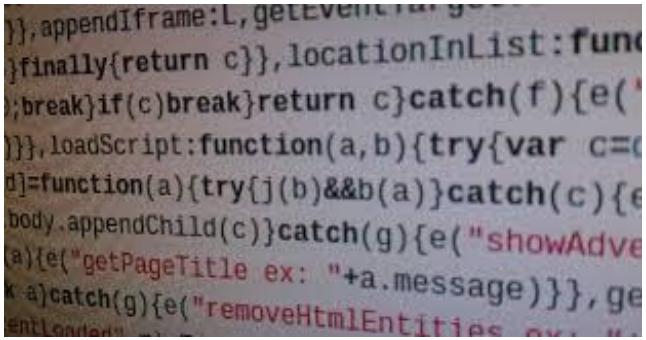


UT 13

REGISTROS, FICHEROS Y OPERACIONES. Primera Parte

IES JUAN DE LA CIERVA
DPTO. INFORMÁTICA

CONTENIDOS
DE LA
PRESENTACIÓN

Objetivos

La unidad permite al alumno conocer los conceptos de los distintos tipos de ficheros con los que nos podemos encontrar (byte y carácter), los distintos tipos de operaciones que podemos realizar con ellos y su manejo.

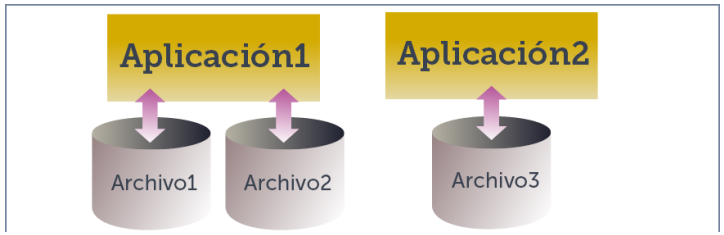
Contenidos

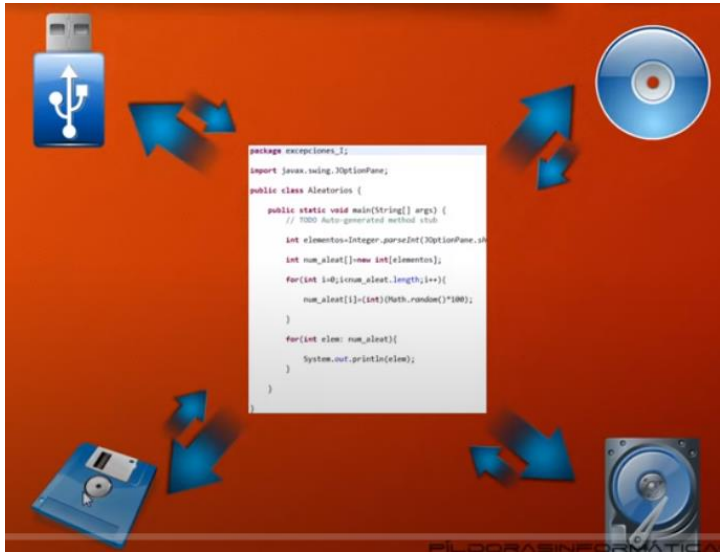
- **Conceptos de ficheros.**
- **Tipos de ficheros (secuenciales y directos).**
- **Operaciones sobre ficheros.**
- **Creación, lectura y escritura ficheros de Texto**.
- Creación, lectura y escritura ficheros binarios.
- Modificación y actualización de ficheros.
- Eliminación de ficheros.

Ficheros

Ficheros

Un archivo o **fichero informático** es un conjunto de bytes que son almacenados en un dispositivo de almacenamiento permanente.





Operaciones con Ficheros

- Creación
- Consulta o Recuperación
- Mantenimiento o actualización
- Borrado o destrucción.

Organización de Ficheros

El acceso a los datos de un fichero está ligado a la organización de éste.

Existen dos tipos de accesos:

- Acceso Secuencial: para acceder al registro n hay que recorrer los n-1 anteriores.
- Acceso Directo: Se puede acceder directamente a un registro a partir de una clave de acceso.

E/S en Java

- Los programas de Java realizan la **E/S a través de flujos (stream).**
- Por tanto, se pueden aplicar **las mismas clases y métodos** E/S **a cualquier** tipo de **dispositivo.**
- **Java implementa los flujos en las clases incluidas en el paquete java.io**

E/S en Java

El acceso a Ficheros se hará mediante operaciones de Entrada/Salida.

