

GUIA RAPIDA DE FICHEROS DE TEXTO EN JAVA:

Los Ficheros de Texto en Java tienen un tratamiento muy sencillo. Para definirlos basta con utilizar la clase *File* situada dentro del paquete *Java.io*.

Para crear un objeto básico *File f* basta con crear mediante el *new* de un objeto de esa clase, utilizando como parámetro un *String* con el nombre del fichero en el disco. El fichero creado de este modo deberá estar situado en el mismo directorio donde se encuentre la clase Java en la que estemos trabajando.

Una vez creado el objeto *f*, debemos decidir si lo usaremos para lectura (tomar datos de él) o para escritura (poner datos en él). Para la comunicación con un fichero se utilizan “*Streams*” que son los flujos de entrada o salida (input, output respectivamente) de datos asociados a un fichero. Sin embargo para la utilización básica de un fichero de texto dichos flujos, aunque se utilizan, son casi transparentes al programador.

Para utilizar un fichero de texto en modo lectura, se utiliza una clase ya conocida y utilizada ampliamente para obtener datos desde teclado: *Scanner*.

Si miramos la documentación de la clase *Scanner* veremos que el constructor que tiene como parámetro un objeto de tipo *File* en su cabecera advierte de que lanza una excepción *FileNotFoundException* que es derivada de la clase *IOException* y por tanto es checked. Es por eso que la instrucción de construcción del *Scanner* con un fichero debe ser siempre incluida dentro de un bloque *try_catch_finally* o de un método que en cuya cabecera tenga la instrucción de propagación de al menos esa excepción (*FileNotFoundException*).

Lo habitual es situar la declaración de la variable de tipo *Scanner* fuera del bloque *try_catch_finally* asignándole un valor inicial igual a *null*. Sin embargo se pondrán dentro de la parte del *try*: La construcción creación de la variable de tipo *Scanner* y las operaciones asociadas a la lectura de cada una de las líneas de texto de los datos de diferentes tipos básicos con los métodos conocidos de la clase *Scanner*, luego la captura de la excepción ya vista con el *catch*, y en el *finally* una instrucción condicional para que en el caso de que el *Scanner* haya sido creado correctamente lo cierre con el método *close()*.

Para utilizar un fichero de texto en modo escritura, se utiliza una clase llamada: *PrintWriter*.

Si miramos la documentación de la clase *PrintWriter* veremos que el constructor que tiene como parámetro un objeto de tipo *File* en su cabecera advierte de que lanza una excepción *FileNotFoundException* que es derivada de la clase *IOException* y por tanto es checked. Es por eso que la instrucción de construcción del *PrintWriter* con un fichero debe ser siempre incluida dentro de un bloque *try_catch_finally* o de un método que en cuya cabecera tenga la instrucción de propagación de al menos esa excepción (*FileNotFoundException*).

Lo habitual es situar la declaración de la variable de tipo *PrintWriter* fuera del bloque *try_catch_finally* asignándole un valor inicial igual a *null*. Sin embargo se pondrán dentro de la parte del *try*: La construcción creación de la variable de tipo *PrintWriter* y las operaciones asociadas a la lectura de cada una de las líneas de texto de los datos de diferentes tipos básicos con los métodos de la clase *PrintWriter* que afortunadamente ya son conocidos puesto que son los mismos que se han utilizado para escribir por pantalla con *System.out*. y sólo cambiaríamos esa clase por la variable de tipo *PrintWriter*, luego la captura de la excepción ya vista con el *catch*, y en el *finally* una instrucción condicional para que en el caso de que el *PrintWriter* haya sido creado correctamente lo cierre con el método *close()*.

El constructor de las clases *Scanner* y *PrintWriter* con un *File* como parámetro en realidad construye el flujo de entrada o salida respectivamente asociado al fichero automáticamente y es esta creación la que puede provocar el lanzamiento de la excepción checked *FileNotFoundException* de ahí la necesidad de la cláusula *throws* con dicha excepción en la cabecera de ambos constructores.

EJERCICIO SOBRE LECTURA Y ESCRITURA DE FICHEROS DE TEXTO EN JAVA:

```
import java.util.*;
import java.io.*;

public class Ejercicio1LecturaYEscrituraFicheroTexto{
    // ocultamos el constructor
    private Ejercicio1LecturaYEscrituraFicheroTexto(){}

    public static void main (String[] args){
        System.out.println("LEYENDO DATOS DESDE TECLADO Y ESCRIBIENDO EN EL FICHERO");
        leerDesdeTecladoYGuardarEnFichero();
        System.out.println("\nESCRIBIENDO DATOS EN LA PANTALLA DESDE EL FICHERO");
        escribirEnPantallaDesdeFichero();
    }

    /**
     * Ejercicio, implementar los métodos invocados desde el main, el primero leerá
     * frases del teclado hasta que escriba el usuario la línea "fin" y las escribirá
     * en un fichero llamado "texto.txt". El segundo leerá el contenido de "texto.txt"
     * y lo mostrará en pantalla.
     */
}
```

Necesario para el Scanner y Locale

Necesario para el File, IOException y PrintWriter

```
private static void leerDesdeTecladoYGuardarEnFichero(){
    Scanner teclado=new Scanner(System.in).useLocale(Locale.US);
    File f=new File("texto.txt");
    PrintWriter salida=null;
    try {
        salida=new PrintWriter(f);
        String s=teclado.nextLine();
        while(!s.equals("fin")){
            salida.println(s);
            s=teclado.nextLine();
        }
    }
    catch (FileNotFoundException e) {
        System.out.println("El fichero no existe o no se puede crear");
    }
    finally{ if (salida!=null) salida.close(); }
```

Crear el fichero

Declarar el PrintWriter

Al construir el PrintWriter con un fichero, el constructor puede lanzar la excepción IO FileNotFoundException, que es checked y por tanto debe ser tratada o propagada. El resto de instrucciones aunque no provocan excepciones no tienen sentido si no se ha creado correctamente

En cualquier caso si se ha creado cerrar el fichero

```
private static void escribirEnPantallaDesdeFichero(){
    File f=new File("texto.txt");
    Scanner entrada= null;
    try {
        entrada=new Scanner(f,"UTF-8");
        String s="";
        while(entrada.hasNext()){
            s=entrada.nextLine();
            System.out.println(s);
        }
    }
    catch (FileNotFoundException e) {System.out.println("El fichero no existe o no se puede abrir");}
    finally{ if (entrada!=null) entrada.close(); }
}
```

Crear el fichero

Declarar el Scanner

Al construir el Scanner con un fichero, el constructor puede lanzar la excepción IO FileNotFoundException, que es checked y por tanto debe ser tratada o propagada. (El otro parámetro es para obligar a utilizar esa codificación. También se puede usar el Locale. El resto de instrucciones como en el caso anterior.

En cualquier caso si se ha creado cerrar el fichero

NOTA: Las instrucciones del PrintWriter "salida" son las mismas que se utilizan en el System.out.... para escribir por pantalla. El scanner "entrada" se utiliza exactamente igual que el Scanner "teclado" salvo la excepción en el constructor.