

Agenda

BNF y EBNF

Brevemente, lo necesario para especificar el lenguaje sobre el que vamos a trabajar.

JLEX y JCUP

Breve descripción de las herramientas para generar el parser o analizador sintáctico.

2do obligatorio

Vamos a ver la letra, el entorno de trabajo, los textos que se imprimen, y un ejemplo de entrada y salida.

BNF

(Backus–Naur Form)

- Notación formal para definir la sintaxis de un lenguaje
- Usada para especificar la mayoría de los lenguajes de programación

- Metasímbolos:

`::=` se define como

`|` or

`{ }` repetición

`[]` opcional

- Los terminales entre comillas y negrita, por ejemplo: **'if'**, **'5'**

Ejemplo BNF

Gramática para representar números con decimales

```
numero ::= entPos  
        | entPos '.' entPos
```

```
entPos ::= digito  
        | digito entPos
```

```
digito ::= '0' | '1' | '2' | '3' | '4'  
        | '5' | '6' | '7' | '8' | '9'
```

E-BNF

(Extended BNF)

Se agrega:

- ? Opcional
- * 0 a n veces
- + 1 a n veces

Ejemplo anterior con E-BNF

numero ::= digito + ('.' digito +) ?

digito ::= '0' | '1' | '2' | '3' | '4'
 | '5' | '6' | '7' | '8' | '9'

JLEX y JCUP

JLEX

Generador de analizadores lexicográficos para Java.

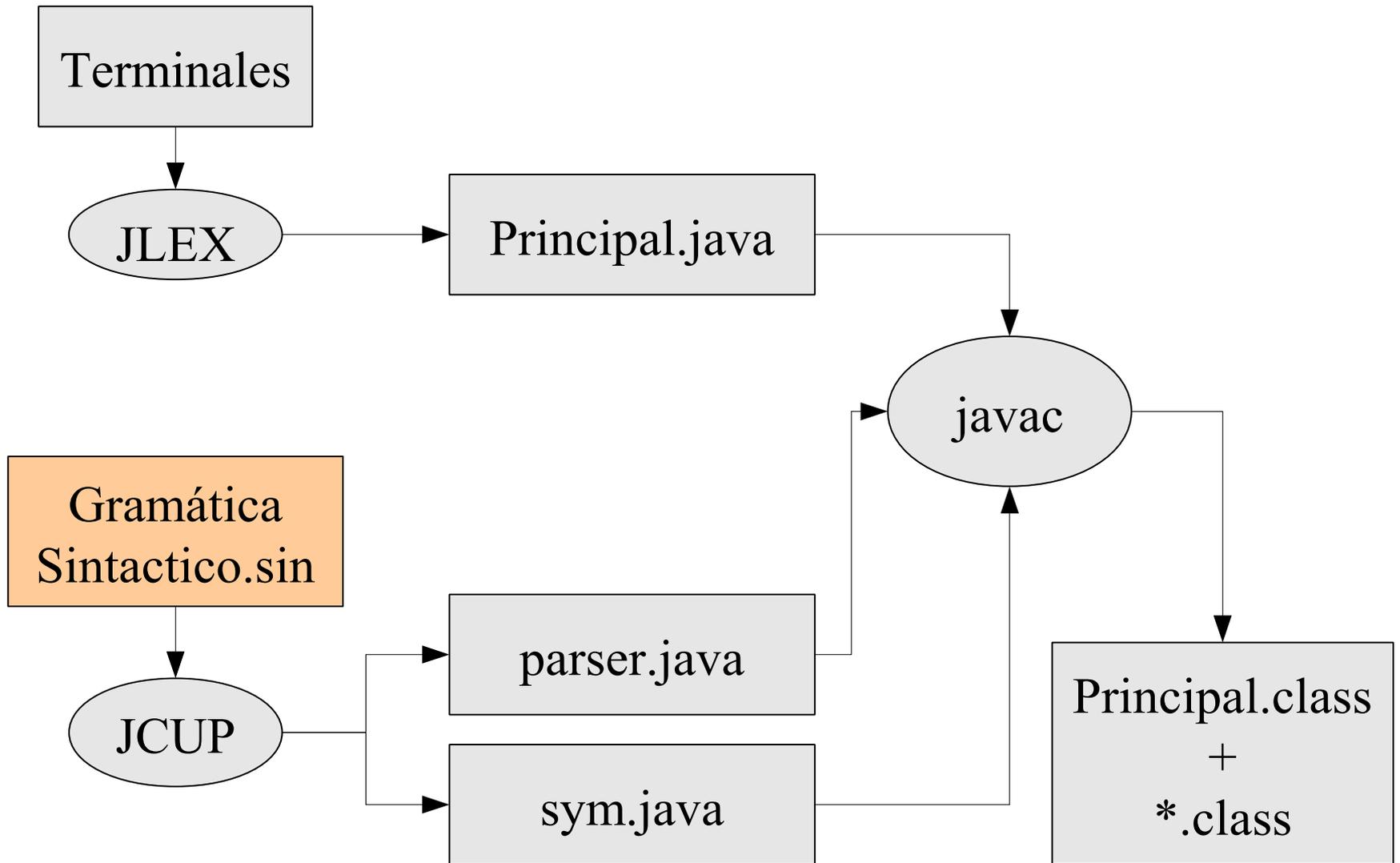
Usa expresiones regulares para reconocer los terminales del lenguaje, por ejemplo: palabras claves, identificadores, números, strings, etc. Se entrega hecho en el archivo:

Principal.java

JCUP

Generador de analizadores sintácticos (parsers) en Java.

[Manual de JCUP \(versión 0.10j\)](#)



Archivo
de entrada

Tira (o código fuente) del
lenguaje definido en la letra
del obligatorio.

