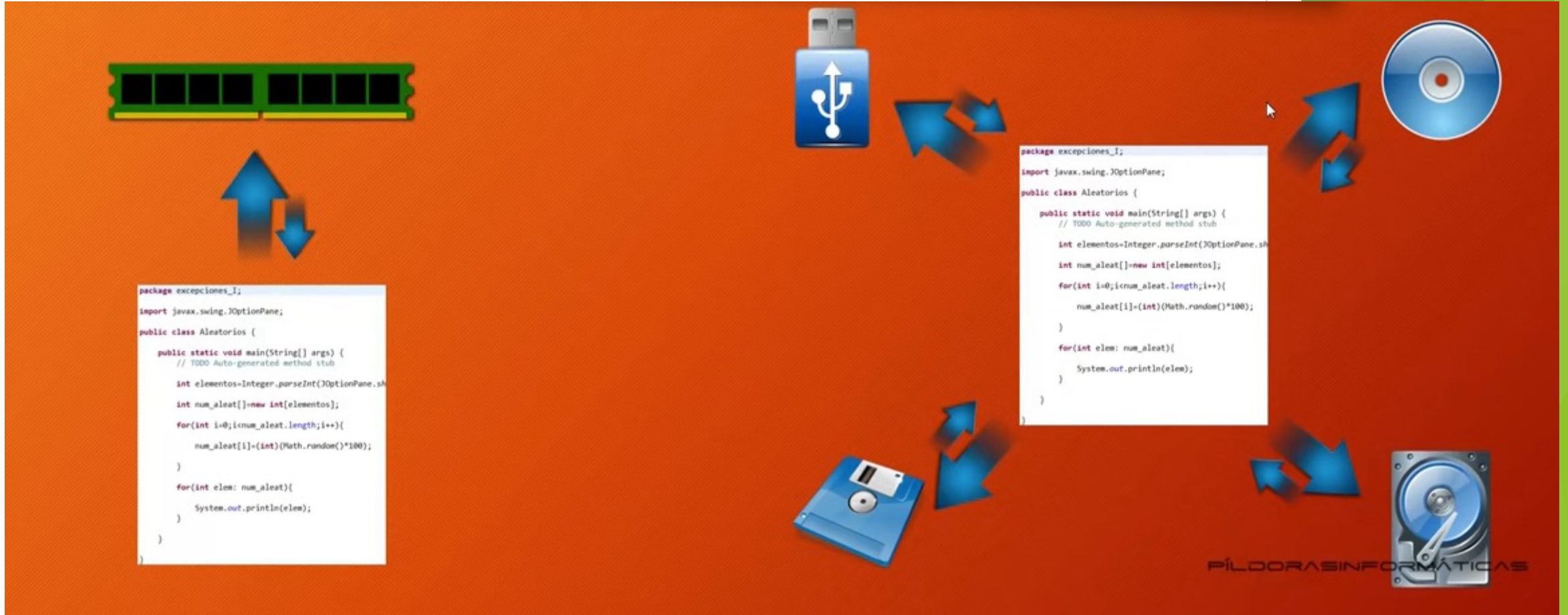


ACCESO A DATOS

UT 1 - MANEJO DE FICHEROS I

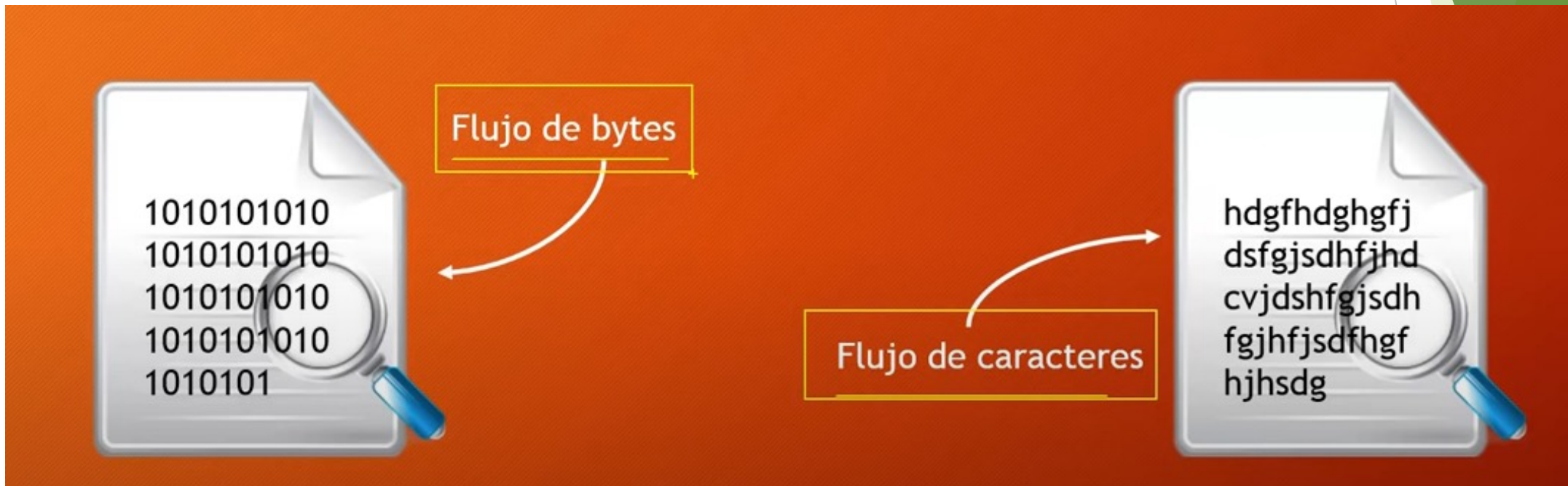
¿Por qué el uso de ficheros?



¿Por qué el uso de ficheros?

- ▶ Cuando guardamos la información en variables o Arrays, esta no persiste en el disco duro.
- ▶ Esto es debido a que la RAM es volátil, su información se pierde al apagar el ordenador
- ▶ Necesitamos guardar la información en otros elementos
 - ▶ Pen drive
 - ▶ Disco duro
 - ▶ ...

Formas de acceso a un fichero

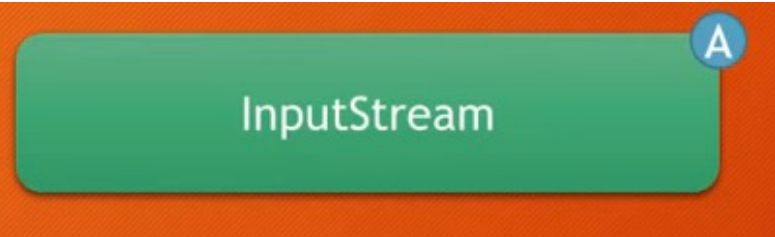


Utilizamos el paquete java.io.*

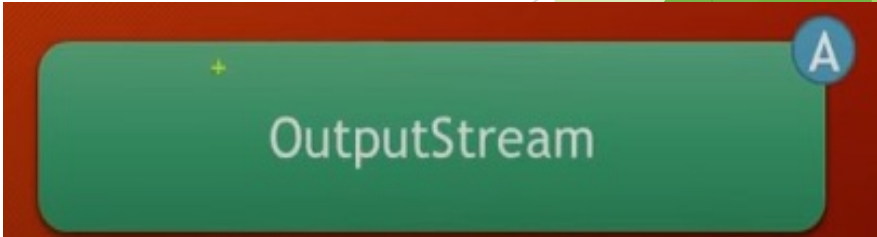
Formas de acceso a un fichero

► Flujo de Bytes:

- Cuando queremos enviar esa información a través de la red (no necesitamos leer la información), viajan 0s y 1s
- Usaremos las clases **abstractas** InputStream y OutputStream.
- InputStream para ver la información en el fichero
- OutputStream para leer la información en el fichero

A diagram showing the 'InputStream' class. It consists of a green rounded rectangle with the text 'InputStream' inside, set against an orange background. A small blue circle with the letter 'A' is in the top right corner.

