

Documentación

Compilador LED

Autores:

García Robles, Carlos Adan
Juárez Amézquita, Melvin Abisaí

Especificaciones Léxicas

Las entidades léxicas (*tokens*) que puedan ser procesadas por el token manager, se definen en el archivo de entrada y se validan mediante las expresiones regulares, detalladas en este mismo documento.

El compilador deberá usar el lenguaje LED e interpretar el código fuente para poder interactuar con varios archivos de hoja de cálculo, mediante una programación funcional y simple. El lenguaje usa palabras en castellano sin tilde, por lo que a simple vista podrá parecer que tiene problemas ortográficos en algunos casos.

EJEMPLO

```
FUNCION NUMERO esImpar (numero):
variables:
    numero a;
codigo:
    si ( num % 2 ) ENTONCES:
        a++;
        retornar 1;

    SINO :
        retornar 0;

//area de declaracion ed funcion de carga principal
/*
sakjdaslkfjkasd

*/
FUNCION NUMERO principal();
variables :
    numera x;
codigo:
    a = 1;
    si ( esImpar(a) ) entonces:
        escribir "El numero encontrado es impar:";
    sino:
        escribir "el numero encontrado no es par";
    retornar 0;
```

//Funciones y Encabezados

Decimal	<code>[1-9][0-9]* 0</code>
Identificador	<code>[a-zA-Z][a-zA-Z0-9_]*</code>

//Gramaticales

TokenFinLinea	<code>;</code>
TokenComillas	<code>"</code>
TokenDosPuntos	<code>:</code>
TokenIdentacion	

//Externos

TokenComentarios	<code>//.*</code>
TokenSout	<code>{TokenComillas}.*{TokenComillas}</code>

//Agrupadores

ParenL	<code>(</code>
ParenR	<code>)</code>

//Preprocesador

TokenInclude	<code>INCLUIR" "incluir</code>
--------------	--------------------------------

//Funciones y encabezados

TokenFunction	FUNCION" "funcion
TokenReturn	RETORNAR" "retornar
TokenEVariables	("VARIABLES" "variables"){TokenDosPuntos}?
TokenECodigo	("CODIGO" "codigo"){TokenDosPuntos}?
TokenEFile	("ARCHIVOS" "archivos"){TokenDosPuntos}?

//Tipos de datos

TokenInt	NUMERO" "numero
TokenString	CADENA" "cadena
TokenBool	BOOLEANO" "booleano
TokenNull	NULL" "null
TokenTipos	{TokenInt} {TokenString} {TokenBool}

//Instrucciones

TokenPrint	ESCRIBIR" "escribir
TokenOpen	ABRIR" "abrir

//Sentencias

TokenIf	SI" "si
TokenElse	("SINO" "sino"){TokenDosPuntos}?
TokenThen	("ENTONCES" "entonces"){TokenDosPuntos}?
TokenFor	DESDE" "desde
TokenWhile	MIENTRAS" "mientras
TokenIterar	("ITERAR" "iterar"){TokenDosPuntos}?
TokenParaCada	PARACADA" "paracada

//Operadores Logicos

TokenAnd	AND" "and
TokenOr	OR" "or

//Operadores Basicos

TokenAsignacion	=
TokenComas	,

//Operadores Aritmeticos

TokenAsignacion	=
TokenComas	,
TokenSuma	+
TokenResta	-
TokenMulti	*
TokenDiv	/
TokenMod	%
TokenExp	^
TokenInc	INCREMENTAR" "incrementar
TokenDec	DECREMENTAR" "decrementar