Goal ::= MainClass ( ClassDeclaration )\* <EOF>  
MainClass ::= "class" Identifier "{" "public" "static" "void" "main" "(" "String" "["  
"]" Identifier ")" "{" Statement "}" "}"  
ClassDeclaration ::= "class" Identifier ( "extends" Identifier )? "{" ( VarDeclaration )\* (  
ConstructorDeclaration )\* ( MethodDeclaration )\* "}"  
VarDeclaration ::= Type Identifier ";"  
ConstructorDeclaration::= Identifier  
"(" ( Type Identifier ( "," Type Identifier )\*)? ")"  
"{" ( VarDeclaration )\* ( Statement )\* "}"  
MethodDeclaration ::= ("public" | "private" | "protected") Type Identifier  
"(" ( Type Identifier ( "," Type Identifier )\*)? ")"  
"{" ( VarDeclaration )\* ( Statement )\* "return" Expression ";" "}"

Type ::= “int” ARR | “Boolean” ARR | “char” ARR | “String” ARR | “Float” ARR

ARR ::= “[” “]” | lamda

===================================================================================

Statement ::= "{" ( Statement )\* "}"  
| "if" "(" Expression ")" Statement ELSE\_PART   
| "while" "(" Expression ")" Statement  
| "System.out.println" "(" Expression ")" ";"  
| Identifier ID

ELSE\_PART ::= "else" Statement | lamda

ID ::= "=" Expression ";" | "[" Expression "]" "=" Expression ";"

===================================================================================

Expression ::=   
| <INTEGER\_LITERAL> EXP\_DASH  
| <FLOAT\_LITERAL> EXP\_DASH  
| "true" EXP\_DASH

| "false" EXP\_DASH  
| Identifier EXP\_DASH  
| "this" EXP\_DASH  
| "new" NEW\_DASH EXP\_DASH  
| "!" Expression EXP\_DASH  
| "(" Expression ")" EXP\_DASH

EXP\_DASH -> X EXP\_DASH | lamda

X -> ( "&&" | "||" | "==" | "!=" | ">" | "<" |"<=" | ">="| "+" | "-" | "\*" | "/" ) Expression

| "[" Expression "]"

| ”.” DOT\_DASH

DOT\_DASH -> "length" | Identifier "(" ( Expression ( "," Expression )\* )? ")"

NEW\_DASH -> Identifier "(" (Expression ( "," Expression)\*)? ")"

| ("int" | "float" | "String" | "char" | "boolean" ) "[" Expression "]"

Identifier ::= <IDENTIFIER>