CURSOS INTERSEMESTRALES

PROTECO

Flujos

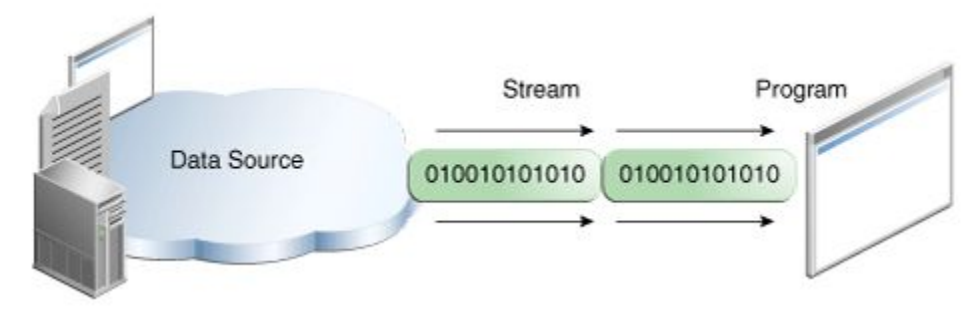
¿Qué es?

Un flujo es una secuencia de datos entre una fuente de entrada y un destino de salida. Un flujo de puede representar diferentes tipos de fuentes y destinos, incluyendo discos, dispositivos y otros programas.

PROTECO

Flujo de Entrada

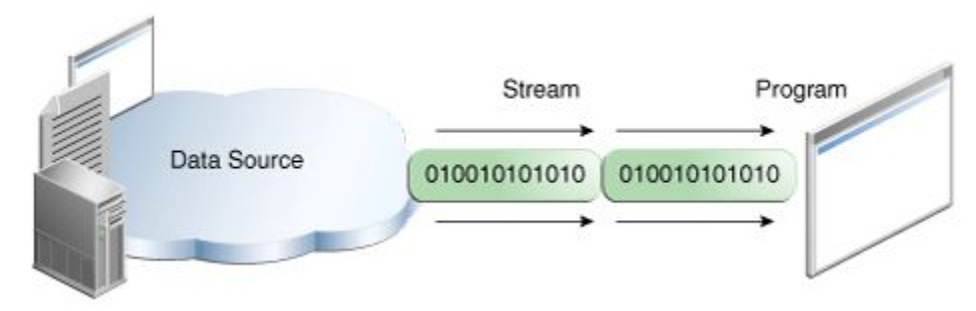
Todos los flujos utilizan un flujo de entrada para leer datos desde una fuente , un elemento a la vez.



PROTECO

Flujo de Salida

Y utilizan un flujo de salida para escribir datos a un destino.



PROTECO

Closeable

Una clase que implemente Closeable una fuente o destino de datos que se puede cerrar. El método de cierre se invoca para liberar los recursos que el objeto mantiene (como los archivos abiertos).

