

1 CHANGE LOG v1.5

InnerStmt lze nově rozgenerovat na: *ReturnStmt*, *ExitStmt* a *ContinueStmt*. To způsobí, že půjde příkazy RETURN, EXIT a CONTINUE použít uvnitř složeného příkazu(if, scope, ...), ale navíc přibude povinnost kontrolovat, zda není RETURN použit mimo funkci (nebo EXIT a CONTINUE mimo cyklus).

2 LL Grammar IFJ17 (v1.5)

- (1) $Line \rightarrow Statement \ LineEnd$
- (2) $LineEnd \rightarrow \mathbf{EOL} \ Line$
- (3) $\quad \quad \quad | \ \varepsilon$
- (4) $Statement \rightarrow InnerStmt$
- (5) $\quad \quad \quad | \ FuncDecl$
- (6) $\quad \quad \quad | \ FuncDef$
- (7) $\underline{InnerStmt} \rightarrow VarDecl$
- (8) $\quad \quad \quad | \ Assignment$
- (9) $\quad \quad \quad | \ IfStmt$
- (10) $\quad \quad \quad | \ ScopeStmt$
- (11) $\quad \quad \quad | \ DoStmt$
- (12) $\quad \quad \quad | \ \textcolor{green}{ForStmt}$
- (13) $\quad \quad \quad | \ PrintStmt$
- (14) $\quad \quad \quad | \ InputStmt$
- (15) $\quad \quad \quad | \ \underline{ReturnStmt}$
- (16) $\quad \quad \quad | \ \underline{\textcolor{green}{ExitStmt}}$
- (17) $\quad \quad \quad | \ \underline{\textcolor{green}{ContinueStmt}}$
- (18) $\quad \quad \quad | \ \varepsilon$
- (19) $StmtSeq \rightarrow InnerStmt \ \mathbf{EOL} \ StmtSeq$
- (20) $\quad \quad \quad | \ \varepsilon$
- (21) $VarDecl \rightarrow \mathbf{DIM} \ VarDeclNext$
- (22) $\quad \quad \quad | \ \textcolor{green}{\mathbf{STATIC}} \ VarDef$
- (23) $VarDeclNext \rightarrow VarDef$

(24) | **SHARED** *VarDef*

(25) *VarDef* → **ID AS Type InitOpt**

(26) *InitOpt* → '=' *Expression*

(27) | ϵ

(28) *FuncDecl* → **DECLARE FUNCTION ID '(' Params ') AS Type**

(29) *Type* → **INTEGER**

(30) | **DOUBLE**

(31) | **STRING**

(32) | **BOOLEAN**

(33) *FuncDef* → **FUNCTION ID '(' Params ') AS Type EOL StmtSeq END FUNCTION**

(34) *ParamDecl* → **ID AS Type**

(35) *Params* → *ParamDecl ParamsNext*

(36) | ϵ

(37) *ParamsNext* → ', ' *ParamDecl ParamsNext*

(38) | ϵ

(39) *ReturnStmt* → **RETURN Expression**

(40) *Assignment* → **ID AssignOperator Expression**

(41) *InputStmt* → **INPUT ID**

(42) *PrintStmt* → **PRINT Expression ';' ExpressionList**

(43) *ExpressionList* → *Expression ';' ExpressionList*

(44) | ϵ

(45) *ScopeStmt* → **SCOPE EOL StmtSeq END SCOPE**

(46) *IfStmt* → **IF Expression THEN EOL StmtSeq IfStmtCont**

(47) *IfStmtCont* → **END IF**

(48) | **ELSE EOL StmtSeq END IF**

(49) | **ELSEIF Expression THEN EOL StmtSeq IfStmtEnd**

(50) *IfStmtEnd* → **END IF**

(51) | **ELSE EOL StmtSeq END IF**

(52) *DoStmt* → **DO DoStmtEnd**

(53) *DoStmtEnd* → *TestType Expression EOL StmtSeq LOOP*

(54)			EOL <u>StmtSeq</u> LOOP <i>TestType</i> <i>Expression</i>
(55)	<i>TestType</i>	→	WHILE
(56)			UNTIL
(57)	<u>ExitStmt</u>	→	EXIT <i>LoopType</i>
(58)	<u>ContinueStmt</u>	→	CONTINUE <i>LoopType</i>
(59)	<u>Looptype</u>	→	DO
(60)			FOR
(61)	<u>ForStmt</u>	→	FOR ID <i>TypeOpt</i> '=' <i>Expression</i> TO <i>Expression</i> <i>StepOpt</i> EOL <u>StmtSeq</u> NEXT <i>IdOpt</i>
(62)	<i>TypeOpt</i>	→	AS <i>Type</i>
(63)			ε
(64)	<i>StepOpt</i>	→	STEP <i>Expression</i>
(65)			ε
(66)	<i>IdOpt</i>	→	ID
(67)			ε
(68)	<i>AssignOperator</i>	→	'='
(69)			'-' =
(70)			'+' =
(71)			'*' =
(72)			'\' =' TOKEN_DIVI_ASSIGN
(73)			'/' =' TOKEN_DIVR_ASSIGN
(74)	<i>Expression</i>	→	ID
(75)			INT
(76)			REAL
(77)			STR TOKEN_STRING
(78)			TRUE
(79)			FALSE
(80)	= number of rules		

3 Komentář

- *Neterminály*: Psány kurzívou (*Statement*, *Expression*, ...).
- **TERMINÁLY(TOKENY)**: Terminály (**IF**, **LOOP**, ...) psány VELKÝMI PÍSMENY a vyznačeny **tučně**. Nepísmenné terminály ('=', '(', ')', ...) vyznačeny 'uvozovkami'.