

Asignatura

Programación



UNIVERSAE
Instituto Superior de FP

Asignatura

Programación

UNIDAD 12

Lectura y escritura de información



UNIVERSAE
Instituto Superior de FP



Flujos (Streams)

¿Que són?

Información de una comunicación de un origen a un destino

- Origen. Teclado, Fichero, Red, etc..
- Destino. Pantalla, Fichero, etc..



Tipos

- Según los datos
 - Bytes o byte streams
 - Caracteres o carácter streams
 - Estándar o predeterminados
- Según la dirección
 - Entrada
 - Salida
 - Entrada/salida
- Según la forma de acceso
 - Secuencial
 - Directo



Flujo estándar o predeterminado

- Son los propios que proporciona java
- Entrada y salida de datos por teclado y pantalla
 - `System.in`
 - `System.out`
 - `print`
 - `println`
 - `System.err`
- Es posible cambiar la salida.
 - `System.setOut`
 - `System.setErr`
 - `Java clase.java > salida.txt 2> salidaError.txt`
- No hay apertura del flujo
- Tampoco hay cierre del flujo.

```
Scanner sc = new Scanner(System.in);
System.out.println("Escriba un dato:");
String dato = sc.nextLine();
System.out.println("El dato introducido es: " + dato);
// la salida err es independiente de out puede ejecutarse antes que out
System.err.println("FIN");
```

```
Escriba un dato:
información
El dato introducido es: información
FIN
```



Uso de los flujos

Apertura

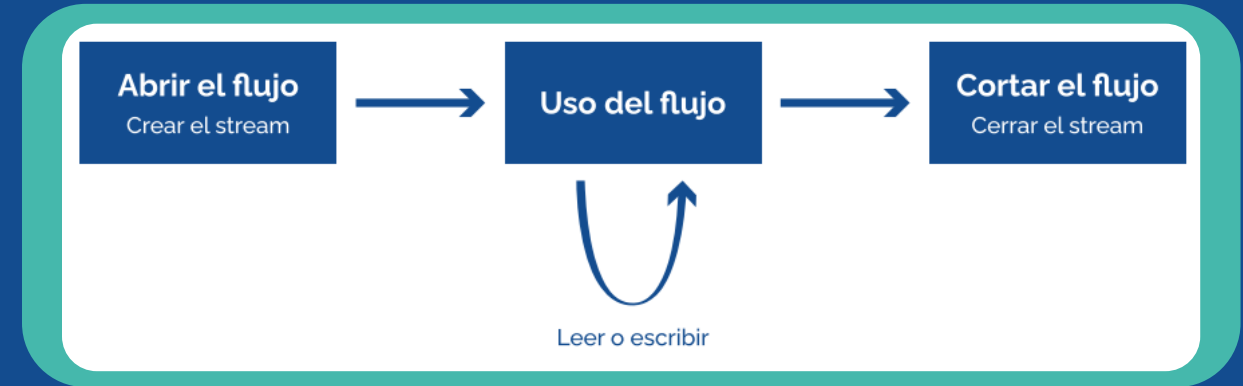
- Crear el fichero si no existe y/o abrirlo para su lectura.
- La apertura puede determinar desde donde se empieza a escribir o leer.

Uso

- Acceso secuencial
 - Tipo texto
 - ❑ FileReader, FileWriter, BufferedReader, ...
 - Tipo binario
 - ❑ FileInputStream, FileOutputStream, ...
- Acceso directo
 - RandomAccessFile