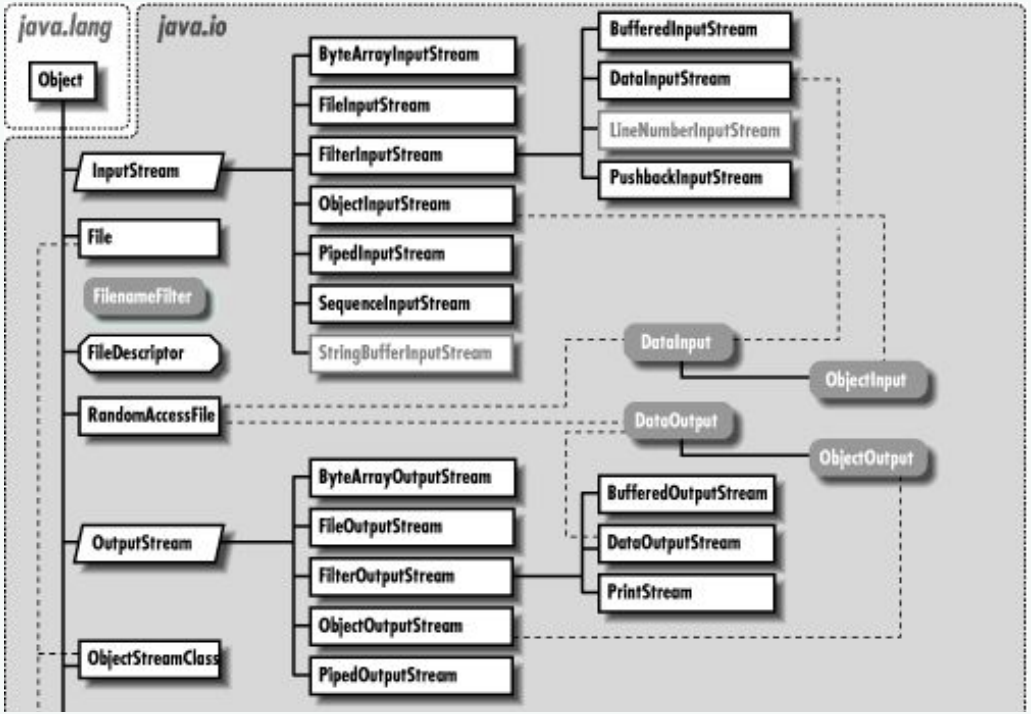
PROTECO

Flushable

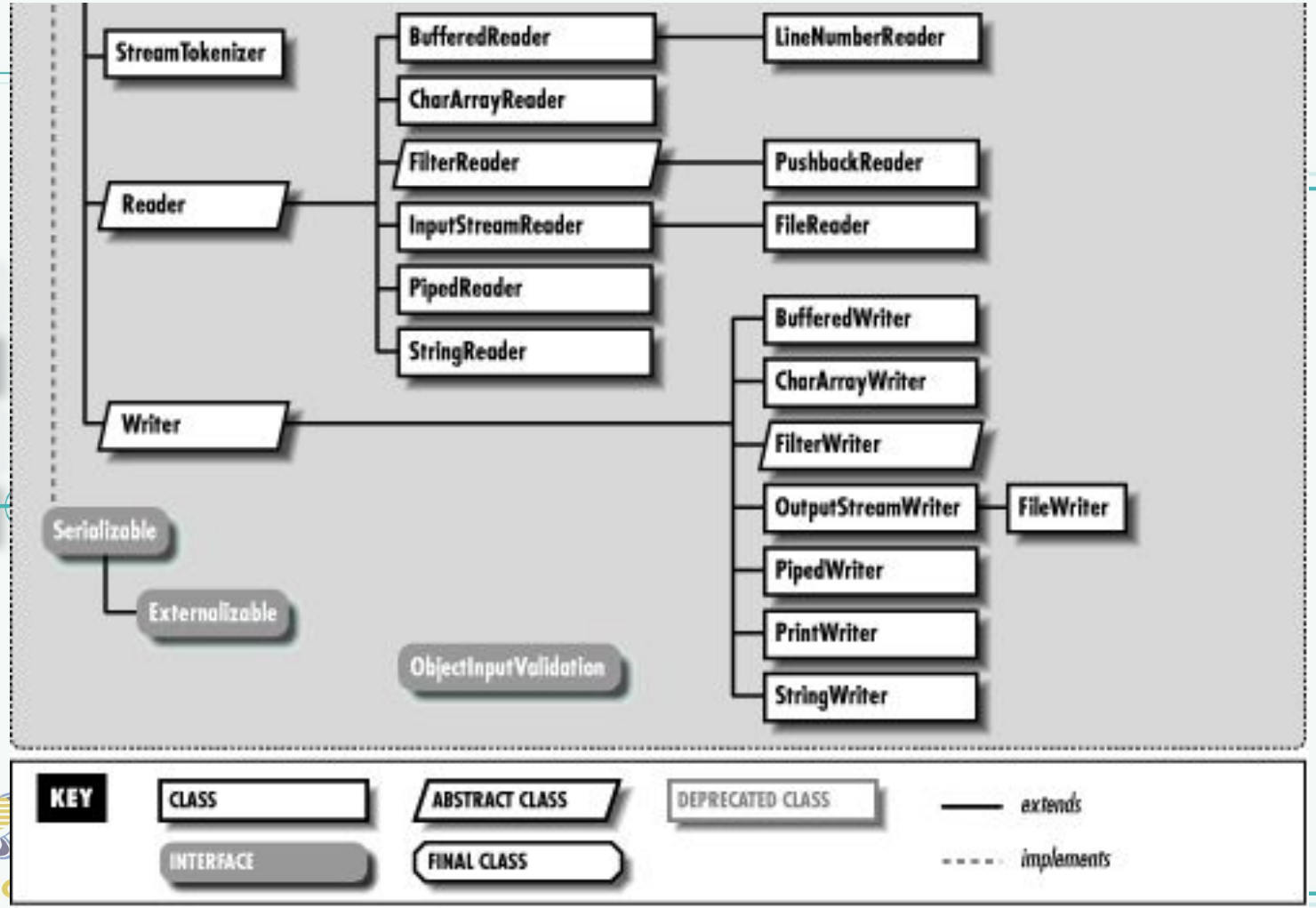
Un flushable es un destino de datos que se puede vaciar. El método de descarga se invoca para escribir cualquier salida en búfer en la secuencia subyacente.

