

Clases `InputStream` y `OutputStream`

Estos métodos, definidos en las clases abstractas, están disponibles en todas las subclases, formando un conjunto mínimo de métodos de entrada y salida en todos los flujos de bytes.

<i>InputStream</i>	
<code>int available()</code>	Devuelve el número de bytes de entrada disponibles para lectura
<code>void close()</code>	Cierra el origen de entrada
<code>int read()</code>	Devuelve el valor entero del siguiente byte de entrada. Devuelve -1 al llegar al final del archivo
<code>int read(byte buffer[])</code>	Intenta leer el tamaño en bytes definidos en la variable <code>buffer</code> . Devuelve el número real de bytes leídos y -1 al llegar al final del archivo
<code>int read(byte buffer[], int, posInicial, int numBytes)</code>	Intenta leer el tamaño en bytes definidos en la variable <code>buffer</code> , comenzando en la posición <code>posInicial</code> y el número de bytes de la variable <code>numBytes</code>
<code>void reset()</code>	Restablece la cabeza de lectura a la marca definida previamente

<i>OutputStream</i>	
<code>void close()</code>	Cierra el flujo de salida
<code>void flush()</code>	Vacia el buffer
<code>void write(int byte)</code>	Escribe 1 byte en el flujo de salida

Clase File

Algunos métodos comunes para ficheros y para carpetas:

boolean exists()	Devuelve true si existe la carpeta o archivo.
boolean canRead()	Devuelve true si el archivo se puede leer
boolean canWrite()	Devuelve true si el archivo se puede escribir
boolean isHidden()	Devuelve true si el objeto File es oculto

Algunos métodos para carpetas:

boolean isDirectory()	Devuelve true si el objeto File es una carpeta y false si es un archivo o si no existe.
boolean mkdir()	Intenta crear una carpeta y devuelve true si fue posible hacerlo
boolean delete()	Borra la carpeta y devuelve true si puedo hacerlo

Algunos métodos para archivos:

boolean isFile()	Devuelve true si el objeto File es un archivo y false si es carpeta o si no existe.
boolean renameTo (File f2)	Cambia el nombre del archivo por el que posee el archivo pasado como argumento. Devuelve true si se pudo completar la operación.
boolean delete()	Borra el archivo y devuelve true si puedo hacerlo
long length()	Devuelve el tamaño del archivo en bytes (en el caso del texto devuelve los caracteres del archivo)

Clases Reader/Writer

Reader	
void close()	Cierra la corriente de entrada. Cualquier acceso posterior generaría una IOException.
int read()	Lee el siguiente byte de la corriente de entrada y le almacena en formato de entero. Devuelve -1 si estamos al final del fichero
int read (byte[] búfer)	Lee de la corriente de entrada bytes y les almacena en el búfer. Lee hasta llenar el búfer.
int read (byte[] bufer, int posInicio, int despl)	Lee de la corriente de entrada bytes y les almacena en el búfer. La lectura la almacena en el array pero a partir de la posición indicada, el número máximo de bytes leídos es el tercer parámetro.
boolean ready()	Devuelve verdadero si la corriente de entrada está lista.
void reset()	Coloca el puntero de lectura en la posición marcada con mark.
long skip (long numCarcteres)	Se salta de la lectura el número de bytes indicados

Writer	
void close()	Cierra la corriente de salida. Cualquier acceso posterior generaría una IOException.
void flush()	Vacía los búferes de salida de la corriente de datos
void write (int byte)	Escribe un byte en la corriente de salida
void write (byte[] bufer)	Escribe todo el array de bytes en la corriente de salida
void write (byte[] buffer, int posInicial, int numBytes)	Escribe el array de bytes en la salida, pero empezando por la posición inicial y sólo la cantidad indicada por numBytes.
void write (String texto)	Escribe los caracteres en el String en la corriente de salida.

Clases `DataInputStream`/`DataOutputStream`

<i>DataInputStream</i>	
<code>void readBoolean()</code>	Lee un valor de tipo booleano
<code>void readByte ()</code>	Lee un valor de tipo byte
<code>void readChar()</code>	Lee un valor de tipo carácter
<code>void readDouble()</code>	Lee un valor de tipo double
<code>void readFloat()</code>	Lee un valor de tipo float
<code>void readInt()</code>	Lee un valor de tipo entero
<code>void readLong()</code>	Escribe un valor de tipo entero largo
<code>void readShort()</code>	Lee un valor de tipo entero corto

<i>DataOutputStream</i>	
<code>void writeBoolean(boolean valor)</code>	Escribe un valor de tipo booleano
<code>void writeByte (int valor)</code>	Escribe un valor de tipo byte
<code>void writeChar(int valor)</code>	Escribe un valor de tipo carácter
<code>void WriteDouble(double valor)</code>	Escribe un valor de tipo double
<code>void writeFloat(float valor)</code>	Escribe un valor de tipo float
<code>void writeInt(int valor)</code>	Escribe un valor de tipo entero
<code>void writeLong(long valor)</code>	Escribe un valor de tipo entero largo
<code>void writeShort(int valor)</code>	Escribe un valor de tipo entero corto