El sistema de E/S de Java es extenso, con numerosas clases, interfaces y métodos.

Java define dos sistemas completos de E/S:

- Para bytes
- Para caracteres

FLUJOS DE BYTES Y DE CARACTERES

En Java hay 2 formas de recibir o enviar datos:

- **Flujo de caracteres** : permiten procesar la E/S de caracteres. Usan UNICODE. Ficheros de texto.
- **Flujo bytes:** permiten procesar la E/S de bytes. Uso por ejemplo para leer y escribir datos binarios, Útiles para trabajar con archivos, enviar paquetes de datos a través de una red,..

FLUJOS DE BYTES Y DE CARACTERES

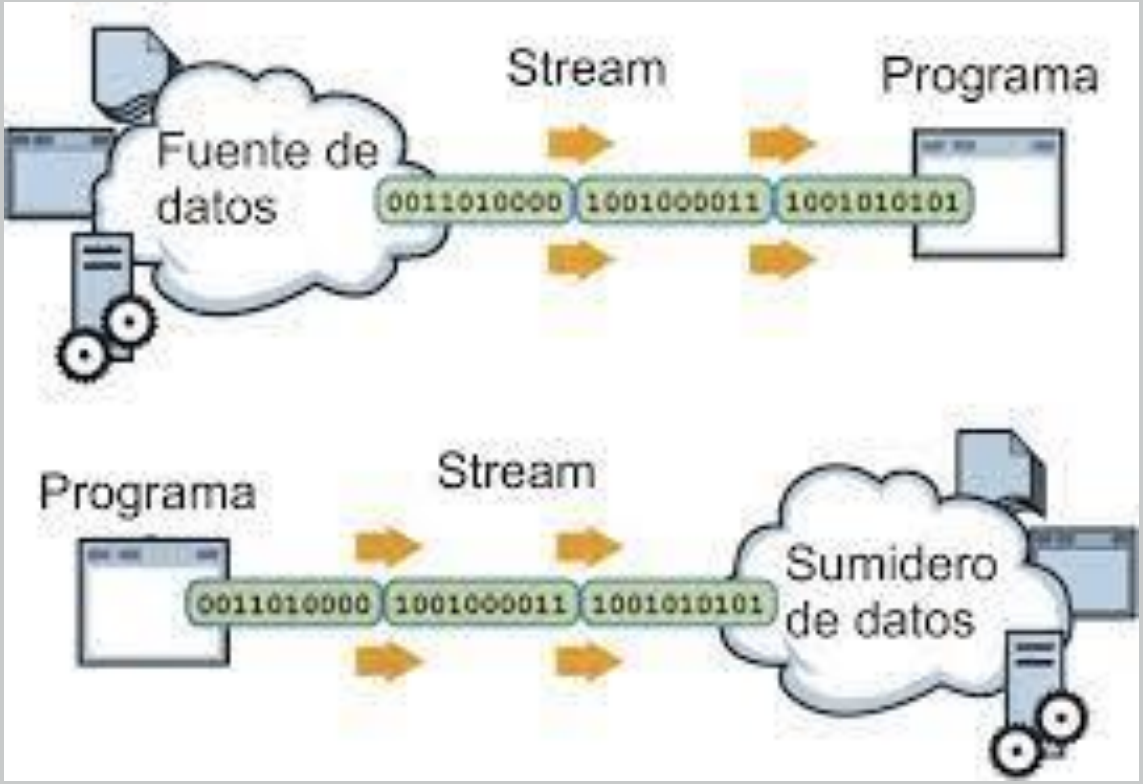
CLASES DE FLUJOS DE BYTES

- Se definen con dos jerarquías de clases:
  - **InputStream** (características flujos entrada)
  - **OutputStream** (comportamiento flujos de salida)

A partir de estas dos clases se crean otras subclases concretas de distinta funcionalidad y que se encargan de los detalles de leer y escribir en distintos dispositivos (ejem. Archivos en disco)

