PROG -> package main eol EOL DEF_FUNC eof

DEF_FUNC -> func id (PARAMS) FUNC_RETLIST_BODY eol EOL DEF_FUNC

DEF_FUNC -> ε

PARAMS -> id TYPE PARAMS_N

PARAMS -> ε /)

PARAMS_N -> , id TYPE PARAMS_N

PARAMS_N -> ε /)

FUNC_RETLIST_BODY -> (FUNC_BODY

FUNC_RETLIST_BODY -> { eol EOL STAT OPTIONAL_RET }

FUNC_BODY ->) { eol EOL STAT OPTIONAL_RET }

FUNC_BODY -> RETURN_TYPE { eol EOL STAT REQUIRED_RET eol EOL}

REQUIRED_RET -> return exp EXP_N

OPTIONAL_RET -> return eol EOL

OPTIONAL_RET -> eol EOL

EXP_N -> , exp EXP_N

EXP_N -> ε

STAT -> id ID_N/CALL_FUNC

STAT -> if exp { eol EOL STAT } else { eol EOL STAT eol EOL } eol EOL STAT

STAT -> for FOR_DEF exp ; id = exp { eol EOL STAT eol EOL } eol EOL STAT

STAT -> ε

FOR_DEF -> id INIT_DEF exp ;

FOR_DEF -> ε / ;

INIT_DEF -> :=

INIT_DEF -> =

ID_N-> , id ID_N

ID_N -> ε

//CALL_FUNC-> id (exp EXP_N) – NAHRADENE DO CALL_FUNC/ASSIGN

TYPE -> int

TYPE -> float64

TYPE -> string

TYPE -> id

EOL -> eol EOL

EOL -> ε

RETURN_TYPE -> TYPE RETURN_TYPE_N

RETURN_TYPE -> ε /)

RETURN_TYPE_N -> , TYPE RETURN_TYPE_N

RETURN_TYPE_N -> ε /)

CALL_FUNC/ASSIGN -> exp EXP_N eol EOL STAT

CALL_FUNC/ASSIGN -> (FUNC_PARAMS) EOL STAT

OPT_ID -> id

OPT_ID -> ε

FUNC_PARAMS -> exp FUNC_PARAMS_N

FUNC_PARAMS -> ε

FUNC_PARAMS_N -> , exp FUNC_PARAMS_N

FUNC_PARAMS_N -> ε

FOR_ASSIGN -> id = exp ;

FOR_ASSIGN -> ε

ID_N/CALL_FUNC -> ID_N INIT_DEF OPT_ID CALL_FUNC/ASSIGN

ID_N/CALL_FUNC -> (FUNC_PARAMS) EOL STAT