Cierre

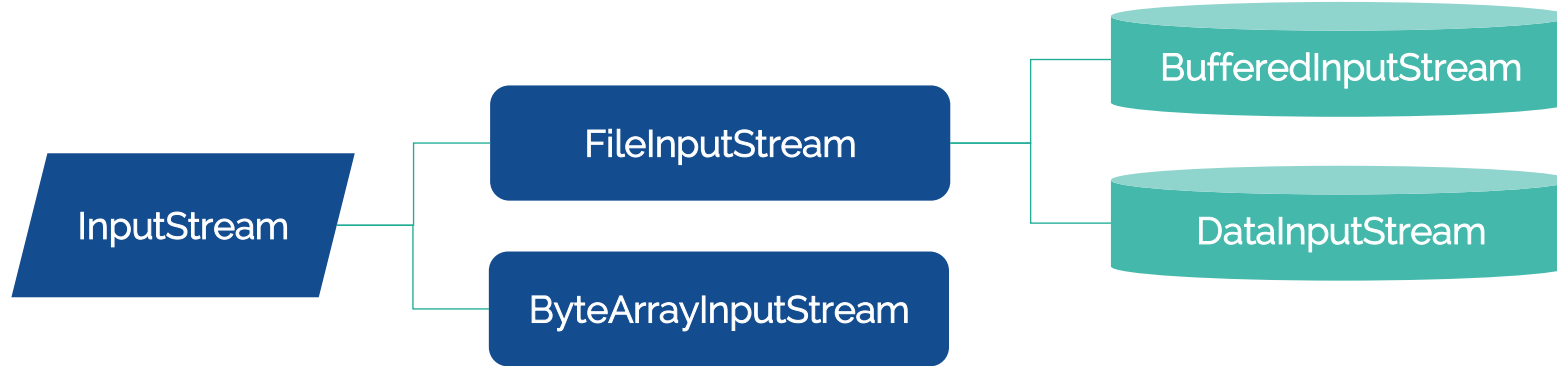
- Fin del uso del fichero
- Liberación de memoria, punteros y otras referencias.
- Close()