PROTECO

Java.io



PROTECO


StreamTokenizer
Reader
BufferedReader
CharArrayReader
FilterReader
InputStreamReader
PipedReader
StringReader
LineNumberReader
PushbackReader
FileReader
Writer
BufferedWriter
CharArrayWriter
FilterWriter
OutputStreamWriter
PipedWriter
PrintWriter
StringWriter
FileWriter
Serializable
Externalizable
ObjectInputValidation
KEY
CLASS
ABSTRACT CLASS
DEPRECATED CLASS
INTERFACE
FINAL CLASS
extends
implements

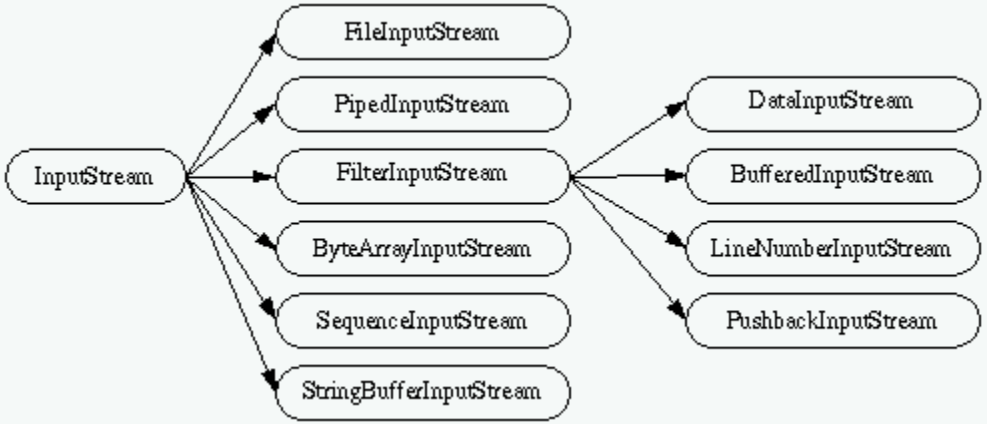
InputStream

Esta clase abstracta es la superclase de todas las clases que representan un flujo de entrada de bytes.

Las aplicaciones que necesitan definir una subclase de InputStream siempre deben proporcionar un método que devuelva el siguiente byte de entrada.