CLASES DE FLUJOS DE CARACTERES

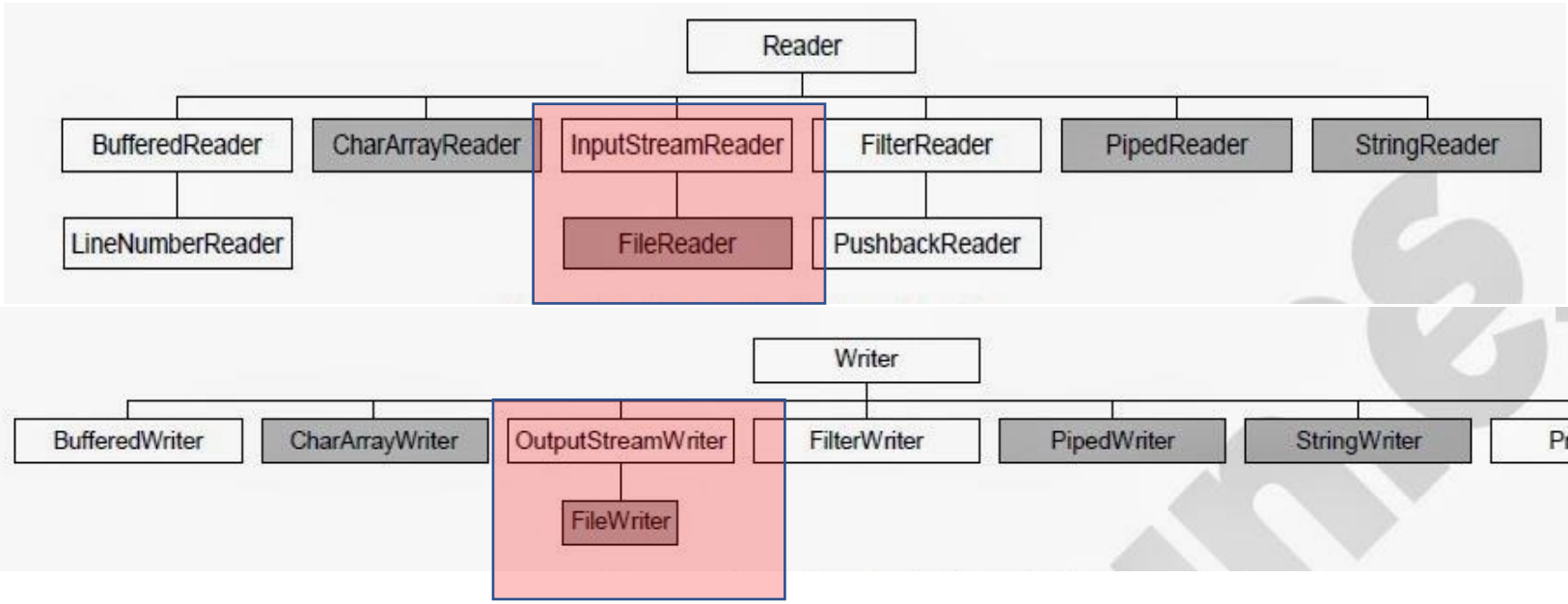
- Se definen con dos jerarquías de clases:
  - **Reader** (características flujos entrada)
  - **Writer** (comportamiento flujos de salida)



Ficheros:Acceso con Flujo de Caracteres

CLASES DE FLUJOS DE CARACTERES

- Se definen con dos jerarquías de clases:
  - **Reader** (características flujos entrada)
  - **Writer** (comportamiento flujos de salida)



FICHEROS DE FLUJO DE CARACTERES

Ventaja: operan directamente con caracteres Unicode. Por tanto, para almacenar un texto Unicode es la mejor opción.

Por lo general, para realizar E/S de archivos basada en caracteres , se usan las clases:

- FileReader
- FileWriter

F.F. CARACTERES – Lectura de Ficheros

Abrimos el fichero de lectura y leeremos la información de forma secuencial.

CONSTRUCTORES DE LA CLASE FileReader:

FileReader nombre = new FileReader(“fichero.extension”);
(Controlar la excepción FileNotFoundException)

MÉTODOS

int read( ): devuelve en ASCII el byte leído. Nos devuelve **-1** si no tiene más información el fichero. En algún caso puede que falle el -1 y debemos preguntar por 65535.

int skip(long): salta n caracteres.

String readLine( ): devuelve toda una línea del fichero.

close() : cierra el stream

F.F. CARACTERES - FILEREADER

Ejemplo:

Leer el fichero de texto pruebaTexto.txt que has creado previamente en el Escritorio.

Lee el texto hasta que alcance final de fichero.

