Streams de Entrada/Salida

- Los Canales de I/O Estándar
  - Entrada estandard: referenciado por System.in
    - utilizado para la entrada del programa, típicamente lee la entrada introducida por el usuario.
  - Salida estandard: referenciado por System.out
    - utilizado para la salida del programa, típicamente muestra información al usuario.
  - Error estandard: referenciado por System.err
    - utilizado para mostrar mensajes de error al usuario.

Streams de Entrada/Salida

- I/O: Leer y Escribir
  - Frecuentemente los programas necesitan traer información desde una fuente externa o enviar información a un fuente externa. La información puede estar en cualquier parte, en un archivo, en disco, en algún lugar de la red, en memoria o en otro programa. También puede ser de cualquier tipo: objetos, caracteres, imágenes o sonidos.

Streams de Entrada/Salida

- En este capítulo solamente analizaremos el caso de los streams de archivos.
- Clase File
  - En esta clase se define un archivo o un directorio.
  - Ejemplo:
    - File archivo = new File(“archivo.txt”);
    - File directorio = new File(“directorio”);

Streams de Entrada/Salida

```
import java.io.*;
public class Copiar {
 public static void main(String[] args) throws IOException {
        File inputFile = new File("entrada.txt");
        File outputFile = new File("salida.txt");
        FileReader in = new FileReader(inputFile);
        FileWriter out = new FileWriter(outputFile);
        int c;
        while ((c = in.read()) != -1)
      out.write(c);
        in.close();
        out.close();
      }
}
```

Streams de Entrada/Salida

- PipedWriter y PipedReader.
- SequenceInputStream
  - Crea un sólo stream de entrada desde varias fuentes de entrada (se utiliza para concatenar).
- FilterInputStream y FilterOutputStream.

Streams de Entrada/Salida

- Ejercicio
  - Escribir un programa que lea los contenidos de un archivo e imprima en la salida estos contenidos en MAYUSCULAS.
  - Hint:
    - Utilice para la lectura:

- BufferedReader input = new BufferedReader(new FileReader(archivo));
- PrintWriter output = new PrintWriter(new FileWriter("copia.txt"));

- ver: LeerArchivo.java y ManipularArchivo.java

Streams de Entrada/Salida

- Contructor:
- File(File parent, String child)
- Creates a new File instance from a parent abstract pathname and a child pathname string.
- File(String pathname)
- Creates a new File instance by converting the given pathname string into an abstract pathname.
- File(String parent, String child)
- Creates a new File instance from a parent pathname string and a child pathname string.

Streams de Entrada/Salida

- getName()
- Returns the name of the file or directory denoted by this abstract pathname.
- getParent()
- Returns the pathname string of this abstract pathname's parent, or null if this pathname does not name a parent directory.
- isDirectory()
- Tests whether the file denoted by this abstract pathname is a directory.
- length()
- Returns the length of the file denoted by this abstract pathname.
- list(FilenameFilter filter)
- Returns an array of strings naming the files and directories in the directory denoted by this abstract pathname that satisfy the specified filter.
- listFiles(FileFilter filter)
- Returns an array of abstract pathnames denoting the files and directories in the directory denoted by this abstract pathname that satisfy the specified filter.

Streams de Entrada/Salida

- Ejercicio
  - Haga un programa que reciba como parámetro el nombre de un directorio.
  - Este programa debe imprimir la lista de los directorio y archivos que los componen.
  - Su programa debe desplegar el tamaño de los archivos.
  - Modifica su programa para que imprima la lista de los sub directorios.

  - ver: readdir.java