**Context Free Grammar**

|  |  |
| --- | --- |
| # | Expression |
|  | <program> → <global> Lead : Start <body> End . <taskdef> |
|  | <global> → <global\_choice> . <global> |
|  | <global> → λ |
|  | <global\_choice> → <varlet> |
|  | <global\_choice> → <array> |
|  | <global\_choice> → <task> |
|  | <global\_choice> → <object> |
|  | <dtype> → Int |
|  | <dtype> → Double |
|  | <dtype> → Char |
|  | <dtype> → String |
|  | <dtype> → Boolean |
|  | <object> → Object id Start Var<dtype> id . <objdec\_choice> End <varname> |
|  | <objdec\_choice> → <objvar> . <objdec\_choice> |
|  | <objdec\_choice> → <var> . <objdec\_choice> |
|  | <objdec\_choice> → λ |
|  | <var> → Var<dtype> id |
|  | <objvar> → Object id id |
|  | <varname> → id <varnames> |
|  | <varname> → λ |
|  | <varnames> → ; <varname> |
|  | <varnames> → λ |
|  | <task> → Task <return> id <tparam> |
|  | <return> → Null |
|  | <return> → <dtype> |
|  | <tparam> → ( Var<dtype> id <tparams> ) |
|  | <tparam> → λ |
|  | <tparams> → ; Var<dtype> id <tparams> |
|  | <tparams> → λ |
|  | <array> → Array <dtype> id of <size> |
|  | <size> → intlit<sizes> |
|  | <sizes> → by intlit |
|  | <sizes> → λ |
|  | <varlet> → Let <vardec> |
|  | <varlet> → Var<vardec> |
|  | <vardec> → <varINT> |
|  | <vardec> → <varDOUBLE> |
|  | <vardec> → <varCHAR> |
|  | <vardec> → <varSTRING> |
|  | <vardec> → <varBOOLEAN> |
|  | <varINT> → Int id <initINT> |
|  | <initINT> → is <value1><ids1> |
|  | <initINT> → <ids1> |
|  | <varDOUBLE> → Double id <initDOUBLE> |
|  | <initDOUBLE> → is <value2><ids2> |
|  | <initDOUBLE> → <ids2> |
|  | <varCHAR> → Char id <initCHAR> |
|  | <initCHAR> → is <value3><ids3> |
|  | <initCHAR> → <ids3> |
|  | <varSTRING> → String id <initSTRING> |
|  | <initSTRING> → is <value4><ids4> |
|  | <initSTRING> → <ids4> |
|  | <varBOOLEAN> → Boolean id <initBOOLEAN> |
|  | <initBOOLEAN> → is <value5><ids5> |
|  | <initBOOLEAN> → <ids5> |
|  | <ids1> → ; id is <ids1\_tail> |
|  | <ids1> → λ |
|  | <ids2> → ; id is <ids2\_tail> |
|  | <ids2> → λ |
|  | <ids3> → ; id is < ids3\_tail> |
|  | <ids3> → λ |
|  | <ids4> → ; id is <ids4\_tail> |
|  | <ids4> → λ |
|  | <ids5> → ; id is <ids5\_tail> |
|  | <ids5> → λ |
|  | <ids1\_tail> → is <value1> <ids1> |
|  | <ids2\_tail> → is <value2> <ids2> |
|  | <ids3\_tail> → is <value3> <ids3> |
|  | <ids4\_tail> → is <value4> <ids4> |
|  | <ids5\_tail> → is <value5> <ids5> |
|  | <value1> → <numvalue><Operations1> |
|  | <value1> → ( <value1> ) <Op1> |
|  | <value2> → <doublevalue><Operations2> |
|  | <value2> → ( <value2> ) <Op2> |
|  | <value3> → charlit |
|  | <value4> → stringlit |
|  | <value5> → boollit |
|  | <numvalue> → intlit |
|  | <numvalue> → <numelement> |
|  | <numelement> → <incdec\_null> id <subelement><incdec\_null> ; |
|  | <incdec\_null> → <incdec> |
|  | <incdec\_null> → λ |
|  | <Operations1> → <mathOp><value1> |
|  | <Operations1> → λ |
|  | <OpInt> → <numvalue><Operations1> |
