⟨*program*⟩ → ⟨*roots*⟩

⟨*roots*⟩ → ⟨*root*⟩
  | ⟨*root*⟩ ⟨*roots*⟩

⟨*root*⟩ → ⟨*function*⟩
  | ⟨*dcl*⟩;

⟨*dcl*⟩ → ⟨*type*⟩ ⟨*id*⟩
  | ⟨*type*⟩ ⟨*assign*⟩

⟨*id*⟩ → ⟨*letter*⟩
  | ⟨*id*⟩ ⟨*letter*⟩
  | ⟨*id*⟩ ⟨*digit*⟩

⟨*letter*⟩ → [a-z]
  | [A-Z]

⟨*digit*⟩ → [0-9]

⟨*assign*⟩ → ⟨*id*⟩ `<--` ⟨*expr*⟩

⟨*type*⟩ → ⟨*primitive-type*⟩
  | ⟨*array-type*⟩

⟨*primitive-type*⟩ → bool
  | double
  | int
  | char

⟨*array-type*⟩ → ⟨*type*⟩ [ ]
  | string

⟨*function*⟩ → function ⟨*id*⟩ returns ⟨*type*⟩ using (⟨*params*⟩) begin ⟨*stmts*⟩ return ⟨*expr*⟩; end
  | function ⟨*id*⟩ returns nothing using (⟨*params*⟩) begin ⟨*stmts*⟩ return nothing; end

⟨*params*⟩ → ⟨*subparams*⟩
  | ε

⟨*subparams*⟩ → ⟨*type*⟩ ⟨*id*⟩, ⟨*subparams*⟩
  | ⟨*type*⟩ ⟨*id*⟩

⟨*stmts*⟩ → ⟨*stmt*⟩
  | ⟨*stmt*⟩⟨*stmts*⟩

⟨*stmt*⟩ → ⟨*assign*⟩;
  | ⟨*if*⟩
  | ⟨*while*⟩
  | ⟨*from*⟩
  | ε
  | ⟨*dcl*⟩;
  | ⟨*functioncall*⟩;
  | ⟨*switch*⟩

⟨*expr*⟩ → ⟨*expr*⟩+⟨*term*⟩
  | ⟨*expr*⟩-⟨*term*⟩
  | ⟨*term*⟩

⟨*term*⟩ → ⟨*term*⟩ * ⟨*factor*⟩
  | ⟨*term*⟩ / ⟨*factor*⟩
  | ⟨*factor*⟩

⟨*factor*⟩ → (⟨*expr*⟩)
  | ⟨*id*⟩
  | ⟨*digit*⟩
  | "⟨*string*⟩"

⟨*string*⟩ → ⟨*letter*⟩
  | ⟨*digit*⟩
  | ⟨*digit*⟩ ⟨*string*⟩
  | ⟨*letter*⟩ ⟨*string*⟩
  | ε

⟨*switch*⟩ → switch (⟨*expr*⟩) begin ⟨*cases*⟩ end

⟨*cases*⟩ → case ⟨*expr*⟩: ⟨*stmts*⟩ ⟨*endcase*⟩

⟨*endcase*⟩ → ⟨*cases*⟩
  | break;
  | break; ⟨*cases*⟩
  | default: ⟨*stmts*⟩ break;
  | break; default: ⟨*stmts*⟩ break;

⟨*from*⟩ → from ⟨*expr*⟩ to ⟨*logexpr*⟩ step ⟨*expr*⟩ begin ⟨*stmts*⟩ end

⟨*while*⟩ → while(⟨*logexpr*⟩) begin ⟨*stmts*⟩ end

⟨*if*⟩ → if(⟨*logexpr*⟩) begin ⟨*stmts*⟩ ⟨*endif*⟩

⟨*endif*⟩ → end else ⟨*if*⟩
  | end else begin ⟨*stmts*⟩ end
  | end

⟨*logexpr*⟩ → ⟨*logexpr*⟩ OR ⟨*andcomp*⟩
  | ⟨*andcomp*⟩

⟨*andcomp*⟩ → ⟨*andcomp*⟩ AND ⟨*comp*⟩
  | ⟨*comp*⟩

⟨*comp*⟩ → ⟨*boolean-operand*⟩ ⟨*comparison-operator*⟩ ⟨*boolean-operand*⟩

⟨*boolean-operand*⟩ → true
  | false
  | ⟨*expr*⟩
  | ⟨*boolean*⟩

⟨*boolean*⟩ → !(⟨*logexpr*⟩)
  | (⟨*logexpr*⟩)

⟨*comparison-operator*⟩ → `>`
  | `<`
  | `<=`
  | `>=`
  | `!=`
  | `=`

⟨*functioncall*⟩ → call ⟨*id*⟩(⟨*params*⟩)