**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> | 🡪 | Object id Start Var id as <dtype>. <objdec> End <varname>. |
|  | <objdec> | 🡪 | <objdec\_choice><objdec> |
|  | <objdec> | 🡪 | Λ |
|  | <objdec\_choice> | 🡪 | <objvar> |
|  | <objdec\_choice> | 🡪 | <var> |
|  | <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> |
|  | <ids2> | 🡪 | ; id <value2><ids2> |
|  | <ids3> | 🡪 | ; id <value3><ids3> |
|  | <ids4> | 🡪 | ; id <value4><ids4> |
|  | <ids5> | 🡪 | ; id <value5><ids5> |
|  | <ids1> | 🡪 | Λ |
|  | <ids2> | 🡪 | Λ |
|  | <ids3> | 🡪 | Λ |
|  | <ids4> | 🡪 | Λ |
|  | <ids5> | 🡪 | Λ |
|  | <size> | 🡪 | Numlit<sizes> |
|  | <sizes> | 🡪 | by Numlit |
|  | <sizes> | 🡪 | Λ |
|  | <dtype1> | 🡪 | Int |
|  | <dtype2> | 🡪 | Double |
|  | <dtype3> | 🡪 | Char |
|  | <dtype4> | 🡪 | String |
|  | <dtype5> | 🡪 | Boolean |
|  | <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> |
|  | <numvalue> | 🡪 | Intlit |
|  | <numvalue> | 🡪 | <incdecvar> |
|  | <numvalue> | 🡪 | <objelement> |
|  | <tvalue1> | 🡪 | (<tvalue1>)<Op1> |
|  | <Operations1> | 🡪 | <mathOp><numvalue><Operations1> |
|  | <Operations1> | 🡪 | Λ |
|  | <Op1> | 🡪 | <Operations1> |
|  | <Op1> | 🡪 | <mathOp>(<tvalue1>) |
|  | <Op1> | 🡪 | Λ |
|  | <tvalue2> | 🡪 | <doublevalue><Operations2> |
|  | <doublevalue> | 🡪 | Doublelit |
|  | <doublevalue> | 🡪 | <incdecvar> |
|  | <doublevalue> | 🡪 | <objelement> |
|  | <tvalue2> | 🡪 | (<tvalue2>)<Op2> |
|  | <Operations2> | 🡪 | <mathOp><doublevalue><Operations2> |
|  | <Operations2> | 🡪 | Λ |
|  | <Op2> | 🡪 | <Operations2> |
|  | <Op2> | 🡪 | <mathOp>(<tvalue2>) |
|  | <Op2> | 🡪 | Λ |
|  | <tvalue3> | 🡪 | Charlit |
|  | <tvalue4> | 🡪 | Stringlit |
|  | <tvalue5> | 🡪 | Boollit |
|  | <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> | 🡪 | <arrayinit> |
|  | <functions> | 🡪 | <io\_statement> |
|  | <functions> | 🡪 | <taskvar> |
|  | <functions> | 🡪 | <objvar> |
|  | <functions> | 🡪 | <objinit> |
|  | <functions> | 🡪 | <condstate> |
|  | <functions> | 🡪 | <loopstate> |
|  | <functions> | 🡪 | <option> |
|  | <functions> | 🡪 | Λ |
|  | <vardec> | 🡪 | Var <init> |
|  | <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> | 🡪 | id |
|  | <task\_id> | 🡪 | id(<param>) |
|  | <taskvar> | 🡪 | <task\_id>. |
|  | <param> | 🡪 | <tvalue><params> |
|  | <param> | 🡪 | Λ |
|  | <params> | 🡪 | ; <tvalue><params> |
|  | <params> | 🡪 | Λ |
|  | <tvalue> | 🡪 | <tvalue1> |
|  | <tvalue> | 🡪 | <tvalue2> |
|  | <tvalue> | 🡪 | <tvalue3> |
|  | <tvalue> | 🡪 | <tvalue4> |
|  | <tvalue> | 🡪 | <tvalue5> |
|  | <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[<size>] = <tvalue>. |
|  | <cond\_loop> | 🡪 | <functions><control><cond\_loop> |
|  | <cond\_loop> | 🡪 | Λ |
|  | <condstate> | 🡪 | If <logop2>(<conditions>)<cond\_loop><if2 ><if3>EndIf. |
|  | <if2> | 🡪 | Or <conditions><cond\_loop><if2> |
|  | <if2> | 🡪 | Λ |
|  | <if3> | 🡪 | Otherwise <cond\_loop><control> |
|  | <conditions> | 🡪 | <ids><Ops><b> |
|  | <conditions> | 🡪 | (<ids>)<Ops><b> |
|  | <conditions> | 🡪 | <logop2><conds> |
|  | <conds> | 🡪 | <input\_id> |
|  | <conds> | 🡪 | Boollit |
|  | <conds> | 🡪 | <task\_id> |
|  | <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> |
|  | <a2> | 🡪 | <tvalue3> |
|  | <a2> | 🡪 | <tvalue4> |
|  | <a2> | 🡪 | <logop2><tvalue5> |
|  | <a2> | 🡪 | <cvalue2> |
|  | <b> | 🡪 | <logop1><conditions> |
|  | <b> | 🡪 | Λ |
|  | <cvalue1> | 🡪 | Intlit |
|  | <cvalue1> | 🡪 | Doublit |
|  | <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 |
|  | <control> | 🡪 | Skip. |
|  | <control> | 🡪 | Stop. |
|  | <control> | 🡪 | 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> | 🡪 | Task id <tparam> as <return>. Start <statements> End. <taskdef> |
|  | <taskdef> | 🡪 | Λ |