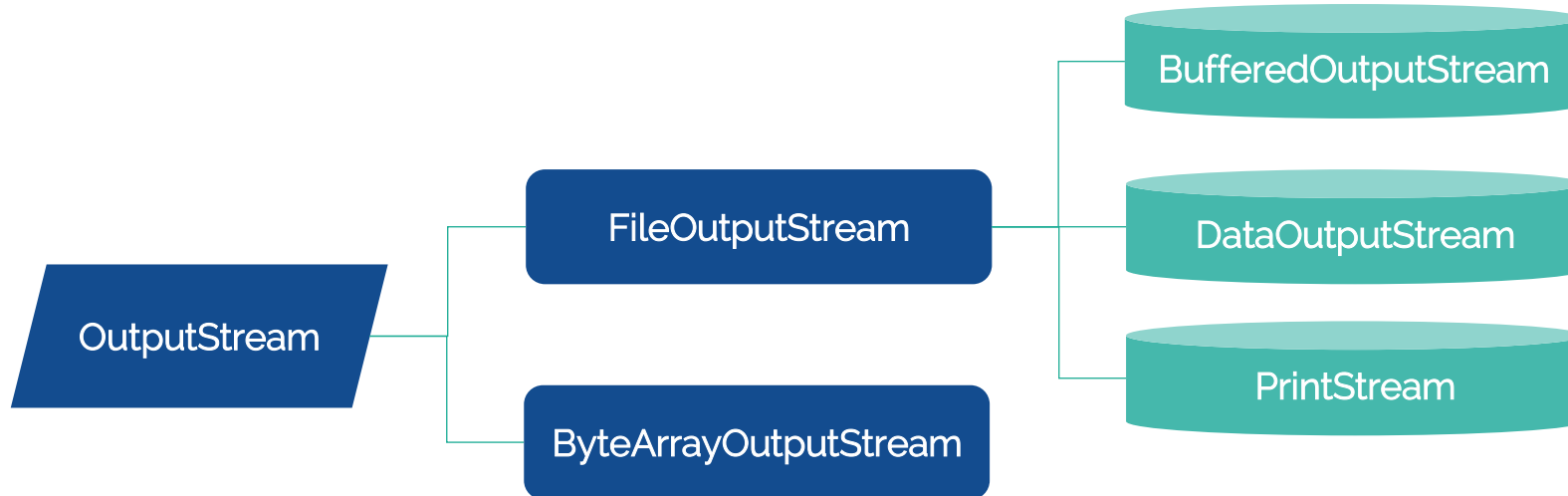


Clases para el manejo de flujos de bytes

Entrada



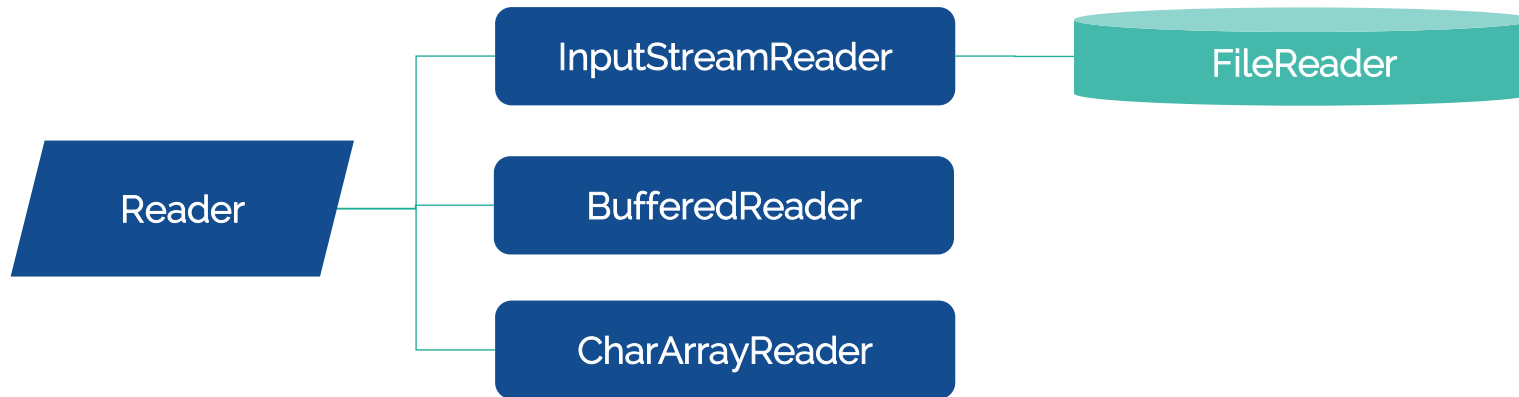
Salida



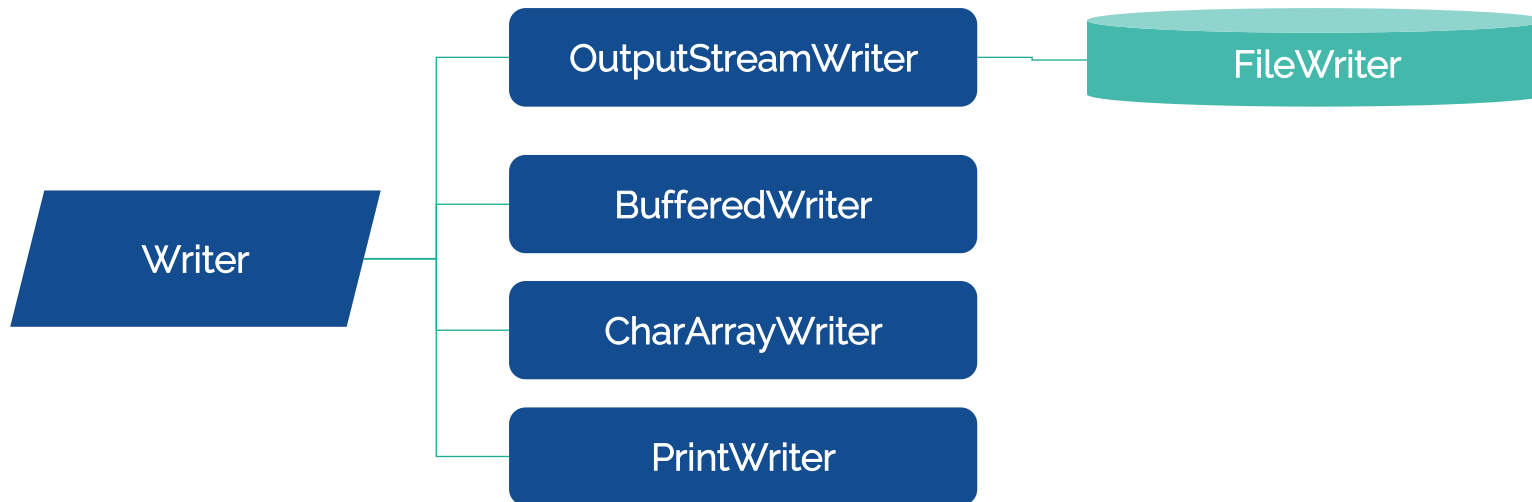


Clases para el manejo de flujos de caracteres

Entrada



Salida





Flujos. Ejemplos

```
char[] vocales = { 'A', 'E', 'I', 'O', 'U' };
String rutaFichero = "C:\\documentos\\fichero.bin";
```

```
FileOutputStream fos = new FileOutputStream(rutaFichero);
BufferedOutputStream bos = new BufferedOutputStream(fos);
for (int i = 0; i < vocales.length; i++) {
    bos.write((byte) vocales[i]);
}

bos.close();
fos.close();
```