Principal.class
+
*.class

java

Archivo
de salida

Si reconoce, devuelve 0 error,
e imprime las estructuras
reconocidas.

Sino, imprime: “Error
sintactico” donde se trancó.

Scripts útiles

seteos.bat

configura el PATH y el CLASSPATH, hay que personalizarlo para cada PC, y ejecutarlo antes de los otros scripts

generar-parser.bat

genera los archivos sym.java y parser.java desde Sintactico.sin

compilar.bat

compila los .java anteriores y el Principal.java que se proporciona

ejecutar.bat

ejecuta todos los test

comparar.bat

compara contra la salida oficial todos los test

E-BNF de ejemplo anterior:

```
numero := digito + ( '.' digito + ) ?
```

```
digito := '0' | '1' | '2' | '3' | '4' | '5' | '6' | '7' | '8' | '9'
```

```
import java_cup.runtime.*;
```

```
import java.io.*;
```

```
terminal      D0, D1, D2, D3, D4;D5, D6, D7, D8, D9, PUNTO;
```

```
non terminal numero, enteroPos, digito;
```

```
/* grmática */
```

```
numero      ::= enteroPos | enteroPos PUNTO enteroPos;
```

```
enteroPos   ::= digito | digito enteroPos;
```

```
digito      ::= D0 | D1 | D2 | D3 | D4  
              | D5 | D6 | D7 | D8 | D9;
```

**Gramática para JCUP
correspondiente**

Respecto a la salida impresa, tener presente que:

- JCUP es un LALR (Look-Ahead Left to Right) parser, que hace reconocimiento bottom-up.
- Es decir reconoce de izquierda a derecha, primero las hojas y luego el nodo padre.

Obligatorio

El trabajo consiste en escribir la gramática para JCUP que genere el analizador sintáctico (parser). Y este reconozca programas (o código fuentes) del lenguaje especificado en la E-BNF presentada en la letra del obligatorio.

Además debe imprimir determinados textos a medida que se van reconociendo ciertas estructuras.

Obligatorio

El lenguaje presentado en la letra es parecido a SQL, pero modificado y más sencillo.

Cada código SQL (o archivo de entrada) es una secuencia de declaraciones SQL. Todas terminan con ';'.
;

Las cuales pueden ser de esquema (schema_stm) o de datos (data_stm), y así sucesivamente como esta en la E-BNF presentada en la letra del obligatorio.

Los mensajes que se imprimen (1)

"OK" luego de reconocer un programa sql completo

"DECLARACION DE ESQUEMA\n---" luego de reconocer una estructura de la forma: *schema_stm*

"DECLARACION DE DATOS\n---" luego de reconocer una estructura de la forma: *data_stm*

Nota: "\n---" representa un salto de línea y tres guiones.

Los mensajes que se imprimen (2)

"CREACION DE TABLA" o

"CREACION DE TABLA CON SELECT"

luego de reconocer una estructura de la forma: *create_table* sin o con select, respectivamente

"CREACION DE INDICE"

luego de reconocer una estructura de la forma: *create_index*

"LISTA DE IDENTIFICADORES"

luego de reconocer una estructura de la forma: 'id' (',' 'id') *

Los mensajes que se imprimen (3)

"ELIMINACION DE TABLA"

"ELIMINACION DE INDICE"

luego de reconocer una estructura de la forma: *drop_table* o *drop_index*, respectivamente

"ACTUALIZACION SIMPLE"

"ACTUALIZACION SIMPLE CONDICIONAL"

luego de reconocer una estructura de la forma: *update_simple* sin o con *where*, respectivamente

Los mensajes que se imprimen (4)

"ACTUALIZACION CON SELECT"

luego de reconocer una estructura de la forma: *update_subquery*

"LISTA DE (ID = EXPRESSION)"

luego de reconocer una estructura de la forma:

*'id' '=' expression (',' 'id' '=' expression) **

Los mensajes que se imprimen (5)

"ELIMINACION DE FILAS"

"ELIMINACION DE FILAS CONDICIONAL"

luego de reconocer una estructura de la forma: *delete_stm* sin o con *where* respectivamente

"INSERCIÓN DE FILA"

"INSERCIÓN DE FILAS CON SELECT"

luego de reconocer una estructura de la forma: *insert_stm* para una fila, o para varias usando *select* respectivamente

Los mensajes que se imprimen (6)

"SELECCION DE FILAS"

"SELECCION DE FILAS CONDICIONAL"

luego de reconocer una estructura de la forma: *select_stm* sin o con *where* respectivamente

"OR"

"AND"

luego de reconocer una estructura con OR o AND y sus respectivos pares de operandos, respectivamente

Los mensajes que se imprimen (7)

"CONSTANTE CADENA DE CARACTERES"

"CONSTANTE NUMERO"

luego de reconocer una constante de la forma 'string' o 'number'
respectivamente

"ES NULL", "NO ES NULL", "EXISTE", "ENTRE", "NO ENTRE"

luego de reconocer una estructura de la forma: *logic_factor*
correspondiente.

Ejemplo de entrada

```
SELECT c1, c2
FROM t1
WHERE (c3 = 100) and
      (c4 = 'A') and
      (c5 >= 50) ;

DROP TABLE t1 ;
```

Ejemplo de salida

```
LISTA DE IDENTIFICADORES
CONSTANTE NUMERO
CONSTANTE CADENA DE CARACTERES
AND
CONSTANTE NUMERO
AND
SELECCION DE FILAS CONDICIONAL
DECLARACION DE DATOS
---
ELIMINACION DE TABLA
DECLARACION DE ESQUEMA
---
OK
```