

Equipo 2 Jueves. Lenguaje BABAC

La descripción del lenguaje con el que vamos a trabajar tiene las siguientes características:

- Es fuertemente tipado.
- Los tipos de datos son: entero, real, carácter y booleano.
- Tiene una sentencia de asignación para todos los tipos de expresiones.
- Tendrá operadores lógicos.
- Las operaciones de entrada y salida serán a través del teclado y pantalla respectivamente, además el operador de entrada permite leer sobre una lista de identificadores y el operador de salida permite escribir sobre una lista de expresiones y/o constantes de tipo cadena.
- Estructuras de control:
 - si-sino
 - mientras
- En cualquier parte del programa se podrán definir bloques y la declaración de variables se realiza al comienzo del bloque delimitado con marcas de inicio y de fin.
- El lenguaje distingue entre mayúsculas y minúsculas.

Características concretas de nuestro lenguaje (BABAC)

- Está basado en el lenguaje C.
- Las palabras reservadas deben estar en castellano.
- Contiene una estructura de datos tipo array 1D y 2D que puede contener datos de tipo entero, real, carácter o booleano. Los métodos que contiene dicha estructura son:
 - De acceso a un elemento.
 - Producto de elemento a elemento.
 - Suma de elemento a elemento.
 - Resta de elemento a elemento.
 - Producto externo array por un escalar.
 - Producto de matrices $C=A*B$.
- Los subprogramas deben ser definidos como funciones.
- La estructura de control a desarrollar es "repeat-until" (repetir_hasta)

BNF

```
<programa> ::= <cabecera_programa> <bloque>
<cabecera_programa> ::= 'principal'
<bloque> ::= <inicio_bloque> <declar_variables_locales>
<declar_subprogs> <sentencias> <fin_bloque>
<inicio_bloque> ::= '{'
<declar_variables_locales> ::= <ini_declar_variables>
<variables_locales> <fin_declar_variables>
<declar_subprogs> ::= <declar_subprogs> <declar_subprog>
<sentencias> ::= <sentencias> <sentencia>
| <sentencia>
<fin_bloque> ::= '}'
<ini_declar_variables> ::= 'var'
<variables_locales> ::= <variables_locales> <cuerpo_declar_variables>
| <cuerpo_declar_variables>

<fin_declar_variables> ::= 'finvar'
<declar_subprog> ::= <cabecera_subprograma> <bloque>
<sentencia> ::= <bloque>
| <asignacion>
| <si>
| <mientras>
| <entrada>
| <salida>
| <devolver>
| <repetir_hasta>
<cuerpo_declar_variables> ::= <tipo_basico> <lista_variables> ';'
<cabecera_subprograma> ::= <tipo_basico> <variable> '('
<lista_parametros> ')'
| <tipo_basico> <identificador> '(' ')'
<asignacion> ::= <var_exp> '=' <expresion> ';'
<variable> ::= <identificador>
| <identificador> <declar_matriz>
<var_exp> ::= <identificador>
| <identificador> '[' <expresion> ']'
| <identificador> '[' <expresion> ';' <expresion> ']'
<si> ::= 'si' '(' <expresion> ')' <sentencia> 'sino' <sentencia>
| 'si' '(' <expresion> ')' <sentencia>
<mientras> ::= 'mientras' '(' <expresion> ')' <sentencia>
<entrada> ::= 'leer' <lista_variables> ';'
<salida> ::= 'escribir' <lista_expresiones_cad> ';'
<devolver> ::= 'devolver' <expresion> ';'
<repetir_hasta> ::= 'repetir' <sentencia> 'hasta' '(' <expresion> ')' ';'
<lista_variables> ::= <lista_variables> ',' <variable>
| <variable>
<identificador> ::= ([a-zA-Z_] [0-9a-zA-Z]*)
<lista_parametros> ::= <lista_parametros> ',' <tipo_basico> <variable>
| <tipo_basico> <variable>
```

```

<expresion> ::= '(' <expresion> ')'
            | <op_unario> <expresion>
            | <expresion> <op_binario> <expresion>
            | <variable>
            | <constante>
            | <llamada_funcion>
<lista_expresiones> ::= <expresion> ',' <lista_expresiones>
            | <expresion>
<lista_expresiones_cad> ::= <lista_expresiones_cad> ',' <exp_cad>
            | <exp_cad>
<exp_cad> ::= <expresion> | <constante_cadena>
<tipo_basico> ::= 'caracter'
            | 'real'
            | 'entero'
            | 'logico'
<declar_matriz> ::= '[' <constante_entera> ']'
            | '[' <constante_entera> ',' <constante_entera> ']'
<op_unario> ::= '+'
            | '-'
            | '!'
<op_binario> ::= '+'
            | '-'
            | '*'
            | '/'
            | '=='
            | '!='
            | '<='
            | '>='
            | '<'
            | '>'
            | '&&'
            | '||'
            | '^'
<constante> ::= <constante_entera>
            | <constante_matriz>
            | <constante_logica>
            | <constante_real>
            | <constante_caracter>
<llamada_funcion> ::= <identificador> '(' <lista_expresiones> ')' ';'
            | <identificador> '(' ')' ';'
<constante_matriz> ::= '{' <auxiliar_matriz> '}'
<auxiliar_matriz> ::= <agregado_entero>
            | <agregado_real>
            | <agregado_logico>
            | <agregado_caracter>

<agregado_entero> ::= '{' <fila_agregado_entero> '}'
<fila_agregado_entero> ::= <fila_agregado_entero> ';' <lista_entera>
            | <lista_entera>

```

```

<lista_entera> ::= <lista_entera> ',' <constante_entera>
                | <constante_entera>
<agregado_real> ::= '{' <fila_agregado_real> '}'
<fila_agregado_real> ::= <fila_agregado_real> ';' <lista_real>
                        | <lista_real>
<lista_real> ::= <lista_real> ',' <constante_real>
                | <constante_real>

<agregado_logico> ::= '{' <fila_agregado_logico> '}'
<fila_agregado_logico> ::= <fila_agregado_logico> ';' <lista_logica>
                        | <lista_logica>
<lista_logico> ::= <lista_logica> ',' <constante_logico>
                | <constante_logico>

<agregado_caracter> ::= '{' <fila_agregado_caracter> '}'
<fila_agregado_caracter> ::= <fila_agregado_caracter> ';'
<lista_caracter>
                        | <lista_caracter>
<lista_caracter> ::= <lista_caracter> ',' <constante_caracter>
                | <constante_caracter>

<constante_entera> ::= [0-9]+
<constante_cadena> ::= '"' [^"]+ '"'
<constante_logica> ::= 'verdadero'
                | 'falso'
<constante_real> ::= <constante_entera> '.' <constante_entera>
<constante_caracter> ::= '"' [^'] '"'

```