InputStream

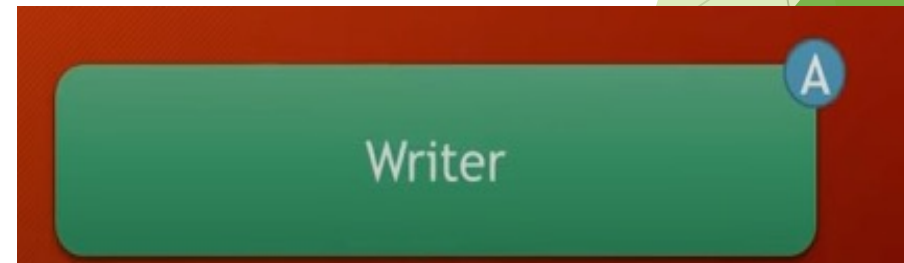
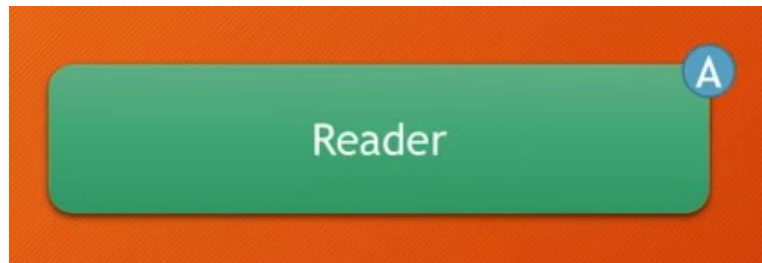
A diagram showing the 'OutputStream' class. It consists of a green rounded rectangle with the text 'OutputStream' inside, set against a dark red background. A small blue circle with the letter 'A' is in the top right corner.

OutputStream

Formas de acceso a un fichero

► Flujo de Caracteres:

- Cuando queremos poder leer la información de un fichero
- Usaremos las clases **abstractas** Reader y Writer
- Reader para leer el fichero
- Writer para escribir el fichero



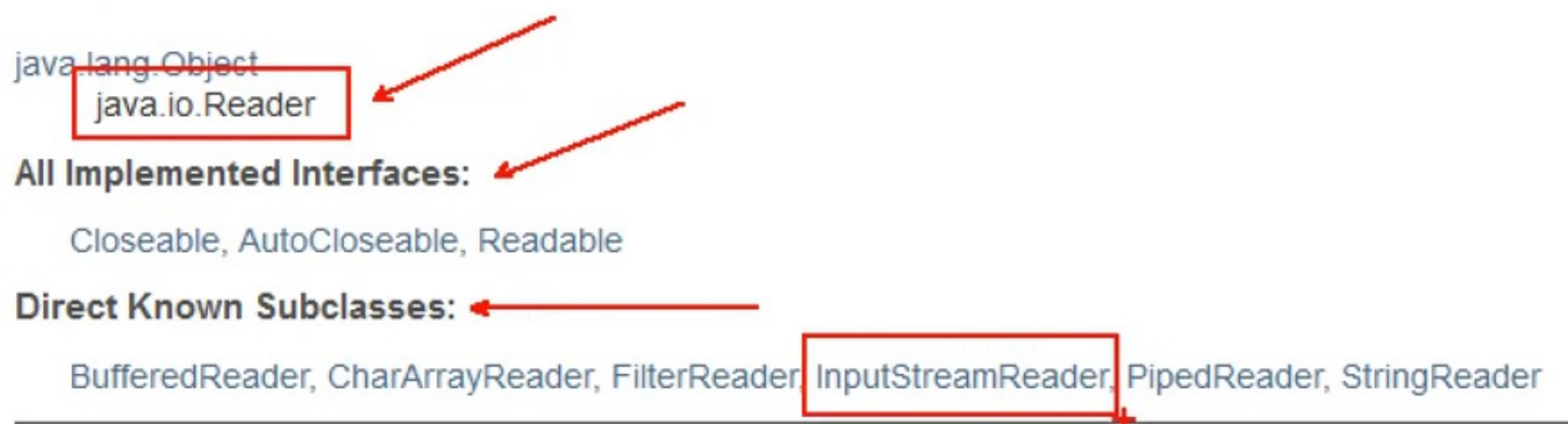
Formas de acceso a un fichero

- Leyendo flujo de caracteres:
 - En la API de Java

java.io

Class Reader

java.lang.Object
java.io.Reader



All Implemented Interfaces:

