Syntax for MiniJava

Notes:
Symbols in **blue** are tokens.
This grammar is *not* unambiguous!
Some work will be needed to avoid conflicts.

program	→	classDecl$^+$
classDecl	→	**class ID** (**extends ID**)$^?$ **{** (instVarDecl | methodDecl)* **}**
instVarDecl	→	type **ID ;**
localVarDecl	→	type **ID** = exp **;**
methodDecl	→	**public** type **ID (** formalList$^?$ **) { ** stmtDecl* **return** exp **; }**
methodDecl	→	**public void ID (** formalList$^?$ **) { ** stmtDecl* **}**
formalList	→	type **ID** (**,** type **ID**)*
baseType	→	**ID**
baseType	→	**int**
baseType	→	**boolean**
type	→	baseType
type	→	type **[ ]**
stmt	→	**{** stmtDecl* **}**
stmt	→	**while (** exp **)** stmt
stmt	→	**if (** exp **)** stmt (**else** stmt)$^?$
stmt	→	**for (** (type **ID** = exp | assign | callExp)$^?$ **;** exp$^?$ **;** (assign | callExp)$^?$ **)** stmt
stmt	→	**switch (** exp **) { ** (stmtDecl | **case** exp **:** | **default :**)* **}**
stmt	→	assign **;**
stmt	→	callExp **;**
stmt	→	**break;**
stmt	→	**;**
stmtDecl	→	stmt | localVarDecl
assign	→	exp1 = exp
assign	→	**ID** ++
assign	→	**ID - -**
assign	→	++ **ID**
assign	→	**- - ID**

Precedence rules (lowest to highest)
8. ||
7. &&
6. ==, !=
5. <, <=, >, >=, instanceof
4. +, -
3. *, /, %
2. unary +, unary -, casting
1. exp1

exp	→	exp || exp	exp	→	exp && exp
exp	→	exp == exp	exp	→	exp != exp
exp	→	exp < exp	exp	→	exp <= exp
exp	→	exp > exp	exp	→	exp >= exp
exp	→	exp + exp	exp	→	exp - exp
exp	→	exp * exp	exp	→	exp / exp
exp	→	exp % exp	exp	→	! exp
exp	→	+ exp	exp	→	- exp
exp	→	(type) exp	exp	→	exp **instanceof** type
exp	→	exp1			
exp1	→	**ID**	exp1	→	callExp
exp1	→	**INTLIT**	exp1	→	**STRINGLIT**
exp1	→	**CHARLIT**	exp1	→	**false**
exp1	→	**true**	exp1	→	**null**
exp1	→	**this**	exp1	→	exp1 [exp]
exp1	→	exp1 . **ID**	exp1	→	(exp)
exp1	→	**new ID** ( )	exp1	→	**new** baseType [exp] ([ ])*

callExp	→	**ID** (expList?)
callExp	→	exp1 . **ID** (expList?)
callExp	→	**super** . **ID** (expList?)
expList	→	exp (, exp)*