Mini Java Grammar

Leela Pakanati, Manoj Kurapati

Program ::= MainClassDecl ClassDeclList

MainClassDecl::= class ID { public static void main (String[ ] ID) { StmtList } }

ClassDeclList = ClassDecl ClassDeclList
        | ε

ClassDecl ::= class ID [extends ID] { ClassVarDeclList MethodDeclList }

ClassVarDeclList = ClassVarDecl ClassVarDeclList
        | ε

ClassVarDecl ::= Formal;

MethodDeclList = MethodDecl MethodDeclList
        | ε

MethodDecl ::= public Formal ([Formal FormalList]) { StmtList return Expr; }

FormalList = , Formal FormalList
        | ε

Formal ::= Type ID

Type ::= int IsArr | boolean | ID

IsArr ::= [ ] | ε

StmtList = Stmt StmtList
        | ε

Stmt ::= if (Expr) BracketStmt else BracketStmt
        | while (Expr) BracketStmt
        | AtomicStmt

BracketStmt ::= { AtomicStmtList }
        | AtomicStmt

AtomicStmtList = AtomicStmt AtomicStmtList
        | ε

AtomicStmt ::= System.out.println (Expr);
        | ID IsArrVal = Expr;

IsArrVal ::= [Expr]
          | ϵ

ExprList = , Expr ExprList
          | ϵ

Expr ::= OrOperand Expr'

Expr' ::= || OrOperand Expr'
          | ϵ

OrOperand ::= AndOperand OrOperand'

OrOperand' ::= && AndOperand OrOperand'
          | ϵ

AndOperand ::= EqualityOperand AndOperand'

AndOperand' ::= == EqualityOperand AndOperand'
          | != EqualityOperand AndOperand'
          | ϵ

EqualityOperand ::= RelOperand EqualityOperand'

EqualityOperand' ::= < RelOperand EqualityOperand'
          | <= RelOperand EqualityOperand'
          | >= RelOperand EqualityOperand'
          | > RelOperand EqualityOperand'
          | ϵ

RelOperand ::= AddOperand RelOperand'

RelOperand' ::= + AddOperand RelOperand'
          | - AddOperand RelOperand'
          | ϵ

AddOperand ::= MulOperand AddOperand'

AddOperand' ::= * MulOperand AddOperand'
          | / MulOperand AddOperand'
          | ϵ

Value ::= new NewCons
          | AtomicValue

AtomicValue ::= ID DotExpr
          | this DotExpr

```
        | Integer
        | null
        | true
        | false
        | ! AtomicValue
        | - AtomicValue
        | new NewWhat


NewWhat ::= ID ( ) DotExpr
        | int [ Expr ]


DotExpr ::= . DotWhat
        | ε


DotWhat ::= ID ( [Expr ExprList] )
        | length
```