**CONTEXT-FREE GRAMMAR**

|  |  |  |  |
| --- | --- | --- | --- |
|  | <program> | 🡪 | <global> Task Lead : Start <body> End . <taskdef> |
|  | <global> | 🡪 | <global\_choice> . <global> |
|  | <global> | 🡪 | λ |
|  | <global\_choice> | 🡪 | <varlet> |
|  | <global\_choice> | 🡪 | <array> |
|  | <global\_choice> | 🡪 | <task> |
|  | <global\_choice> | 🡪 | <object> |
|  | <global\_choice> | 🡪 | λ |
|  | <object> | 🡪 | <object1> |
|  | <object> | 🡪 | λ |
|  | <object1> | 🡪 | Object id Start Var id as <dtype> . <objdec\_choice> End <varname> |
|  | <objdec\_choice> | 🡪 | <objvar> . <objdec\_choice> |
|  | <objdec\_choice> | 🡪 | <var> . <objdec\_choice> |
|  | <objdec\_choice> | 🡪 | λ |
|  | <dtype> | 🡪 | Int |
|  | <dtype> | 🡪 | Double |
|  | <dtype> | 🡪 | Char |
|  | <dtype> | 🡪 | String |
|  | <dtype> | 🡪 | Boolean |
|  | <var> | 🡪 | Var id as <dtype> |
|  | <objvar> | 🡪 | Object id id |
|  | <varname> | 🡪 | id <varnames> |
|  | <varname> | 🡪 | λ |
|  | <varnames> | 🡪 | ; <varname> |
|  | <varnames> | 🡪 | λ |
|  | <task> | 🡪 | Task id <tparam> as <return> |
|  | <return> | 🡪 | Null |
|  | <return> | 🡪 | <dtype> |
|  | <tparam> | 🡪 | ( Var id as <dtype> <tparams> ) |
|  | <tparam> | 🡪 | λ |
|  | <tparams> | 🡪 | ; Var id as <dtype> <tparams> |
|  | <tparams> | 🡪 | λ |
|  | <array> | 🡪 | Array id of <size> as <dtype> |
|  | <varlet> | 🡪 | <type> id <init> |
|  | <type> | 🡪 | Var |
|  | <type> | 🡪 | Let |
|  | <init> | 🡪 | <value1> <ids1> as Int |
|  | <init> | 🡪 | <value2> <ids2> as Double |
|  | <init> | 🡪 | <value3> <ids3> as Char |
|  | <init> | 🡪 | <value4> <ids4> as String |
|  | <init> | 🡪 | <value5> <ids5> as Boolean |
|  | <ids1> | 🡪 | ; id <value1> <ids1> |
|  | <ids1> | 🡪 | λ |
|  | <ids2> | 🡪 | ; id <value2> <ids2> |
|  | <ids2> | 🡪 | λ |
|  | <ids3> | 🡪 | ; id <value3> <ids3> |
|  | <ids3> | 🡪 | λ |
|  | <ids4> | 🡪 | ; id <value4> <ids4> |
|  | <ids4> | 🡪 | λ |
|  | <ids5> | 🡪 | ; id <value5> <ids5> |
|  | <ids5> | 🡪 | λ |
|  | <size> | 🡪 | numlit <sizes> |
|  | <sizes> | 🡪 | by numlit |
|  | <sizes> | 🡪 | λ |
|  | <value1> | 🡪 | is <tvalue1> |
|  | <value1> | 🡪 | λ |
|  | <value2> | 🡪 | is <tvalue2> |
|  | <value2> | 🡪 | λ |
|  | <value3> | 🡪 | is <tvalue3> |
|  | <value3> | 🡪 | λ |
|  | <value4> | 🡪 | is <tvalue4> |
|  | <value4> | 🡪 | λ |
|  | <value5> | 🡪 | is <tvalue5> |
|  | <value5> | 🡪 | λ |
|  | <tvalue1> | 🡪 | <numvalue> <Operations1> |
|  | <tvalue1> | 🡪 | ( <tvalue1> ) <Op1> |
|  | <tvalue2> | 🡪 | <doublevalue> <Operations2> |
|  | <tvalue2> | 🡪 | ( <tvalue2> ) <Op2> |
|  | <tvalue3> | 🡪 | charlit |
|  | <tvalue4> | 🡪 | stringlit |
|  | <tvalue5> | 🡪 | boollit |
|  | <numvalue> | 🡪 | intlit |
|  | <numvalue> | 🡪 | <incdecvar> |
|  | <numvalue> | 🡪 | <objelement> |