PROTECO

InputStream



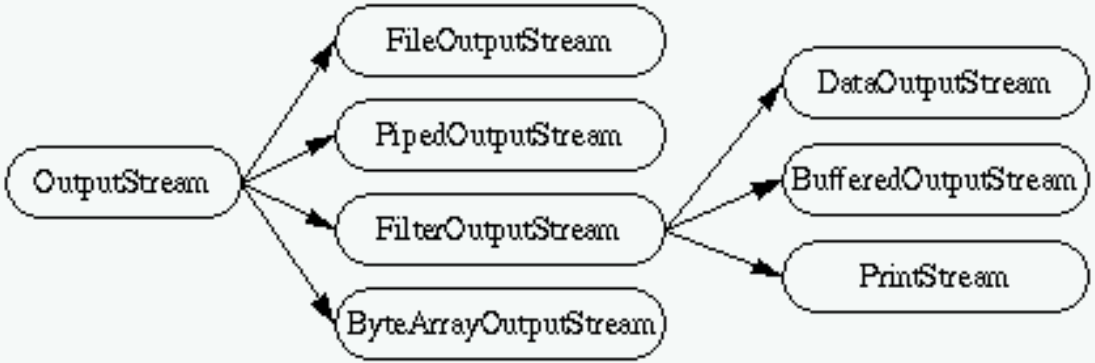
PROTECO

OutputStream

Esta clase abstracta es la superclase de todas las clases que representan un flujo de salida de bytes. Un flujo de salida acepta bytes de salida y los envía a algún destino.

PROTECO

OutputStream



PROTECO

AutoCloseable

Un flujo que debe cerrarse cuando ya no se necesita.

Interfaz.

FileInputStream

Un FileInputStream obtiene bytes de entrada de un archivo en un sistema de archivos. Los archivos disponibles dependen del entorno del host.
FileInputStream está diseñado para leer flujos de bytes en bruto, como datos de imagen. Para leer secuencias de caracteres, considere usar FileReader.

PROTECO

FileInputStream

-Constructor:

- Recibe un objeto tipo cadena de la dirección del archivo.
- Un objeto tipo File.

PROTECO

FileOutputStream

Un flujo de salida de archivo es un flujo de salida para escribir datos en un archivo o en un FileDescriptor. Si un archivo está o no disponible o puede crearse depende de la plataforma subyacente.

PROTECO

Métodos

- available() Define si el flujo está disponible
- close() Cierra el Flujo
- read() Lee un byte
- flush() Limpia el buffer

PROTECO

Java.system

- static PrintStream ***err***: Error estándar del flujo de salida.
- static InputStream ***in***: Flujo de entrada estándar, línea de comandos.
- static PrintStream ***out***: Flujo de salida estándar, línea de comandos.

PROTECO

BufferedInputStream

Un BufferedInputStream agrega funcionalidad a otro flujo de entrada, es decir, la capacidad de almacenar en búfer la entrada y admitir los métodos de marca y restablecimiento.

PROTECO

BufferedOutputStream

La clase implementa un flujo de salida en búfer. Al configurar un flujo de salida de este tipo, una aplicación puede escribir bytes en el flujo de salida sin necesariamente generar una llamada al sistema para cada byte escrito.

PROTECO

Buffers

Tanto de salida como de entrada ambos buffers al momento de construirse necesitan un flujo para poder hacer algo.

PROTECO

Serialización

- La serialización de objetos permite que un objeto sea transformado en una secuencia de bytes.
- Más tarde esa secuencia puede ser usada para reconstruir (deserializar) el objeto original.

PROTECO

Persistencia

- Después de la deserialización el objeto tiene el mismo estado que tenía al ser serializado, excepto los miembros de instancia que no eran serializables.
- Este mecanismo se conoce generalmente como persistencia.

