1. program→program`
2. program`-->classDeclaration End Program`|Ɛ
3. ClassDeclaration→ Category ID CD’
4. CD’ → { Class\_Implementation}|Derive { Class\_Implementation}
5. Class\_Implementation→ Type Var\_Method Class\_Implementation | Comment Class\_Implementation | using\_command Class\_Implementation| Func \_Call Class\_Implementation |empty
6. Var\_Method→ ID Var\_Method `
7. Var\_Method `→ID\_List ; | MethodDeclaration
8. MethodDeclaration→ Func Decl Me\_Declaration
9. Me\_Declaration→ ; |{ VarDeclaration Statements }
10. Func Decl → (ParameterList)
11. Type → Ilap | Silap | Clop | Series | Ilapf | Silapf | None | Logical
12. ParameterList →empty | None | Non-Empty List
13. Non-Empty List -->Type ID Non-Empty List`
14. Non-Empty List`-->, Type ID Non-Empty List`|Ɛ
15. VarDeclaration→ empty | Type ID ID\_List |ID ID\_List ; VarDeclaration
16. ID\_List-->ID\_List`
17. ID\_List` -->, ID ID\_List`|Ɛ
18. Statements→empty | Statement Statements
19. Statement→Assignment | If \_Statement | Rotatewhen \_Statement | Continuewhen\_Statement | Replywith \_ Statement | terminatethis \_Statement|read (ID ); | write (Expression);
20. Assignment→ VarDeclaration = Expression;
21. Func \_Call → ID (Argument\_List) ;
22. Argument\_List →empty | NonEmpty\_Argument\_List
23. NonEmpty\_Argument\_List-->Expression NonEmpty\_Argument\_List`
24. NonEmpty\_Argument\_List`--> , Expression NonEmpty\_Argument\_List`|ε
25. Block Statements→{ statements }
26. If \_Statement→ if (Condition \_Expression) Block Statements
27. Condition \_Expression→ Condition Co\_Expression
28. Co\_Expression→ Condition \_Op Condition| ε
29. Condition \_Op → and | or
30. Condition→ Expression Comparison \_Op Expression
31. Comparison \_Op → == | != | > | >= | < | <=
32. Rotate \_Statement → Rotate when(Condition \_Expression) Block Statements
33. Continuewhen \_Statement →Continuewhen ( expression ; expression ; expression ) Block Statements
34. Replywith \_Statement→ Replywith Expression ; | returnID ;
35. terminatethis \_Statement→ terminatethis;
36. Expression-->Term Expression`
37. Expression`-->Add\_Op Term Expression`|ε
38. Add\_Op → + | -
39. Term -->Factor Term`
40. Term` -->Mul\_Op Factor Term` |ε
41. Mul\_Op→\* | /
42. Factor→ ID| Number
43. Comment → <\* STR \*>| -- STR
44. using\_command →using(F\_name.txt);
45. F\_name →STR