|  | <Operations1> | 🡪 | <mathOp> <numvalue> <Operations1> |
|  | <Operations1> | 🡪 | λ |
|  | <Op1> | 🡪 | <Operations1> |
|  | <Op1> | 🡪 | <mathOp> ( <tvalue1> ) |
|  | <Op1> | 🡪 | λ |
|  | <doublevalue> | 🡪 | doublelit |
|  | <doublevalue> | 🡪 | <incdecvar> |
|  | <doublevalue> | 🡪 | <objelement> |
|  | <Operations2> | 🡪 | <mathOp> <doublevalue> <Operations2> |
|  | <Operations2> | 🡪 | λ |
|  | <Op2> | 🡪 | <Operations2> |
|  | <Op2> | 🡪 | <mathOp> ( <tvalue2> ) |
|  | <Op2> | 🡪 | λ |
|  | <mathOp> | 🡪 | + |
|  | <mathOp> | 🡪 | - |
|  | <mathOp> | 🡪 | \* |
|  | <mathOp> | 🡪 | / |
|  | <mathOp> | 🡪 | % |
|  | <incdec> | 🡪 | ++ |
|  | <incdec> | 🡪 | -- |
|  | <relop1> | 🡪 | == |
|  | <relop1> | 🡪 | != |
|  | <relop1> | 🡪 | > <equal> |
|  | <relop1> | 🡪 | < <equal> |
|  | <relop2> | 🡪 | == |
|  | <relop2> | 🡪 | != |
|  | <logop1> | 🡪 | && |
|  | <logop1> | 🡪 | || |
|  | <logop2> | 🡪 | ! |
|  | <logop2> | 🡪 | λ |
|  | <equal> | 🡪 | = |
|  | <equal> | 🡪 | λ |
|  | <body> | 🡪 | <statements> |
|  | <statements> | 🡪 | <functions> . <statements> |
|  | <statements> | 🡪 | λ |
|  | <functions> | 🡪 | <vardec> |
|  | <functions> | 🡪 | <varinit> |
|  | <functions> | 🡪 | <incdecvar> |
|  | <functions> | 🡪 | <array> |
|  | <functions> | 🡪 | Clear |
|  | <functions> | 🡪 | <arrayinit> |
|  | <functions> | 🡪 | <io\_statement> |
|  | <functions> | 🡪 | <taskvar> |
|  | <functions> | 🡪 | <objvar> |
|  | <functions> | 🡪 | <objinit> |
|  | <functions> | 🡪 | <condstate> |
|  | <functions> | 🡪 | <loopstate> |
|  | <functions> | 🡪 | <option> |
|  | <functions> | 🡪 | λ |
|  | <vardec> | 🡪 | Var id <vardec\_init> |
|  | <vardec\_init> | 🡪 | <init> |
|  | <vardec\_init> | 🡪 | as <dtype> |
|  | <varinit> | 🡪 | id = <varinits> |
|  | <varinits> | 🡪 | <tvalue> |
|  | <varinits> | 🡪 | <task\_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> |
|  | <task\_id> | 🡪 | id ( <param> ) |
|  | <taskvar> | 🡪 | <task\_id> |
|  | <param> | 🡪 | <tvalue> <params> |
|  | <param> | 🡪 | λ |
|  | <params> | 🡪 | ; <param> |
|  | <params> | 🡪 | λ |
|  | <tvalue> | 🡪 | intlit |
|  | <tvalue> | 🡪 | doublelit |
|  | <tvalue> | 🡪 | charlit |
|  | <tvalue> | 🡪 | stringlit |
|  | <tvalue> | 🡪 | boollit |
|  | <tvalue> | 🡪 | <input\_id> |
|  | <incdecvar> | 🡪 | <incdec> id |
|  | <incdecvar> | 🡪 | id <incdec> |
|  | <objelement> | 🡪 | <incdec> <element> |
|  | <objelement> | 🡪 | <element> <incdec> |
|  | <element> | 🡪 | id @ id <subelement> |
|  | <subelement> | 🡪 | @ id <subelement> |
|  | <subelement> | 🡪 | λ |
|  | <input\_id> | 🡪 | id <subelement> |
|  | <objinit> | 🡪 | <element> = <tvalue> |
|  | <arrayinit> | 🡪 | id [ <index> ] <multi> = <tvalue> |
|  | <multi> | 🡪 | [ <index> ] |
|  | <multi> | 🡪 | λ |
|  | <index> | 🡪 | numlit |
|  | <index> | 🡪 | id |
