<program> -> <dec> <main> <dec>
<main> -> main ( ) { <mst> }
<dec> -> <class_dec> | <fn_dec> | ε

<class_dec> -> class ID <extends> { <class_body>}
<extends> -> extends:ID | null

<class_body> -> <class_chidlren> <class_body> | null
<class_ chidlren> -> <dt_dec> | <func_dec> | <Constructor>

<func_dec> -> DT ID (<param_list>) <Body>
<param_list> -> <param> | <param> , <param_list> | ε
<param> -> DT ID

<Body> -> ; | { <MST> }

<Constructor> -> ID (<param_list>) <Body>

<dt_dec> -> DT ID = <Const_or_ID> ; // int x = 5; or int x = x;
| DT ID ; // int x;
| DT <multi_dec> ; // int x,b,c; or int a,b;
| DT <multi_dec_init> ; // int x = 10,b=20; or int x = a, y = b; or int x=a,y=3;
| <arr_type> ID [] = { <arr_const_or_id> }; // int arr[] = {}; or int arr[] = {1,2,a}; obj arr[]

<Const_or_ID> -> Const | ID
<multi_dec> -> ID , <multi_dec> | ID
<multi_dec_init> -> ID = Const, <multi_dec_init> | ID = ID, <multi_dec_init> | ID = <Const_or_ID>
<arr_type> -> DT | ID
<arr_const_or_id> -> ε | <Const_or_ID> | ID , | Const ,

<SST> -> <while_loop> | <for_loop> | <if_else> | <do_while> | <command> | <exp> | <try> | <throw> | <return> | <continue> | <break> | <dt_dec> | <func_dec>

<MST> -> <SST><MST> | ε

<while_loop> -> while (<cond>)<loop_body>
<cond> -> <Const_or_ID> | <Const_or_ID> <ROP> <Const_or_ID> | <exp>
<ROP> -> RO1 | RO2
<loop_body> -> ; | <SST> | {<MST>}

<for_loop> -> for (<F1><F2>;<F3>) <loop_body>
<F1> -> <dt_dec> | <assign_st> | ;
<F2> -> <cond> | null
<F3> -> <inc_dec> | <assign_st> | null

<inc_dec> -> ID increase_decrease
<assign_st> -> ID = <assign_options> ;
<assign_options> -> <Const_or_ID> | <exp>

<if> -> if (<cond>) <loop_body> <else>
<else> -> else <loop_body> | null

<do_while> -> do <loop_body> while (<cond>);

<this> -> this ID ;| this <func_call> ;
<func_call>-> ID (<param_list>)

<exp> -> <OE>
<OE>-> <AE><OE'>
<OE'> -> OR<AE><OE'> | null
<AE>-> <RE2><AE'>
<AE'> -> AND<RE2><AE'> | null
<RE2>-> <RE1><RE2'>
<RE2'> -> RO2<RE1><RE2'> | null
<RE1> -> <E><RE1'>
<RE1'> -> RO1<E><RE1'> | null
<E> -> <T><E'>
<E'> -> PM<T><E'> | null
<T> -> <F><T'>
<T'> -> MDM<F><T'> | null
<F> -> ID | const | (<OE>) | -<F> | NOT <F>

<return> -> return <return_options> ;
<return_options> -> ID | <const> | <exp> |null

<continue> -> continue;

<break> -> break;

<try> → try { <MST> } catch (ID) { <MST> }

<throw> -> throw <throw_options>:
<throw_options> -> ID | Const | new ID (<param_list>)