

## BOLETÍN PRÁCTICA 1

### JFLEX COMO HERRAMIENTA GENÉRICA PARA EL PROCESAMIENTO DE ARCHIVOS

#### OBJETIVOS



Tras la realización de este boletín, y del estudio de la parte correspondiente de la asignatura, deberás ser capaz de:

- Utilizar JFlex como herramienta genérica, para realizar filtrado de archivos o implementar funcionalidades ad hoc.

### TEMÁTICA

Aunque jflex es una herramienta que combinado con java cup permite la implementación de compiladores, también puede utilizarse para realizar otras funciones. En esta práctica, utilizaremos jflex para implementar pequeñas aplicaciones que permitan realizar algunas operaciones específicas relacionadas con el procesamiento de archivos.

### TRABAJO PREVIO

- Lectura del manual de JFLEX: <http://jflex.de/docu.html>
- Diseño de las estrategias que se utilizarán para resolver el problema

La sesión de prácticas se dedicará a implementar los enfoques, y no a diseñar las estrategias necesarias para la resolución de los ejercicios. Asimismo, para evitar pérdidas de tiempo debidas al uso de la herramienta, será necesario haber utilizado JFlex fuera del aula con anterioridad, probando algún ejemplo simple de los vistos en las sesiones lectivas.

También es necesario leer la documentación relativa a la clase FileWriter, que permite escribir datos de tipo String en un fichero de forma relativamente sencilla, utilizando el método write.

### ENUNCIADO

Se trata de producir las siguientes clases:

- **ContarPalabrasYCaracteres**. Deberemos generar una clase que, **dado el nombre de un archivo, muestre por pantalla los siguientes datos**:
  - Número de **líneas**.
  - Número de **palabras**, entendiendo como tal **cualquier cadena que comience por una letra**.
  - Número de **cantidades numéricas**.
  - Número de **caracteres totales**.
  - Número de **caracteres totales sin contar los espacios ni los saltos de línea**.

- **ProcesarComentarios.** Crear una clase llamada ProcesarComentarios que permita computar y mostrar por pantalla la densidad de comentarios en un código fuente java. Definimos la densidad de comentarios como la proporción de caracteres (sin computar saltos de línea) dedicada a comentar el código, en forma de porcentaje. Además, el código deberá generar dos ficheros. El primero llamado "salida.cod" contendrá solo el código fuente sin los comentarios. El segundo se llamará "salida.com" y contendrá únicamente los comentarios.

**Notas:**

Los comentarios de bloque comienzan por `/*` y acaban en `*/`. En java no se permiten los comentarios de bloque anidados. `/* /* */ */` daría error, puesto que el comentario que abre el primer `/*` se cerraría en cuanto se encuentre el primer `*/`, quedando el segundo cierre como parte del código fuente.

Recuerda que también existen comentarios de línea, que comienzan por `/*` y se extienden hasta el final de la línea y los comentarios de javadoc, que comienzan por `/**` y se cierran igual que los comentarios de bloque

- **ReconocedorExpresiones.** Se pide construir un reconocedor de expresiones válidas en java utilizando jflex: dada una lista de expresiones java separadas por un salto de línea, el reconocedor deberá mostrar cuales de ellas son válidas.

Ejemplo:

Entrada	Salida
X+2*3+(7-4/6)	VÁLIDA
((Var*k)*9)	VÁLIDA
2x	NO VÁLIDA: VARIABLE INESPERADA
2+3)*(3+3)	NO VÁLIDA: PARÉNTESIS CERRADO INESPERADAMENTE
2+8*\$	NO VÁLIDA: SÍMBOLO \$ INESPERADO
5*x	VÁLIDA

**Nota:** Opcionalmente, puedes no contemplar el menos unario si lo consideras más sencillo, asumiendo por tanto que las expresiones `-2*x` o `4+(-2)` son ambas inválidas.