**assignment3.g**

grammar assignment3;

@header{

package sil;

import java.util.HashMap;

}

@lexer::header{

package sil;

}

@members{

/\*\* Map variable name to Integer object holding value \*/

HashMap memory = new HashMap();

}

program: stat+ ;

stat: expr NEWLINE {System.out.println($expr.value);}

| 'LET' ID '=' expr NEWLINE

{memory.put($ID.text, new Integer($expr.value));}

| 'PRINT' ID {System.out.print(memory.get($ID.text));}

| 'PRINT' STRING {System.out.print($STRING.text.replace("\"", ""));}

| 'PRINTLN' ID {System.out.println(memory.get($ID.text));}

| 'PRINTLN' STRING {System.out.println($STRING.text.replace("\"", ""));}

| NEWLINE

;

expr returns [int value]

: e=multExpr {$value = $e.value;}

( '+' e=multExpr {$value += $e.value; }

| '-' e=multExpr {$value -= $e.value;}

)\*

;

multExpr returns [int value]

: e=atom {$value = $e.value;} ('\*' e=atom {$value \*= $e.value;})\*

;

atom returns [int value]

: INT {$value = Integer.parseInt($INT.text);}

| ID

{

Integer v = (Integer)memory.get($ID.text);

if ( v!=null ) $value = v.intValue();

else System.err.println("undefined variable "+$ID.text);

}

| '(' expr ')' {$value = $expr.value;}

;

ID : ('a'..'z'|'A'..'Z'|'\_') ('a'..'z'|'A'..'Z'|'0'..'9'|'\_')\*

;

INT : '0'..'9'+

;

FLOAT

: ('0'..'9')+ '.' ('0'..'9')\* EXPONENT?

| '.' ('0'..'9')+ EXPONENT?

| ('0'..'9')+ EXPONENT

;

COMMENT

: '//' ~('\n'|'\r')\* '\r'? '\n' {$channel=HIDDEN;}

| '/\*' ( options {greedy=false;} : . )\* '\*/' {$channel=HIDDEN;}

;

WS : ( ' '

| '\t'

| '\r'

| '\n'

) {$channel=HIDDEN;}

;

NEWLINE:'\r'? '\n' ;

STRING

: '"' ( ESC\_SEQ | ~('\\'|'"') )\* '"'

;

CHAR: '\'' ( ESC\_SEQ | ~('\''|'\\') ) '\''

;

fragment

EXPONENT : ('e'|'E') ('+'|'-')? ('0'..'9')+ ;

fragment

HEX\_DIGIT : ('0'..'9'|'a'..'f'|'A'..'F') ;

fragment

ESC\_SEQ

: '\\' ('b'|'t'|'n'|'f'|'r'|'\"'|'\''|'\\')

| UNICODE\_ESC

| OCTAL\_ESC

;

fragment

OCTAL\_ESC

: '\\' ('0'..'3') ('0'..'7') ('0'..'7')

| '\\' ('0'..'7') ('0'..'7')

| '\\' ('0'..'7')

;

fragment

UNICODE\_ESC

: '\\' 'u' HEX\_DIGIT HEX\_DIGIT HEX\_DIGIT HEX\_DIGIT

;

**Input File:**



