1 GRAMÁTICA DA LINGUAGEM X++

1. *Program* → *ClassList*
   | ϵ
2. *ClassList* → *ClassDecl ClassList*
   | *ClassDecl*
3. *ClassDecl* → **class ID** *ClassBody*
   | **class ID extends ID** *ClassBody*
4. *ClassBody* → **{** *VarDeclListOpt ConstructDeclListOpt MethodDeclListOpt* **}**
5. *VarDeclListOpt* → *VarDeclList*
   | ϵ
6. *VarDeclList* → *VarDeclList VarDecl*
   | *VarDecl*
7. *VarDecl* → *Type* **ID** `VarDeclOpt` **;**
   | *Type* **[]ID** `VarDeclOpt` **;**
8. *VarDeclOpt* → **, ID** *VarDeclOpt*
   | ϵ
9. *Type* → **int**
   | **string**
   | **ID**
10. *ConstructDeclListOpt* → *ConstructDeclList*
    | ϵ
11. *ConstructDeclList* → *ConstructDeclList ConstructDecl*
    | *ConstructDecl*
12. *ConstructDecl* → **constructor** *MethodBody*
13. *MethodDeclListOpt* → *MethodDeclList*
    | ϵ
14. *MethodDeclList* → *MethodDeclList MethodDecl*
    | *MethodDecl*
15. *MethodDecl* → *Type* **ID** *MethodBody*
    | *Type* **[] id** *MethodBody*
16. *MethodBody* → **(** *ParamListOpt* **){** *StatementsOpt* **}**
17. *ParamListOpt* → *ParamList*
    | ϵ
18. *ParamList* → *ParamList* **,** *Param*
    | *Param*
19. *Param* → *Type* **ID**
    | *Type* **[] ID**
20. *StatementsOpt* → *Statements*

```
        |  ε
21.  Statements → Statements Statement
        |  Statement
22.  Statement → VarDeclList
        |  AtribStat ;
        |  PrintStat ;
        |  ReadStat ;
        |  ReturnStat ;
        |  SuperStat ;
        |  IfStat
        |  ForStat
        |  break ;
        |  ;
23.  AtribStat → LValue = Expression
        |  LValue = AllocExpression
24.  PrintStat → print Expression
25.  ReadStat → read LValue
26.  ReturnStat → return Expression
27.  SuperStat → super ( ArgListOpt )
28.  IfStat → if ( Expression ) { Statements }
        |  if ( Expression ) { Statements } else { Statements }
29.  ForStat → for ( AtribStatOpt ; ExpressionOpt ; AtribStatOpt ) { Statements }
30.  AtribStatOpt → AtribStat
        |  ε
31.  ExpressionOpt → Expression
        |  ε
32.  LValue → ID LValueComp
33.  LValueComp → . ID LValueComp
        |  . ID [ Expression ] LValueComp
        |  . ID ( ArgListOpt ) LValueComp
        |  [ Expression ] LValueComp
        |  ε
34.  Expression → NumExpression
        |  NumExpression RelOp NumExpression
35.  AllocExpression → new ID ( ArgListOpt )
        |  Type [ Expression ]
36.  NumExpression → Term + Term
        |  Term - Term
        |  Term
37.  Term → UnaryExpression * UnaryExpression
        |  UnaryExpression / UnaryExpression
        |  UnaryExpression % UnaryExpression
        |  UnaryExpression
38.  UnaryExpression → + Factor
        |  - Factor
```

39. *Factor* → **INTEGER_LITERAL**
    | **STRING_LITERAL**
    | *LValue*
    | **(** *Expression* **)**
40. *ArgListOpt* → *ArgList*
    | ϵ
41. *ArgList* → *ArgList* **,** *Expression*
    | *Expression*