

Grammar

Implemented: (sort of)

$$\begin{aligned}[prog] &\rightarrow [stmt]^* \\[stmt] &\rightarrow \begin{cases} \{[type]\} \langle ident \rangle = [expr]; & \text{variable (re)assignment} \\ ret [expr]; & \text{exit/return} \end{cases} \\[expr] &\rightarrow \begin{cases} [term] \\ [binexpr] \\ -[expr] & \text{negative expression (underscore)} \end{cases} \\[binexpr] &\rightarrow \begin{cases} [expr] + [expr] & prec = 0 \\ [expr] - [expr] & prec = 0 \\ [expr] * [expr] & prec = 1 \\ [expr] / [expr] & prec = 1 \end{cases} \\[term] &\rightarrow \begin{cases} \langle int_lit \rangle \\ \langle ident \rangle \end{cases} \\[type] &\rightarrow \{int\end{aligned}$$

Todo:

$$\begin{aligned} [prog] &\rightarrow [stmt]^* \\ [stmt] &\rightarrow \begin{cases} ret \ [expr]; \\ [type] \ ident = [expr]; \\ if([expr])[stmt] \\ [scope] \end{cases} \\ [scope] &\rightarrow \{[stmt]^*\} \\ [expr] &\rightarrow \begin{cases} [binexpr] \\ [term] \end{cases} \\ [binexpr] &\rightarrow \begin{cases} [expr] * [expr] & prec = 1 \\ [expr] / [expr] & prec = 1 \\ [expr] + [expr] & prec = 0 \\ [expr] - [expr] & prec = 0 \end{cases} \\ [term] &\rightarrow \begin{cases} int_lit \\ ident \\ ([Expr]) \\ TRUE \\ FALSE \\ 'c' \end{cases} \\ [type] &\rightarrow \begin{cases} int \\ bool \\ char \\ float \end{cases} \end{aligned}$$