FORMATO DEL FICHERO CON LA TABLA DE SÍMBOLOS

Para la corrección de la Práctica de Procesadores de Lenguajes, y como se indica en la especificación de la misma (http://dlsiisv.fi.upm.es/procesadores/Practica.html), el Procesador deberá leer el programa fuente de un archivo de texto y entregar obligatoriamente los resultados generados en varios archivos de texto (lista de *tokens*, Tabla de Símbolos, *parse*, errores). Se detalla aquí el formato que ha de tener el **fichero** donde se almacenará la **Tabla de Símbolos**. Dicho formato es compatible con la librería TS-2006 de Tabla de Símbolos disponible en la web de la asignatura.

Las Tablas de Símbolos (TS) deben estar contenidas en un único fichero. El fichero deberá generarse inmediatamente antes de destruir cada TS, de forma que tenga toda la información recopilada durante el análisis.

El lenguaje que describe las TS no distingue minúsculas y mayúsculas, salvo en las cadenas. El lenguaje tiene varias palabras reservadas: "LEXEMA", "ATRIBUTOS" y los nombres de los atributos estándar indicados en la sección "Lista de atributos predefinidos".

Un fichero con la TS podrá tener líneas en blanco o líneas con la información de la TS. Cada línea de información debe acabar obligatoriamente con un salto de línea. Existen tres tipos de líneas obligatorias, cada una de ellas con un formato diferente:

Línea con el número de la TS:

Esta línea se utiliza como encabezado para comenzar cada TS. El formato de esta línea es:

```
pal* # del* núm del* : del* RC
```

Siendo:

- **pal** → Secuencia de caracteres que empieza por letra y puede contener letras[1] o dígitos.
- **pal*** → Conjunto de pal separadas por espacios o tabuladores. Se puede utilizar opcionalmente para poner un título a cada TS.
- **#** → Carácter 'almohadilla' obligatorio que indica que a continuación aparece el número identificador de la TS.
- **del*** → Cualquier cantidad de espacios en blanco, tabuladores o nada.
- **núm** → número entero sin signo formado por dígitos. Este número deberá ser distinto para cada TS generada y deberá identificar a cada TS reflejando el orden de creación de las distintas TS. Cada vez que se cree una nueva TS, su número deberá ser superior a las TS anteriores. Es decir, la TS global o principal deberá tener el número más bajo de todas las TS.
- **:** → Carácter 'dos puntos' obligatorio.
- **RC** → Salto de línea obligatorio.

Línea del lexema:

Esta línea se utiliza para indicar cada entrada de la TS. El formato de esta línea es:

```
del* * del* [lexema del* :] del* 'nombre' del* RC
```

[1] Cuando en este documento se menciona 'letras' se refiere a los caracteres de las letras sin acentos ni eñe.

Siendo:

- **del*** → Cualquier cantidad de espacios en blanco, tabuladores o nada.
- ***** → Carácter 'asterisco' obligatorio que indica que a continuación aparece el lexema de un identificador de la TS.
- **lexema del* :** → La secuencia de caracteres "LEXEMA" seguida de 'dos puntos' (quizás con blancos o tabuladores entre ellos) puede aparecer o no.
- **'nombre'** → Representa el nombre o lexema de uno de los identificadores que aparece en la TS. El nombre será una cadena encerrada entre comillas simples y puede estar formado por cualquier carácter (excepto blancos, tabuladores, saltos de línea o comillas simples).
- **RC** → Salto de línea obligatorio.

Línea de atributo:

Esta línea se utiliza para indicar cada atributo perteneciente a la entrada previa indicada en la 'línea del lexema' de la TS. El formato de esta línea es:

del* **+** del* **atributo** del* **:** del* **valor RC**

Siendo:

- **del*** → Cualquier cantidad de espacios en blanco, tabuladores o nada.
- **+** → Carácter 'más' obligatorio que indica que a continuación aparece un atributo de un identificador de la TS.
- **atributo** → Secuencia de caracteres que empieza por letra y puede contener letras o dígitos. Indica el nombre del atributo y es obligatorio. Hay una serie de nombres de atributo que deben utilizarse obligatoriamente para ciertos elementos de la tabla de símbolos, y que se indican en el siguiente apartado.
- **:** → Carácter 'dos puntos' obligatorio.
- **valor** → Valor del atributo. Puede tener dos formatos:
  - **'cadena'**: el valor del atributo puede ser una cadena encerrada entre comillas simples y puede estar formado por cualquier carácter (excepto blancos, tabuladores, saltos de línea o comillas simples). Si se quiere representar un valor de atributo nulo (vacío, sin valor) se puede utilizar una cadena con un guión ('-').
  - **núm**: número entero formado por dígitos; opcionalmente, puede ir precedido del signo menos.
- **RC** → Salto de línea obligatorio.

