Context Free Grammar for COOL Compiler

$$
\begin{array}{rcl}
\textit{program} & ::= & [\![\textit{class};]\!]^+ \\
\textit{class} & ::= & \textbf{class}\ \text{TYPE}\ [\textbf{inherits}\ \text{TYPE}]\ \{\ [\![\textit{feature};]\!]^*\} \\
\textit{feature} & ::= & \text{ID}(\ [\ \textit{formal}\ [\![, \textit{formal}]\!]^*\ ]\ ) : \text{TYPE}\ \{\ \textit{expr}\ \} \\
 & | & \text{ID} : \text{TYPE}\ [\ \texttt{<-}\ \textit{expr}\ ] \\
\textit{formal} & ::= & \text{ID} : \text{TYPE} \\
\textit{expr} & ::= & \text{ID}\ \texttt{<-}\ \textit{expr} \\
 & | & \textit{expr}[@\text{TYPE}].\text{ID}(\ [\ \textit{expr}\ [\![, \textit{expr}]\!]^*\ ]\ ) \\
 & | & \text{ID}(\ [\ \textit{expr}\ [\![, \textit{expr}]\!]^*\ ]\ ) \\
 & | & \textbf{if}\ \textit{expr}\ \textbf{then}\ \textit{expr}\ \textbf{else}\ \textit{expr}\ \textbf{fi} \\
 & | & \textbf{while}\ \textit{expr}\ \textbf{loop}\ \textit{expr}\ \textbf{pool} \\
 & | & \{\ [\![\textit{expr};]\!]^+\} \\
 & | & \textbf{let}\ \text{ID} : \text{TYPE}\ [\ \texttt{<-}\ \textit{expr}\ ]\ [\![, \text{ID} : \text{TYPE}\ [\ \texttt{<-}\ \textit{expr}\ ]]\!]^*\ \textbf{in}\ \textit{expr} \\
 & | & \textbf{case}\ \textit{expr}\ \textbf{of}\ [\![\text{ID} : \text{TYPE} \Rightarrow \textit{expr};]\!]^+\textbf{esac} \\
 & | & \textbf{new}\ \text{TYPE} \\
 & | & \textbf{isvoid}\ \textit{expr} \\
 & | & \textit{expr} + \textit{expr} \\
 & | & \textit{expr} - \textit{expr} \\
 & | & \textit{expr} * \textit{expr} \\
 & | & \textit{expr} / \textit{expr} \\
 & | & \tilde{}\,\textit{expr} \\
 & | & \textit{expr} < \textit{expr} \\
 & | & \textit{expr} <= \textit{expr} \\
 & | & \textit{expr} = \textit{expr} \\
 & | & \textbf{not}\ \textit{expr} \\
 & | & (\textit{expr}) \\
 & | & \text{ID} \\
 & | & \text{integer} \\
 & | & \text{string} \\
 & | & \textbf{true} \\
 & | & \textbf{false}
\end{array}
$$