GRAMATICA

PROYECTO2

ORGANIZACIÓN DE COMPILADORES Y LENGUAJES 1

David Roberto Díaz Prado

201807420

La Gramatica que utilizada para la realización de este proyecto

La sintaxis que se utilizara para todo este proyecto esta dada en el enunciado:

**Patrones que acepta la gramatica**

**Numeros Enteros (0-9)+**

**Numeros Decimales [(0-9)+]?“.”(0-9)+**

**Cadena “\””*todo lo que venga* “\””**

**Caracteres “ ‘ “ *[Todo lo que venga]? “ ‘ “***

***Identificador ([a-z]|[A-Z])+(\_ |[a-z]|[A-Z]|[0-9])\****

Iniciamos la Gramatica:

Los no terminales los pondré en comillas y los terminales en mayúsculas

Nuestro estado Inicial: INIT

*INIT -> BLOQUEGLOBAL*

Producción que define el cuerpo del archivo

*BLOQUEGLOBAL->BLOQUEGLOBAL MAIN*

*| BLOQUEGLOBAL METODO*

*|BLOQUEGLOBAL DECLARACION*

*|BLOQUEGLOBAL ASIGNACION*

*| BLOQUEGLOBAL FUNCION*

*|FUNCION*

*|MAIN*

*|METODO*

*|DECLARACION*

*|AISGNACION*

Producción recursiva a la izquierda para que acepta todas las instrucciones globales del archivo de practica.

*MAIN -> “exec” “(“ IDENTIFICADOR “)” “;“*

Producción de definición del método main que recibe como parametro el nombre de un método o función.

*DECLARACION -> TIPO IDENTIFICADOR “=” EXP “;”*

*|TIPO IDENTIFICADOR “;”*

Producción que declara una variable nueva

*ASIGNACION->IDENTIFICADOR “=” EXP “;”*

Producción de para asignar un valor a un identificador

*METODO -> “void” IDENTIFICADOR “(“ PARAMETROS ”)” “{“ CUERPO2 “}”*

*|“void” IDENTIFICADOR “(“ ”)” “{“ CUERPO2 “}”*

Producción para declarar métodos con y sin parámetros

*PARAMETROS ->PARAMETROS “,” TIPO IDENTIFICADOR*

*|TIPO IDENTIFICADOR*

Producción para los parámetros

*TIPO -> “int”*

*|“double”*

*|”string”*

*|”char”*

*|”boolean”*

Producción que define los tipos de valores existenetes en el lenguaje

*CUERPO2->CUERPO2 INSTRUCCIONES*

*|INSTRUCCIONES*

Producción para definir las instrucciones locales

*INSTRUCCIONES->ASIGNACION*

*|DECLARACION*

*|CICLOS*

*|CONTROLES*

*|BREAK*

*|LLAMADA*

Producción para definir las instrucciones de los entornos locales

*BREAK->”break” “;”*

Producción para definir un break

*LLAMADA-> IDENTIFICADOR “(“ VALLAMADA ”)” “;”*

*| IDENTIFICADOR “(“”)” “;”*

Producción para definir las llamadas de metodos

*VALLAMADA -> VALLAMADA “,“ EXP*

*|EXP*

Producción para mandar valores en lso métodos llamados

*CONTROLES->IF*

*|SWITCH*

Producción para definir las controles o sentencias de control

*CICLOS->WHILE*

*|DOWHILE*

*|FOR*

Producción para definir las instrucciones cíclicas

IF-> “if” “(”EXP”)” “{” CUERPO2 ”}”

|“if” “(”EXP”)” “{” CUERPO2 ”}” “else” “{” CUERPO2 “}”

|“if” “(”EXP”)” “{” CUERPO2 ”}” “else” IF

Producción para definir la sentencia IF

SWITCH -> “switch” “(”EXP”)” CASOS

|“switch” “(”EXP”)” CASOS “default” “:” CUERPO2

|“switch” “(”EXP”)” “default” “:” CUERPO2

Producción para definir la sentencia SWITCH

CASOS-> CASOS “case” EXP “:” CUERPO2

|“case” EXP “:” CUERPO2

Producción para definir la casos para switch

*FOR->”for” “(”ASIGNACION “;” EXP “;” ASIGNACION ”)” “{“ CUERPO2 “}”*

*|”for” “(” DECLARACION “;” EXP “;” ASIGNACION ”)” “{“ CUERPO2 “}”*

Producción para definir el ciclo for

WHILE-> “while” “(“ EXP ”)” “{“ CUERPO “}”

Producción para definir el ciclo while

DOWHILE->”do” “{“ CUERPO “}” “while” “(“ EXP ”)”

Producción para definir el ciclo dowhile

FUNCION->TIPO IDENTIFICADOR “(“ PARAMETROS “)” “{” CUERPO “}”

| TIPO IDENTIFICADOR “(“ “)” “{” CUERPO “}”

Producción para definir FUNCIONES

*EXP->”(”EXP”)”*

*|ARITMETICAS*

*|RELACIONALES*

*|LOGICAS*

*|”true”*

*|”false”*

*|IDENTIFICADOR*

*|ENTERO*

*|DECIMAL*

*|CARÁCTER*

*|CADENA*

Producción donde se declaran las posibles operaciones

ARITMETICAS-> EXP + EXP

|EXP – EXP

|EXP \* EXP

|EXP / EXP

|EXP % EXP

|EXP ^ EXP

Producción donde se declaran las posibles operaciones Aritmeticas

RELACIONALES -> EXP == EXP

|EXP != EXP

|EXP > EXP

|EXP < EXP

|EXP >= EXP

|EXP <= EXP

Producción donde se declaran las posibles operaciones RELACIONALES

LOGICAS -> EXP || EXP

|EXP && EXP

|! EXP

Producción donde se declaran las posibles operaciones RELACIONALES