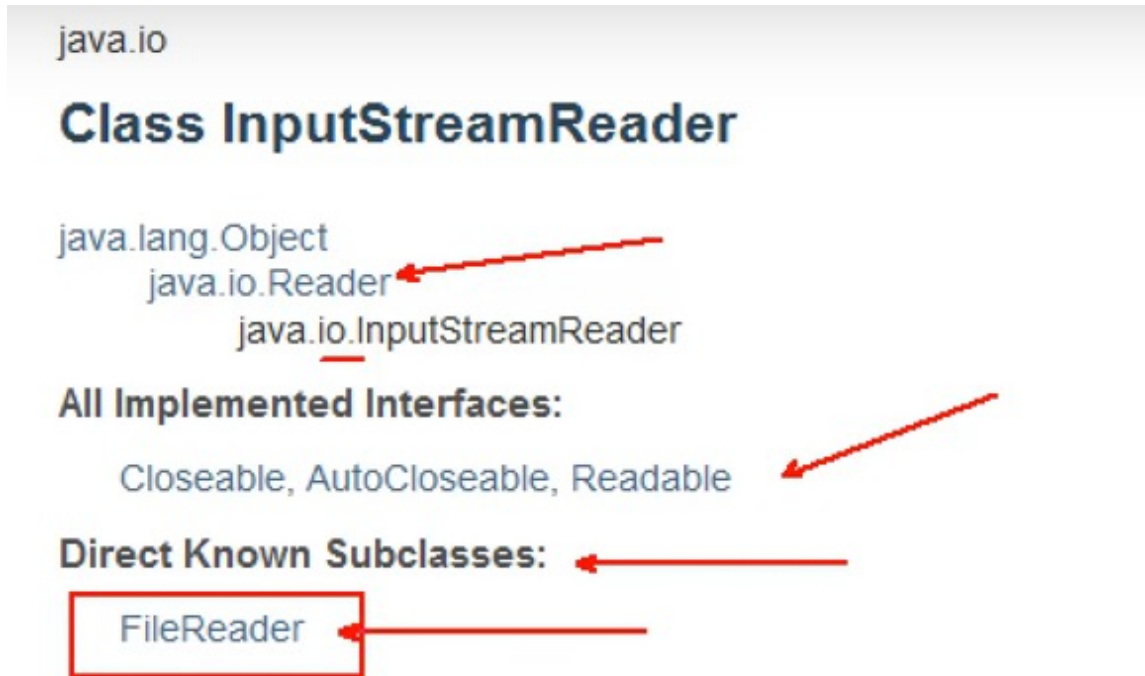
Closeable, AutoCloseable, Readable

Direct Known Subclasses:

BufferedReader, CharArrayReader, FilterReader, **InputStreamReader**, PipedReader, StringReader

Formas de acceso a un fichero

- Leyendo flujo de caracteres:
 - En la API de Java



Formas de acceso a un fichero

- ▶ Leyendo flujo de caracteres:
 - ▶ FileReader:
 - ▶ Convierte el flujo de bytes a caracteres
 - ▶ Usaremos esta clase para especificar donde se encuentra nuestro archivo

Constructor Summary

Constructors

Constructor and Description

`FileReader(File file)`

Creates a new `FileReader`, given the `File` to read from.

`FileReader(FileDescriptor fd)`

Creates a new `FileReader`, given the `FileDescriptor` to read from.

`FileReader(String fileName)`

Creates a new `FileReader`, given the name of the file to read from.

Formas de acceso a un fichero

► Leyendo flujo de caracteres:

► FileReader:

- Este constructor lanza una excepción `FileNotFoundException`
- Estamos obligados a capturar la excepción

FileReader

```
public FileReader(File file)  
+      throws FileNotFoundException
```

Creates a new `FileReader`, given the `File` to read from.

Parameters:

`file` - the `File` to read from

Throws:

`FileNotFoundException` - if the file does not exist, is a directory rather than a regular file,

Formas de acceso a un fichero

- ▶ Leyendo flujo de caracteres:
 - ▶ FileReader:
 - ▶ Usaremos el método `read()` para leer caracteres.
 - ▶ Devuelve entero, -1 al final del fichero y lanza `IOException`.

read

```
public int read()
    throws IOException
```

Reads a single character.