Lista de atributos predefinidos:

La TS debe tener una serie de atributos fijos que tienen un nombre predefinido que deberán utilizarse en el fichero de TS, con el significado que se indica a continuación:

- **Tipo** → representa el tipo del identificador.
- **Despl** → valor numérico que representa la dirección relativa que tendrá cada variable (el desplazamiento).
- **numParam** → valor numérico que representa el número de parámetros formales que tiene un identificador de tipo subprograma.
- **TipoParam*XX*** → representa el tipo del *XX*º parámetro de un subprograma. *XX* representa un número de hasta dos dígitos, cuyos valores irán desde el 1 hasta el valor del atributo numParam.
- **ModoParam*XX*** → representa el modo de paso del *XX*º parámetro de un subprograma. *XX* representa un número de hasta dos dígitos, cuyos valores irán desde el 1 hasta el valor del atributo numParam. Este atributo será necesario solamente cuando el lenguaje tenga distintos modos de paso de parámetros.

- **`TipoRetorno`** → representa el tipo que devuelve un identificador de tipo función.
- **`EtiqFuncion`** → representa la etiqueta que se asocia a un identificador de tipo función.
- **`Param`** → representa que un identificador local es un parámetro formal del subprograma donde está declarado. Se usa normalmente cuando el modo del paso del parámetro no es por valor.

Se pueden utilizar otros nombres de parámetros con otros significados, si se considera necesario, pero se deben utilizar obligatoriamente los nombres de parámetros dados con el significado indicado en cada caso, no pudiéndose utilizar nombres de atributos distintos con dicho significado.

Comentarios:

Se pueden incluir comentarios en cualquier parte del fichero. El inicio del comentario se indica con un paréntesis abierto "(" y el final del comentario con un paréntesis cerrado ")". Un comentario puede contener cualquier carácter, excepto el salto de línea o el paréntesis cerrado. Hay que notar que los comentarios no deben contener datos de la TS propiamente dicha, sino solamente información para aclarar su contenido al humano que lo lea.

Líneas Opcionales:

Después de la 'línea del lexema' y antes de la primera 'línea de atributos', puede aparecer opcionalmente la secuencia "ATRIBUTOS del* :" (la palabra "ATRIBUTOS", seguida de espacios o tabuladores opcionales y de 'dos puntos') seguida de un salto de línea.

También se puede poner una línea conteniendo únicamente uno o más guiones ("-"), blancos o tabuladores, para realizar separaciones que faciliten la legibilidad del fichero de Tabla de Símbolos.

Ejemplos:

Se indican a continuación algunos ejemplos de cómo podría ser unos ficheros de TS. Hay que hacer notar que son solamente ejemplos del formato, pero no indican los contenidos reales que debería tener un fichero de TS para un fichero fuente de la práctica.

Ejemplo 1: Tabla de Símbolos con un formato correcto:

```
CONTENIDOS DE LA TABLA # 100 :

* LEXEMA : 'edad2'
  Atributos :
  + tipo : (esto es de tipo "int") 'entero'
  + despl : 33
  --------- ----------
* lexema: 'valido#'
  ATRIBUTOS:
  + tipo: 'logico'  (**es decir, boolean**)
  + despl: 32
  --------- ----------
*      'nombre'
  + tipo : 'cadena'
  + despl : 0
```

Ejemplo 2: Misma Tabla de Símbolos que en el ejemplo 1, pero con un formato más compacto:

```
#100:
*'edad2'
+tipo:'entero'
+despl:33
*'valido#'
+tipo:'logico'
+despl:32
*'nombre'
+tipo:'cadena'
+despl:0
```

