión a la base de datos anterior y muestre un mensaje si se ha hecho o no con éxito.

27. Amplía el programa anterior para que implemente el código necesario para interrogar a la base de datos con una sentencia SQL. Implementa una sentencia que permita recuperar todo el contenido de una tabla de la base de datos.
28. Para comprobar que efectivamente se han recuperado todos los datos, implementar el código para recorrer el resultado de la consulta y mostrar por pantalla las entradas de la tabla. Puedes utilizar concatenación de strings para que el formato de salida sea más comprensible.
29. Implementa un método que permita leer por teclado los atributos para una nueva entrada en la tabla (puedes utilizar lo que ya has desarrollado en el bloque anterior), cree la consulta adecuada de inserción e introduzca los nuevos datos en la base de datos. Comprueba después que la inserción se ha realizado correctamente.
30. Implementa un método que permita, dado un id de la tabla, actualizar alguno de sus campos (el nuevo valor del campo se introducirá por teclado).
31. Implementa un método que permita, dado un id de la tabla, borrar la entrada.
32. Modifica los dos ejercicios anteriores para que pidan una confirmación del usuario antes de pasar a realizar la acción.