

MEMORIA CASO PRACTICO

PROCESADORES DEL LENGUAJE

Proyecto realizado por José María Fernández Gómez

Se pedía :

Desarrolle un traductor predictivo, recursivo y descendente para un subconjunto de instrucciones de un lenguaje similar a C. El traductor debe generar código intermedio para una máquina abstracta de pila.

Organización del código:

Se ha dividido el proyecto en 4 paquetes distintos.

Launcher: Contiene la clase ejecutable que creará el analizador

Analizador: Contiene la clase Analizador

TipoDatos: Contiene varias clases para dar formato a los tipos de datos y una que se encarga de mapear e introducir en un mapa hash los lexemas proporcionados por archivo externo.

Léxico: Contiene la clase que implementa un analizador léxico

Clases mas significativas:

TestAnalizador: Se le proporciona por consola el programa a testear, crea en analizador, llama a su método programa que se encarga de inicializar el analizador e ir llamando a las subrutinas que lo componen. Por último saca por consola el contenido de la tabla de símbolos generada.

Analizador: Analizador completo que se encarga de reconocer la secuencia de comienzo de un programa de tipo void main{.....}. Una vez hechas estas comprobaciones va analizando en primer lugar el tipo de variables declaradas y luego las clasifica en función de esta. Después se dispone a analizar la parte operativa o de instrucciones, Esto se hace por medio de las reglas gramaticales proporcionadas en el enunciado.

Léxico: Analizador léxico, el analizador principal creará un objeto de este dentro de sí para analizar sintácticamente el código proporcionado. Este objeto recibe el programa a analizar en un archivo txt y una forma de codificación standard de caracteres. A grandes rasgos se encarga de extraer los componentes léxicos del programa y clasificarlos en función de sus etiquetas, que son strings que se le asignan a cada componente para poder identificarlos y compararlos.

ComponenteLexico: Clase que se encarga de darle formato a todos los componentes léxicos, indicando que a parte de su valor deben de tener una etiqueta para poder ser identificados.

hashCompLexicos: Clase que se encarga de extraer todas las cadenas de caracteres o símbolos del archivo proporcionado de programa y guardarlos en un Hash table para luego poder identificarlos con sus etiquetas .

Resto de clases en TipoDatos : Son clases que sirven para dar formato a otros tipos de datos en función de su sintaxis, se usan para tener todo más organizado. Todos heredan de componente léxico para que hereden la etiqueta y las funciones básicas. Luego Primitivo y Array heredarán de la clase Abstracta TipoDato ya que no son datos simples y no tienen una etiqueta definida.

EJECUCION:

Hay dos opciones para probar el analizador.

1: Corriendo el programa desde algún IDE como Eclipse, se le pedirá el número del programa a probar en la consola, estos programas van del 1 al 12, se encuentran en la carpeta adjunta programas.

2: Por medio de la terminal y comandos, se ha adjuntado un ejecutable CasoPractico.jar, este ejecutable se correrá por medio del comando `java -jar CasoPractico.jar`. después el funcionamiento es el mismo, se le pedirá el programa a testear.

Si se quiere probar con un programa que no figure en la lista de programas de test que se adjunta en la entrega se tendrá que indicar el nombre modificando el código en la línea 21 de la clase TestAnalizador que se encuentra dentro del paquete Launcher que a su vez está en la carpeta src, se tendría que sustituir toda esa línea por:

```
String fichero="NOMBREDELARCHIVO.txt";
```