

Cool: Gramática

```
program ::= [class ;]+
class    ::= class TYPE [inherits TYPE] { [feature ;]* }
feature ::= ID( [formal [, formal]* ) : TYPE { expr }
           ID : TYPE [<- expr]
formal   ::= ID : TYPE
expr     ::= ID <- expr
           expr[@ TYPE].ID ( [expr [, expr]* ] )
           ID ( [expr [, expr]* ] )
           if expr then expr else expr fi
           while expr loop expr pool
           { [expr;]+ }
           let ID : TYPE [<- expr] [, ID : TYPE [<- expr]]* in expr
           case expr of [ID : TYPE => expr ;]+ esac
           new TYPE
           isvoid expr
           expr + expr
           expr - expr
           expr * expr
           expr / expr
           expr
           expr < expr
           expr <= expr
           expr = expr
           not expr
           ( expr )
           ID
           INTEGER
           STRING
           true
           false
```
