Lenguajes y Autómatas I

TAREA 27

1. En esta tarea se realizará un sencillo ejercicio para conocer el uso de Javacc, para ello solamente deberá realizar los pasos siguientes:
2. En la sección de descargas localice el archivo ***javacc-5.0.zip***, también lo puede descargar directamente del sitio oficial: ***https://javacc.java.net/***.
3. Descomprima el archivo en la carpeta de documentos.
4. En la carpeta que descomprimió busque la carpeta ***examples*** y ahí cree una nueva carpeta a la que llamará ***prueba***.
5. Usando el bloc de notas debe crear un documento de texto nuevo, ahí escriba lo siguiente:

```
PARSER_BEGIN(ejemplo)
// Donde ejemplo es el nombre de la clase:
class ejemplo {
    public static void main(String[] args) throws
ParseException {
// Esta instrucción maneja las excepciones del compilador.
        try {
            ejemplo analizador= new ejemplo(System.in);
            analizador.Programa();
            System.out.println("Se ha compilado con éxito");
        }
        catch(ParseException e) {
            System.out.println("Ocurrió un error: ");
            System.out.println(e.getMessage());
        }
    }
}

PARSER_END(ejemplo)

// A continuación se definen los TOKENS del léxico:
// Junto al nombre del token va la acción que se va a ejecutar
// image permite mostrar el TOKEN que se está compilando.
// Es conveniente crear diferentes bloques de tokens.
// Programa es el método principal de la sintaxis a definir.
TOKEN: {
    <MAIN:"public static void Main()" >
{System.out.println("MAIN -> " + image);}
    | <PROGRAMA: "Programa" > {System.out.println("PROGRAMA ->
" + image);}
    | <IF: "ien" > {System.out.println("IF -> " + image);}
}

TOKEN: {
    <PAR_IZQ: "(">{System.out.println("PARENTESIS IZQUIERDO ->
" + image);}
    | <PAR_DER: ")">{System.out.println("PARENTESIS DERECHO ->
" + image);}
```

```
    | <LLAVE_IZQ: "{">{System.out.println("LLAVE IZQUIERDA -> "
+ image);}
    | <LLAVE_DER: "}">{System.out.println("LLAVE DERECHA -> " +
image);}
    | <PUNTO_COMA: ";">{System.out.println("PUNTO Y COMA -> " +
image);}
}

TOKEN: {
    <ASIG: "=">{System.out.println("ASIGNACION -> " + image);}
    | <MENOR: "<">{System.out.println("MENOR QUE -> " +
image);}
    | <MAYOR: ">">{System.out.println("MAYOR QUE -> " +
image);}
}

TOKEN: {
    <INT: "inum">{System.out.println("ENTERO -> " + image);}
    | <NUM: (["0"-"9"])+ >{System.out.println("NUMERO -> " +
image);}
    | <IDENT: ["A"-"Z","a"-"z"]([ "A"-"Z","a"-"z","0"-
"9","_"])* > {System.out.println("IDENTIFICADOR -> " + image);}
}
// Se emplean las expresiones regulares para estos tokens.
// Se ignoran los espacios en blanco, tabuladores y saltos.

SKIP: {
    " " | "\r\n" | "\t"
}
// Sigue la sintaxis, para ello defina el método Programa.

void Programa(): {} {
    <PROGRAMA><IDENT><LLAVE_IZQ>Principal()<LLAVE_DER><EOF>
}
// El primer juego de llaves no se va utiliza en este ejemplo.
// Continúe con los demás métodos:
void Principal(): {} {
    <MAIN>
    <LLAVE_IZQ>Sentencias()<LLAVE_DER>
}

void Sentencias(): {} {
    ( SentenciaIf() | Declaracion() | Asignacion() )*
}

void SentenciaIf(): {} {
    <IF><PAR_IZQ>Comparaciones()<PAR_DER><LLAVE_IZQ><LLAVE_DER>
}

void Declaracion(): {} {
    <INT><IDENT><PUNTO_COMA>
}
```

```
void Asignacion(): {} {
    <IDENT><ASIG>Valor()<PUNTO_COMA>
}
void Comparaciones(): {} {
    Valor()Operadores()Valor()
}

void Valor(): {} {
<IDENT> | <NUM>
}

void Operadores(): {} {
<MENOR> | <MAYOR>
}
```

6. Dentro de la carpeta ***prueba*** guarde el documento recién creado, con el nombre de ***ejemplo.jj*** (cerciórese que se guarde con la extensión jj).
7. Ahora deberá compilar el compilador, para ello abra la ventana de comandos y ubíquese en la dirección correspondiente a la carpeta de ***prueba***, y entonces teclee la siguiente instrucción:
   `..\..\bin\javacc ejemplo.jj`
8. Si no presenta errores en el código, (aparecerán dos Warnings) dentro de la carpeta de prueba se generarán varios archivos ***.java***, que forman parte del analizador sintáctico (parser):
   `ejemplo.java`
   `ejemploConstants.java`
   `ejemploTokenManager.java`
   `ParserException.java`
   `SimpleCharStream.java`
   `Token.java`
   `TokenMgrError.java`
9. Verifique que la ruta: ***C:\Program Files\Java\jdk1.8.0_152\bin*** (puede variar la versión de jdk disponible en su sistema) está disponible mediante la declaración ***PATH***.
10. Ahora compile todos esos archivos mediante la siguiente instrucción:
    `javac *.java`
11. Si hubo errores en la compilación revise su código fuente.
12. A continuación, escriba un programa de prueba mediante el bloc de notas:

```
Programa ejemplo {
    public static void Main() {
        inum num2;
        ien (num2 < 4) {
        }
        num2 = 4;
    }
}
```

13. Y guárdelo con el nombre de ***prueba.txt*** en la misma carpeta donde está trabajando.
14. Para probar el compilador ejecute la siguiente sentencia:
    `java ejemplo < prueba.txt`

15. Genere una imagen que evidencie el resultado de este último paso.
16. Fin de la tarea.