

Análisis Léxico

Preguntas más frecuentes

LEX.01. JFlex parece funcionar bien y genera Scanner.java pero al ejecutar los scripts de Ant se produce un error.

R. El fichero JFlex que te hemos proporcionado contiene una referencia a la clase sym para ilustrar el mecanismo de integración con cup. Esto implica que para que funcione el scanner debes primero especificar los símbolos terminales en la especificación de Cup y ejecutar la tarea Cup para que te genere las clases que éste necesita (parser y sym). Lo que te está diciendo es que esas clases aun no están generadas. Dado que siempre antes de ejecutar la tarea flex hay que ejecutar cup, hemos diseñado la tarea build que las ejecuta de forma secuencial en dicho orden.

LEX.02. ¿Cómo se referencian las comillas en las expresiones regulares de JFlex?

R. Puedes hacer referencia a las dobles comillas colocando una barra '\' delante de cada aparición de " que quieras hacer referencia en el patrón.

LEX.03. ¿Cómo puedo hacer que JFlex reconozca comentarios anidados correctamente balanceados?

R. La única forma de hacerlo es utilizar los estados que la herramienta JFlex permite definir. Inicialmente las reglas patrón – acción de tu especificación Flex que se revisan son aquellas asociadas al estado YYINITIAL. Lo que necesitas es crearte un nuevo estado COMMENT y asociarle las reglas de reconocimientos de comentarios anidados. En concreto los pasos son:

1. Desde YYINITIAL, al encontrarte un patrón de comienzo de comentario ejecutas la acción de JFlex yybegin (COMMENT). Esta instrucción permite saltar al autómata definido en el estado COMMENT.
2. Desde COMMENT, al encontrar un comienzo de comentario se suma 1 a una variable global que cuenta el número de comentarios abiertos
3. Desde COMMENT, al encontrar un cierre de comentario se resta una a esa variable. Si esa variable alcanza el valor 0, entonces se ejecuta yybegin (YYINITIAL) para que JFlex salte al autómata que procesa el lenguaje.

LEX.04. Al procesar comentarios mal balanceados obtengo un error cuando llega el fin de fichero

R. Utiliza la directiva %eofval para definir la lógica de tratamiento del error. Típicamente emitir un error léxico de comentarios mal balanceados.

LEX.05. ¿De qué tipo de errores debe informar el analizador léxico?

R. Esta es una decisión de diseño. En general, los errores léxicos están relacionados con el reconocimiento de caracteres no permitidos en los identificadores (ñ, ç, caracteres acentuados, etc.) y el mal balanceado de las cadenas de caracteres, por la emisión o abuso, de algún carácter comillas.

LEX.06. ¿Cómo puedo especificar un analizador léxico en JFlex que no distinga entre mayúsculas y minúsculas?

R. Usa la directiva %%ignorecase

LEX.07. La especificación de JFlex parece tener problemas con los saltos de línea. ¿Cómo puedo solucionarlo?

R. Este problema es debido a la diferencia de codificación de los saltos de nueva línea por parte de los sistemas tipo DOS y los sistemas UNIX. Los primeros identifican el carácter de nueva línea como la secuencia consecutiva de caracteres de escape RETORNO DE CARRO + SALTO DE LINEA (\r\n) mientras que en UNIX es sólo mediante el carácter SALTO DE LINEA (\n). Utiliza la directiva %%notunix si tu entorno de ejecución es DOS.

LEX.08. Obtengo errores al introducir caracteres como ñ o ç en mis casos de prueba.

R. Incluye a la especificación de JFlex la directiva %full.

LEX.09. No consigo detectar los retornos de carro.

R. Por simplicidad mete todo lo que haya que ignorar dentro de una única macro. Por ejemplo, pon: espacios = [\n\r\f\t]+. Después en la sección de reglas de producción utiliza esa macro para hacer el tratamiento oportuno.

LEX.10. ¿Debo considerar caracteres especiales dentro de las cadenas de caracteres?

R. No es necesario que consideres caracteres especiales tales como secuencias de escape o ? ! etc.

LEX.11. Al generar un token CADENA DE CARACTERES, deben incluirse las comillas.

R. Sí y no. Sí porque las comillas forman parte del patrón léxico que reconoce dicho token. A cup le debe llegar un solo token CADENA DE CARACTERES y no 3 (COMILLAS, CADENA, COMILLAS). Y no, el valor de ese token debe ser un objeto String de java que represente la cadena de caracteres sin las comillas.