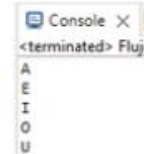


```
// Tiene que existir previamente el fichero, si no, lanzará una excepción
String rutaFichero = "C:\\documentos\\fichero.bin";
```

```
FileInputStream fis = new FileInputStream(rutaFichero);
BufferedInputStream bis = new BufferedInputStream(fis);

int i;
while ((i = bis.read()) != -1) {
    System.out.println((char) i);
}

bis.close();
fis.close();
```

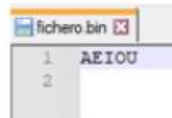


```
char[] vocales = { 'A', 'E', 'I', 'O', 'U' };
String rutaFichero = "C:\\documentos\\fichero.bin";
```

```
FileOutputStream fos = new FileOutputStream(rutaFichero);
ByteArrayOutputStream bos = new ByteArrayOutputStream();
```

```
for (int i = 0; i < vocales.length; i++) {
    bos.write((byte) vocales[i]);
}
```

```
bos.writeTo(fos);
bos.flush();
bos.close();
fos.close();
```

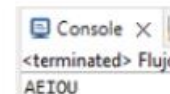


```
// Tiene que existir previamente el fichero, si no, lanzará una excepción
String rutaFichero = "C:\\documentos\\fichero.bin";
```

```
FileInputStream fis = new FileInputStream(rutaFichero);
ByteArrayInputStream bis = new ByteArrayInputStream(fis.readAllBytes());
```

```
int contenido = 0;
while ((contenido = bis.read()) != -1) {
    System.out.print((char) contenido);
}
```

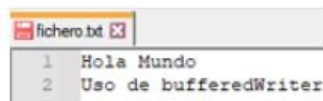
```
bis.close();
fis.close();
```



```
String rutaFichero = "C:\\documentos\\fichero.txt";
```

```
FileWriter fw = new FileWriter(rutaFichero);
BufferedWriter bw = new BufferedWriter(fw);
```

```
bw.write("Hola Mundo");
bw.newLine();
bw.write("Uso de bufferedWriter");
bw.close();
fw.close();
```

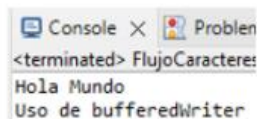


```
// Tiene que existir previamente el fichero, si no, lanzará una excepción
String rutaFichero = "C:\\documentos\\fichero.txt";
```

```
FileReader fr = new FileReader(rutaFichero);
BufferedReader br = new BufferedReader(fr);
```

```
int i;
while ((i = br.read()) != -1) {
    System.out.print((char) i);
}
```

```
br.close();
fr.close();
```



```
String ruta = "C:\\documentos\\configuracion\\config.bin";
RandomAccessFile raf = new RandomAccessFile(ruta, "rw");
raf.writeChars("HOLA MUNDO");
```

```
// LECTURA
```

```
raf.seek(0);
for (int i = 0; i < raf.length() / 2; i++)
    System.out.print(" " + raf.readChar());
System.out.println();
```

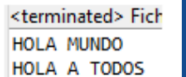
```
// ESCRITURA
```

```
raf.seek(8);
raf.writeChars(" A TODOS");
```

```
// LECTURA
```

```
raf.seek(0);
for (int i = 0; i < raf.length() / 2; i++) {
    System.out.print(" " + raf.readChar());
}
```

```
raf.close();
```



Los ficheros

Almacenamiento en ficheros

- Inicio de las bases de datos
- Dar persistencia a los datos
- Permiten consultar y/o modificar en cualquier momento
- Los ficheros también guardan metainformación (tamaño, permisos, fecha, etc.

Tipos de ficheros



Texto

- Contienen cadena de caracteres
- Forma de acceso secuencial.
- Sus datos son legibles.



Binarios

- Contienen bytes
- Forma de acceso: secuencial o directa.
- Sus datos son legibles para los programas o sistemas.



