Jay Lexical Specification

Identifiers	[a-z_][a-zA-Z0-9_]∗
capitalized identifiers	[A-Z_][a-zA-Z0-9_]∗
Integer literals	$0\mid[1-9][0-9]*$
Boolean literals	true false
Separators	( ) { } ; ,
Operators	= > < ! == <= >= != || && + - ∗ /
Keywords	public class static String[] args
	void main System.out.println
	boolean else if int while

Jay Concrete Syntax

$$
\begin{array}{rcl}
\mathit{Program} & \rightarrow & \texttt{public class}\ \mathsf{CAPID}\ `\{' \\
 & & \texttt{public static void main (String[] args)} \\
 & & `\{'\ \mathit{Declarations}\ \mathit{Statements}\ `\}'\ `\}' \\
\mathit{Declarations} & \rightarrow & \mathit{Declartion}\ * \\
\mathit{Declaration} & \rightarrow & \mathit{Type}\ \mathit{Identifiers}\ ; \\
\mathit{Type} & \rightarrow & \texttt{int}\ \mid \texttt{boolean} \\
\mathit{Identifiers} & \rightarrow & \mathsf{ID}\ \{\ ,\ \mathsf{ID}\ \}\ * \\
\mathit{Statements} & \rightarrow & \mathit{Statement}\ * \\
\mathit{Statement} & \rightarrow & \mathit{Skip}\ \mid \mathit{Block}\ \mid \mathit{Assignment}\ \mid \mathit{IfStatement} \\
 & & \mid \mathit{WhileStatement} \mid \mathit{PrintStatement} \\
\mathit{Skip} & \rightarrow & ; \\
\mathit{Block} & \rightarrow & `\{'\ \mathit{Statements}\ `\}' \\
\mathit{Assignment} & \rightarrow & \mathsf{ID}\ =\ \mathit{Expression}\ ; \\
\mathit{IfStatement} & \rightarrow & \texttt{if}\ (\ \mathit{Expression}\ )\ \mathit{Statement}\ \{\ \texttt{else}\ \mathit{Statement}\ \}? \\
\mathit{WhileStatement} & \rightarrow & \texttt{while}\ (\ \mathit{Expression}\ )\ \mathit{Statement} \\
\mathit{PrintStatement} & \rightarrow & \texttt{System.out.println}\ (\ \mathit{Expression}\ )\ ;
\end{array}
$$

Jay Concrete Syntax, continued

$$
\begin{array}{rcl}
\mathit{Expression} & \rightarrow & \mathit{Conjunction}\ \{\ \texttt{||}\ \mathit{Conjunction}\ \}* \\
\mathit{Conjunction} & \rightarrow & \mathit{Relation}\ \{\ \texttt{\&\&}\ \mathit{Relation}\ \}* \\
\mathit{Relation} & \rightarrow & \mathit{Addition}\ \{\ \mathit{RelOp}\ \mathit{Addition}\ \}? \\
\mathit{RelOp} & \rightarrow & \texttt{<}\ \mid \texttt{<=}\ \mid \texttt{>}\ \mid \texttt{>=}\ \mid \texttt{==}\ \mid \texttt{!=} \\
\mathit{Addition} & \rightarrow & \mathit{Term}\ \{\ \mathit{AddOp}\ \mathit{Term}\ \}\ * \\
\mathrm{AddOp} & \rightarrow & \texttt{+}\ \mid \texttt{-} \\
\mathit{Term} & \rightarrow & \mathit{Negation}\ \{\ \mathit{MulOp}\ \mathit{Negation}\ \}\ * \\
\mathrm{MulOp} & \rightarrow & `*'\ \mid \texttt{/} \\
\mathit{Negation} & \rightarrow & \mathit{NegOp}?\ \mathit{Factor} \\
\mathit{NegOp} & \rightarrow & \texttt{!}\ \mid \texttt{-} \\
\mathit{Factor} & \rightarrow & \mathrm{ID}\ \mid \mathrm{LITERAL}\ \mid (\ \mathit{Expression}\ )
\end{array}
$$

Jay Abstract Syntax

$$
\begin{array}{rcl}
\mathit{Program} & \rightarrow & \mathit{Declaration}{*}\ \mathit{Statement} \\
\mathit{Declaration} & \rightarrow & \mathit{Type}\ \mathrm{ID}{*} \\
\mathit{Statement} & \rightarrow & \mathit{Skip}\ \mid \mathit{Block}\ \mid \mathit{Assignment}\ \mid \mathit{Conditional} \\
 & & \mid \mathit{Loop}\ \mid \mathit{Print} \\
\mathit{Block} & \rightarrow & \mathit{Statement}{*} \\
\mathit{Assignment} & \rightarrow & \mathrm{ID}\ \mathit{Expression} \\
\mathit{Conditional} & \rightarrow & \mathit{Expression}\ \mathit{Statement}\ \mathit{Statement} \\
\mathit{Loop} & \rightarrow & \mathit{Expression}\ \mathit{Statement} \\
\mathit{Print} & \rightarrow & \mathit{Expression} \\
\mathit{Expression} & \rightarrow & \mathit{Variable}\ \mid \mathit{IntLitExpr}\ \mid \mathit{BoolLitExpr} \\
 & & \mid \mathit{BinaryExpr}\ \mid \mathit{UnaryExpr} \\
\mathit{Variable} & \rightarrow & \mathrm{ID} \\
\mathit{IntLitExpr} & \rightarrow & \mathrm{INT\_LIT} \\
\mathit{BoolLitExpr} & \rightarrow & \mathrm{BOOL\_LIT} \\
\mathit{BinaryExpr} & \rightarrow & \mathit{Expression}\ \mathrm{OPERATOR}\ \mathit{Expression} \\
\mathit{UnaryExpr} & \rightarrow & \mathrm{OPERATOR}\ \mathit{Expression}
\end{array}
$$