F.F. CARACTERES - FILEWRITER

Abrimos el fichero de escritura, y guardaremos la información carácter a carácter.

CONSTRUCTORES :
FileWriter nombre = new FileWriter(“fichero.extension”);
FileWriter nombre = new FileWriter(“fichero.extension”, boolean a);
a= true añade datos al fichero que ya existe
a= false sobreescribe el fichero

MÉTODOS
void write(caracter): escribe el carácter en el fichero.

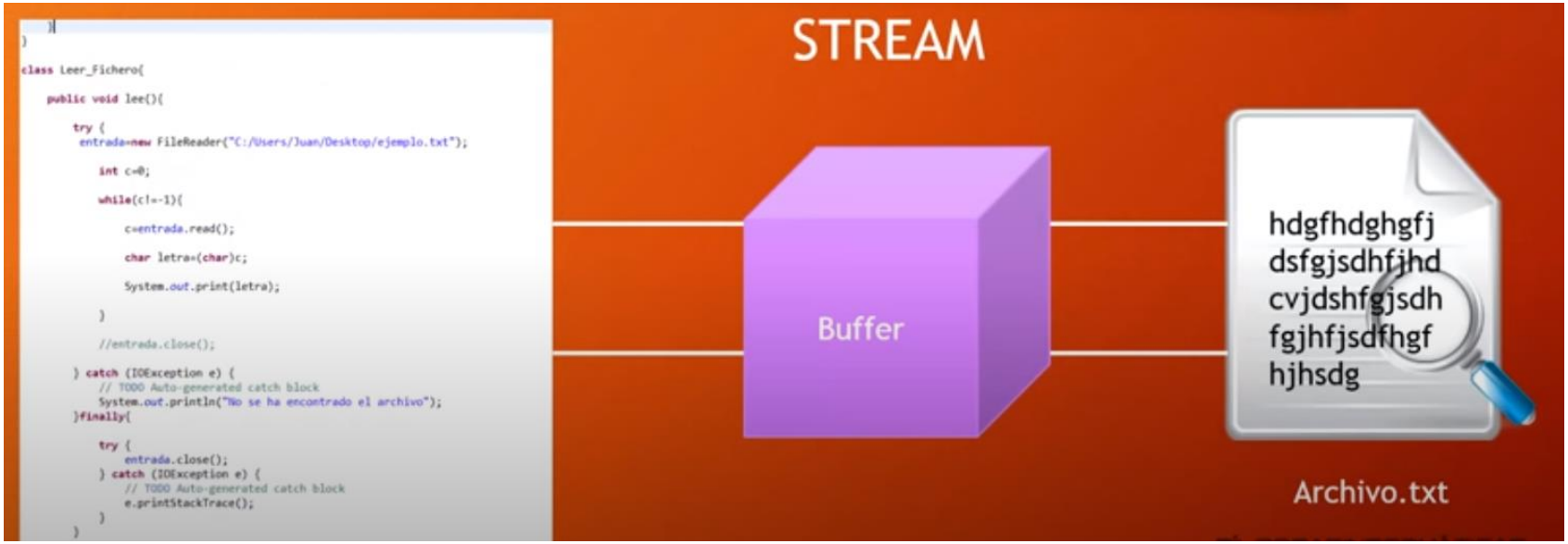
PARA CERRAR EL CANAL
void close( ), cierra el fichero.

