Adrian Samuel Molina Cabrera

201903850

Gramática

Expresiones Regulares

Expresiones en blanco:

1. \s+
2. [\t\r\n\f]
3. \n
4. "/""/".*
5. [/][*][^*/]*[*][/]

Numero Decimal:

1. [0-9]+("."[0-9]+)?\b

Id:

1. ([a-zA-Z])[a-zA-Z0-9_]*

Carácter:

1. (\'[^']?\')

Terminales

Tipos:

Int	Boolean	String
Double	Char	Void

Palabras Reservadas:

True	New
False	Add
List	

Operadores Aritmeticos:

++	--	*	^
+	-	/	%

Operadores Relacionales:

==	<=	>=
¡=	<	>

Operadores Lógicos:

||	&&	!

Agrupación:

(	[	{
)	]	}

Otros Operadores:

=	:	.
?	;	,

Sentencias:

If	Case	While	For
Else	Default	Do	Continue
Switch	Break	Print	Return

Funciones Nativas:

toLower	Truncate	toString
toUpper	Round	toCharArray
length	TypeOf	exec

Producciones

"Ini -> instrucciones EOF"

Esta es la producción inicial en la cual se crea el árbol que maneja todo el programa

"Instrucciones -> instrucciones instrucción

|instrucción"

Esta producción es la que lee y almacena todo en un arreglo, para luego agregarlo al árbol

"Instrucción -> declaracionVar

|metodos

|llamada

|sentencia_if

|sentencia_switch

|sentencia_while

|sentencia_dowhile

|sentencia_for

|sentencia_print

|Id INC PTCOMA

|Id DEC PTCOMA

|Continue PTCOMA

|Break PTCOMA

|sentencia_Return

|Exec llamada"

Aquí se manejaban todas las instrucciones que se podían generar con el programa Typesty.

“metodos -> tipos ID PARIZQ parametros PARDER LLAIZQ instrucciones LLADER

|tipos ID PARIZQ PARDER LLAIZQ instrucciones LLADER”

En los metodos se pueden declarar de 2 formas, con o sin parametros.

“llamada -> llamar PTCOMA”

Esta producción sirve únicamente como una forma fácil de llamar a la siguiente producción.

“llamar -> ID PARIZQ parametros_llamada PARDER

|ID PARIZQ PARDER

Esta producción sirve para ejecutar el método almacenado en la tabla de simbolos, dependiendo de si trae o no parámetros.

“parametros_llamada -> parametros_llamada COMA expresión

|expresión”

En esta producción se envían las expresiones necesarias para ejecutar el método a llamar.

“parametros -> parametros COMA tipos ID

|tipos ID”

En esta producción se declaran los parametros que serán utilizados en los metodos o funciones.

“sentencia_if -> IF PARIZQ expresión PARDER LLAIZQ instrucciones LLADER

|IF PARIZQ expresión PARDER LLAIZQ LLADER

|IF PARIZQ expresión PARDER LLAIZQ instrucciones LLADER ELSE LLAIZQ instrucciones LLADER

|IF PARIZQ expresión PARDER LLAIZQ instrucciones LLADER ELSE sentencia_if”

Hay 4 formas de declarar un if, las cuales son:

1. Únicamente el if e instrucciones
2. Un if y un else
3. If anidados
4. If sin condición e instrucciones.

“sentencia_switch -> SWITCH PARIZQ expresion PARDER LLAIZQ caseList defaultList LLADER

|SWITCH PARIZQ expresion PARDER LLAIZQ caseList LLADER

|SWITCH PARIZQ expresion PARDER LLAIZQ defaultList LLADER”

Hay 3 formas de declarar un switch, las cuales son:

1. Con un caseList y un defaultList
2. Únicamente CaseList
3. Únicamente DefaultList

“caseList -> caseList CASE expresion DPUNTOS instrucciones

|CASE expresion DPUNTOS instrucciones”

Esta producción retorna un arreglo con en caseList a utilizar en el switch.

“defaultList -> DEFAULT DPUNTOS instrucciones”

Esta producción retorna las instrucciones del defaultList a utilizar en el switch.

