|  |
| --- |
| 1- <start> 🡪 <defs> <VI> main (<NV>) {<MST>}$ | $  <VI> 🡪 void  <NV> 🡪~ | void  <defs>🡪<class><defs> | ID {T=DT.VP} <defs1><defs> |DT {T=DT.VP} ID <defs2><defs> |static <type> (<fn\_dec> | ~  <type> 🡪ID{T=ID.VP} ID | DT{T=DT.VP} ID  <defs1> 🡪<Ass\_st> | ID <X>  <X> 🡪<obj\_dec> | (<terminal>  <terminal> 🡪<fn\_dec> | <constructor\_dec>  <defs2> 🡪 <DT\_dec> | (<fn\_dec> |
| 2- <MST>🡪~ | <SST><MST> |
| 3- <SST>🡪<if\_else> | <while\_st> | <switch> | <for\_st> |<return> | continue;| break; | ID{T=ID.VP} <SST1> | DT{T=DT.VP} ID <SST2>  <SST1> 🡪<array> <L2>| ID <Xxx> //L1 IS REMOVED  <L2>🡪<fn\_call> | inc\_dec;|<Ass\_st>  <Xxx> 🡪<obj\_dec> | ( <constructor\_dec>  <SST2> 🡪 <DT\_dec> |
| 4- <return> 🡪 return <OE1> ;  <OE1> 🡪 <OE> | ~ |
| 5- <fn\_call> 🡪 <fn\_call 1>;  <fn\_call1>🡪<checkID> (<arg> {fn\_lookup(N,AL,NULL,FALSE)})  <checkID>🡪. ID{T=ID.VP, compatinility(T,T,.)}<array> <checkID> | ~  <arg> 🡪 <OE> {N=ID.VP T=LOOKUP(N), AL=T} <arg1>| ~  <arg1> 🡪 ,<OE>{N=ID.VP T=LOOKUP(N), AL=T}<arg1>| ~ |
| 9- <inc\_dec> 🡪 <inc\_dec1>;  <inc\_dec1> 🡪 inc\_dec |
| 6- <DT\_dec> 🡪 {N=ID.VP,IF(!INSERT(N,T,S)) IF(FLAG=1) INSERT\_CDT(N,T,AM,TM) } <new\_arr> | {N=ID.VP,IF(!INSERT(N,T,S)) IF (FLAG=1)INSERT\_CDT(N,T,AM,TM) } <init>  <new\_arr> 🡪 [<OE>]<array2>  <array2>🡪= {<array3>} ; | ;  <array3> 🡪 <OE><array4>  <array4> 🡪 , <OE> <array4> | ~  <init> -->AOP <init2> | <list>  <init2> --><OE> ;| <init4>  <init3> -->AOP <init4>| ~  <init4> --> ID <init3> <list> | <const> < list>  <list> -->,ID{N=ID.VP,IF(!INSERT(N,T,S)) IF (FLAF=1) INSERT\_CDT(N,T,AM,TM) } <init3> <list> | ; |
| 7- <fn\_dec> --> <para> { ,IF(!INSERT(N,S,T)) IF(FLAG=1)INSERT\_CDT(N,T,AM,TM)} <inherit> {<MST>}  <word1> --> virtual | static | const  <para> --> <def> <E> | void | ~  <def> --> ID ID|DT {T=DT.VP,PL=T} ID{N=ID.VP,IF(!INSERT(N,T,S))}  <E> --> ,<def><E> | ~  <inherit> -->: ID{N=ID.VP, LOOKUP\_CT(N)} <fn\_call1> //removed |
| 8- <Ass\_st> --> <Ass\_st1><Xx> ;  <Ass\_st1> --> AOP <OE>  <Xx> --> <Ass\_st1><Xx> | ~  <array> --> [<OE>]| ~ |
| 10- <OE> --> <AE> <OE'> //  <OE'> --> {TL}||<AE>{Tr}T=compatibility(tR,tL,O)} <OE'> | ~ //  <AE> --> <RE> <AE'> //  <AE'> --> {TL} &&<RE> {Tr}T=compatibility(tR,tL,O)} <AE'> | ~ //  <RE> --> <PE><RE'> //  <RE'> -->{TL} ROP <PE> {Tr}T=compatibility(tR,tL,O)} <RE'>|~  <PE> --> <ME><PE'> //  <PE'> --> {TL} PM <ME> {Tr}T=compatibility(tR,tL,O)} <PE'>|~  <ME> --> <F><ME'> //  <ME'> --> {TL} MDM <F> {Tr}T=compatibility(tR,tL,O)} <ME'>|~  <F> --> <CONST> |(<OE>) |!<PE> |ID <XOE1> //  <XOE1> 🡪<fn\_call> | <array> <nt2> inc\_dec|~ //  <nt2> 🡪 .ID<array><nt2>|~ // |
| 11- <obj\_dec> --> {N=ID.VP INSERT(N,S,T) IF (FLAG=1) INSERT\_CDT(N,T,AM,TM)} <array><new\_init> <list2>  <new\_init> --> = ID < new\_init > | ~  <list2> --> ,ID{N=ID.VP INSERT(N,S,T) IF (FLAG=1) INSERT\_CDT(N,T,AM,TM)}  <array><new\_init><list2> | ; |
| 12-<constructor\_dec> --> <arg>); |
| 12- <if\_else> --> if(<OE>){<MST>} <o\_else>  <o\_else> -->else {<MST>}|~ |
| 13- <while\_st> --> while (<OE>){<MST>} |
| 14- <switch> --> switch(<OE>){<case><default>}  <case> --> case <OE>: {<MST>} <case>| ~  <default> --> default:{<MST>} | ~ |
| 15- <class> --> class ID{N=ID.VP} <chk\_inhrt> {IF(!INSERT\_CT(N,TYPE,P,REF)) {<class\_body>};  <chk\_inhrt> -->: AM ID{N=ID.VP,P=N} | ~  <class\_body> -->ID {T=DT.VP} <X1> <class\_body> | DT{T=DT.VP} ID <X2><class\_body> | AM: <class\_body> | <word><type> (<fn\_dec> | ~  <word> --> virtual | const | static  <X1> --> <constructor\_fn> | ID <X3>  <X2> --> ( <fn\_dec> {FLAG=1} | <DT\_dec>{FLAG=1}  <X3> --> ( <fn\_dec> {FLAG=1} | <obj\_dec>{FLAG=1} |
| 16- <for\_st> -->for(<C1><C2>;<C3>) { <MST>}  <C1> --> DT{T=DT.VP} ID <DT\_dec>|ID {N=ID.VP, T=LOOKUP(N)}<Ass\_st>|;  <C2> --> <OE> | ~  <C3> --> ID{N=ID.VP, T=LOOKUP(N)} <X11> | inc\_dec | ~  <X11> --> <Ass\_st1> | inc\_dec |
| 17)<const> --> int\_const | Float\_const | string\_const | bool\_const |char\_const |
| 18) <constructor\_fn> --> {N=IP.VP} (<para>{INSERT(N,S,T)} {<MST>} |