denotational semantics:

<STMT>:
[[<STMT>]]: Stmt → Env → Env
[[<STMT>]] = [[<IF_STMT>]] | [[<BLOCK>]] | [[<ASSIGN>]] | [[<DECLARE>]] |
[[<WHILE_LOOP>]]

<STMT_LIST>:
[[<STMT_LIST>]]: List<Stmt> → Env → Env

<WHILE_LOOP>:
[[<WHILE_LOOP>]]: BExpr × Stmt → Env → Env
[while (b) S] = if ([b] = true) then [[S]]([while (b) S]) else env

<IF_STMT>:
[[<IF_STMT>]]: BExpr × Stmt × Stmt → Env → Env
[if (b) S1 else S2] = if ([b] = true) then [S1] else [S2]

<BLOCK>:
[[<BLOCK>]]: List<Stmt> → Env → Env
[{SL}] = [<STMT_LIST>]

<DECLARE>:
[[<DECLARE>]]: DataType × List<ID> → Env → Env
[DataType ID1, ID2, ...] = env[ID1 ← DefaultValue(DataType), ID2 ← DefaultValue(DataType),
...]

<ASSIGN>:
[[<ASSIGN>]]: ID × Expr → Env → Env
[ID = E] = env[ID ← [E]]

<EXPR>:
[[<EXPR>]]: Expr → Env → Value
[E1 + E2] = [E1] + [E2]
[E1 - E2] = [E1] - [E2]

<TERM>:
[[<TERM>]]: Expr → Env → Value
[E1 * E2] = [E1] * [E2]
[E1 / E2] = [E1] / [E2]
[E1 % E2] = [E1] % [E2]

<FACT>:
[[<FACT>]]: Expr → Env → Value
[ID] = env(ID)
[INT_LIT] = INT_LIT

[FLOAT_LIT] = FLOAT_LIT
[(E)] = [E]

<BOOL_EXPR>:
[[<BOOL_EXPR>]]: BExpr → Env → Bool
[E1 > E2] = ([E1] > [E2])
[E1 < E2] = ([E1] < [E2])
[E1 >= E2] = ([E1] >= [E2])
[E1 <= E2] = ([E1] <= [E2])

<BTERM>:
[[<BTERM>]]: BExpr → Env → Bool
[E1 == E2] = ([E1] == [E2])
[E1 != E2] = ([E1] != [E2])

<BAND>:
[[<BAND>]]: BExpr → Env → Bool
[E1 && E2] = ([E1] && [E2])

<BOR>:
[[<BOR>]]: BExpr → Env → Bool
[E1 || E2] = ([E1] || [E2])