“sentencia_while -> WHILE PARIZQ expresion PARDER LLAIZQ instrucciones LLADER”

La sentencia While recibe únicamente la expresion de validación y las instrucciones a ejecutar mientras la condición se cumpla.

“sentencia_dowhile -> DO LLAIZQ instrucciones LLADER WHILE PARIZQ expresion PARDER PTCOMA”

La sentencia DoWhile recibe únicamente la expresion de validación y las instrucciones a ejecutar mientras la condición se cumpla.

“sentencia_for -> FOR PARIZQ forVar PTCOMA expresion PTCOMA for_increment PARDER LLAIZQ instrucciones LLADER”

La sentencia For obtiene la variable que se utilizara en la iteración, las instrucciones a realizar y el incremento de la variable.

“forVar -> tipos ID ASIGNAR expresion

|ID ASIGNAR expresion”

Esta producción es para asignar la variable de incremento del for.

“for_increment -> ID INC

|ID DEC

|ID ASIGNAR expresion”

Esta producción sirve para decidir cómo será el incremento del for en cada iteración.

“sentencia_print -> PRINT PARIZQ expresion PARDER PTCOMA”

La sentencia print recibe únicamente la expresion a mostrar en consola.

“sentencia_return -> RETURN expresion PTCOMA

|RETURN PTCOMA”

La sentencia return devuelve el valor indicado.

“declaracionVar -> tipos ID PTCOMA

|tipos ID ASIGNAR expresion PTCOMA

|ID ASIGNAR expresion PTCOMA

|tipos CORIZQ CORDER ID ASIGNAR NEW tipos CORIZQ expresion CORDER PTCOMA

|tipos CORIZQ CORDER ID ASIGNAR LLAIZQ listaValores LLADER PTCOMA

|ID CORIZQ expresion CORDER ASIGNAR expresion PTCOMA

|LIST MENORQ tipos MAYORA ID ASIGNAR NEW LIST MENORQ tipos MAYORA PTCOMA

|LIST MENORQ tipos MAYORA ID ASIGNAR tocha PTCOMA

|ID PUNTO ADD PARIZQ expresion PARDER PTCOMA

|ID CORIZQ CORIZQ expresion CORDER CORDER ASIGNAR expresion PTCOMA

En esta producción se declaran variables, vectores, y listas, también sirve para asignar valores a esas variables.

“expresion -> MENOS expresion

|NOT expresion

|ID INC

|expresion MAS expresion

|ID DEC

|expresion MENOS expresion

|expresion POR expresion

|expresion DIVIDIDO expresion

|expresion MOD expresion

|expresion POT expresion

|expresion MENORIGUALQ expresion

|expresion MENORQ expresion

|expresion MAYORIGUALQ expresion

|expresion MAYORA expresion

|expresion IGUALA expresion

|expresion DIFERENTED expresion

|expresion OR expresion

|expresion AND expresion

|ID

|DECIMAL

|TRUE

|FALSE

|CADENA

|CARACTER

|ID CORIZQ expresion CORDER

|ID CORIZQ CORIZQ expresion CORDER CORDER

|PARIZQ expresion PARDER

|PARIZQ tipos PARDER expresion

|expresion INTERROGACION expresion DPUNTOS expresion

|LENGTH PARIZQ expresion PARDER

|TOLOWER PARIZQ expresion PARDER

|TOUPPER PARIZQ expresion PARDER

|TRUNCATE PARIZQ expresion PARDER

|ROUND PARIZQ expresion PARDER

|TYPEOF PARIZQ expresion PARDER

|TOSTRING PARIZQ expresion PARDER

|llamar”

En esta producción es donde se encuentran todas las operaciones y valores, los cuales son utilizados en todas las instrucciones de la gramática.

“listaValores -> listaValores COMA expresion

|expresion

Esta expresion retorna un arreglo con valores.

“tocha -> TOCHARARRAY PARIZQ expresion PARDER”

Esta producción obtiene una expresion y devuelve un arreglo de caracteres.

“tipos -> TINT

|TDOUBLE

|TBOOLEAN

|TCHAR

|TSTRING

|TVOID”

Esta producción devuelve el tipo que será utilizado en las instrucciones.