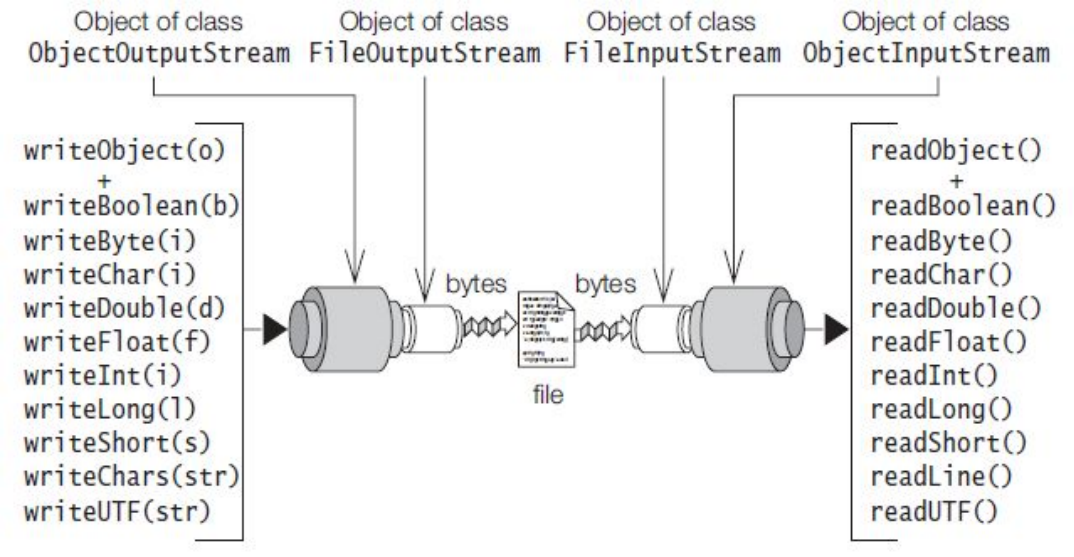
PROTECO

Interfaz Serializable

- Para que los objetos de una clase puedan ser serializados dicha clase debe implementar la interfaz *java.io.Serializable*.
- Esta interfaz se conoce como *marker interface* debido a que no posee métodos que implementar.

PROTECO

Proceso



PROTECO

Transient

- El modificador *transient* puede ser aplicado a cualquier variable de instancia, en caso de serialización el valor de esa variable no será almacenado.
- Al deserializar un objeto que posea variables de instancia marcadas como *transient*, éstas recibirán el valor por defecto.

PROTECO

¿Qué se serializa?

- La información sobre la clase necesaria para reconstruir el objeto.
- Los valores de todos los miembros que sean serializables, no *transient* y no *static*. Incluyendo aquellos que son heredados.

PROTECO

File

Las interfaces de usuario y los sistemas operativos utilizan cadenas de rutas de acceso dependientes del sistema para nombrar archivos y directorios. Esta clase presenta una vista abstracta e independiente del sistema de las rutas jerárquicas.

PROTECO

Abstract Path

- Una cadena de prefijo opcional dependiente del sistema, como un especificador de unidad de disco, "/" para el directorio raíz de UNIX, o "\\\\" para una ruta de acceso de Microsoft Windows UNC.
- Una secuencia de cero o más nombres de cadena.

