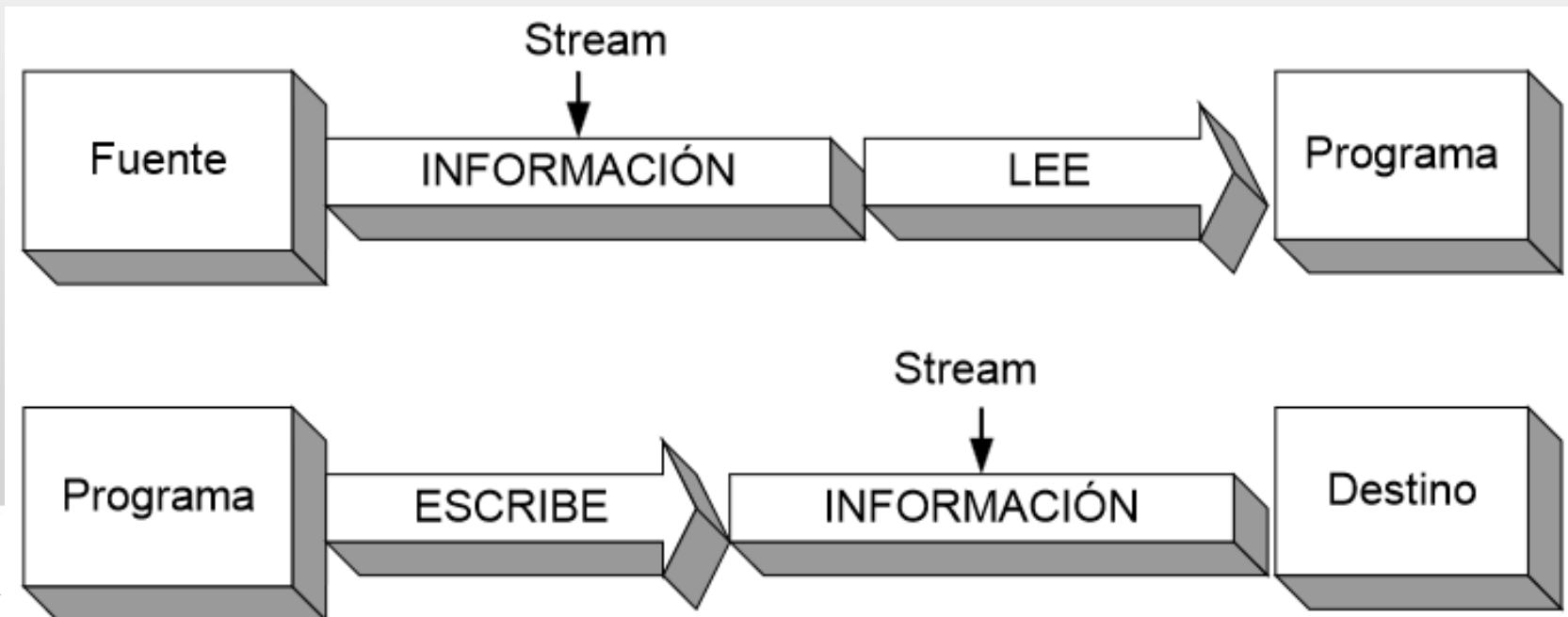


Stream

- Es el flujo de datos desde una fuente hacia un programa
- Permiten obtener información de una fuente secuencialmente.
- Permiten enviar información a un destino secuencialmente.
- Los stream son unidireccionales.
- La operación de lectura es conocida como Input
- La operación de escritura es conocida como Output

