

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERIA
ESCUELA DE CIENCIAS Y SISTEMAS
LENGUAJES Y COMPILADORES 1

Manual de Iflex y Cup

Autor: Iris Benita Us Rojas
Septiembre de 2010

INTRODUCCION

Jflex y CUP es una herramienta ampliamente utilizada para la programación de sistemas informáticos ya que permite generar compiladores a través de gramáticas independientes del contexto mediante analizadores ascendentes, aunado al gran potencial que brinda el lenguaje de programación Java.

Este manual tiene como objetivo explicar el proceso de instalación y utilización de esta herramienta, mas no entrar en detalles de cómo se programan; gramáticas y analizadores sintácticos con ella, es decir describe como utilizarla para generar compiladores a través de un ejemplo sencillo. Es importante tomar en cuenta que se describe una de tantas formas de cómo se puede llevar a cabo esta tarea, ya que existen muchas otras formas.

Lea detenidamente este documento y siga cada uno de los pasos que se describen ya que de ello depende que pueda o no generar una salida correcta.

INSTALACION

1. Instalación del JDK

Lo primero que necesitamos es instalar el JDK, independientemente si estamos en Windows o en Linux, ya que el código de JFlex y Cup es compilado desde consola, a través de comandos que nos proporciona java.

Descargamos el jdk y a continuación lo instalamos para poder usar el código Java.

2. DESCARGA DE LIBRERIAS

Descargamos las fuentes de JFlex en la pagina: <http://jflex.de/download.html>

Descargamos las fuentes de Cup en la pagina: <http://www2.cs.tum.edu/projects/cup/>

3. CREACIÓN DE CARPETA CONTENEDORA DE FUENTES

Creamos una carpeta nueva con el nombre "Librerias, Fuentes, etc" dentro de la unidad de disco que estamos utilizando.

Descomprimos el contenido de la carpeta descargada: jFlex-1.4.3 y java_cup_v10k dentro de la carpeta creada.

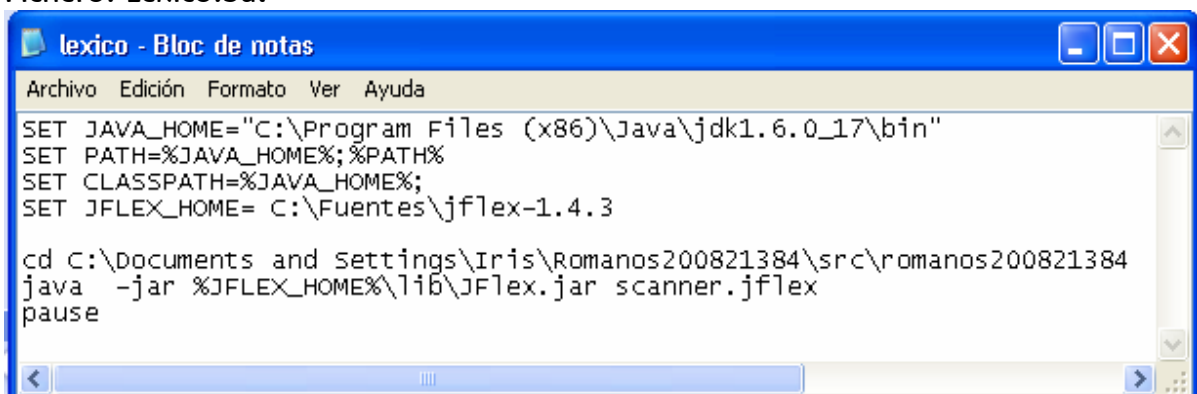
4. Creación del Archivo de Configuración Lexico.bat

Este tipo de archivos puede ser compilado desde consola o tambien desde un archivo.bat; en esta ocasión utilizaremos el archivo.bat ya que es mas facil de manejar.

Escribimos en el archivo lo que se presenta en la figura, cambiando la direccion de JAVA_HOME, por la direccion actual JDK.

Agregamos la direccion de nuestro actual proyecto.