PROTECO

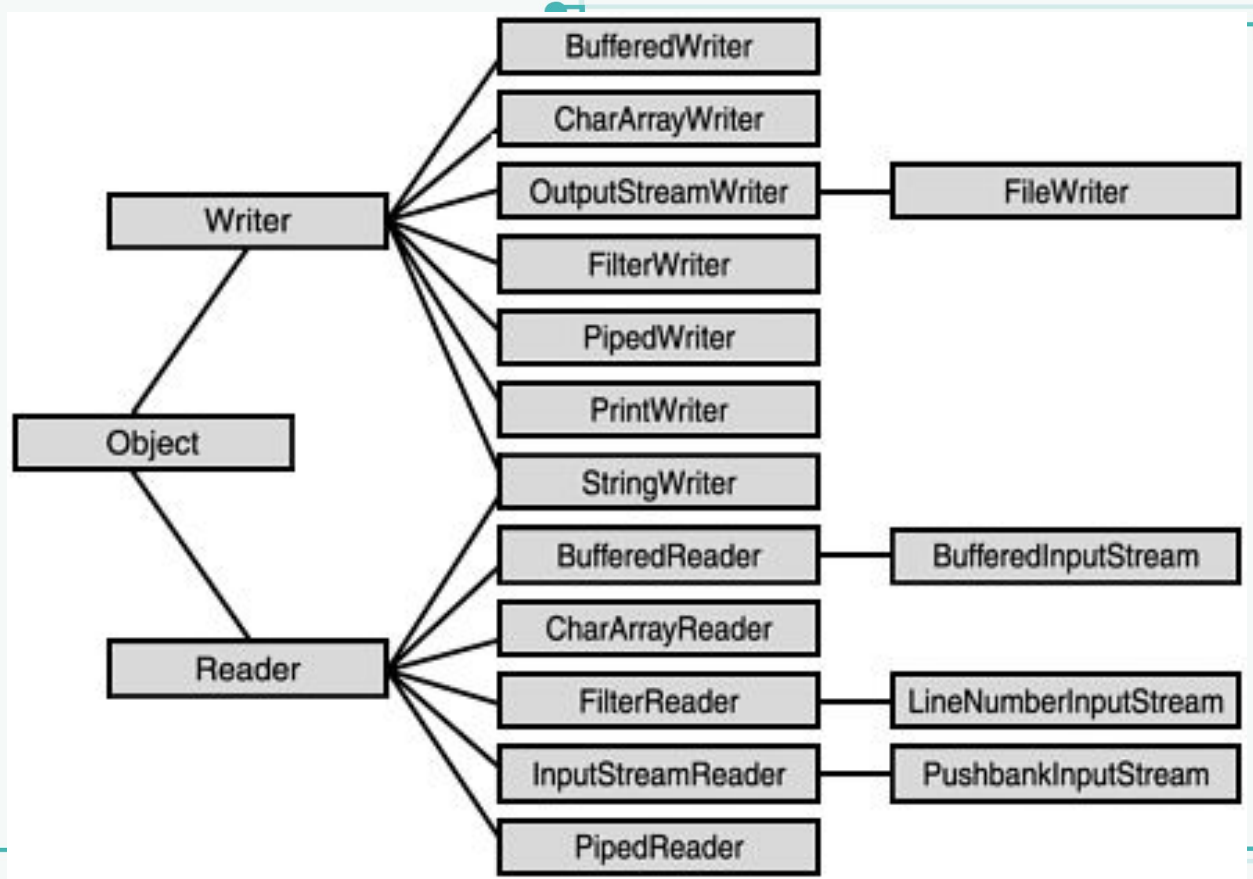
Writer y Reader

- Writer: es una clase abstracta especializada en la escritura(output) de flujo de caracteres.

- Reader: es una clase abstracta especializada en lectura(input) de flujo de caracteres.

PROTECO

Herencia



PROTECO

InputStreamReader

Un InputStreamReader es un puente de flujos de bytes a flujos de caracteres: lee bytes y los decodifica en caracteres usando un juego de caracteres específico.

FileReader

Los constructores de esta clase asumen que la codificación de caracteres es predeterminada y el tamaño predeterminado del byte-buffer son apropiados.

Para especificar estos valores, construir con un InputStreamReader en un FileInputStream.

FileReader está diseñado para leer flujos de caracteres. Para leer secuencias de bytes sin procesar, considere usar un FileInputStream.

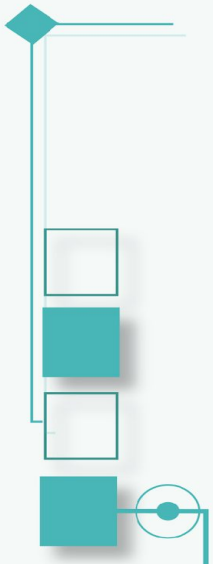
PROTECO

BufferedReader

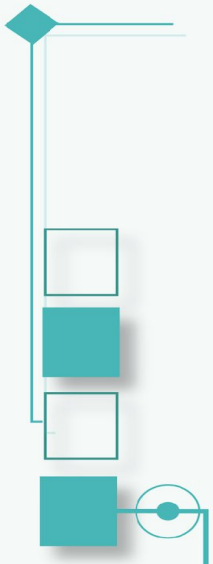
Lee el texto de una secuencia de entrada de caracteres y almacena en búfer los caracteres para proporcionar una lectura eficiente de los caracteres, matrices y líneas.

PROTECO

OutputStreamWriter



OuputStreamReader



StreamTokenizer

La clase StreamTokenizer toma un flujo de entrada y lo analiza en "tokens", permitiendo que los tokens se lean de uno en uno. El proceso de análisis se controla mediante una tabla y una serie de indicadores que se pueden establecer en varios estados. El tokenizer de flujo puede reconocer identificadores, números, cadenas entre comillas y varios estilos de comentarios.

PROTECO


PROTECO


PROTECO