Phrase Structure

<Program> ::= <ImportList> class IDENT <Block>

<ImportList> ::=(import IDENT (. IDENT)* ;) *

<Block> ::= { (<Declaration> ; | <Statement> ;)* }

<Declaration> ::= def <VarDec> | def <ClosureDec>

<VarDec> ::= IDENT (: <Type> | ϵ)

<Type> ::= <SimpleType> | <KeyValueType> | <ListType>

<SimpleType> ::= int | boolean | string

<KeyValueType> ::= @@ [<SimpleType> : <Type>]

<ListType> ::= @ [<Type>]

<ClosureDec> ::= IDENT = <Closure>

<Closure> ::= { <FormalArgList> - > <Statement> *}

<FormalArgList> ::= ϵ | <VarDec> (, <VarDec>)*

<Statement> ::= <LValue> = <Expression>
| print <Expression>
| while (<Expression>) <Block>
| while* (<Expression>) <Block>
| while* (<RangeExpression>) < Block>
| if (<Expression>) <Block>
| if (<Expression>) <Block> else <Block>
| %<Expression>
| return <Expression>
| ϵ

<ClosureEvalExpression> ::= IDENT (<ExpressionList>)

<LValue> ::= IDENT | IDENT [<Expression>]

<List> ::=@ [<ExpressionList> |

<ExpressionList> ::= ϵ | <Expression> (, <Expression>)*

<KeyValueExpression> ::= <Expression> : <Expression>

<KeyValueList> ::= ϵ | <KeyValueExpression> (, <KeyValueExpression>) *

<MapList> ::= @@[<KeyValueList>]

<RangeExpr> :: <Expression> .. <Expression>

<Expression> ::= <Term> (<RelOp> <Term>)*

<Term> ::= <Elem> (<WeakOp> <Elem>)*

<Elem> ::= <Thing> (<StrongOp> <Thing>)*

<Thing> ::= <Factor> (<VeryStrongOp> <Factor)*

<Factor>::= IDENT | IDENT [<Expression>] | INT_LIT | true | false | STRING_LIT
| (<Expression>) | ! <Factor> | -<Factor> | size(<Expression>) |
key(<Expression) | value(<Expression >) | <ClosureEvalExpression> | <Closure> |
<List> | <MapList>

<RelOp> ::= | | & | == | != | < | > | ≤ | ≥

<WeakOp> ::= + | -

<StrongOp> ::= * | /

<VeryStrongOp> ::= ≪ | ≫