

ACCESO A DATOS: Trabajo de investigacion.

B - Investigación sobre clases del paquete java.io:

File:

Clase pensada para poder realizar operaciones de información sobre archivos. No proporciona métodos de acceso a los archivos, sino operaciones a nivel de sistema de archivos (listado de archivos, crear carpetas, borrar ficheros, cambiar nombre,...).

FileReader:

Clase del paquete java.io que puede utilizarse para leer un flujo de caracteres de los archivos. Puede implementar un objeto de clase File para la lectura.

BufferedReader:

Clase que nos permite leer 'texto' en un InputStream, utilizando un buffer para proporcionar una escritura eficiente de caracteres, arrays y strings. Puede implementar un objeto de la clase FileReader para la lectura.

FileWriter:

Clase del paquete java.io que puede utilizarse para escribir un flujo de caracteres desde nuestro programa a un archivo. Puede implementar un objeto de clase File para la escritura.

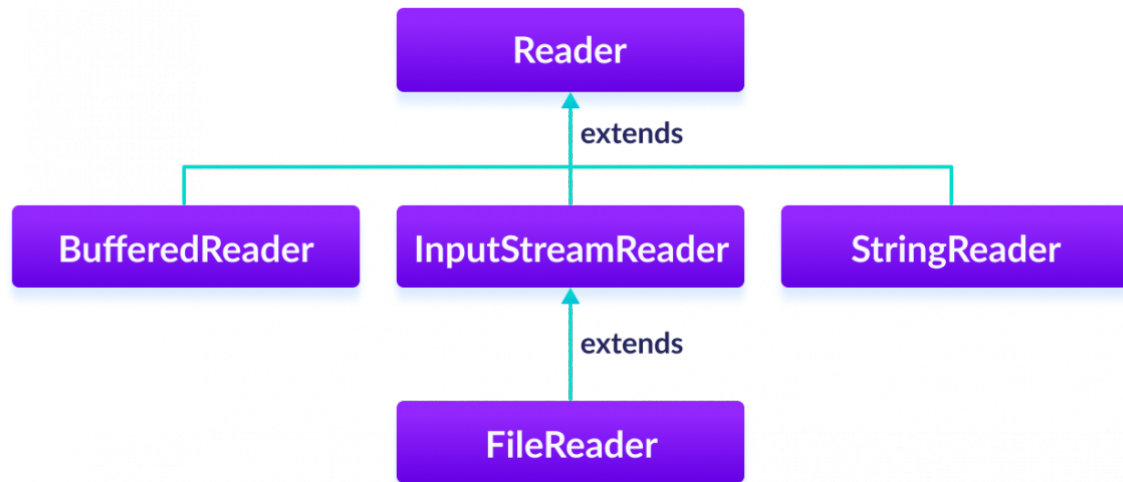
BufferedWriter:

Clase que nos permite escribir 'texto' en un OutputStream, utilizando un buffer para proporcionar una escritura eficiente de caracteres, arrays y strings. Puede implementar un objeto de la clase FileWriter para la escritura.

PrintWriter:

PrintWriter es un objeto que nos permite imprimir representaciones formateadas de una salida de stream de texto.

¿Qué diferencia hay entre FileReader y BufferedReader?
¿Cuál es más eficiente en el tratamiento de información? ¿Por qué?

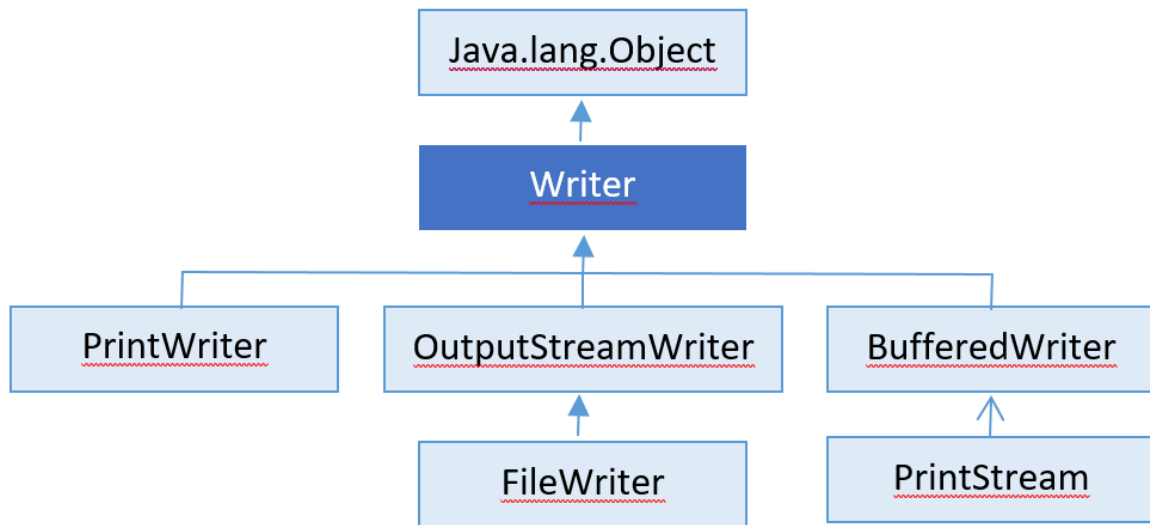


FileReader se usa para leer un archivo desde una unidad de disco, mientras que BufferedReader no está obligado a leer solo archivos.

Para el tratamiento de información BufferedReader sería una mejor opción, acepta como parámetros el origen de la lectura y el tamaño del buffer de lectura. Esto en archivos muy grandes sería una gran ventaja porque podría leerlos en pequeños 'trozos'.

Si se pretende leer archivos 'pequeños' de texto y solamente se quiere hacer operaciones que no impliquen un uso de memoria muy grande, FileReader podría ser una mejor opción.

¿Qué diferencia hay entre FileWriter, BufferedWriter y PrintWriter?



En el caso de `FileWriter` y `BufferedWriter`.

`BufferedWriter` es más eficiente si usted tiene múltiples escrituras entre `flush/close` las escrituras son pequeñas comparadas con el tamaño del buffer.

En ambos casos, la cadena se escribe a la vez. `BufferedWriter` mejora la eficiencia en múltiples escrituras pequeñas.

En cuanto a `PrintWriter` y `FileWriter` ambas son clases de representación de caracteres, lo que significa que trabajan con caracteres y los convierten a bytes utilizando el conjunto de caracteres por defecto. Se diferencian en que `PrintWriter` no arroja `IOException`, debes usar un indicador booleano y usar `checkError()`;

`PrintWriter` tiene un constructor opcional que puede utilizar para activar el auto-flushing cuando se llaman métodos específicos. Esta opción no existe en `FileWriter`.

Entre `FileWriter` y `PrintWriter` ¿Cuál de las dos clases lanza excepciones de tipo `IOException`? ¿Cómo gestionará los errores la otra clase? ¿Cuál de las dos clases usarías para añadir/escribir texto al final de un fichero?

`FileWriter` lanza las excepciones tipo `IOException`. Como había comentado antes lo Gestionara con indicadores booleanos y el metodo `checkError()`; .
Usaria `FileWriter` por que podria usarla con `BufferedReader` sinsobreescribir el contenido.