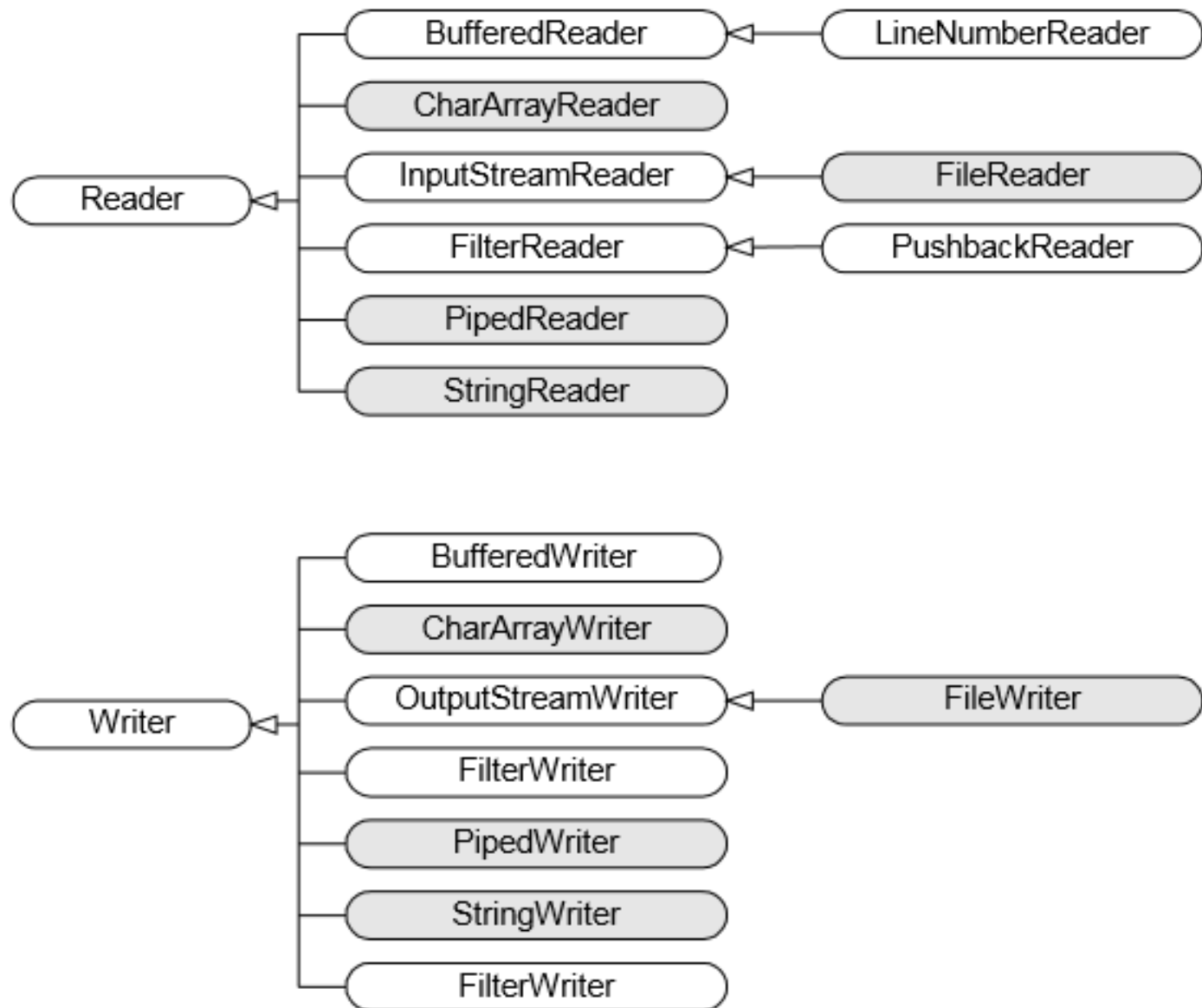


Tipos de Streams

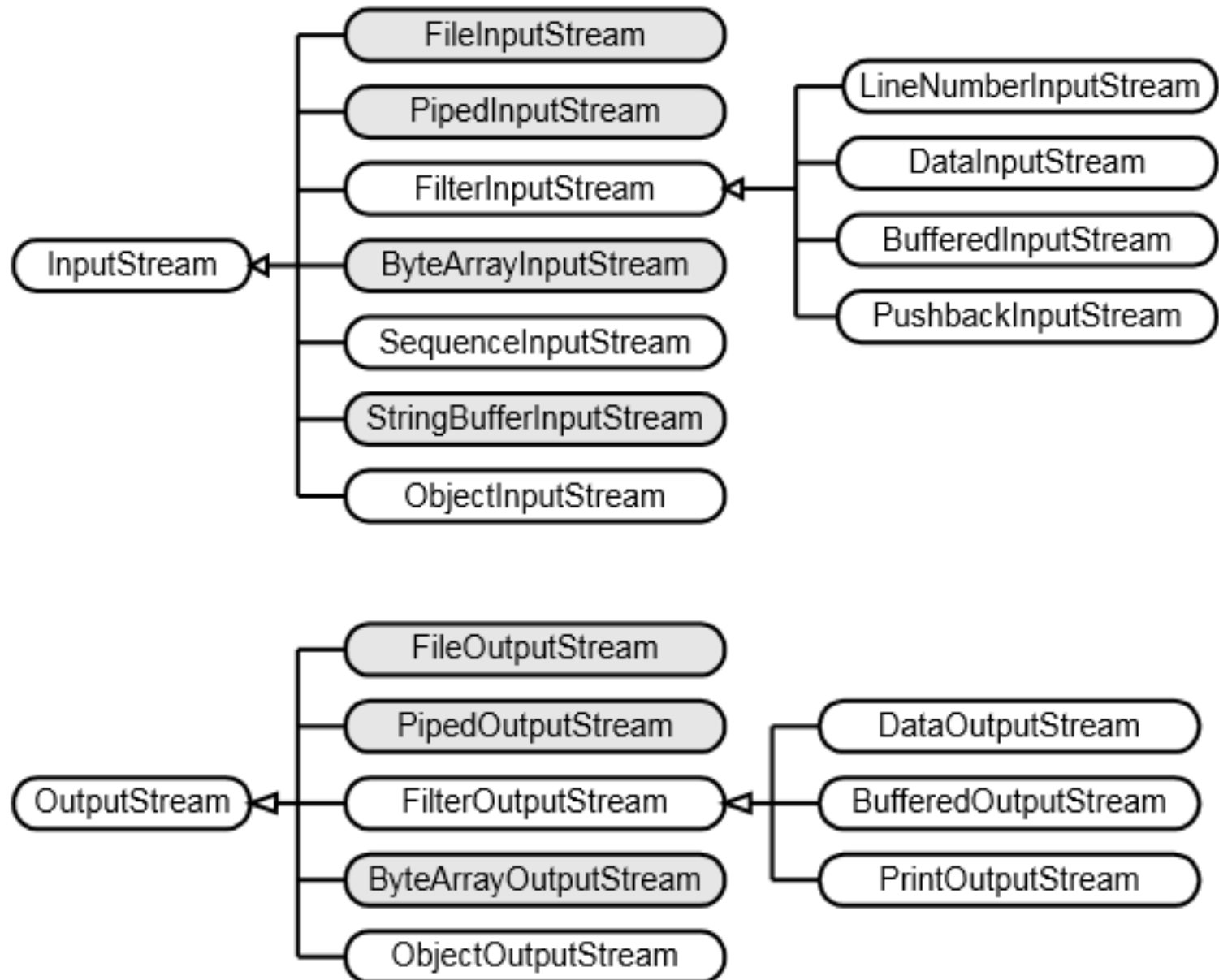
Java contiene dos tipos de Streams, los cuales están organizados en dos grandes grupos:

- Streams orientados a caracteres (Character Streams)
 - Manejan caracteres como unidad de trabajo (Los caracteres están formados por 2 bytes)
 - Lectura y escritura de archivos en formato texto (txt, ini, csv, etc)
 - Las superclases son: Reader y Writer
- Streams orientados a bytes (Byte Streams)
 - Manejan bytes como unidad de trabajo (Cada byte se lee de forma unitaria 1 byte por vez)
 - Lectura y escritura de archivos en formato binario (jpeg, mpeg, xls, etc)
 - Las superclases son: InputStream, OutputStream

Manipulación de Caracteres



Manipulación de Bytes



Superclasses

- Reader
 - `int read()`
 - `int read(char cbuf[])`
 - `int read(char cbuf[], int offset, int length)`
- InputStream
 - `int read()`
 - `int read(byte cbuf[])`
 - `int read(byte cbuf[], int offset, int length)`
- Writer
 - `int write(int c)`
 - `int write(char cbuf[])`
 - `int write(char cbuf[], int offset, int length)`
- OutputStream
 - `int write(int c)`
 - `int write(byte cbuf[])`
 - `int write(byte cbuf[], int offset, int length)`

File Stream

Los File Stream son los stream utilizados para la lectura y escritura de (particularmente) archivos, es una categoría que agrupa tanto a los streams orientados a caracteres como a los streams orientados a byte.

En general se utilizan en conjunto con un objeto del tipo File, que es una representación abstracta de un archivo. La clase File modela tanto archivos como directorios.

FileStream:

- FileReader
- FileWriter
- FileInputStream
- FileOutputStream

Categorías

| Tipo de E/S | Streams |
|--------------------------|---|
| Memoria | CharArrayReader CharArrayWriter |
| | ByteArrayReader ByteArrayWriter |
| | StringReader StringWriter |
| | StringBufferInputStream |
| | |
| Pipe | PipedReader PipedWriter |
| | PipedInputStream PipedOutputStream |
| File | FileReader FileWriter |
| | FileInputStream FileOutputStream |
| Concatenación | No hay disponibles |
| | SequenceInputStream |
| Serialización de Objetos | No hay disponible |
| | ObjectInputStream ObjectOutputStream |
| Conversión de datos | No hay disponible |
| | DataInputStream DataOutputStream |

| Tipo de E/S | Streams |
|-------------------------------------|---|
| Contadores | LineNumberReaders |
| | LineNumberInputStream |
| Datos por adelantado | PushbackReader |
| | PushbackInputStream |
| Impresión | PrintWriter |
| | PrintStream |
| Buffering | BufferedReader BufferedWriter |
| | BufferedInputStream BufferedOutputStream |
| Filtrado | FilterReader FilterWriter |
| | FilterInputStream FilterOutputStream |
| Conversión entre bytes y caracteres | InputStreamReader |
| | OutputStreamWriter |

Resumen

- Los Streams son el medio por el cual se realiza la entrada/salida de datos en Java.
- Existen dos tipos de Streams:
 - Streams que manejan caracteres (Reader, Writer)
 - Streams que manejan bytes (InputStream, OutputStream)
- Los Streams de envío de datos y procesamiento de datos:
 - Streams que pasan datos (FileInputStream, FileReader)
 - Streams que procesan datos (FilterInputStream, FilterReader)
- Dentro de cada tipo de Stream existen dos jerarquías:
 - Streams de entrada (InputStream, Reader)
 - Streams de salida (OutputStream, Writer)
- La serialización de objetos permite leer y escribir objetos