// MiniLan.lex

import java.lang.System;

import java\_cup.runtime.Symbol;

%%

%{

// This variable determines if the scanner prints

// messages on the screen (true) or if it only returns

// the token to the parser (false). You can change

// this value depending on whether you are interested

// in knowing what the scanner does or not.

boolean output = true;

%}

%char

%public

%cup

%full

%type java\_cup.runtime.Symbol

DIGIT=([0-9])

INTEGER=({DIGIT}+)

LETTER=([A-Za-z])

BLANK=(" "|\t)

COMMENT=("//"({LETTER}|{DIGIT}|{BLANK})\*)

REAL=(({DIGIT}\*"."{DIGIT}+)|({DIGIT}+"."{DIGIT}\*))

IDENT=({LETTER}(({LETTER}|{DIGIT}|"\_")\*({LETTER}|{DIGIT}))?)

%%

begin {if (output) System.out.println("SCANNER:: BEGIN");

return new Symbol(sym.BEGIN);}

end {if (output) System.out.println("SCANNER:: END");

return new Symbol(sym.END);}

print {if (output) System.out.println("SCANNER:: PRINT");

return new Symbol(sym.PRINT);}

; {if (output) System.out.println("SCANNER:: EOS");

return new Symbol(sym.EOS);}

"(" {if (output) System.out.println("SCANNER:: LP");

return new Symbol(sym.LP);}

")" {if (output) System.out.println("SCANNER:: RP");

return new Symbol(sym.RP);}

"+" {if (output) System.out.println("SCANNER:: PLUS");

return new Symbol(sym.PLUS);}

"-" {if (output) System.out.println("SCANNER:: MINUS");

return new Symbol(sym.MINUS);}

"/" {if (output) System.out.println("SCANNER:: DIV");

return new Symbol(sym.DIV);}

\\* {if (output) System.out.println("SCANNER:: MULT");

return new Symbol(sym.MULT);}

{INTEGER} {if (output) System.out.println("SCANNER:: NUMBER <"+yytext()+">");

return new Symbol(sym.NUMBER);}

{REAL} {if (output) System.out.println("SCANNER:: NUMBER <"+yytext()+">");

return new Symbol(sym.NUMBER);}

{IDENT} {if (output) System.out.println("SCANNER:: IDENT <"+yytext()+">");

return new Symbol(sym.IDENT);}

{COMMENT} {if (output) System.out.println("SCANNER:: COMMENT <"+yytext()+">");}

(" "|\t|\n|\r)+ {}

. {if (output) System.out.println("SCANNER:: Unmatched input "+ yytext());}

import java.io.FileReader;

import java\_cup.runtime.\*;

parser code{:

public static void main(String args[]) throws Exception {

new parser(new Yylex(new FileReader(args[0]))).parse();

} // End Main

:}

terminal BEGIN, END, PRINT;

terminal PLUS, MINUS, MULT, DIV;

terminal EOS, LP, RP;

terminal NUMBER;

terminal IDENT;

non terminal program, instructions, instruction, sentence, printSentence;

non terminal arithExpr, term, factor;

program ::= BEGIN instructions END {: System.out.println(" PARSER:: program <== BEGIN instructions END"); :}

;

instructions ::= instructions instruction {: System.out.println(" PARSER:: instructions <== instructions instruction");:}

| instruction {: System.out.println(" PARSER:: instructions <== instruction");:}

;

instruction ::= sentence EOS {: System.out.println(" PARSER:: instruction <== sentence EOS");:}

;

sentence ::= printSentence {: System.out.println(" PARSER:: sentence <== printSentence ");:}

;

printSentence ::= PRINT LP arithExpr RP {: System.out.println(" PARSER:: printSentence <== PRINT LP arithExpr RP"); :}

;

arithExpr ::= arithExpr PLUS term {: System.out.println(" PARSER:: arithExpr <== arithExpr PLUS term"); :}

| arithExpr MINUS term {: System.out.println(" PARSER:: arithExpr <== arithExpr MINUS term"); :}

| term {: System.out.println(" PARSER:: arithExpr <== term"); :}

;

term ::= term MULT factor {: System.out.println(" PARSER:: term <== term MULT factor"); :}

| term DIV factor {: System.out.println(" PARSER:: term <== term DIV factor"); :}

| factor {: System.out.println(" PARSER:: term <== factor"); :}

;

factor ::= NUMBER {: System.out.println(" PARSER:: factor <== NUMBER "); :}

| LP arithExpr RP {: System.out.println(" PARSER:: factor <== LP arithExpr RP"); :}

| MINUS factor {: System.out.println(" PARSER:: factor <== MINUS factor"); :}

| PLUS factor {: System.out.println(" PARSER:: factor <== PLUS factor"); :}

;

begin

print(3 + 2 \* 5);

print(2 \* 4 + 6);

print(6-3);

print(6-3/2.0);

print((6-3)/2.0);

print(-(6-3)/2.0);

end