Fichero: Lexico.bat

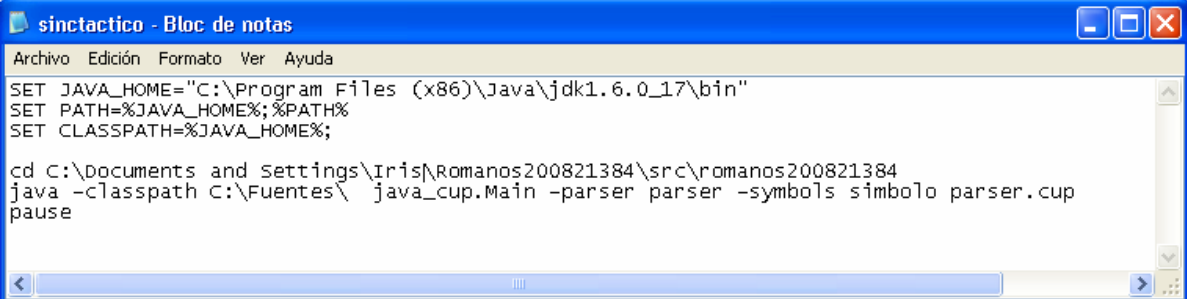


```
lexico - Bloc de notas
Archivo  Edición  Formato  Ver  Ayuda
SET JAVA_HOME="C:\Program Files (x86)\Java\jdk1.6.0_17\bin"
SET PATH=%JAVA_HOME%;%PATH%
SET CLASSPATH=%JAVA_HOME%;
SET JFLEX_HOME= C:\Fuentes\jflex-1.4.3

cd C:\Documents and Settings\Iris\Romanos200821384\src\romanos200821384
java -jar %JFLEX_HOME%\lib\JFlex.jar scanner.jflex
pause
```

Hacemos el mismo procedimiento para el sintáctico

Fichero: sintactico.bat

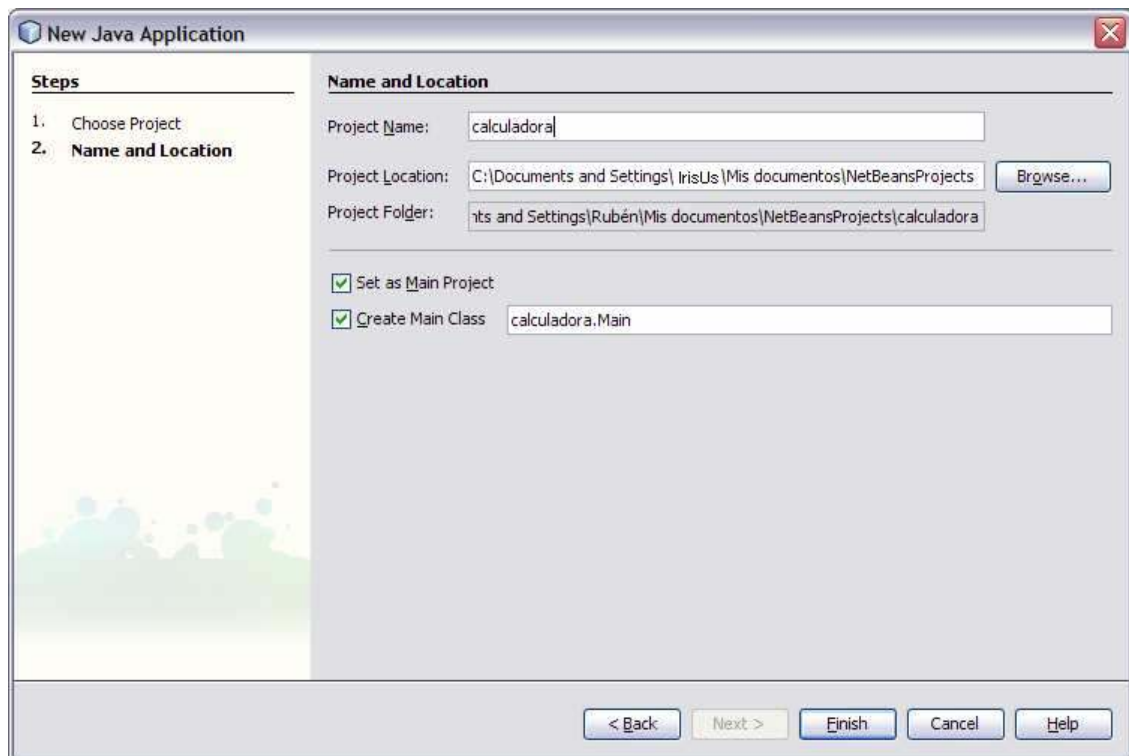


```
SET JAVA_HOME="C:\Program Files (x86)\Java\jdk1.6.0_17\bin"
SET PATH=%JAVA_HOME%;%PATH%
SET CLASSPATH=%JAVA_HOME%;

cd C:\Documents and Settings\Iris\Romanos200821384\src\romanos200821384
java -classpath C:\Fuentes\ java_cup.Main -parser parser -symbols simbolo parser.cup
pause
```

