# **Grammar**

|  |  |  |
| --- | --- | --- |
| Program | ::= | **Programstart** Block **Programend** |
| Block | ::= | **declare** Declatations **declareend** **do** Comand\* **doend** |
| Declaration | ::= | Function\* InitializationTo0\* |
| Function | ::= | Type FunctionName **(**Type\***)** Block |
| InitializationTo0 | ::= | Type VarName (,VarName)\***;** |
| Comand | ::= | (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**(**varName\***);** |
| Operation | ::= | ((varName| LiteralNumber|Negation) (Operator(varName| LiteralNumber|Negation| FunctionCall))\*)**;** |
| Negation | ::= | **.not** Expression |
| ForLoop | ::= | **for(**Assignment**;** Operation**;** Operator LiteralNumber**) start** Comand **forend** |
| WhileLoop | ::= | **while(**Operation**) start** Comand\* **whileend** |
| IfStatement | ::= | **if(**Operation**) start** Comand\* (**else if (**Operation**) start** Comand\*)\*(**else** Comand\*)? **fi** |
| SwitchStatement | ::= | **switch (**VarName**):** (**case** Value **start** Comand\* **end**)+ (**default start** Comand\* **end**)? **switchend** |
| Operator | ::= | \**.**(\**+**|**-**|\**\***|**/**|**%**|**and**|**or**|**<**|**<=**|**>**|**>=**|**==**) |
| LiteralNumber | ::= | (**+**|**-**)?[**0**-**9**]+ |
| VarName | ::= | ([**a**-**z**]|[**A**-**Z**]|\_) ([**a**-**z**]|[**A**-**Z**]|[**0**-**9**]|\_)\* |