**<def>main{<mst>}<def>**  
  
**<def>** -> <class\_dec> | <fn\_dec> | <interface>

**<class\_dec>** -> class ID <inherit> { <class\_body>}  
<inherit> -> extends ID | null  
  
<class\_body> -> <class\_member><class\_body> | null  
<class\_members> -> <Dec> | <fn\_dec> | <Constructor>  
  
**<fn\_dec>** -> DT ID (<param\_list>) <Body>  
<param\_list> -> <param> <more\_params> | null  
<more\_params> -> , <param> <more\_params> | , | null  
<param> -> DT ID  
  
<Body> -> ; | {<MST>}  
  
**<Constructor>** -> ID (<param\_list>)<Body> **(its MST will not have return)**

<Body> -> {<CMST>} | null

<CMST> -> <CSST><CMST> | null

<CSST> -> <while\_loop> | <for\_loop> | <if\_else> | <do\_while> | <command> | <exp> | <try> | <throw> | <continue> | <break>

**<interface>** -> interface ID <interface\_inherit> {<interface\_body>}

<interface\_inherit> -> extends ID <interface\_inherit2> | null

<interface\_inherit2> -> , ID <interface\_inherit2> | null  
<interface\_body> -> <Dec> | DT ID (<param\_list>);

**<Dec>** -> DT ID <INIT> <list>  
<list> -> , ID <INIT> <list> | ;  
<INIT> -> = ID <INIT> | = <const> | null  
<const> -> int\_const | float\_const | str\_const | char\_const | bool\_const

**<array\_dec>** -> <array\_type> ID = [<arr\_content>];  
<array\_type> -> DT | ID

<arr\_content> -> DT <more\_content>| ID <more\_content> |null

<more\_content> -> , DT <more\_content> | , ID <more\_content> | null

**<united\_dec>** -> <array\_type> ID <INIT><list>

<INIT> -> = ID() <INIT> | = ID <INIT> | = <const> | = [<arr\_content>] | null

<list> -> , ID <INIT><list> | ;

**<SST>** -> <while\_loop> | <for\_loop> | <if\_else> | <do\_while> | <command> | <exp> | <try> | <throw> | <return> | <continue> | <break>

**<MST>** -> <SST><MST> | null  
  
**<while\_loop>** -> while (<cond>)<loop\_body>  
<cond> -> ID | <const> | <operand><ROP><operand>  
<operand> -> ID | const  
<ROP> -> RO1 | RO2  
<loop\_body> -> ; | <SST> | {<MST>}

**<for\_loop>** -> for (<F1><F2>;<F3>)<loop\_body>  
<F1> -> <Dec> | <assign\_st> | ;  
<F2> -> <cond> | null  
<F3> -> <inc\_dec> | <assign\_st> | null

**<if\_else>** -> if (<cond>) <loop\_body> <OElse>  
<OElse> -> else <loop\_body> | null

**<do\_while>** -> do <loop\_body> while (<exp>) ;

**<command>** -> <TS>ID<command2> | ID <command3> | inc\_dec<TS>ID<opt>  
<command2> -> <opt><command4> | (<param\_list>)<fn\_chain>;  
<command4> -> inc\_dec; | <assign\_op><exp>;  
<command3> -> <opt><command4> | (<param\_list><fn\_chain> | ID = ID (<param\_list>);

<TS> -> super. | this. | ID.

<opt> -> .ID<opt> | [<exp>] | (<param\_list>)<opt2> | null  
<opt2> -> .ID<opt>

**<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’>;

<return’> -> ID | <const> | <exp> |null

**<continue>** -> continue;

**<break>** -> break;

**<try>** -> try{<MST>}<catch><finally>

<catch> -> catch(ID){<MST>} | null

<finally> -> finally{<MST>} | null

**<throw>** -> throw <throw’>:

<throw’> -> ID | new ID() | ID() | ID.ID | <const> | null

**<const>** -> str\_const | int\_const |float\_const | char\_const | bool\_const