**<DEC>** : DT ID <init> <list>

<INIT> : = <init2> | e

<INIT2> : <const> | ID <init>

<const> : int-const |str-const | bool-const | char-const | double-const

<list> : , ID <init><list> | ;

First set:

<dec> : {DT}

<init> : {=, e}

<init2> : {int-const, str-const, bool-const, char-const, double-const, ID}

<const> : {int-const, str-const, bool-const, char-const, double-const}

<list> : {(,), ;}

Follow Set:

<dec> : {$}

<init> : {(,), ;}

<init2> : {(,), ;}

<const> : {(,), ;, ( } ) }

<list> : {$}

**<For\_st>** : for : <c1> <c2> ; <c3> : <body>

<c1> : <dec> | <asgn\_st> | ;

<c2> : <OE> | e

<c3> : ID <X> <c4> | inc\_dec ID <X> | e

<c4> : <asgn\_op> <exp> | inc\_dec

First set:

<For\_st> : {for}

<c1> : {DT, ID, ;}

<c2> : {this, ID, int-const, str-const, bool-const, char-const, double-const, (, inc\_dec, :, !, e }

<c3> : {ID, inc\_dec, e}

<c4> : {=, RO, AO, inc\_dec}

Follow set:

<For\_st> : {$}

<c1> : {this, ID, int-const, str-const, bool-const, char-const, double-const, (, inc\_dec, ! }, ;}

<c2> : {;}

<c3> : {:}

<c4> : {:}

**<Body>** : <SST> | ( <MST> ) | ;

<SST> : <For\_st> | <while\_st> |<foreach\_st> |<if\_else> | <return\_st>| <ArrList\_st> | DT<SST03> |inc\_dec ID <X> ; |ID <SST01>

<SST01> : ID = new ID:<Arg\_st>:; | <Y> :<Argu\_st>: ; |<X> <SST02>

<SST02> : <asgn\_op> <exp>; | inc\_dec;

<SST03> : ID <init> <list> | [] ID <list3> ; |

<MST> : <SST><MST> | e

First set:

<Body> : {For, while, foreach, if, ret, ArrayList, DT, inc\_dec, ID, (, ; }

<SST> : {For, while, foreach, if, ret, ArrayList, DT, inc\_dec, ID }

<SST01> : {ID, (.), :, [, =, RO, AO, inc\_dec }

<SST02> : {=, RO, AO, inc\_dec}

<SST03> : {ID, [ }

<MST> : {For, while, foreach, if, ret, ArrayList, DT, inc\_dec, ID, e }

Follow set:

<Body> : {$, elif, else, For, while, foreach, if, ret, ArrayList, DT, inc\_dec, ID, (, ; }

<SST> : { $, For, while, foreach, if, ret, ArrayList, DT, inc\_dec, ID, ) }

<SST01> : { $, For, while, foreach, if, ret, ArrayList, DT, inc\_dec, ID, ) }

<SST02> : { $, For, while, foreach, if, ret, ArrayList, DT, inc\_dec, ID, ) }

<SST03> : { $, For, while, foreach, if, ret, ArrayList, DT, inc\_dec, ID, ) }

<MST> : { ) }

**<obj\_dec> :** ID ID = new ID : <Arg\_st> : ;

<Argu\_st> : <OE> <Arg\_list2> | e

<Argu\_st2> : ,<OE><Arg\_list2> | e

<funct\_call> : ID <Y> : <Argu\_st> : ;

<Y> : .ID <Y> | e

<Asgn\_st> : ID <X> <asgn\_op> <E>;

<Asgn\_op> : = | <CoP>

<CoP> : RO | AO

<X> : <Handle\_Array> <Handle\_Dot>

<Handle\_Array> : [int-constant] | e

<Handle\_Dot> : .ID<X2> | e

<X2> : <X> | :<Argu\_st>: <Handle\_Dot>

First set:

<obj\_Dec> : {ID}

<Argu\_st> : {ID, int-const, str-const, bool-const, char-const, double-const}

<Argu\_st2> : {,}

<funct\_call> : {ID}

<Y> : {(.), e}

<asgn\_st> : {ID}

<asgn\_op> : {=, RO, AO }

<CoP> : {RO, AO}

<X> : { [, (.), e }

<Handle\_Array> : { [, e}

<Handle\_Dot> : {(.), e}

<X2> : { [, (.), e, : }

Follow set:

<obj\_dec> : {$}

<Argu\_st> : {:}

<Argu\_st2> : {:}

<funct\_call> : {$}

<Y> : {:}

<asg\_st> : { ; }

<asgn\_op> : {this, ID, int-const, str-const, bool-const, char-const, double-const, (, : ,inc\_dec, ! }

<CoP> : {this, ID, int-const, str-const, bool-const, char-const, double-const, (, : ,inc\_dec, ! }

<X> : { =, RO, AO }

<Handle\_Array>: { =, RO, AO }

<Handle\_Dot> : { =, RO, AO }

<X2> : { =, RO, AO }

**<OE>** : <AE><OE2>

<OE2> : || <AE> <OE2> | e

<AE> : <RE> <AE2>

<AE2> : && <RE> <AE2> | e

<RE> : <E> <RE2>

<RE2> : RO <E> <RE2> | e

<E> : <T> <E2>

<E2> : PM <T> <E2> | e

<T> : <F> <T2>

<T2> : MDM <F> <T2> | e

<F> : <Th> ID <X> | <Const> | ( <OE> ) | INC\_DEC ID | ! <F>

<Th> : this. | Ɛ

First set:

<OE> : {this, ID, int-const, str-const, bool-const, char-const, double-const, (, : ,inc\_dec, ! }

<OE2> : { ||, e}

<AE> : {this, ID, int-const, str-const, bool-const, char-const, double-const, (, : ,inc\_dec, ! }

<AE2> : {&&, e}

<RE> : {this, ID, int-const, str-const, bool-const, char-const, double-const, (, : ,inc\_dec, ! }

<RE2> : {RO, e}

<E> : {this, ID, int-const, str-const, bool-const, char-const, double-const, (, : ,inc\_dec, !}

<E2> : {PM, e}

<T> : {this, ID, int-const, str-const, bool-const, char-const, double-const, (, : ,inc\_dec, !}

<T2> : {MDM, e}

<F> : {this, ID, int-const, str-const, bool-const, char-const, double-const, (, : ,inc\_dec, ! }

<Th> : { this, e }

Follow:

<OE> : {$, : , ), ]}

<OE2> : {$, : , ), ]}

<AE> : {||, $, : , ), ]}

<AE2> : {||, $, : , ), ]}

<RE> : {&&,||, $, : , ), ]}

<RE2> : {&&,||, $, : , ), ]}

<E> : {RO, &&, ||, $, : , ), ], ;}

<E2> : {RO, &&, ||, $, : , ), ], ;}

<T> : {PM, RO, &&, ||, $, : , ), ], ;}

<T2> : {PM, RO, &&, ||, $, : , ), ], ;}

<F> : {MDM, PM, RO, &&, ||, $, : , ), ], ;}

<Th> : {ID}

**<Cond>** : <ID\_const> <RO\_LO> <ID\_const> | ID

<RO\_LO> : RO | LO

<ID\_const> : ID |<const>

<Print\_St> : print(<ID\_const>);

First set:

<cond> : {ID, int-const, str-const, bool-const, char-const, double-const}

<RO\_LO> : {+=, -=, \*=, /=, &&, ||, !=}

<ID\_const> : {ID, int-const, str-const, bool-const, char-const, double-const}

<Print\_St> : {print}

Follow set:

<cond> : {$, :}

<RO\_LO> : {ID, int-const, str-const, bool-const, char-const, double-const}

<ID\_const> : {$, : }

<Print\_St> : {$}

**<foreach\_st>** : foreach : DT ID in <DS> : <body>

<DS> : <Arr\_St> | <ArrList\_St>

First set:

<foreach\_st> : {foreach}

<DS> : {DT, ArrayList}

Follow set:

<foreach\_St> : {:}

<DS> : {:}

**<While\_st>** : while : <OE> : <body>

First set:

<while\_St> : {while}

Follow set:

<while\_st> : {:}

**<if\_else>** : if : <OE> : <body> <elif\_st>

<elif\_st> : elif : <OE> : <body> | e | <else>

<else> : else <body> | e

First set:

<if\_else> : {if}

<elif\_St> : {elif, else, e}

<else> : {else, e}

Follow set:

