Program = Global\*; Staments

Global = FunctionDeclaration | Declarations

FunctionDeclaration = Type; String id; ParamDeclaration\*; Statements

ParamDeclaration = Type; String Id

Statements = Declaration\*; Statement\*

Declaration = Init\*

Init = ArrayInit | NoArrayInit

ArrayInit = Type; String id; int size; (Expression initList)\*

NoArrayInit = Type; String id; (Expression init)?

Statement = IfSatement | Block | WhileSatement | SwitchStatement | ForStatement | Return | Expression | Break | Continue | Skip

IfStatement = Expression condition; Block; (Expression elseif; Block)\*; Block?

Block = Statement\*

WhileStatement = Expression condition; Block

SwitchStatement = Expression condition; (Literal case; Statement\*)\*; (Statement\*)?

ForStatement = Expression\*; Expression; Expression\*; Block

Return = Expression?

Break =

Continue =

Skip =

Expression = VariableRef | Value | Binary | Unary | Assignments | Function | TypeCast

VariableRef = Variable | ArrayRef

Variable = String id

ArrayRef = String id; Expression index

Value = IntValue | BoolValue | FloatValue | CharValue | TimeValue | DateValue

Binary = Operator; Expression e1, e2

Unary = Operator; Expression

Assignments = Assignment | ArrayAssign

Assignment = String Id; Operator; Expression

ArrayAssign = String Id; Expression index; Operator; Expression

Function = String name; Expression\*

TypeCast = Type type; Expression

Type = ‘int’ | ‘bool’ | ‘void’ | ‘char’ | ‘float’ | ‘time’ | ‘date’

Operator = BooleanOp | RelationalOp | ArithmeticOp | UnaryOp | AssignOp

RelationalOp = ‘==’ | ‘!=’ | ‘<’ | ‘<=’ | ‘>’ | ‘>=’

ArithmeticOp = ‘+’ | ‘-‘ | ‘\*’ | ‘/’ | ‘%’ | ‘++’ | ‘—‘

RelationalOp = ‘&&’ | ‘||’

UnaryOp = ‘!’ | ‘-‘

AssignOp = ‘+=’ | ‘-=’ | ‘\*=’ | ‘/=’ | ‘%=’ | ‘=’

IntValue = int

BoolValue = bool

CharValue = String

FloatValue = float

DateValue = String year, month, day

TimeValue = String hour, minute, second