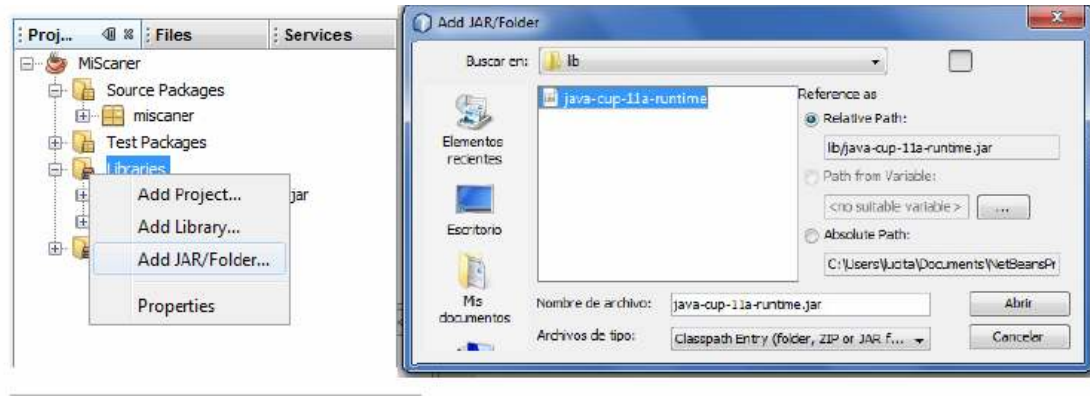
5. CREACIÓN DE UN PROYECTO EN NETBEANS

Damos clic en File, luego en New Project y Java Application
Por ultimo Clic en Finish



Dentro de la carpeta del proyecto, agregamos una carpeta llamada "lib". En ella copiamos la java-cup-11a-runtime.jar

Damos clic secundario en Librerías, y le damos Agregar nuevo JAR/FOLDER.



6. CREACIÓN DE LEXICO.JFLEX

```
package calculadora;  
import java_cup.runtime.Symbol;
```

```
%%
```

```
%public
```

```
%class Lexico
```

```
%cup
```

```
%%
```

```
"+" { return new Symbol(Simbolos.PLUS); }
```

```
"-" { return new Symbol(Simbolos.MINUS); }
```

```
"*" { return new Symbol(Simbolos.TIMES); }
```

```
"/" { return new Symbol(Simbolos.DIV); }
```

```
"(" { return new Symbol(Simbolos.LPAREN); }
```

```
")" { return new Symbol(Simbolos.RPAREN); }
```

```
[0-9]+ { return new Symbol(Simbolos.NUMBER, new Integer(yytext())); }
```

```
[ \t\r\n] { /* ignore white space. */ }
```

```
. { System.err.println("Illegal character: "+yytext()); }
```