<if\_else> : {$}

<elif\_st> : {$, :}

<else> : {$, :}

**<funct\_st>**  : <static\_st> <DT\_Void> ID : <param> : <body>

<static\_st> : static | e

<DT\_Void> : void | DT

<param> : DT ID <multiParam> | e

<MultiParam> : , DT ID <multiParam> | e

<return\_st> : ret <const>;| e

First set:

<funct\_St> : {static, DT, void}

<static\_st> : {static, e}

<DT\_void> : {void, DT}

<param> : {DT, e}

<Multiparam> : {(,), e}

<return\_st> : {ret, e}

Follow set:

<funct\_st> : {$}

<static\_st> : {void, DT, class}

<DT\_void> : {ID}

<param> : {:}

<multiparam> : {:}

<return \_st> : {;}

**<Arr\_St>** : DT[] ID <list3> ;

<list3> : = <list2> | e

<list2> : new DT [int] | { <const> <multi\_value> }

<Multi\_Value> : ,<const> <multi\_value> | e

// <const> of same DT

**<ArrList\_St>** : ArrayList ID <list4> ;

<list4> : = new ArrayList() <list5> | e

<list5> : { <multi\_value> } | e

First set:

<Arr\_st> : { DT }

<ArrList\_St> : { ArrayList}

<list2> : { new, ( { ) }

<list3> : { =, e}

<list4> : { =, e}

<list5> : { ( { ) , e}

<multiValue> : { ( , ) , e}

Follow set:

<Arr\_st > : {$}

<ArrList\_st> : {$}

<list2> : {;}

<list3> : {;}

<list4> : {;}

<list5> : {;}

<multiValue> : { ( } )}

**<class\_st>** : <AccessModifier> <static\_abstract> class ID <inherit\_inter> <body2>

<AccessModifier> : AccessModifier | e

<static\_abstract> : static | abstract | e

<Body2> : { <MST2> }

<SST2> : <body> | <funct\_st> | <class\_st> | e

<MST2> : <SST2><MST2> |e

<inherit\_inter> : <inherit\_st> | <interface\_imple>

First set:

<class\_st> : {AccessModifier, static, abstract, class}

<AccessModifier> : {AccessModifier, e}

<static\_abstract> : {static, abstract, e}

<Body2> : { ( { ) }

<SST2> : {For, while, foreach, if, ret, ArrayList, DT, inc\_dec, ID, (, ;, static, DT, void, AccessModifier, abstract, e}

<MST2> : { For, while, foreach, if, ret, ArrayList, DT, inc\_dec, ID, (, ;, static, DT, void, AccessModifier, abstract, e}

<inherit\_inter> : { ( , ), :, e}

Follow set:

<class\_st> : {$}

<AccessModifier> : {static, abstract, class}

<static\_abstract> : {class}

<Body2> : {$}

<MST2> : { ( } ) }

<SST2> : {( } )}

<inherit\_inter> : { ({) }

**<inherit\_st>** : (: ) ID | e

First set:

<inherit\_St> : { :, e }

Follow set:

<inherit\_st > : { ( , ), :, ({) }

**<Poly\_st>** : <AccessModifier2> VO <funct\_st2>

<AccessModifier2> : public | protected

<funct\_st2> : <DT\_Void> ID : <param> : <body>

First set:

<Poly\_st> : {public, protected}

<AccessModifier2> : {public, protected}

<Funct\_St2> : {DT, void}

Follow set:

<Poly\_st> : {$}

<AccessModifier2> : {VO}

<funct\_st2> : {$}

**<interface\_imple>** : <col\_comma> ID <interface\_imple> | e

<col\_comma> : : | ,

First set:

<interface\_imple> : {:, ( , ) , e}

<col\_comma> : {: , ( , )}

Follow set:

<interface\_imple> : { ( { ) }

<col\_comma> : {ID}

**<Interface\_st>** : interface ID <Body3>

<Body3> : { <MST3> }

<MST3> : <SST3><MST3> | e

<SST3> : <funct\_St3> | e

<funct\_st3> : <DT\_void> ID : <params> : ;

First set:

<interface\_st> : {interface}

<Body3> : { ( { )}

<MST3> : {DT, void, e}

<SST3> : {DT, void, e}

<funct\_St3> : {DT, void }

Follow set:

<interface\_st> : {$}

<Body3> : {$}

<MST3> : { ( } ) }

<SST3> : { ( } ) }

<funct\_St3> : { ( } ) }