Ejemplo 3: Tabla de Símbolos con un formato correcto:

```
TABLA PRINCIPAL #1:
* (‘suma’ es una función, con 2 parámetros formales) LEXEMA : 'suma'
  ATRIButos:
  + tipo:                 'funcion'
    + numParam:           2
      + TipoParam01:      'ent'
       + ModoParam1:      1      (es por valor)
      + TipoParam2:       'real'
       + ModoParam02:     2      (por referencia)
      + TipoRetorno:      'ent'
    + EtiqFuncion:        'Etsuma01’
-------------------------------------------------
TABLA de la FUNCION suma #2:
* LeXeMa: 'a1' (tipo de entrada ‘parámetro’ pasado por valor)
   + despl : 0
   +  tipo   : 'ent'

* LEXEMA: 'b2' (es un ‘parámetro’ por referencia)
+tipo : 'real'
+despl : -2
+param: 1 (hay que usar este atributo para indicar que es un parámetro por referencia)

* Lexema: 'c_3' (es una ‘variable’ que es un vector de 500 enteros)
   + tipo :     'vector'
   + elementos :'ent'
   + tam:        500
   + despl : -8
```

Ejemplo 4: Tabla de Símbolos equivalente a la del ejemplo 3, con otro formato correcto:

```
TS de suma #2:
* 'a1'
 + despl : 0
 +  tipo     : 'ent'

* 'b2'
  + param: 1
  + tipo : 'real'
  + despl: -2
  + tiporetorno:  '-'     (se puede omitir)

* 'c_3'
  + tipo: 'vector'
  + elementos: 'ent'
  + tam: 500
  + despl: -8
```

```
#0:
* 'suma'
  + EtiqFuncion: 'Etsuma01'
  + tiporetorno: 'ent'
  + TIPO: 'funcion'
  + numParam: 2
  + TipoParam1: 'ent'
  + ModoParam01: 'PorValor'
  + TipoParam02: 'real'
  + ModoParam2: 'PorReferencia'
```

Ejemplo 5: Fichero con varios errores:

```
CONTENIDO ACTUAL DE LA TABLA  1 :                    (Error 1)
   + despl : 0                                       (Error 2)
* LEXEMA : apellidos (esto es un comentario)         (Error 3)
  ATRIBUTOS :
   + tipo 'string'                                   (Error 4)
   despl : 16                                        (Error 5)
* LEX : 'valido#'                                    (Error 6)
   + tipo : bool                                     (Error 7)
   + despl : +48                                     (Error 8)
   + 3arg : 'int'                                    (Error 9)
   + NumeroArgumentos: 1                             (Error 10)
   + tipoParam3: 'entero'                            (Error 11)
* "nombre"                                           (Error 12)
   + tipo : 'string'  + param : 0                    (Error 13)
   + Numero argumentos: 0                            (Error 14)
   + desplazamiento: 5                               (Error 15)
   + numParam: -1                                    (Error 16)
```

Error 1: No aparece el carácter especial # antes del número de la tabla.

Error 2: Se ha puesto un atributo sin haber indicado antes el lexema de la entrada de la TS.

Error 3: El lexema no va entre comillas.

Error 4: Falta el símbolo de dos puntos para separar el atributo del valor.

Error 5: Una línea de atributo debe comenzar por un signo +.

Error 6: La cadena “LEX :” no es una cadena válida en la línea del lexema. Debe poner “LEXEMA :” o nada.

Error 7: El valor del atributo (bool) no aparece entre comillas.

Error 8: Un número no puede llevar un signo positivo.

Error 9: El nombre del atributo no puede empezar por dígito.

Error 10: Se utiliza un nombre de atributo no estándar para representar el número de argumentos de un subprograma.

Error 11: El numeral del parámetro debe comenzar siempre por el 1. No puede estar el 3 sin el 1 y el 2.

Error 12: El lexema no puede ir entre comillas dobles.

Error 13: No hay salto de línea entre las dos líneas de atributo.

Error 14: El nombre de un atributo no puede contener espacios. Además, no se estaría utilizando el nombre estándar de atributo para representar el número de argumentos de un subprograma.

Error 15: Se utiliza un nombre de atributo no estándar para representar la dirección de una variable.

Error 16: No tiene sentido que el número de parámetros sea un valor negativo.