

1 CHANGE LOG v1.6

Soubor s programem může být buď prázdný, a nebo pokud v něm něco je, tak musí obsahovat **scope** a **end scope**. Deklarace/definice funkcí (případně glob. proměnných) musí být právě před výše zmíněným **scope**. Odstranil jsem pravidlo *Expression*, protože již nikdy nebude použito.

2 LL Grammar IFJ17 (v1.6)

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

(25) | **STATIC** *VarDef*

(26) *SharedVar* → **DIM SHARED** *VarDef*

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

(28) *InitOpt* → '=' *Expression*

(29) | ϵ

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

(31) *Type* → **INTEGER**

(32) | **DOUBLE**

(33) | **STRING**

(34) | **BOOLEAN**

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

(36) *ParamDecl* → **ID AS** *Type*

(37) *Params* → *ParamDecl ParamsNext*

(38) | ϵ

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

(40) | ϵ

(41) *ReturnStmt* → **RETURN** *Expression*

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

(43) *InputStmt* → **INPUT ID**

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

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

(46) | ϵ

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

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

(49) *IfStmtCont* → **END IF**

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

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

(52) *IfStmtEnd* → **END IF**

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

(54) *DoStmt* → **DO** *DoStmtEnd*

(55)	<i>DoStmtEnd</i>	→	<i>TestType Expression</i>	EOL	<i>StmtSeq</i>	LOOP
(56)			EOL	<i>StmtSeq</i>	LOOP	<i>TestType Expression</i>
(57)	<i>TestType</i>	→	WHILE			
(58)			UNTIL			
(59)	<i>ExitStmt</i>	→	EXIT	<i>LoopType</i>		
(60)	<i>ContinueStmt</i>	→	CONTINUE	<i>LoopType</i>		
(61)	<i>Looptype</i>	→	DO			
(62)			FOR			
(63)	<i>ForStmt</i>	→	FOR	ID	<i>TypeOpt</i>	‘=‘ <i>Expression</i> TO <i>Expression</i> <i>StepOpt</i> EOL <i>StmtSeq</i> NEXT <i>IdOpt</i>
(64)	<i>TypeOpt</i>	→	AS	<i>Type</i>		
(65)			ε			
(66)	<i>StepOpt</i>	→	STEP	<i>Expression</i>		
(67)			ε			
(68)	<i>IdOpt</i>	→	ID			
(69)			ε			
(70)	<i>AssignOperator</i>	→	‘=‘			
(71)			‘−=‘			
(72)			‘+=‘			
(73)			‘*=‘			
(74)			‘\=‘	TOKEN_DIVI_ASSIGN		
(75)			‘/=‘	TOKEN_DIVR_ASSIGN		
(76)	= number of rules					

3 Komentář

- *Neterminály*: Psány kurzívou (*Global*, *ScopeStmt*, ...).
- **TERMINÁLY(TOKENY)**: Terminály (**IF**, **LOOP**, ...) psány VELKÝMI PÍSMENY a vyznačeny **tučně**. Nepísmenné terminály (**‘=‘**, **‘(‘**, **‘)’**, ...) vyznačeny **‘uvozovkami‘**.