|  | <OpInt> → ( <value1> ) <Op1> ; |
|  | <Op1> → <mathOp><OpInt> |
|  | <Op1> → λ |
|  | <doublevalue> → doublelit |
|  | <doublevalue> → <numelement> |
|  | <Operations2> → <mathOp><value2> |
|  | <Operations2> → λ |
|  | <OpDouble> → <doublevalue><Operations2> |
|  | <OpDouble> → ( <value2> ) <Op2> |
|  | <Op2> → <mathOp><OpDouble> |
|  | <Op2> → λ |
|  | <mathOp> → + |
|  | <mathOp> → - |
|  | <mathOp> → \* |
|  | <mathOp> → / |
|  | <mathOp> → % |
|  | <incdec> → ++ |
|  | <incdec> → -- |
|  | <relop1> → == |
|  | <relop1> → != |
|  | <relop1> → > |
|  | <relop1> → < |
|  | <relop1> → >= |
|  | <relop1> → <= |
|  | <logop1> → && |
|  | <logop1> → || |
|  | <logop2> → ! |
|  | <logop2> → λ |
|  | <body> → <statements> |
|  | <statements> → <functions> . <statements> |
|  | <statements> → λ |
|  | <functions> → Var<vardec> |
|  | <functions> → <varinit> |
|  | <functions> → <array> |
|  | <functions> → Clear |
|  | <functions> → id <id\_choices> |
|  | <functions> → <incdec> id <subelement> |
|  | <functions> → <io\_statement> |
|  | <functions> → <objvar> |
|  | <functions> → <IfOtherwise> |
|  | <functions> → <loopstate> |
|  | <functions> → <option> |
|  | <varinit> → <varinitINT> |
|  | <varinit> → <varinitDOUBLE> |
|  | <varinit> → <varinitCHAR> |
|  | <varinit> → <varinitSTRING> |
|  | <varinit> → <varinitBOOLEAN> |
|  | <varinitINT> → Int id = <INT> |
|  | <varinitDOUBLE> → Double id = <DOUBLE> |
|  | <varinitCHAR> → Char id = <CHAR> |
|  | <varinitSTRING> → String id = <STRING> |
|  | <varinitBOOLEAN> → Boolean id = <BOOLEAN> |
|  | <INT> → <value1> |
|  | <INT> → <task\_id> |
|  | <DOUBLE> → <value2> |
|  | <DOUBLE> → <task\_id> |
|  | <CHAR> → <value3> |
|  | <CHAR> → <task\_id> |
|  | <STRING> → <value4> |
|  | <STRING> → <task\_id> |
|  | <BOOLEAN> → <value5> |
|  | <BOOLEAN> → <task\_id> |
|  | <task\_id> → id ( <param> ) |
|  | <param> → <value><params> |
|  | <param> → λ |
|  | <params> → ; <param> |
|  | <params> → λ |
|  | <value> → intlit |
|  | <value> → doublelit |
|  | <value> → charlit |
|  | <value> → stringlit |
|  | <value> → boollit |
|  | <value> → <input\_id> |
|  | <io\_statement> → <input> |
|  | <io\_statement> → <output> |
|  | <input> → Read id |
|  | <output> → Say <input\_statement> |
|  | <input\_statement> → stringlit<concat> |
|  | <input\_statement> → <input\_id><concat> |
|  | <concat> → , <concat\_value><concat> |
|  | <concat> → λ |
|  | <concat\_value> → stringlit |
|  | <concat\_value> → <input\_id> |
|  | <subelement> → @ id <subelement> |
|  | <subelement> → λ |
|  | <input\_id> → id <subelement> |
|  | <multi> → [ <index> ] |
|  | <multi> → λ |
|  | <index> → intlit |
|  | <index> → id |
|  | <IfOtherwise> → If ( <conditions> ) <cond\_loop><or><otherwise>EndIf |
|  | <or> → Or ( <conditions> ) <cond\_loop><or> |
|  | <or> → λ |
|  | <otherwise> → Otherwise <cond\_loop><control> |
|  | <otherwise> → λ |
