

Program → ClassDeclaration *

ClassDeclaration → **class IDENT** { ClassMember* }

ClassMember → Field | Method | MainMethod

Field → **public** Type **IDENT** ;

MainMethod → **public static void IDENT (String [] IDENT)** Block

Method → **public** Type **IDENT (Parameters?)** Block

Parameters → Parameter | Parameter , Parameters

Parameter → Type **IDENT**

Type → BasicType ([])*

BasicType → **int** | **boolean** | **void** | **IDENT**

Statement → Block | EmptyStatement | IfStatement | ExpressionStatement | WhileStatement
| ReturnStatement

Block → { BlockStatement* }

BlockStatement → Statement | LocalVariableDeclarationStatement

LocalVariableDeclarationStatement → Type **IDENT** (= Expression)? ;

EmptyStatement → ;

WhileStatement → **while** (Expression) Statement

IfStatement → **if** (Expression) Statement (**else** Statement) ?

ExpressionStatement → Expression ;

ReturnStatement → **return** Expression? ;

Expression → AssignmentExpression

AssignmentExpression → LogicalOrExpression (= AssignmentExpression) ?

LogicalOrExpression → LogicalAndExpression (|| LogicalAndExpression)*

LogicalAndExpression → EqualityExpression (&& EqualityExpression)*

EqualityExpression → RelationalExpression ((== | !=) RelationalExpression)*

RelationalExpression → AdditiveExpression ((< | <= | > | >=) AdditiveExpression)*

AdditiveExpression → MultiplicativeExpression ((+ | -) MultiplicativeExpression)*

MultiplicativeExpression → UnaryExpression ((* | / | %) UnaryExpression)*

UnaryExpression → PostfixExpression | (! | -) UnaryExpression

PostfixExpression → PrimaryExpression (PostfixOp)*

PostfixOp → MethodInvocation | FieldAccess | ArrayAccess

MethodInvocation → . **IDENT** (Arguments)

FieldAccess → . **IDENT**

ArrayAccess → [Expression]

Arguments → (Expression (, Expression) *) ?

PrimaryExpression → **null** | **false** | **true** | **INTEGER_LITERAL** | **IDENT** | **IDENT** (Arguments) | **this** | (Expression) | NewObjectExpression | NewArrayExpression

NewObjectExpression → **new IDENT** ()

NewArrayExpression → **new** BasicType [Expression] ([])*