F.F. CARACTERES - FILEREADER

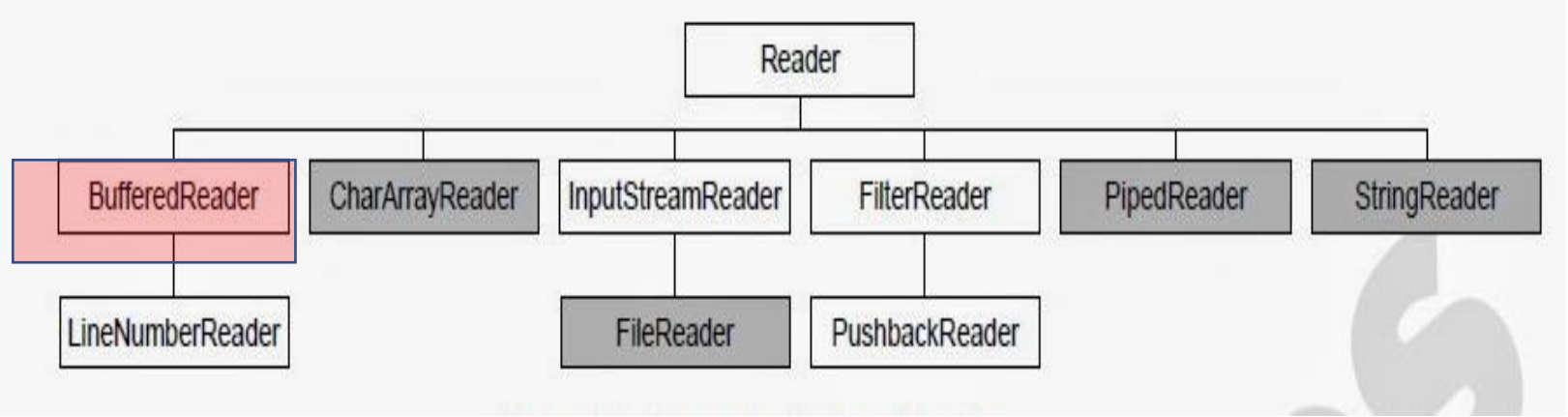
Ejemplo:

Sobreescribir y añadir un texto capturado por teclado, en un fichero de texto pruebaTexto.txt y almacénalo en el Escritorio.

F.F. CARACTERES – FILEREADER
Con Buffer



F.F. CARACTERES – FILEREADER
Con Buffer



Método: readLine();

F.F. CARACTERES – FILEREADER
Con Buffer

Si cada registro está almacenado en una fila del fichero podemos leer todos los datos de una línea y no carácter a carácter.

Debemos utilizar **BufferedReader**

Recordar uso de BufferedReader

```
import java.io.*;

BufferedReader <nombre_objeto> =new
BufferedReader(objeto clase FileReader);

<variable>=<nombre_objeto>.readLine();
```

Preguntaremos por distinto de null, en la lectura del fichero.

F.F. CARACTERES - BufferedReader

Abrimos el fichero, y guardaremos la información carácter a carácter.

CONSTRUCTORES :

BufferedReader(objeto Reader);
BufferedReader(objeto Reader, int tamaño);

MÉTODOS

Int read( ): leer un carácter del fichero.
String readLine(): leer una línea de texto.

PARA CERRAR EL CANAL

void close( ), cierra el fichero.

F.F. CARACTERES - FILEREADER

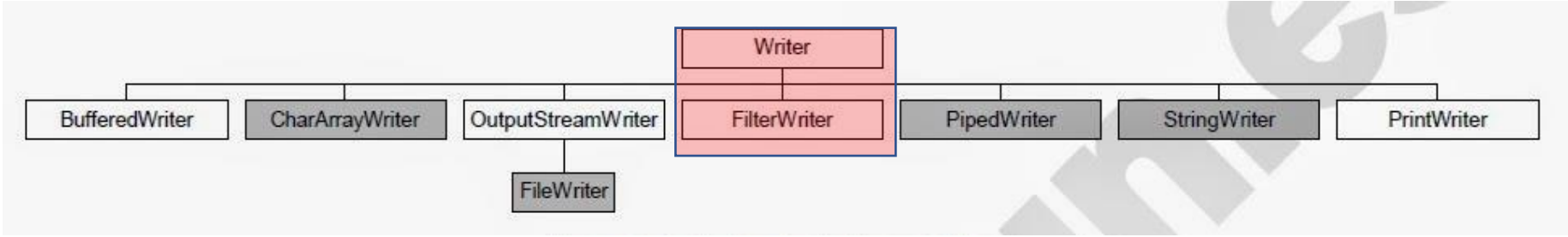
Ejemplo:

Haz un programa que lea un archivo de texto utilizando buffer de entrada.

(*) Ten en cuenta:

- readLine() te devuelve el texto que encuentra hasta un \n o \r.
- Cuando acaba el fichero retorna un null;

CLASES DE FLUJOS DE CARACTERES FILEWRITER



F.F. CARACTERES - FILEWRITER

Ejemplo:

Realiza un programa que escriba en un fichero de texto la información introducida desde teclado.

Lee líneas de texto que acaban al introducir un retorno de carro.

El programa termina cuando el usuario escribe la palabra stop.

Se usa FileWriter para la salida en el archivo.