FloaJay

$$
\begin{array}{rcl}
\mathit{Type} & \longrightarrow & \texttt{int} \mid \texttt{boolean} \mid \underline{\texttt{float}} \\
\mathit{Term} & \longrightarrow & \underline{\mathit{Cast}}\ \{\ \mathit{MulOp}\ \underline{\mathit{Cast}}\ \}\ * \\
\mathrm{MulOp} & \longrightarrow & \text{`}*\text{'} \mid / \\
\underline{\mathit{Cast}} & \longrightarrow & \underline{(\ \mathit{Type}\ )\ \mathrm{Negation}}
\end{array}
$$

FunJay

Program	$\rightarrow$	`public class` ID '{' *GlobalDeclarations* `public static void main ( String[] args )` '{' *Declarations Statements* '}' *Procedures* '}'
GlobalDeclarations	$\rightarrow$	*GlobalDeclartion* $*$
GlobalDeclaration	$\rightarrow$	`static` *Type Identifiers* ;
Declarations	$\rightarrow$	*Declartion* $*$
Declaration	$\rightarrow$	*Type Identifiers* ;
Type	$\rightarrow$	`int` | `boolean`
Identifiers	$\rightarrow$	ID { , ID } $*$
Statements	$\rightarrow$	*Statement* $*$
Statement	$\rightarrow$	; | *Block* | *Assignment* | *IfStatement* | *WhileStatement* | *PrintStatement* | *CallStatement* | *ReturnStatement*
Block	$\rightarrow$	'{' *Declarations Statements* '}'
Assignment	$\rightarrow$	ID = *Expression* ;

FunJay

IfStatement	$\rightarrow$	`if` (*Expression*) *Statement* { `else` *Statement* }
WhileStatement	$\rightarrow$	`while` (*Expression*) *Statement*
PrintStatement	$\rightarrow$	`System.out.println` (*Expression*) ;
CallStatement	$\rightarrow$	*Identifier* (*ActParams*?) ;
ActParams	$\rightarrow$	*Expression* { , *Expression* } *
ReturnStatement	$\rightarrow$	`return` *Expression*? ;
Expression	$\rightarrow$	*Conjunction* { `||` *Conjunction* }*
Conjunction	$\rightarrow$	*Relation* { `&&` *Relation* }*
Relation	$\rightarrow$	*Addition* { *RelOp* *Addition* }?
RelOp	$\rightarrow$	`<` | `<=` | `>` | `>=` | `==` | `!=`
Addition	$\rightarrow$	*Term* { *AddOp* *Term* } *
AddOp	$\rightarrow$	`+` | `-`
Term	$\rightarrow$	*Negation* { *MulOp* *Negation* } *
MulOp	$\rightarrow$	'*' | /
Negation	$\rightarrow$	*NegOp*? *Factor*
NegOp	$\rightarrow$	`!` | `-`
Factor	$\rightarrow$	ID | LITERAL | (*Expression*) | *Call*

FunJay

Call	$\longrightarrow$	*Identifier* (*ActParams* ?)
Procedures	$\longrightarrow$	*Procedure* *
Procedure	$\longrightarrow$	`static` *ReturnType* ID (*FormParams*?)
		'{' *Declarations Statements* '}'
ReturnType	$\longrightarrow$	*Type* | `void`
FormParams	$\longrightarrow$	*FormParam* { , *FormParam* } *
FormParam	$\longrightarrow$	*Type* ID