# **Grammar**

|  |  |  |
| --- | --- | --- |
| Program | ::= | **Programstart** Block **Programend** |
| Block | ::= | **declare** Declatations **declareend** **do** ComandList **doend** |
| Declaration | ::= | Function InitializationTo0 |
|  | | | InitializationTo0 |
|  | | | Function |
|  | | | Ε |
| Function | ::= | Type FunctionName **(**ArgList**)** Block |
| ArgList | := | Type |
|  | | | Type**,** ArgList |
|  | | | E |
| InitializationTo0 | ::= | Type listName |
| listName | ::= | listName **,** varName ; |
|  | | | varName **;** |
| ComandList | ::= | SingleComand |
|  | | | SingleComand ComandList |
|  | | | E |
| SingleComand | ::= | Assignment |
|  | | | LoopBlock |
|  | | | ConditionBlock |
|  | | | FunctionCall |
|  | | | **end ;** |
|  | | | **giveBackWith** Expression **;** |
|  | | | **giveBackWith nothing ;** |
| Expression | ::= | varName**;** |
|  | | | Operation |
|  | | | LiteralNumber |
|  | | | FunctionCall |
| Assignment | ::= | VarName **<-** Expression **;** |
|  | := | Expression **->** VarName **;** |
| LoopBlock | ::= | ForLoop |
|  | | | WhileLoop |
| ConditionBlock | ::= | IfStatement |
|  | ::= | SwitchStatement |
| FunctionCall | ::= | FunctionName**(**listName**);** |
| Operation | ::= | Operation **.Operator** Operation |
|  | | | **.Operator** Operation |
|  | | | varName |
|  | | | **Number** |
| ForLoop | ::= | **for(**Assignment**;** Condition**;** **.Operator** **Number) start** ComandList **forend** |
| WhileLoop | ::= | **while(**Operation**) start** ComandList **whileend** |
| IfStatement | ::= | **if(**Operation**) start** ComandList ElseIFBlock **fi** |
| ElsifBlock | ::= | ElseIfBlock **else** ComandList |
|  | | | **else if (**Operation**) start** ComandList ElseIfBlock |
|  | | | E |
| SwitchStatement | ::= | **switch (**varName**):** CaseBlock DefaultBlock **switchend** |
| CaseBlock | ::= | **case** Value **start** ComandList **end** CaseBlock |
|  | | | **case** Value **start** CaseBlock |
|  | | | E |
| DefaultBlock | ::= | **default start** ComandList **end** |
|  |  | E |
| LiteralNumber | ::= | Number**;** |
|  | | | **|**char**|;** |
|  | | | **|**char**|,** LiteralNumber**;** |