گرامر های اصلاح شده:

1) num -> [0-9] num'
   num' -> num num'
   | ∈
2) letter -> [a-zA-Z] letter'
   letter' -> letter letter'
   | ∈
3) numOrletter -> num | letter | ∈
4) program -> list
5) list -> declaration list'
   list' -> declaration list'
   | ∈
6) declaration ->function | varDeclaration
7) varDeclaration -> type variableList ;
8) ScopedVariableDec -> scopedSpecifier variableList;
9) variableList -> varInitialization variableList'
   variableList' -> , varInitialization variableList'
   | ∈
10) varInitialization -> varForm varInitialization'
    varInitialization' -> ∈
    | : (eachExpression)
11) varForm -> letter numOrletter varForm'
    varForm' -> [num]
    | ∈
12) scopedSpecifier-> static type | type
13) type -> Boolean | character | integer | char | bool |int
14) function -> void numOrletter (parameter) { statement }
    | type letter numOrletter (parameter) statement
15) parameter -> listOfParameters | ϵ

16) listOfParameters -> paramTypeList listOfParameters'
    listOfParameters' -> ; paramTypeList listOfParameters'

```
                                | ∈
17)  paramTypeList -> type paramList
18)  paramList -> paramId paramList'
     paramList' -> , paramId paramList'
                             | ∈
19)  localDeclarations -> localDeclarations'
     localDeclarations' -> ScopedVariableDec localDeclarations'
                                | ∈
20)  paramId -> letter numOrletter paramId'
      paramId' -> ∈
                        | [ ]
21)  statement -> phrase | compoundPhrase | selectPhrase |
iterationPhrase | returnPhrase | continue
22)  compoundPhrase -> { localDeclarations statementList }
23)  statementList -> statementList'
     statementList' -> statement statementList'
                             | ∈
24)  phrase -> allExpression ; | ;
25)  selectPhrase -> if ( eachExpression ) selectPhrase'
     selectPhrase' -> ifBody
                            | { ifBody ifBody }
26)  ifBody -> statement ifBody'
                  | ;
     ifBody' -> ∈
                       | other statement
27)  iterationPhrase -> till ( eachExpression ) statement
28)  returnPhrase -> comeback; | giveback allExpression ; | giveBack
numOrlette ;
29)  continue → continue ;
30)  allExpression -> alterable allExpression''
                            | eachExpression
     allExpression' -> allExpression
                            | alterable
```

```
     allExpression'' -> mathOp allExpression'
                      | ++
                      | --
31)  eachExpression -> logicOp eachExpression eachExpression'
                           | relExpression eachExpression'
     eachExpression' -> logicOp eachExpression''
                            | ∈
     eachExpression'' -> eachExpression eachExpression'
                           | then eachExpression eachExpression'
                           | else eachExpression eachExpression'
32)  relExpression -> mathEXP relExpression'
      relExpression' -> compareType mathEXP
                      | ∈
33)  compareType ->equal | nonEqual
34)  equal -><= | >= | ==
35)  nonEqual-> <|>|!=
36)  mathEXP -> unaryExpression mathEXP'
     mathEXP' -> op mathEXP mathEXP'
                 | ∈
37)  op -> + | - | * | / | %
38)  unaryExpression -> unaryop unaryExpression | factor
39)  unaryop->- |* |?
40)  factor -> inalterable | alterable
41)  alterable -> letter numOrletter alterable''
     alterable'' -> alterable' alterable''
                      | ∈
    alterable' -> [ allExpression ]
                 | . letter numOrletter
42)  inalterable -> ( allExpression ) | constant | letter numOrletter
( args )
43)  args -> arguments | ϵ
44)  arguments -> allExpression arguments'
     arguments' -> , allExpression arguments'
```

| ∈

45) constant -> CONST | true | false

46) logicOp -> && | || | ~| and| or