



CUADERNO PRÁCTICAS TEORÍA DE LA COMPUTACIÓN

ÁLVARO FRAIDIAS MONTEAGUDO

Índice:

Como lanzar las prácticas desde una clase java externa.....	pág 2
Práctica 2.....	pág 3
Práctica 4.....	pág 4

Como lanzar las prácticas desde una clase java externa:

1º Compilamos el archivo.flex con la herramienta proporcionada por el profesor. Esto generará un archivo.java

2º Compilamos el archivo.cup. Para ello descargamos la herramienta CUP proporcionada por el profesor y la guardamos en la misma carpeta que el archivo.cup.

Una vez hecho esto ejecutamos la terminal y nos metemos en la ruta donde están estos archivos y ejecutamos la siguiente instrucción: `java -jar java-cup-11b.jar <archivo.cup>`

Si todo va bien nos generará dos archivos: El parser.java y el sym.java

3º Creamos un proyecto java y en la carpeta src pegamos los 3 archivos generados en los dos pasos previos. Una vez hecho esto, en el main ponemos:

```
Analizador analizador = new Analizador (new  
FileReader (args[0]));  
analizador.yylex();
```

Nos saldrán más de una bombilla con un círculo rojo. En ese caso vamos siguiendo los consejos q nos da NetBeans o Eclipse.

4º Añadir las librerías. Para hacer esto nos dirigimos a la parte de la izquierda de la pantalla, donde nos aparecen los proyectos y le damos clic derecho encima de las librerías y nos saldrá la opción "Add library " y ahí añadimos los .jar del JFLEX y CUP.

5º Por último, añadir el texto.txt. Para ello nos vamos a la parte de arriba de NetBeans o Eclipse al apartado "RUN" y dentro de este apartado vamos a "Set Project Configuration" y allí añadimos el texto.txt .

Práctica 2

Esta práctica consiste en leer de un fichero todos los símbolos y categorizarlos dependiendo de los tokens definidos en otro archivo con la extensión .flex.

Las expresiones regulares que definen los tokens que he declarado son los siguientes:

En mi caso yo he declarado 10 tokens que son los siguientes:

1º Un token que he declarado con el nombre "nombre" que engloba todos aquellos símbolos que cumplan la expresión regular:

[A-ZÁÉÍÓÚ] [A-Za-zÁÉÍÓÚáéíóú] *

2º Un token con el nombre " número de teléfono" que engloba todos aquellos símbolos que empiecen con un "+" seguido de cualquier combinación de números (clausura de todos los números comprendidos entre 0 y 9). Su expresión regular es:

[+34] [6] ? [7] ? [0-9] * {8}

3º Un token con el nombre "cifra" que a diferencia de " número de teléfono" este engloba los símbolos que sean una clausura desde el 1 al 9.

Su expresión regular es: **[0-9] ***

4º Un token que he declarado con el nombre "correo" que abarca todos aquellos símbolos que cumplan con la expresión regular:

[_a-z0-9-]+ (. [_a-z0-9-]+) * @ [a-z0-9-]+ (. [a-z0-9-]+) * (. [a-z] {2,4})

5º Un token con el nombre "contraseña" que engloba todas las palabras alfanuméricas que están dentro del símbolo "&" incluido este símbolo.

Su expresión regular es: **[& (a-zA-Z0-9) &] ***

6º Un token " [" que se utiliza para saber cuándo empieza un campo dentro del fichero.

7º Un token "]" que se usa para saber cuándo finaliza un campo dentro del fichero.

8º Un token ";" que sirve para separar distintos lexemas dentro del fichero.

9º Un token ", " que también sirve para separar distintos lexemas dentro del fichero.

10º Los tokens "\n" "\t" "\s" "\r" que se usan para realizar saltos de línea dentro del fichero.

A parte de estos 10 tokens he declarado 3 palabras reservadas que son: "Cargos", "Departamentos" y "Trabajadores".

Práctica 4

En esta práctica se utilizará también la herramienta cup para hacer un análisis sintáctico aparte del análisis léxico.

Primero debemos encontrar todos los tokens con el analizador léxico Jflex y pasárselos al cup como terminales para que pueda utilizarlos para crear las reglas sintácticas.

La herramienta cup se encarga de coger los tokens proporcionados por Jflex y crear una gramática que defina un lenguaje. Para ello hay que declarar la gramática en el archivo .cup y para eso usaremos un metalenguaje empleado para definir lenguajes libres de contexto, es decir, una manera formal de describir lenguajes formales. Este metalenguaje es EBNF.

Una vez definido esta gramática y recogidos los tokens mediante Jflex, lanzaremos el programa mediante una clase externa.

Los tokens que nos proporciona Jflex serán los símbolos terminales y los que creamos nosotros como pasos intermedios en la definición de la gramática serán los símbolos no terminales. En mi caso he creado 10 no terminales que son:

***S:** Símbolo inicial.

***MIS_CARGOS, MIS_DEPARTAMENTOS, MIS TRABAJADORES:** Son los 3 grandes apartados que tiene que tener el texto que analicemos para que este correcto.

***CONJUNTO_CARGOS, CONJUNTO_DEPAR, CONJUNTO_TRABAJ, NOMBRE:** En estos no terminales es donde se lleva a cabo la recursividad.

***INFORMACION_DEPAR, PERSONA:** Los he utilizado para poner los terminales sin que quede tan largo la gramática.