Overrides:

read in class Reader

Returns:

The character read, or -1 if the end of the stream has been reached

Throws:

IOException - If an I/O error occurs

Formas de acceso a un fichero

► ACTIVIDAD 1: Realiza el siguiente programa

```
package accesoafichero;
import java.io.FileReader;
import java.io.IOException;

public class LeerFichero {

    public static void lee(){
        try {
            //Si estamos en LINUX
            FileReader miLector = new FileReader("/Users/alejandrogonzalezredondo/Desktop/"
                + "EjemplosLecturaFicheros/Ejemplo.txt");
            //Si estamos en Windows
            //FileReader miLector = new FileReader("C:/Users/NombreUsuario/Desktop/ejemplo.txt);
            int caracter = miLector.read();
            while (caracter!=-1){
                System.out.println(caracter);
                caracter=miLector.read();
            }
        }
        catch (IOException e) {
            System.out.println("Archivo no encontrado");
        }
    }
}
```

Formas de acceso a un fichero

► ACTIVIDAD 1:

- ¿Ocorre algo extraño?
- Soluciona el problema
- ¿Qué pasa si quitamos el bloque try catch?
- Los flujos deben ser cerrados, hay que llamar al método close(), pero donde?...

```
miLector.close();|
```

Formas de acceso a un fichero

- Escribiendo flujo de caracteres:
 - En la API de Java



Formas de acceso a un fichero

- ▶ Escribiendo flujo de caracteres:
 - ▶ **FileWriter:**
 - ▶ Escribe un flujo de caracteres.

Constructor Summary

Constructors

Constructor and Description

FileWriter(File file)

Constructs a FileWriter object given a File object.

FileWriter(File file, boolean append)

Constructs a FileWriter object given a File object.

FileWriter(FileDescriptor fd)

Constructs a FileWriter object associated with a file descriptor.

FileWriter(String fileName)

Constructs a FileWriter object given a file name.

FileWriter(String fileName, boolean append)

Constructs a FileWriter object given a file name with a boolean indicating whether or not to append the data written.

Formas de acceso a un fichero

- ▶ Leyendo flujo de caracteres:
 - ▶ **FileWriter:**
 - ▶ Este constructor lanza una excepción `IOException`
 - ▶ Estamos obligados a capturar la excepción

FileWriter

```
public FileWriter(String fileName)  
    throws IOException
```

Constructs a `FileWriter` object given a file name.

Parameters:

`fileName` - String The system-dependent filename.

Throws:

`IOException` - if the named file exists but is a directory rather than a regular file, does not exist but cannot be created, or cannot be opened for any other reason

Formas de acceso a un fichero

- ▶ Leyendo flujo de caracteres:
 - ▶ **FileWriter:**
 - ▶ Usaremos el método `write(String str)` para escribir.
 - ▶ Lanza `IOException`.

write

```
public void write(String str)  
           throws IOException
```

Writes a string.

Parameters:

`str` - String to be written

Throws:

`IOException` - If an I/O error occurs

Formas de acceso a un fichero

► ACTIVIDAD 2: Realiza el siguiente programa

```
package accesoafichero;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
public class EscribirFichero {

    public static void escribe(){
        try {
            //Si estamos en LINUX
            FileWriter miEscriotor = new FileWriter("/Users/alejandrogonzalezredondo/Desktop/"
                + "EjemplosLecturaFicheros/Ejemplo.txt");
            //Si estamos en Windows
            //FileWriter miEscriotor = new FileWriter("C:/Users/NombreUsuario/Desktop/ejemplo.tx
            String laFrase="Hola estamos escribiendo en un archivo";
            for (int x=0; x<laFrase.length();x++){
                miEscriotor.write(laFrase.charAt(x));
            }
            System.out.println("Escritura terminada");
            miEscriotor.close();
        }
        catch (IOException e) {
            System.out.println("Archivo no encontrado");
        }
    }
}
```

Formas de acceso a un fichero

► ACTIVIDAD 2:

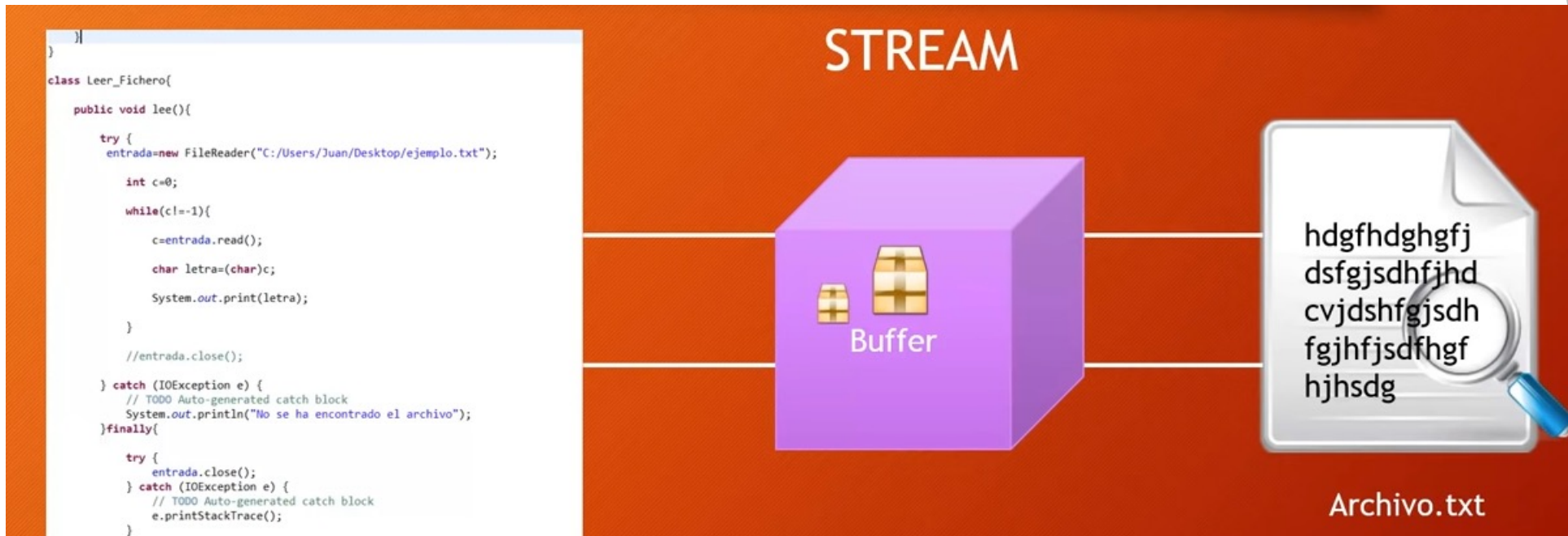
- ¿qué ocurrió con lo que había escrito ya?
- ¿Qué ocurre si el archivo a escribir no existía?
- Podemos usar el constructor sobrecargado para que escriba a continuación del texto anterior si el archivo existía previamente.

```
FileWriter miEscrivor = new FileWriter("/Users/alejandrogonzalezredondo/Desktop/"  
+ "EjemplosLecturaFicheros/Ejemplo.txt", true);
```

Formas de acceso a un fichero

- ▶ Flujo de Caracteres, INCONVENIENTES:
 - ▶ Cada vez que escribimos o leemos un carácter se genera comunicación entre el programa y el archivo, si tenemos que hacer millones de accesos el proceso no será óptimo.
 - ▶ Solución: Usar un buffer.
 - ▶ ¿Qué es un buffer?
 - ▶ Es una memoria intermedia que almacena todo el contenido del archivo, de forma que nuestro programa leerá o escribirá en el buffer en vez de acceder constantemente al archivo.

Formas de acceso a un fichero



Formas de acceso a un fichero

► ACTIVIDAD 3:

- Visita la API de Java y lee detenidamente las clases **BufferedReader** y **BufferedWriter**.
- Familiarízate con sus constructores y métodos.

Formas de acceso a un fichero

► ACTIVIDAD 4: Realiza el siguiente programa

```
package accesoafichero;
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;
public class LeerConBuffer {
    public static void leeBuffer(){
        try {
            FileReader miLector = new FileReader("/Users/alejandrogonzalezredondo/Desktop/
            + "EjemplosLecturaFicheros/EjemploBuffered.txt");
            BufferedReader miBuffer = new BufferedReader(miLector);
            String linea="";

            while (linea!=null){
                linea=miBuffer.readLine();
                if (linea!=null)
                    System.out.println(linea);
            }
        }
        catch (IOException e) {
            System.out.println("Archivo no encontrado");
        }
    }
}
```

Formas de acceso a un fichero

► ACTIVIDAD 5:

- Realiza un programa que permita escribir en un fichero utilizando la clase `BufferedWriter`.
- Realiza otro programa que permita escribir frases mientras el usuario lo deseé.