Uso del sistema de ficheros

| Operación | Descripción |
|-------------------------|---|
| File (String pathname) | Constructor, inicializa el objeto file con la ruta indicada. |
| boolean createNewFile() | Crea un fichero vacío en la ruta que contiene <i>File</i> |
| boolean mkdir() | Crea un directorio en la ruta que contiene <i>File</i> |
| boolean delete() | Elimina el fichero o directorio en la ruta que contiene <i>File</i> |

| Operación | Descripción |
|-----------------------|--|
| boolean exists() | Comprueba si la ruta existe. |
| boolean isFile() | Comprueba si es un fichero. |
| boolean isDirectory() | Comprueba si es un directorio. |
| File[] listFiles() | Obtiene un listado de tipo <i>File</i> de los ficheros que hay en la ruta. |
| String getName() | Obtiene el nombre del fichero o directorio. |
| String getParent() | Obtiene el directorio padre. |
| String getPath() | Obtiene la ruta. |
| Boolean canRead() | Comprueba si se puede leer. |
| Boolean canWrite() | Comprueba si se puede escribir. |
| Boolean canExecute() | Comprueba si se puede ejecutar. |

```
String ruta = "C:\\documentos\\configuracion";

File rutaPrincipal = new File(ruta);
System.out.println("Creando directorio: " + rutaPrincipal.mkdir());

File ficheroConfiguracion = new File(ruta, "config.txt");
System.out.println("Creando fichero: " + ficheroConfiguracion.createNewFile());

System.out.println(ficheroConfiguracion.isFile());
System.out.println(ficheroConfiguracion.canRead());
System.out.println(ficheroConfiguracion.canWrite());
System.out.println(ficheroConfiguracion.canExecute());
```

Serialización

¿Qué es?

- Proceso de almacenar objetos en ficheros
- Se transforma en un conjunto de bytes

Condiciones que debe tener una clase para serializarla

- Debe implementar la interfaz *Serializable*
- Todos los campos deben ser serializables
 - Los primitivos ya son serializables
 - Otras clases deben implementar la interfaz Serializable
- Los campos static no serán serializados, para cualquier otro que no sea serializable hay que usar la palabra reservada **transient**
- Debe disponer de un campo estático y privado **serialVersionUID**

Clases para el flujo de serialización

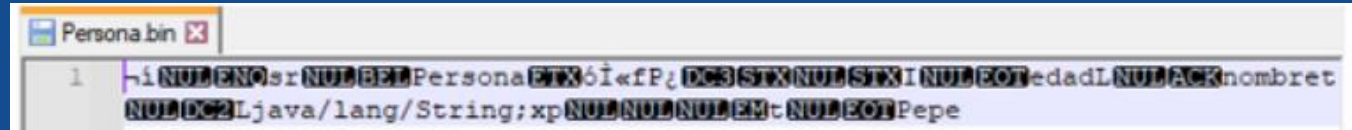
- ObjectOutputStream
 - writeObject()
- ObjectInputStream
 - readObject()



```
String rutaFichero = "C:\\documentos\\Persona.bin";

// Escritura
FileOutputStream fos = new FileOutputStream(rutaFichero);
ObjectOutputStream oos = new ObjectOutputStream(fos);
Persona persona = new Persona("Pepe", 25);
oos.writeObject(persona);
oos.close();

// Lectura
FileInputStream fis = new FileInputStream(rutaFichero);
ObjectInputStream ois = new ObjectInputStream(fis);
Persona objeto = (Persona) ois.readObject();
System.out.println("Nombre: " + objeto.getNombre() + " Edad: " + objeto.getEdad());
ois.close();
```





Ejercicios propuestos.



Ejercicio 1

Crear un fichero con el abecedario completo. Se pide que se lea el fichero y donde haya una consonante se sustituya por un asterisco *. Se tiene que trabajar siempre con el mismo fichero.



Ejercicio 2

Dada una ruta (path) cualquiera, se solicita que muestre por pantalla:

- El total de ficheros.
- El total de carpetas.

A tener en cuenta, se pide todos los ficheros que hay en la ruta y dentro de cada carpeta que hubiera. A demás puede ser que se introduzca mal una ruta (path).



Resumen

1. Flujos o streams
2. Flujo estándar o predeterminado
3. Uso de los flujos
4. Clases para el manejo de flujos de bytes
5. Clases para el manejo de flujos de caracteres
6. Flujos. Ejemplos
7. Los ficheros
8. Uso del sistema de ficheros
9. Serialización
10. Serialización. Ejemplo
11. Ejercicios propuestos

The background is a solid blue color. Overlaid on this are several faint, light-blue geometric patterns. These include a grid of small squares that form larger, irregular shapes, and numerous small, light-blue arrows pointing in various directions. The overall effect is a sense of movement and digital connectivity.

UNIVERSAE

— CHANGE YOUR WAY —