|  | <cond\_loop> | 🡪 | <functions> . <control> . <cond\_loop> |
|  | <cond\_loop> | 🡪 | λ |
|  | <control> | 🡪 | Skip |
|  | <control> | 🡪 | Stop |
|  | <control> | 🡪 | λ |
|  | <condstate> | 🡪 | If <logop2> ( <conditions> ) <cond\_loop> <if2> <if3> EndIf |
|  | <if2> | 🡪 | Or <conditions> <cond\_loop> <if2> |
|  | <if2> | 🡪 | λ |
|  | <if3> | 🡪 | Otherwise <cond\_loop> <control> |
|  | <if3> | 🡪 | λ |
|  | <conditions> | 🡪 | <ids> <b> |
|  | <conditions> | 🡪 | <logop2> <conds> |
|  | <conds> | 🡪 | ( <conds\_choice> ) <b> |
|  | <conds> | 🡪 | <input\_id> |
|  | <conds> | 🡪 | boollit |
|  | <conds> | 🡪 | <task\_id> |
|  | <conds\_choice> | 🡪 | <ids> |
|  | <conds\_choice> | 🡪 | <task\_id> <Ops> |
|  | <ids> | 🡪 | <input\_id> <a\_null> |
|  | <ids> | 🡪 | <logop2> <ids\_null> <a\_null> |
|  | <ids> | 🡪 | <tvalue> <a> |
|  | <ids> | 🡪 | <task\_id> |
|  | <ids\_null> | 🡪 | <tvalue5> |
|  | <ids\_null> | 🡪 | <task\_id> |
|  | <Ops> | 🡪 | <Op1> |
|  | <Ops> | 🡪 | <Op2> |
|  | <Ops> | 🡪 | λ |
|  | <a\_null> | 🡪 | <a> |
|  | <a\_null> | 🡪 | λ |
|  | <a> | 🡪 | <relop1> <a1> |
|  | <a> | 🡪 | <relop2> <a2> |
|  | <a1> | 🡪 | id |
|  | <a1> | 🡪 | <tvalue1> |
|  | <a1> | 🡪 | <tvalue2> |
|  | <a1> | 🡪 | <tvalue3> |
|  | <a1> | 🡪 | <tvalue4> |
|  | <a2> | 🡪 | <logop2> <tvalue5> |
|  | <a2> | 🡪 | <cvalue2> |
|  | <b> | 🡪 | <logop1> <conditions> |
|  | <b> | 🡪 | λ |
|  | <cvalue2> | 🡪 | <logop2> boollit |
|  | <option> | 🡪 | <option1> |
|  | <option> | 🡪 | <option2> |
|  | <option> | 🡪 | <option3> |
|  | <option1> | 🡪 | Option <input\_id> Start State intlit : <statements> Stop . <state1> <default> End |
|  | <option2> | 🡪 | Option <input\_id> Start State charlit : <statements> Stop . <state2> <default> End |
|  | <option3> | 🡪 | Option <input\_id> Start State 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 |
|  | <loopstate> | 🡪 | Until ( <conditions> ) <cond\_loop> Loop |
|  | <loopstate> | 🡪 | Do <cond\_loop> Until ( <conditions> ) |
|  | <loopstate> | 🡪 | For ( <initialize> ; <cond> ; <inc\_dec> ) <cond\_loop> Loop |
|  | <initialize> | 🡪 | id = intlit |
|  | <initialize> | 🡪 | λ |
|  | <cond> | 🡪 | id <relop1> intlit |
|  | <cond> | 🡪 | λ |
|  | <inc\_dec> | 🡪 | <incdecvar> |
|  | <inc\_dec> | 🡪 | λ |
|  | <taskdef> | 🡪 | <taskhead> <returntype> <dtype> End . <taskdef> |
|  | <taskdef> | 🡪 | λ |
|  | <taskhead> | 🡪 | Task id <tparam> as |
|  | <returntype> | 🡪 | Int <taskbody> . Response intlit . |
|  | <returntype> | 🡪 | Double <taskbody> . Response doublelit . |
|  | <returntype> | 🡪 | Char <taskbody> . Response charlit . |
|  | <returntype> | 🡪 | String <taskbody> . Response stringlit . |
|  | <returntype> | 🡪 | Null <taskbody> . |
|  | <taskbody> | 🡪 | Start <statements> |
|  | <Start> | 🡪 | <program> # |