

- Program:

Program -> **Functions**

Functions -> **function functions** | empty

- Function:

Function -> "function" "identifier" ";" "beginparams" **Dec** "endparams" "beginlocals" **Dec** "endlocals"
"beginbody" **Statements** "endbody"

- Declaration:

Declaration -> **Identifiers** ":" "integer" | **Identifiers** ":" "array" "[" "number" "]" "of" "integer" | **Identifiers** ";" "enum"
"(" **Identifiers** ")" **Dec** "endparams" "beginlocals" **Dec** "endlocals" "beginbody" **Statements** "endbody"

Dec -> Declaration ";" **Dec** | empty

Identifiers -> **Ident** | **Identifiers** ";" **Ident**

Ident -> "ident"

- Statement:

Statement -> **var** ":" **expression** | "if" **bool-expr** "then" **statements** "endif" | "if" **bool-expr** "then" **statements**
"else" **statements** "endif" | "while" **bool-expr** "beginloop" **statements** "endloop" | "do" "beginloop" **statements**
"endloop" "while" **bool-expr** | "for" **var** ":" "number" ";" **bool-expr** ";" **var** ":" **expression** "beginloop" **statements**
"endloop" | "read" **var varLoop** | "write" **var varLoop** | "continue" | "return" **expression**

Statements -> **Statement** ";" **statements** | empty

- Bool-expr:

Bool-expr -> **relation-and-exprs** | **bool-expr** "or" **relation-and-expr**

- Relation-and-expr:

Relation-and-expr -> **relation-expr** | **relation-and-expr** "and" **relation-expr**

- Relation-expr:

Relation-expr -> **expression comp expression** | "not" **expression comp expression** | "true" | "not" "true" |
"false" | "not" "false" | "(" **bool-expr** ")" | "not" "(" **bool-expr** ")"

- Comp:

Comp -> "==" | "<>" | "<" | ">" | "<=" | ">="

- Expression:

Expression -> **multiplicative_expr addSubExpr**

addSubExpr -> "+" **expression** | "-" **expression** | empty

- Multiplicative-expr:

Multiplicative-expr -> **term** | **term "*" multiplicative_expr** | **term "/" multiplicative_expr** | **term "%" multiplicative_expr**

- Term:

Term -> **var** | "number" | "(" **expression** ")" | "-" **var** | "-" "number" | "-" "(" **expression** ")" | **Ident** "(" **expression** **expressionLoop** ")"

expressionLoop -> "," **expression expressionLoop** | empty

- Var:

Var -> **Ident** | **Ident** "[" **expression** "]"

varLoop -> "," **var varLoop** | empty