Backtrack-Free BNF for the Eck Programming Language

<*classDec*> → **class** <*className*> **{** <*classVarDec*>* <*subroutineDec*>* **}**

<*classVarDec* > → <*classVarModifier*> <*type*> <*varList*> **;**

<*classVarModifier*> → **static** | **field**

< *subroutineDec* > → <*subroutineSpecifier*> <*subroutineType*>
<*subroutineName*> **(**<*formalParameters*>**)** **{** <*subroutineBody*> **}**

< *subroutineSpecifier* > → **constructor** | **function** | **method**

<*subroutineType*> → **void** | <*type*>

<*formalParameters*> → <*parameterList*> | ε

<*parameterList*> → <*type*> <*varName*> <*parameterList1*>

<*parameterList1*> → **,** <*parameterList*> | ε

<*subroutineBody*> → <*varDec*>* <*statements*>

<*varDec*> → <*type*> <*varList*> **;**

<*type*> → **int** | **char** | **boolean** | **int[]** | **char[]** | **boolean[]** | <*className*>

<*varList*> → <*varName*> <*varList1*>

<*varList1*> → **,** <*varList*> | ε

<*className*> → *identifier*

<*subroutineName*> → *identifier*

<*varName*> → *identifier*

<*statements* > → <*statement* >*

<*statement* > → <*assignmentStatement*> |<*ifStatement*> | <*whileStatement*> | <*doStatement*> |
<*returnStatement*>

< *assignmentStatement* > → <*varName*> <*varArray*> **=** <*expression*> **;**

< *varArray* > → **[** <*expression*> **]** | ε

<*ifStatement*> → **if** **(** <*expression*> **)** **{** <*statements*> **}** <*elseStatement*>

<*elseStatement*> → **else** **{** <*statements*> **}** **|** ε

< *whileStatement* > → **while** **(** <*expression*> **)** **{** <*statements*> **}**

< *doStatement* > → **do** *identifier* <*doStatement1*> <*actualParameters*> **;**

<*doStatement1*> → **.** *identifier* **|** ε

< *returnStatement* > → **return** < *returnStatement1*> **;**

< *returnStatement1*> → <*expression*> **|** ε

< *expression* > → <*exp1*> <*expression'*>
< *expression*'> → **&** <*exp1*> < *expression*'> | **|** <*exp1*> <*expression'*> | ε

<*exp1*> → <*exp2*> <*exp1'*>
<*exp1'*> → **<** <*exp2*> <*exp1'*> | **>** <*exp2*> <*exp1'*> | **=** <*exp2*> <*exp1'*> | ε

<*exp2*> → <*exp3*> <*exp2'*>
<*exp2'*> → **+** <*exp3*> <*exp2'*> | **−** <*exp3*> <*exp2'*> | ε

<*exp3*> → <*exp4*> <*exp3'*>
<*exp3'*> → ***** <*exp4*> <*exp3'*> | **/** <*exp4*> <*exp3'*> | ε

<*exp4*> → **−** <*exp4*> | **~** <*exp4*> |
integerConstant | *stringConstant* | <*keywordConstant*> |
identifier <*exp4Id*> | **(** <expression> **)**

<*exp4Id*> →
[<*expression*> **]** **|**
. <*subroutineName*> <*actualParameters*> |
<*actualParameters*> |
ε

< *actualParameters* > → **(** <expressionList> **)**

<*expressionList*> → <*expressionList1*> | ε

<*expressionList1*> → <*expression*> <*expressionList2*>

<*expressionList2*> → **,** <*expressionList1*> | ε

<*keywordConstant*> → **true** | **false** | **null** | **this**