7. CREACIÓN DEL ARCHIVO CAL_CUP.CUP

```
package calculadora;

import java_cup.runtime.*;

parser code
{::}

terminal PLUS, MINUS, TIMES, DIV, LPAREN, RPAREN;
terminal Integer NUMBER;
non terminal root;
non terminal Integer expr;
precedence left PLUS, MINUS;
precedence left TIMES, DIV;
start with root;

root ::= expr:e { System.out.println(" = "+e+");" };
;
expr ::= NUMBER:n
{ RESULT=new Integer(n.intValue()); ;}

| expr:l PLUS expr:r
{ RESULT=new Integer(l.intValue() + r.intValue()); ;}

| expr:l MINUS expr:r
{ RESULT=new Integer(l.intValue() - r.intValue()); ;}

| expr:l TIMES expr:r
{ RESULT=new Integer(l.intValue() * r.intValue()); ;}

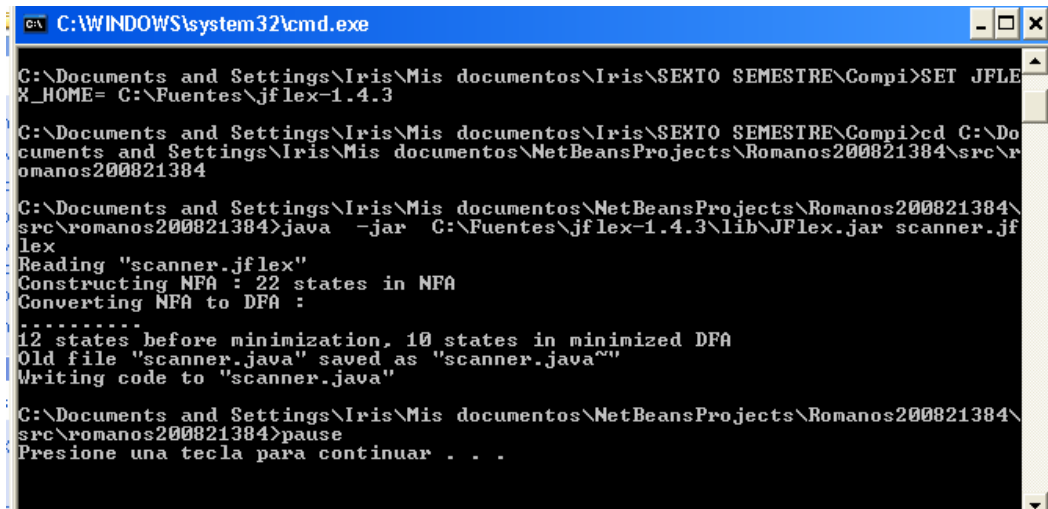
| expr:l DIV expr:r
{ RESULT=new Integer(l.intValue() / r.intValue()); ;}

| LPAREN expr:e RPAREN
{ RESULT=e; ;};
```

8. COMPILACIÓN

Le damos doble clic a los ficheros léxico.bat y sintáctico.bat y luego compilamos el proyecto de netbeans.

