Universidad Simón Bolívar

Traductores e Interpretadores

Informe del Analizador Lexicográfico para el lenguaje BasicTran

Autores: Irina Marcano 13-10805 :: Fernando González 08-10464

**Estructura de la entrega:**

Se crea una clase llamada **ElementoTexto** la cual presenta como atributos línea, columna y contenido. A partir de la premisa de que los elementos a representar del lenguaje son “objetos de tipo texto” se crean dos clases centrales que son **Token** y **Error**, ambas clases heredan sus atributos de objeto de texto. Dichos atributos contendrán la información pertinente para llevar a cabo la impresión de la información deseada.

Seguido a esto pasamos a definir el diccionario de las RegExp para los tokens definidos en el enunciado del lenguaje. Cada elemento del diccionario está definido de la forma:

**... , <Token\_name> => <RegExp> , …**

Donde **Token\_name** puede tomar los valores de:

**Punto || Num || Caracter || Id || Coma || DosPuntos || ParAbre || ParCierra || CorcheteAbre || CorcheteCierre || LlaveAbre || LlaveCierra || Resta || Hacer || Asignacion || Suma || Desigualdad || PuntoYComa ||Mult || Div || Mod || Conjuncion || Disyuncion || Negacion || Menor || MenorIgual ||Mayor || MayorIgual || Igual || SiguienteCar || AnteriorCar || ValorAscii || Concatenacion || Shift**

y **RegExp** sus respectivas expresiones regulares.

Escribimos las palabras reservadas para nuestro lenguaje:

**( with , true , false , var , begin , end , int , while , if , else , bool , char , array , read , of )**

Y de igual manera las introducimos en el diccionario antes definido para la correcta identificación de nuestros Tokens.

Y para cada uno de nuestros tipos de tokens definidos en el diccionario se crea una clase, dichas clases pertenecen a la clase Token y se inicializan para tener el contexto del token que serían **Línea**, **Columna** de su posición y su posible **contenido** y procedemos a renombrar cada clase anteponiendo la abreviatura Tk a las palabras de nuestro diccionario.

Luego de ello, se agregan a las clases Token en general los respectivos formatos de impresión haciendo una modificación de las mismas. Ya que la mayoría de los tokens solo son mencionado, pero algunos como **TkNum, TkId, TkCaracter** tienen diferentes formatos, se modifican sus clases para darles un método de impresión adaptado a cada caso.

Luego declaramos la clase **Lexer**, la cual se encargará de llevar la data de la salida mientras es recorrida la entrada de texto, se inicializa con listas vacías de tokens y errores para ser almacenados allí, además de los índices de filas y columnas quienes posteriormente darán la situación de los dos elementos de texto definidos (tokens o errores)

Es aquí donde declaramos el método **buscar** el cual se encarga de revisar cada una de sus entradas y clasificarlas en Tokens o Errores y agregarlos a su respectiva lista del **Lexer** con su situación indicada por la fila y la columna.

Luego se define el método **mostrarResultado** que se encarga de llamar a la impresión de la salida según sea el caso requerido, (errores o tokens)

Finalmente tenemos el método de lectura el cual…