

Grammar Productions

empty

July 31, 2015

1 Grammar Productions

```
program ::= classDecl*
classDecl ::= class id extends id '{' (fieldDecl | methDecl)* '}'
fieldDecl ::= type id '(' id)* ';'
methDecl ::= (type | void) id '(' ([formals])' block
formals ::= type id '(' type id)*
type ::= int | boolean | string
block ::= '{' varDecl* stmt* '}'
varDecl ::= type id ';'
stmt ::= skip ';'
      | id '=' expr ';'
      | location '=' expr ';'
      | call ';'
      | return [expr] ';'
      | if '(' expr ')' then block else block
      | while '(' expr ')' block
location ::= expr '.' id
call ::= id '(' [actuals] ')'
      | location '(' [actuals] ')'
actuals ::= expr '(' expr)*
expr ::= id
      | location
      | call
      | this
      | new id '(' ')'
      | new type '[' expr ']'
      | expr binary expr
      | unary expr
      | literal
      | '(' expr ')'
binary ::= '+' | '-' | '*' | '/' | '&&' | '||'
      | '<' | '<=' | '>' | '>=' | '==' | '!='
unary ::= '-' | '!'
literal ::= integer - literal | string - literal | true | false
```