|  | <cond\_loop> → <functions> . <control><cond\_loop> |
|  | <cond\_loop> → λ |
|  | <control> → Skip . |
|  | <control> → Stop . |
|  | <control> → λ |
|  | <conditions> → <ids><condsTail> |
|  | <conditions> → <logop2><multiconds> |
|  | <multiconds> → ( <conditions> ) <condsTail> |
|  | <ids> → <input\_id> |
|  | <ids> → <logop2><ids\_null> |
|  | <ids\_null> → <value5> |
|  | <ids\_null> → <task\_id> |
|  | <condsTail> → <logOps> |
|  | <condsTail> → <relOps> |
|  | <condsTail> → λ |
|  | <logOps> → <logop1><conditions> |
|  | <logOps> → λ |
|  | <relOps> → <relopNum><numval> |
|  | <relOps> → <relopText><value> |
|  | <relopNum> → >= |
|  | <relopNum> → <= |
|  | <relopNum> → < |
|  | <relopNum> → > |
|  | <relopText> → == |
|  | <relopText> → != |
|  | <numval> → intlit |
|  | <numval> → doublelit |
|  | <option> → Option <input\_id> Start State <optiontails> |
|  | <optiontails> → <optiontail1> |
|  | <optiontails> → <optiontail2> |
|  | <optiontails> → <optiontail3> |
|  | <optiontail1> → intlit : <statements> Stop . <state1><default> End |
|  | <optiontail2> → charlit : <statements> Stop . <state2><default> End |
|  | <optiontail3> → stringlit : <statements> Stop . <state3><default> End |
|  | <state1> → State intlit : <statements> Stop . <state1> |
|  | <state1> → λ |
|  | <state2> → State charlit : <statements> Stop . <state2> |
|  | <state2> → λ |
|  | <state3> → State stringlit : <statements> Stop . <state3> |
|  | <state3> → λ |
|  | <default> → Default : <statements> |
|  | <default> → λ |
|  | <incdecvar> → <incdec> id |
|  | <incdecvar> → id <incdec> |
|  | <incdecvar> → λ |
|  | <loopstate> → Until ( <conditions> ) <cond\_loop> Loop |
|  | <loopstate> → Do <cond\_loop>LoopIf ( <conditions> ) |
|  | <loopstate> → For ( <initialize> ; <cond> ; <incdecvar> ) <cond\_loop> Loop |
|  | <initialize> → id = intlit |
|  | <initialize> → λ |
|  | <cond> → id <relop1>intlit |
|  | <cond> → λ |
|  | <taskdef> → Task <returntype> End . <taskdef> |
|  | <taskdef> → λ |
|  | <returntype> → Int id : <taskbody> . Response <returnINT> . |
|  | <returntype> → Double id : <taskbody> . Response <returnDOUBLE> . |
|  | <returntype> → Char id : <taskbody> . Response <returnCHAR> . |
|  | <returntype> → String id : <taskbody> . Response <returnSTRING> . |
|  | <returntype> → Boolean id : <taskbody> . Response <returnBOOLEAN> . |
|  | <returntype> → Null id : <taskbody> . |
|  | <taskbody> → Start <statements> |
|  | <returnINT> → intlit |
|  | <returnINT> → id <returntail> |
|  | <returnDOUBLE> → doublelit |
|  | <returnDOUBLE> → id <returntail> |
|  | <returnCHAR> → charlit |
|  | <returnCHAR> → id <returntail> |
|  | <returnSTRING> → stringlit |
|  | <returnSTRING> → id <returntail> |
|  | <returnBOOLEAN> → boollit |
|  | <returnBOOLEAN> → id <returntail> |
|  | <returntail> → ( <param> ) |
|  | <returntail> → <subelement> |
|  | <returntail> → λ |
|  | <StartProgram> → <program># |