ClassDeclaration -> "class" id "{" ClassVarDecList SubrutineDecList "}"

ClassVarDecList -> ClassVarDec ClassVarDecList |

SubrutineDecList -> SubrutineDec SubrutineDecList |

ClassVarDec -> classVarDescriptor Type idList ";"

classVarDescriptor -> "static" | "field"

idList -> id idListCont

idListCont -> "," idList |

Type -> primitiveType | id

SubrutineDec -> SubrutineDescriptor ReturnType id "(" ParameterList ")" SubrutineBody

SubrutineDescriptor -> "constructor" | "function" | "method"

ReturnType -> "void" | Type

ParameterList -> ParamDec ParamListCont |

ParamListCont -> "," ParameterList |

ParamDec -> Type id

SubrutineBody -> "{" VarDecList Statements "}"

VarDecList -> VarDeclaration VarDecList |

VarDeclaration -> "var" Type idList ";"

Statements -> Statement Statements |

Statement -> LetStatement|DoStatement|IfStatement|WhileStatement|ReturnStatement

LetStatement -> "let" id AtIndex "=" Expression ";"

AtIndex -> "[" Expression "]" |

DoStatement -> "do" SubrutineCall ";"

IfStatement -> "if" "(" Expression ")" "{" Statements "}" ElseStatement

ElseStatement -> "else" "{" Statements "}" |

WhileStatement -> "while" "(" Expression ")" "{" Statements "}"

ReturnStatement -> "return" ExpressionOrVoid ";"

SubrutineCall -> NamePath ArgsList

ExpressionOrVoid -> Expression |

Expression -> Operand RightSide

RightSide -> Operator Expression |

Operand -> integerConstant | stringLiteral | keywordConstant | idOrFunction | "(" Expression ")" | unaryOperator Operand

idOrFunction -> id maybeFunction

maybeFunction -> ArgsList | "." NamePath ArgsList | AtIndex |

ArgsList -> "(" ExpressionListOrEmpty ")"

ExpressionListOrEmpty -> ExpressionList |

NamePath -> id NamePathCont

NamePathCont -> "." NamePath |

ExpressionList -> Expression ExpressionListCont

ExpressionListCont -> "," Expression ExpressionList |

Operator -> "+" | "-" | "\*" | "/" | "&" | "|" | "<" | ">" | "="

unaryOperator -> "-" | "~"

keywordConstant -> "true" | "false" | "this" | "null"