

BUFFEREDREADER & BUFFEREDWRITER

Lucia Benitez Martinez y Rafael Manzano Medina

TRADUCCIÓN LITERAL

BufferedReader (buffered = amortiguado, reader = lector)
BufferedWriter (buffered = amortiguado, writer = escritor)



DEFINICIÓN

- Estas clases tienen la misma función que FileReader y FileWriter, leer y escribir en ficheros.
- La diferencia es que nos permite leer una línea completa, en lugar ir carácter por carácter por tanto, devuelve null y no un -1 como en FileReader.
- Con BufferedWriter también podemos añadir una línea, como si pulsáramos un Enter.
- Otra de las grandes ventajas de BufferedReader y BufferedWriter es que esta clase está sincronizada lo cual es sinónimo de seguridad al utilizarla en programación concurrente.

INSTANCIACIÓN

- Forma normal

```
FileWriter fw = new FileWriter("Dirección del fichero");
```

```
FileReader fr = new FileReader("Dirección del fichero");
```

```
BufferedReader br = new BufferedReader(fr);
```

```
BufferedWriter bw = new BufferedWriter(fw);
```

- Forma abreviada

```
BufferedReader br = new BufferedReader(new FileReader("Dirección del fichero"));
```

```
BufferedWriter bw = new BufferedWriter(new FileWriter("Dirección del fichero"));
```

MÉTODOS BUFFEREDREADER

Método	Tipo de Retorno	Descripción
mark(int limiteCaracteres)	void	Marca la posición actual en la que se encuentre el apuntador en el flujo, el parámetro del método indica el número de caracteres que pueden ser leídos mientras se mantenga esta marca.
read()	void	Lee un solo carácter del flujo, lo curioso de este método es que retorna un número entero, ¿Por qué? Más adelante la respuesta.
readLine()	String	Lee una línea completa de texto.
ready()	boolean	Este método es utilizado para saber si aún hay caracteres en el flujo para ser leídos, detalles más adelante.
reset()	void	Reinicia el flujo hasta la marca más reciente que se haya hecho.
skip(long n)	long	Mueve el apuntador del flujo las posiciones necesarios para evitar la cantidad de caracteres de n.

MÉTODOS BUFFEREDWRITER

Método	Tipo de Retorno	Descripción
close()	void	Cierra la corriente. Una vez que la secuencia se ha cerrado no puedes ni usar write () ni flush () pues saltará una excepción de IOException.
newLine()	Void	Escribe un separador de línea. La cadena del separador de línea está definida por la propiedad del sistema line.separator, y no es necesariamente un carácter de nueva línea ('\n').
write(String s, int off, int len)	void	Escribe una porción de una cadena. Si el valor del parámetro "len" es negativo, entonces no escribe. Utiliza la excepción IndexOutOfBoundsException.
write(int c)	void	Escribe un solo carácter.
flush()	void	Sirve para guardar todas las modificaciones realizadas en el fichero

ENLACES A LA API

<https://docs.oracle.com/javase/8/docs/api/index.html?java/io/BufferedReader.html>

<https://docs.oracle.com/javase/7/docs/api/java/io/BufferedWriter.html>