

Tratamiento básico de archivos de texto en Java

Este material presenta en forma práctica el manejo de archivos de texto en el lenguaje Java. Nos enfocaremos en la lectura y escritura de archivos.

Cabe aclarar que todas las clases aquí utilizadas, son parte de la librería de *java.io*, la cual debemos importarla para usar sus clases.

Lectura de un fichero de texto en Java

Para abrir un un archivo de texto en modo lectura, debemos usar la clases ***FileReader*** y ***File***. Dado que ***FileReader*** no permite leer líneas completas de un archivo (que es lo habitual en lectura de archivos), es que debemos concentrarnos en clases utilitarias para ello, como lo es ***BufferedReader***.

Es posible construir un ***BufferedReader*** a partir de un ***FileReader*** de la siguiente forma:

```
File archivo = new File ("c:\\texto.txt");
FileReader fr = new FileReader (archivo);
BufferedReader br = new BufferedReader (fr);
```

Luego de abierto el archivo, podemos leer cada una de las líneas contenido en el mismo mediante el método ***readLine()*** de la clase ***BufferedReader***. Por lo general queremos leer todas las líneas del archivo, por lo que debemos iterar hasta que no existan más líneas por leer.

```
String linea = br.readLine();
while (linea != null) {
    System.out.println(linea);
    linea = br.readLine();
}
```

Dado que la apertura y lectura de un archivo puede generar excepciones, es que las debemos capturar a través de un bloque *try-catch*.

A continuación, se presenta un método para la lectura de archivos.

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

public void leerArchivo(String nombreCompletoArchivo){
    FileReader fr;
    try {
        fr = new FileReader(nombreCompletoArchivo);
        BufferedReader br = new BufferedReader(fr);
        String lineaActual = br.readLine();
        while (lineaActual != null){
            System.out.println(lineaActual);
            lineaActual = br.readLine();
        }
    }
    catch (FileNotFoundException e) {
        System.out.println("Error al leer el archivo "+nombreCompletoArchivo);
        e.printStackTrace();
    } catch (IOException e)
    {
        System.out.println("Error al leer el archivo "+nombreCompletoArchivo);
        e.printStackTrace();
    }
    System.out.println("Archivo leído satisfactoriamente");
}
```

Escritura de un archivo de texto en Java

Para escribir en un archivo de texto, utilizaremos la clase *BufferedWriter*, la cual nos ayuda a escribir líneas dentro del archivo de texto.

A su vez, el archivo debe abrirse en modo escritura, por lo que utilizaremos la clase *FileWriter*. El segundo parámetro del constructor del *FileWriter*, indica si queremos añadir información al final del archivo (true), o queremos eliminar lo ya contenido (false).

El siguiente código de ejemplo, escribe dentro de un archivo, el contenido de un array de Strings. Dicho array de Strings contiene las líneas a escribir dentro del archivo.

```
public void escribirArchivo(String nombreCompletoArchivo, String[] listaLineasArchivo) {
    FileWriter fw;
    try {
        fw = new FileWriter(nombreCompletoArchivo,true);
        BufferedWriter bw = new BufferedWriter(fw);
        for (int i = 0; i < listaLineasArchivo.length; i++){
            String lineaActual = listaLineasArchivo[i];
            bw.write(lineaActual);
            bw.newLine();
        }
        bw.close();
        fw.close();
    } catch (IOException e) {
        System.out.println("Error al escribir el archivo "+nombreCompletoArchivo);
        e.printStackTrace();
    }
}
```

Ejercicio sugerido

- Generar una clase Java llamada “*ManejadorArchivos*”, que contenga los dos métodos de escritura y lectura de archivos.
- Desarrollar una clase *Principal* donde se pueda escribir un archivo y luego leerlo. Ejemplo del “*main*”:

```
public static void main (String[] args) {
    ManejadorArchivos mgr = new ManejadorArchivos();
    String[] listaLineasArchivo = {"Linea 1", "Linea 2", "Linea 3", "Linea 4"};
    mgr.escribirArchivo("c:\\texto.txt", listaLineasArchivo);
    mgr.leerArchivo("c:\\text.txt");
}
```