import java\_cup.runtime.\*;

%%

%class AnalizadorLexico

%unicode

%cup

%%

program {return new Symbol(sym.program);}

begin {return new Symbol(sym.begin);}

end {return new Symbol(sym.end);}

const {return new Symbol(sym.const\_);}

var {return new Symbol(sym.var);}

procedure {return new Symbol(sym.procedure);}

function {return new Symbol(sym.function);}

integer {return new Symbol(sym.integer);}

real {return new Symbol(sym.real);}

char {return new Symbol(sym.char\_);}

type {return new Symbol(sym.type);}

array {return new Symbol(sym.array);}

of {return new Symbol(sym.of);}

record {return new Symbol(sym.record);}

if {return new Symbol(sym.if\_);}

then {return new Symbol(sym.then);}

else {return new Symbol(sym.else\_);}

while {return new Symbol(sym.while\_);}

do {return new Symbol(sym.do\_);}

for {return new Symbol(sym.for\_);}

to {return new Symbol(sym.to);}

case {return new Symbol(sym.case\_);}

:= {return new Symbol(sym.op\_asignacion);}

\< {return new Symbol(sym.op\_menor);}

\> {return new Symbol(sym.op\_mayor);}

\<= {return new Symbol(sym.op\_menorIgual);}

\>= {return new Symbol(sym.op\_mayorIgual);}

= {return new Symbol(sym.op\_igual);}

\<\> {return new Symbol(sym.op\_distinto);}

\+ {return new Symbol(sym.op\_suma);}

- {return new Symbol(sym.op\_resta);}

\\* {return new Symbol(sym.op\_mult);}

div {return new Symbol(sym.op\_divEntero);}

mod {return new Symbol(sym.op\_mod);}

\/ {return new Symbol(sym.op\_divReal);}

or {return new Symbol(sym.op\_or);}

and {return new Symbol(sym.op\_and);}

not {return new Symbol(sym.op\_not);}

\[ {return new Symbol(sym.abrir\_corchete);}

\] {return new Symbol(sym.cerrar\_corchete);}

\( {return new Symbol(sym.abrir\_paren);}

\) {return new Symbol(sym.cerrar\_paren);}

\. {return new Symbol(sym.punto);}

\, {return new Symbol(sym.coma);}

\: {return new Symbol(sym.dos\_puntos);}

\; {return new Symbol(sym.punto\_coma);}

[+-]?[0-9]+(\.[0-9]+)? {return new Symbol(sym.numeric\_real\_const,yytext());}

[+-]?[0-9A-F]+(\.[0-9A-F]+)? {return new Symbol(sym.numeric\_realHex\_const,yytext());}

[+-]?[0-9]+ {return new Symbol(sym.numeric\_integer\_const,yytext());}

[+-]?[0-9A-F]+ {return new Symbol(sym.numeric\_integerHex\_const,yytext());}

'[^']\*' {return new Symbol(sym.string\_const, yytext());}

\{[^}]\*\} | \(\\*[^(\)\\*)]\*\\*\) {return new Symbol(sym.comentario, yytext());}

[a-zA-Z\_]+[a-zA-Z\_0-9]\* {return new Symbol(sym.identifier, yytext());}

\. | \r\n {;}