

Tran = {Interface } | [Class]

Interface = "interface" Identifier NEWLINE INDENT {MethodHeader NEWLINE } DEDENT

MethodHeader = Identifier "(" VariableDeclarations ")" [":" VariableDeclaration { ";"
VariableDeclaration }]

VariableDeclarations = [VariableDeclaration] | VariableDeclaration { ";"
VariableDeclaration }

VariableDeclaration = Identifier Identifier

Class = "class" Identifier ["implements" Identifier { ";" Identifier }] NEWLINE INDENT {
Constructor NEWLINE | MethodDeclaration NEWLINE | Member NEWLINE } DEDENT

Constructor = "construct" "(" VariableDeclarations ")" NEWLINE MethodBody

Member = VariableDeclaration ["accessor:" Statements] ["mutator:" Statements]

MethodDeclaration = ["private"] ["shared"] MethodHeader NEWLINE MethodBody

MethodBody = INDENT { VariableDeclaration NEWLINE } {Statement} DEDENT

Statements = INDENT {Statement NEWLINE } DEDENT

Statement = If | Loop | MethodCall | Assignment

If = "if" BoolExp NEWLINE Statements ["else" NEWLINE (Statement | Statements)]

BoolExpTerm = BoolExpFactor { ("and" | "or") BoolExpTerm } | "not" BoolExpTerm

BoolExpFactor = MethodCallExpression | (Expression ("==" | "!=" | "<=" | ">=" | ">" | "<")
Expression) | VariableReference

Loop = [VariableReference "="] "loop" (BoolExpTerm) NEWLINE Statements

Assignment = VariableReference "=" Expression

MethodCall = [VariableReference { ";" VariableReference } "=" MethodCallExpression

MethodCallExpression = [Identifier "."] Identifier "(" [Expression { ";" Expression }] ")"

Expression = Term { ("+" | "-") Term }

Term = Factor { ("*" | "/" | "%") Factor }

Factor = NumberLiteral | VariableReference | "true" | "false" | StringLiteral | CharacterLiteral
| MethodCallExpression | "(" Expression ")" | "new" Identifier "(" [Expression {" " Expression }]
")"

VariableReference = Identifier

NumberLiteral = { 0-9 } [. { 0-9 } }

StringLiteral = " { any non- " } "

CharacterLiteral = ' (one character not a ') '

Identifier = A-z { A-z 0-9 }