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

Figure 1: Cool syntax.