



Universidad de Alcalá

Escuela Politécnica Superior

Universidad de Alcalá

Procesadores del lenguaje

PECL2

Martes 17:00 – 19:00

Grado en Ingeniería Informática – Curso 2019/2020

Marcos Barranquero Fernández – 51129104N

INTRODUCCIÓN

Se ha realizado dos apartados de la práctica:

1. Se ha creado una gramática capaz de identificar el archivo especificado y generar el árbol gramatical correspondiente.
2. A partir del programa Java generado por ANTLR, se ha implementado un nuevo Listener adecuado para la gramática, que permite visualizar el árbol por consola y escribirlo en un archivo de texto.

Se da por hecho que se tiene ANTLR configurado como se propone en la web de cara a ejecutar los comandos en la ejecución secuencial.

EJECUCIÓN SECUENCIAL – PARTE GRAMÁTICAS

1. Sobre el directorio con las gramáticas, se llama a ANTLR para que genere los archivos necesarios:

```
antlr FiboLexer.g4
antlr FiboParser.g4
```

2. Una vez generados los ficheros Java, los compilamos a bytecode con:

```
Javac *.java
```

3. Llamamos al analizador con el fichero con el código:

```
grun Fibo programa -gui -tokens progfibo.prog
```

4. Podemos visualizar los tokens reconocidos en la consola y el árbol en la interfaz de usuario.

EXPLICACIÓN MODULAR – PARTE GRAMÁTICAS

La gramática se divide en dos componentes:

FIBOLEXER

El lexer contiene todas las asociaciones de Tokens a caracteres de la cadena de entrada. También especifica qué caracteres debe ignorar, como los comentarios o saltos de línea.

FIBOPARSER

Contiene las estructuras de tokens reconocibles, tales como llamadas o definiciones de funciones, estructuras de control, operaciones matemáticas, returns, etc.

EJECUCIÓN SECUENCIAL – PARTE JAVA

1. Compilamos los archivos:

```
Javac *.java
```

2. Llamamos al main pasándole como argumento el archivo del código:

```
Java Main "./progfibo.prog"
```

3. Podemos observar la estructura del árbol tanto por consola como viendo el .txt generado.

EXPLICACIÓN MODULAR – PARTE JAVA

Se explican los ficheros modificados, los otros son generados por ANTLR y no se han cambiado.

LISTENER

La clase listener tiene un ArrayList árbol, el cual almacena las cadenas de tokens. Para añadir cadenas, se tienen una gran cantidad de métodos del Listener que permiten añadir la cadena del token asociado de cada método al árbol.

MAIN

El archivo ejecutable instancia todas las clases necesarias para leer el documento de código y convertirlo en el árbol de gramática. Después, se llama al Walker y se recorre el árbol con el listener creado. Después, se recoge el árbol creado por el listener, y se imprime por pantalla y se escribe en el archivo de texto.