

Goal ::= MainClass (ClassDeclaration)* <EOF>

MainClass ::= "class" Identifier "{" "public" "static" "void" "main" "("
"String" "[" "]" Identifier ")" "{" Statement "}" "}"

ClassDeclaration ::= "class" Identifier ("extends" Identifier)? "{" (VarDeclaration)* (MethodDeclaration)* "}"

VarDeclaration ::= Type Identifier ";"

MethodDeclaration ::= ("public" | "private") Type Identifier "(" (Type
Identifier ("," Type Identifier)*
)? ")" "{" (VarDeclaration)* (Statement)* "return" Expression ";" "}"

Type ::= "boolean" bracket
| "int" bracket
| "float" bracket
| "String" bracket
| "char" bracket

bracket ::= "[" "]" | lambda

Statement ::= "{" (Statement)* "}"
| "if" "(" Expression ")" Statement IfRest
| "while" "(" Expression ")" Statement
| "System.out.println" "(" Expression ")" ";"
| Identifier Statement Rest

IfRest ::= "else" Statement

Statement Rest ::= "=" Expression ";" | "[" Expression "]" "=" Expression ";"

Expression ::= ("true" | "false" | "this" | Identifier | "new"
ExpressionRestNew | <INTEGER_LITERAL> | "!" Expression | "(" Expression ")")
ExpressionBar

ExpressionBar ::= ExpressionRest ExpressionBar | lambda

Identifier ::= <IDENTIFIER>

ExpressionRest ::= ("&&" | "<" | "+" | "-" | "*") Expression
| "[" Expression "]"
| "." DotRest

ExpressionRestNew ::= "int" "[" Expression "]"
| Identifier "(" ")"

DotRest ::= "length"
| Identifier "(" (Expression ("," Expression)*)? ")"