

*Program:*

*ProgramDeclaration ClassDeclarations*

*ProgramDeclaration:*

*PROGRAM COLON ClassName ProgramArguments*

*ClassName:*

*IDENTIFIER*

*| IDENTIFIER LEFT\_SQUARE\_BRACKET ClassName*

*RIGHT\_SQUARE\_BRACKET*

*ProgramArguments:*

*/empty/*

*| LEFT\_PAREN Literals RIGHT\_PAREN*

*Literals:*

*| Literal*

*| Literal COMMA Literals*

*Literal:*

*INTEGER*

*| REAL*

*| BOOLEAN*

*Arguments:*

*/empty/*

*| LEFT\_PAREN Expressions RIGHT\_PAREN*

*Expressions:*

*| Expression*

*| Expression COMMA Expressions*

*Expression:*

*Primary*

*| CompoundExpression*

*Primary:*

- INTEGER
- | REAL
- | BOOLEAN
- | THIS
- | ClassName

*CompoundExpression:*

- IDENTIFIER
- | IDENTIFIER LEFT\_PAREN *Arguments* RIGHT\_PAREN
- | IDENTIFIER DOT *CompoundExpression*
- | IDENTIFIER LEFT\_PAREN *Arguments* RIGHT\_PAREN DOT  
*CompoundExpression*

*ClassDeclarations:*

- ClassDeclaration*
- | *ClassDeclaration ClassDeclarations*

*ClassDeclaration:*

- CLASS *ClassName Extension* IS *ClassBody* END

*Extension:*

- /empty/
- | EXTENDS *ClassName*

*ClassBody:*

- MemberDeclarations*

*MemberDeclarations:*

- /empty/
- | *MemberDeclaration*
- | *MemberDeclaration MemberDeclarations*

*MemberDeclaration:*

- VariableDeclaration*
- | *MethodDeclaration*
- | *ConstructorDeclaration*

*VariableDeclaration:*

VAR *VariableName* COLON *Expression*

*VariableName:*

IDENTIFIER

*MethodDeclaration:*

METHOD *MethodName* *Parameters* *ReturnType* IS *Body*  
END

*MethodName:*

IDENTIFIER

*Parameters:*

/empty/  
| *Parameter*  
| *Parameter* COMMA *Parameters*

*Parameter:*

IDENTIFIER COLON *ClassName*

*ReturnType:*

/empty/  
| *ClassName*

*ConstructorDeclaration:*

THIS *Parameters* IS *Body* END

*Body:*

*BodyDeclaration*  
| *BodyDeclaration* *BodyDeclarations*

*BodyDeclaration:*

*VariableDeclaration*  
| *Statement*

*Statement:*

*Assignment*  
*| WhileLoop*  
*| IfStatement*  
*| ReturnStatement*

*Assignment:*

*VariableName COLON\_EQUAL Expression*

*WhileLoop:*

*WHILE Expression LOOP Body END*

*IfStatement:*

*IF Expression IfBranch ElseBranch END*

*IfBranch:*

*THEN Body*

*ElseBranch:*

*/empty/*  
*| ELSE Body*

*ReturnStatement:*

*RETURN*  
*| RETURN Expression*