

Key:

- Italics: terminal
- Capitalized: non-terminal

Program

PROGRAM -> FUNCTION PROGRAM | ϵ

Function

FUNCTION -> *function identifier ; beginparams* DECLARATION_LOOP *endparams beginlocals*
DECLARATION_LOOP *endlocals beginbody* STATEMENT_LOOP *endbody*

Declaration

DECLARATION_LOOP -> DECLARATION ; DECLARATION_LOOP | ϵ

DECLARATION -> IDENTIFIER_LOOP : *array [number] of integer* | IDENTIFIER_LOOP :
integer | IDENTIFIER_LOOP : *enum (IDENTIFIER_LOOP)*

Identifier Loop

IDENTIFIER_LOOP -> *identifier* | *identifier , IDENTIFIER_LOOP*

Statement

STATEMENT_LOOP -> STATEMENT ; STATEMENT_LOOP | STATEMENT ;

STATEMENT -> VAR := EXPRESSION |

if BOOL-EXPR *then* STATEMENT_LOOP *ELSE endif* |

while BOOL-EXPR *beginloop* STATEMENT_LOOP *endloop* |

do beginloop STATEMENT_LOOP *endloop while* BOOL-EXPR |

read VAR_LOOP |

write VAR_LOOP |

continue |

return EXPRESSION

ELSE -> *else* STATEMENT_LOOP | ϵ

Bool-Expr

BOOL-EXPR -> RELATION-AND-EXPR | RELATION-AND-EXPR *or* BOOL-EXPR

Relation-And-Expr

RELATION-AND-EXPR -> RELATION-EXPR | RELATION-EXPR *and* RELATION-AND-EXPR

Relation-Expr

RELATION-EXPR -> *not* RELATION_EXPR_BODY | RELATION_EXPR_BODY

RELATION_EXPR_BODY -> EXPRESSION COMP EXPRESSION |

true |

false |

(BOOL-EXPR)

Comp

COMP -> == | <> | < | > | <= | >=

Expression

EXPRESSION -> MULTIPLICATIVE_EXPR | MULTIPLICATIVE_EXPR + EXPRESSION |

MULTIPLICATIVE_EXPR - EXPRESSION

EXPRESSION_LOOP -> EXPRESSION | EXPRESSION , EXPRESSION_LOOP | ϵ

Multiplicative-Expr

MULTIPLICATIVE_EXPR -> TERM | TERM * MULTIPLICATIVE_EXPR | TERM /

MULTIPLICATIVE_EXPR | TERM % MULTIPLICATIVE_EXPR

Term

TERM -> VAR | NUMBER | (EXPRESSION) |

-VAR | -NUMBER | - (EXPRESSION) |

identifier (EXPRESSION_LOOP)

Var

VAR_LOOP -> VAR | VAR , VAR_LOOP

VAR -> *identifier* | *identifier* [EXPRESSION]