

# MANEJO DE FICHEROS STREAMS DE CARACTERES

Javier García-Retamero Redondo

# ¿QUÉ SON?

- **Flujo o Stream:**
- Canal de comunicación que se establece para cualquier envío o recepción de información entre una fuente y un destino.
- Trabajaremos con dos tipos:
  - Flujos de 16 bits: Operaciones E/S de **caracteres** (Unicode).
    - Clases descendientes de **Reader** y **Writer**.
  - Flujos de 8 bits: Operaciones E/S de **bytes**.
    - Clases descendientes de **InputStream** y **OutputStream**.

# TRABAJANDO CON CARACTERES

LECTURA



# ¿QUÉ PODEMOS HACER?

- La clase **Reader** nos permite:
  - **Leer** caracteres Unicode de un fichero de texto.
  - La subclase más utilizada es **FileReader**: Lee información de un fichero.

# LECTURA

Suponiendo que el fichero ejemploprueba.txt existe.

- **Abrimos el fichero:**

```
File ficheroPrueba = new File("C:\\acceso_a_datos\\tema1\\ejemploprueba.txt");
```

- **Creamos el flujo de lectura:**

```
FileReader ficheroIn = new FileReader(ficheroPrueba);
```

- **Leemos el código ASCII del carácter contenido en el fichero:**

```
int i;
```

```
i= ficheroIn.read();
```

Devuelve -1 si es final de fichero

# LECTURA

- **Lo pasamos a carácter:**

```
System.out.print((char) i);
```

- **Cerramos el flujo:**

```
ficheroIn.close();
```

# LECTURA

Si queremos leer varios caracteres de golpe utilizaremos:

```
int read(char[] cadena)
```

- Devuelve -1 si llega al final del fichero
- Devuelve en **cadena** los caracteres leídos.

```
int i;
```

```
char cadena[] = new char[20];
```

```
i= ficheroIn.read(cadena);
```

Especificamos el tamaño de la cadena → los caracteres a leer



# LECTURA

Si queremos leer varias líneas a la vez utilizaremos **BufferedReader**:

```
File ficheroPrueba = new File("C:\\acceso_a_datos\\tema1\\ejemploprueba.txt");  
FileReader ficheroIn = new FileReader(ficheroPrueba);  
BufferedReader bufferficheroIn = new BufferedReader( ficheroIn);  
String linea;  
linea = bufferficheroIn.readLine();
```

Devuelve null cuando no  
quedan líneas por leer



# TRABAJANDO CON CARACTERES

ESCRITURA



# ¿QUÉ PODEMOS HACER?

- La clase **Writer** nos permite:
  - **Escribir** caracteres Unicode en un fichero de texto.
  - La subclase más utilizada es **FileWriter**: Escribir información en un fichero.

# ESCRITURA

- **Abrimos el fichero:**

```
File ficheroPrueba = new File("C:\\acceso_a_datos\\tema1\\ejemploprueba.txt");
```

- **Creamos el flujo de escritura:**

- **Opción 1:**

```
FileWriter ficheroOut = new FileWriter(ficheroPrueba);
```

Si el fichero no existe lo crea.  
Si existe lo destruye y lo vuelve a crear.

- **Opción 2:**

```
FileWriter ficheroOut = new FileWriter(ficheroPrueba, true);
```

Añadimos datos y no destruimos el fichero ya creado.

# ESCRITURA

- **Escribir un carácter:**

```
char i="A";  
ficheroOut.write(i);
```

- **Escribir un array con caracteres:**

```
char[] array= {'H','o','l','a'};  
ficheroOut.write(array);
```

# ESCRITURA

- **Escribir un String:**

```
String cadena="Adios";  
ficheroOut.write(cadena);
```

- **Añadir un carácter al final del archivo:**

```
char c='Z';  
ficheroOut.append(c);
```

# ESCRITURA

- **Cerramos el flujo:**

```
ficheroOut.close();
```

# ESCRITURA

Si queremos escribir varias líneas a la vez utilizaremos **BufferedWriter**:

```
File ficheroPrueba = new File("C:\\acceso_a_datos\\tema1\\ejemploprueba.txt");  
FileWriter ficheroOut = new FileWriter(ficheroPrueba);  
BufferedWriter bufferficheroOut = new BufferedWriter(ficheroOut);  
bufferficheroOut.write("otra fila");  
bufferficheroOut.newLine();
```

Salto de línea



# ESCRITURA

También podemos utilizar **PrintWriter**:

```
File ficheroPrueba = new File("C:\\acceso_a_datos\\tema1\\ejemploprueba.txt");  
FileWriter ficheroOut = new FileWriter(ficheroPrueba);  
PrintWriter bufferficheroOut = new PrintWriter(ficheroOut);  
bufferficheroOut.println("otra fila");
```

Escribe la fila + Salto de línea