Las clases abstractas son una plantilla para implementar métodos en ellas.

Usamos los flujos de datos para escribir datos por el teclado y que salgan por pantalla.

**Un STREAM o flujo de datos es el canal de comunicación entre el programa y los archivos que queremos leer o escribir.**

**Flujos basados en bytes ENVIAR Y RECIBIR**

Los flujos basados en bytes están orientados tanto a la lectura como a la escritura de información organizada en bytes.

Este tipo de flujos, por su configuración y funcionamiento, son ideales para gestionar ficheros que contengan información estructurada de forma binaria.

El lenguaje de programación Java provee dos clases que nos van a proporcionar toda la funcionalidad necesaria para poder tratar con este tipo de flujos:

• InputStream

• OutputStream

Con la clase **InputStream**, podremos crear flujos de entrada de datos a ficheros basados en bytes, es decir, podremos leer información de ellos.

Con la clase **OutputStream**, podremos crear flujos de salida de datos a ficheros basados en bytes, es decir, podremos escribir información en ellos.

**Flujos basados en caracteres LEER Y ESCRIBIR**

Con los flujos orientados a bytes, podemos tener la funcionalidad necesaria para poder trabajar con la información de una forma muy sencilla, pero estos tienen una limitación importante, y es que no pueden trabajar con caracteres Unicode, ya que estos utilizan 2 bytes para su codificación.

Dependiendo de la funcionalidad que necesitemos, existen muchos tipos de flujos de datos.

Para la implementación de los flujos de caracteres, Java nos va a proporcionar dos clases base:

1. Reader.

2. Writer.

Estas dos clases pueden tratar estos flujos de carácter en carácter, lo cual hace que su tratamiento sea muy lento, y más teniendo en cuenta la ineficiencia del acceso a disco.

Para solucionar esto, lo que haremos será crear otros flujos a partir de estos dos. Estos nuevos flujos van a ser:

1. BufferedReader

2. PrintWriter

Con la clase BufferedReader, podremos crear flujos de entrada de datos a ficheros basados en caracteres, es decir, podremos leer información de ellos.

El BufferedReader reader = new BufferedReader(Ruta del archivo) = Nos indica que es lo que va a leer, la ubicación. Es lo mismo que el FileReader solo que este va más rápido cuando la cantidad de datos que haya a leer sea mucho mayor

El Buffer es una memoria interna que alamacena y lee datos de manera rápida sin gastar recursos de la aplicación.

Cuando tenemos el mismo método con ….. se llama sobrecarga.

IOException indica que hay que controlar al método y que tiene que estar en un Try-Catch.

.

PREGUNTA DE TEORIA FIJA

PERSISTENCIA DE DATOS: Almacenamiento de datos de forma permanente.

¿Que hacen unico a un fichero?

Nombre, ruta de acceso y extension.