Estos archivos terminan su compilación con esta imagen.



```
C:\WINDOWS\system32\cmd.exe
C:\Documents and Settings\Iris\Mis documentos\Iris\SEXTO SEMESTRE\Compi>SET JFLEX_HOME= C:\Fuentes\jflex-1.4.3
C:\Documents and Settings\Iris\Mis documentos\Iris\SEXTO SEMESTRE\Compi>cd C:\Documents and Settings\Iris\Mis documentos\NetBeansProjects\Romanos200821384\src\Romanos200821384
C:\Documents and Settings\Iris\Mis documentos\NetBeansProjects\Romanos200821384\src\Romanos200821384>java -jar C:\Fuentes\jflex-1.4.3\lib\JFlex.jar scanner.jflex
Reading "scanner.jflex"
Constructing NFA : 22 states in NFA
Converting NFA to DFA :
.....
12 states before minimization, 10 states in minimized DFA
Old file "scanner.java" saved as "scanner.java~"
Writing code to "scanner.java"
C:\Documents and Settings\Iris\Mis documentos\NetBeansProjects\Romanos200821384\src\Romanos200821384>pause
Presione una tecla para continuar . . .
```

```
CA C:\WINDOWS\system32\cmd.exe
C:\Documents and Settings\Iris\Mis documentos\Iris\SEXTO SEMESTRE\Compi>SET JAVA_HOME="C:\Program Files (x86)\Java\jdk1.6.0_17\bin"
C:\Documents and Settings\Iris\Mis documentos\Iris\SEXTO SEMESTRE\Compi>SET PATH="C:\Program Files (x86)\Java\jdk1.6.0_17\bin";C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\System32\Wbem
C:\Documents and Settings\Iris\Mis documentos\Iris\SEXTO SEMESTRE\Compi>SET CLASSPATH="C:\Program Files (x86)\Java\jdk1.6.0_17\bin";
C:\Documents and Settings\Iris\Mis documentos\Iris\SEXTO SEMESTRE\Compi>cd C:\Documents and Settings\Iris\Mis documentos\NetBeansProjects\Romanos200821384\src\Romanos200821384
C:\Documents and Settings\Iris\Mis documentos\NetBeansProjects\Romanos200821384\src\Romanos200821384>java -classpath C:\Fuentes\ java_cup.Main -parser parser -symbols simbolo parser.cup
Opening files...
Parsing specification from standard input...
Checking specification...
Building parse tables...
  Computing non-terminal nullability...
  Computing first sets...
  Building state machine...
  Filling in tables...
  Checking for non-reduced productions...
Writing parser...
Closing files...
----- CUP v0.10k Parser Generation Summary -----
 0 errors and 0 warnings
 9 terminals, 15 non-terminals, and 61 productions declared,
 producing 66 unique parse states.
 0 terminals declared but not used.
 0 non-terminals declared but not used.
 0 productions never reduced.
 0 conflicts detected (0 expected).
 Code written to "parser.java", and "simbolo.java".
----- (v0.10k)
C:\Documents and Settings\Iris\Mis documentos\NetBeansProjects\Romanos200821384\src\Romanos200821384>pause
Presione una tecla para continuar . . . _
```

Especificaciones de Jflex

Un archivo de especificación para Jflex esta organizado en tres secciones, separadas por “%%” y tiene el siguiente formato:

Código de usuario

%%

Directivas Jflex

%%

Expresiones regulares

Código de usuario:

En esta sección se crean las clases necesarias para nuestro analizador léxico, las cuales serán copiadas directamente al inicio del archivo de salida. Se importan los paquetes necesarios.

Directiva

En esta sección se incluyen algunas reglas propias de Jflex también se podrán una definición de expresiones regulares para facilitar su uso después.

Definición Regular:

<nombre macro>= <expresión regular valida>

%char; %line	Activar el conteo de caracteres y de líneas.
%class	permite cambiar el nombre de la clase del analizador lexico, por default copia el definido en el archivo.
%cupsym	Esta cambiara la clase de símbolos que genera por defecto Jflex sym por otro nombre.
%public	permite que la clase del analizador léxico generada por Jflex sea una clase pública.
%cup	Activa el uso de cup.
%init{ Código %init}	Permite agregar código al constructor de la clase generada por jflex, aquí se pueden inicializar las variables.
%unicode	Usamos el formato unicode para trabajar con los ficheros
%standalone	Indicamos a JFlex que queremos que genere su propio método <i>main</i> .
%ignorecase	De las expresiones léxicas definidas diferencia entre mayúsculas y minúsculas.
%type String	Indicamos el tipo que queremos que nos devuelva para los tokens. Dejará el token actual con la función <i>yytext()</i>

Expresiones Regulares:

En esta sección se definen las expresiones regulares que representan los tokens de nuestro lenguaje, y que debemos hacer una vez detectados dichos tokens. Las reglas tienen 3 partes:

[<estado(s)>] <expresión> {<acción>}

Definición de Expresiones:

\t	Representa un tabulador	\r	Representa el "regreso al inicio" o sea el lugar en que la línea vuelve a iniciar.
\n	Representa la nueva línea. Como también el carácter que le dará inicio.	\f	Representa un salto de página
\d	Representa un dígito del 0 al 9	\v	Representa un tabulador vertical
\s	Representa un espacio en blanco.	\w	Representa cualquier carácter alfanumérico.