Escritura de un archivo BufferedWriter - FileWriter

¡Comencemos!

¿Qué aprenderás?

- Comprender la Clase FileWriter para el uso de los métodos en la construcción de los archivos.
- Comprender la Clase BufferedWriter para el uso de los métodos en la escritura de archivos.

Introducción

Aprenderemos a escribir en archivos físicos utilizando las mejores prácticas dentro de la programación en Java, para esto utilizaremos las clases `FileWriter` y `BufferedWriter`, las cuales optimizan el uso de los archivos.

Todo el contenido estudiado en este capítulo nos servirá para manejar, conocer y comprender el uso práctico de los archivos que pueden ir desde un texto plano hasta un PDF.

¡Vamos con todo!

Clase FileWriter

Esta clase es usada para escribir caracteres en archivos. Su método `write()` permite escribir caracteres o strings a un fichero. Esta clase normalmente está envuelta en objetos Writer de más alto nivel, como BufferedWriter.

Para crear un FileWriter, luego de realizar la importación `import java.io.FileWriter;`, se necesita un String como ruta de archivo o una clase File. Para este caso, y siguiendo la línea del curso, utilizaremos un File:

```
File archivo = new File("src/carpeta/texto.txt");
FileWriter fileW = new FileWriter(archivo);
```

Clase BufferedWriter

Esta clase es usada para hacer clases de bajo nivel como FileWriters de una manera más eficiente y más fácil de usar. Comparado con las clases FileWriter, los BufferedWriters escriben relativamente grandes cantidades de información en un archivo, lo que minimiza el número de veces que las operaciones de escritura de archivos, que son operaciones más lentas, se llevan a cabo. La clase BufferedWriter también provee un método llamado `newLine()` creando separadores de línea específicos de la plataforma de manera automática.

Para crear un BufferedWriter, luego de realizar la importación `import java.io.BufferedWriter;`, se necesita un objeto `FileWriter`:

```
File archivo = new File("src/carpeta/texto.txt");
FileWriter fileW = new FileWriter(archivo);
BufferedWriter bufferedWriter = new BufferedWriter(fileW);
```

Para escribir utilizaremos el método `write("texto")`, donde el texto es lo que se escribirá en el archivo:

```
bufferedWriter.write("texto");
```

Para que se guarde la información debemos cerrar el archivo, esto hace que guarde automáticamente los cambios. Utilizaremos el método `close()`:

```
bufferedWriter.close();
```