

# IFJ2020 Grammar rules – *by Poli && Coacher*

## 1. Terminals

ID, K\_WORD, PLUS, MINUS, TIMES, DIV, EQ, L\_EQ, G\_EQ, LESS, GREAT, N\_EQ, L\_VINCL, R\_VINCL, LEFT\_BRAC{, RIGHT\_BRAC}, IF, ELSE, FOR, IS, ASSIGN, STRING\_LITERAL, INT\_LITERAL, COMMA, DOUBLE\_LITERAL, RETURN, PACKAGE\_MAIN, Q\_MARK (uvozovky), F\_MAIN (povinná funkce main), TRUE, FALSE, EOF, EOL, FUNC(jen speciální K word..), ASSIGN, IF, ELSE, FOR, SEMICOLON

## 2. Non-Terminals

<prog>, <func>, <param>, <param\_n>, <retval>, <retvals>, <body>, <f\_retval>, <f\_retvals>, <expr>, <func\_call>, <call-params>, <if>, <for>, <bool\_expr>, <for\_init>

## 3. Rules

-----Funkce, tělo programu, definice funkcí a návratových parametrů-----

- a. <prog> *// jelikož v IFJ20 nejsou globální proměnné tak povolujeme na začátku jen definice fci*
  - i. <prog> -> PACKAGE\_MAIN <func>
  - ii. <prog> -> EOF
- b. <func> *// definice funkce: func id (params) (retvals){ body } může následovat další funkce*
  - i. <func>-> FUNC ID L\_VINCL <param\_n> R\_VINCL <retvals> L\_BRAC <body> <f\_retvals> R\_BRAC EOL<func>
  - ii. <func> -> ε
- c. <param\_n> *// buď nejsou žádné parametry: () nebo (param)*
  - i. <param\_n> -> ε
  - ii. <param\_n> -> <param>
- d. <param> *// buď jeden parametr i, nebo víc parametrů ii*
  - i. <param> -> [INT, DOUBLE, STRING] ID
  - ii. <param>-> [INT, DOUBLE, STRING] ID COMMA <param>
- e. <retvals> *// návratové hodnoty nemusí být přítomny a nebo pokračuj na retval*
  - i. <retvals> -> L\_VINCL <retval> R\_VINCL
  - ii. <retvals> -> ε
- f. <retval> *// buď jedna návratová hodnota nebo více*
  - i. <retval> -> [INT, DOUBLE, STRING]
  - ii. <retval>-> [INT, DOUBLE, STRING] COMMA <retval>
- g. <f\_retvals> *// návratové hodnoty uvnitř těla funkce, nemusí být nebo pokračuj na f\_retval*
  - i. <f\_retval> -> ε
  - ii. <f\_retval> -> <f\_retval>
- h. <f\_retval> *// návratové hodnoty mohou být jeden či více výrazů*
  - i. <f\_retval> -> <expr>
  - ii. <f\_retval> -> <expr> COMMA <f\_retval>

-----Tělo funkce, cykly, aritmetické výrazy-----

- i. <body> *// jsme v těle funkce, buď přiřazujeme do proměnné výraz nebo volání funkce, také se může vyskytnout for, if nebo EOL*
  - i. <body> -> ID ASSIGN <expr> EOL <body>

- ii. `<body> -> ε`
- iii. `<body> -> ID ASSIGN <func-call> EOL <body>`
- iv. `<body> -> <exp> EOL <body>`
- v. `<body> -> <if> <body>`
- vi. `<body> -> <for> <body>`
- vii. `<body> -> EOL`
- j. `<func-call>` *// přiřazujeme z volání funkce*
  - i. `<func-call> -> ID L_VINCL <call_params> R_VINCL`
- k. `<call_params>` *// nemusí být žádné parametry z volání funkce, jinak pokračuj na call-param*
  - i. `<call_params> -> ε`
  - ii. `<call_params> -> <call-param>`
- l. `<call-param>` *// buď máme jeden parametr nebo více*
  - i. `<call-param> -> <expr>`
  - ii. `<call-param> -> <expr> COMMA <call-param>`
- m. `<if>` *// if obsahuje bool podmínku zároky tělo, else další závorky a tělo*
  - i. `<if> -> IF <bool_expr> R_BRAC EOL <body> L_BRAC ELSE R_BRAC EOL <body> L_BRAC`
- n. `<for>` *// for (init; condition; after)*
  - i. `<for> -> for L_VINCL <for_init> SEMICOLON <bool_expr> SEMICOLON <expr> R_VINCL L_BRAC EOL <body> R_BRAC`
- o. `<for_init>` *// to co může být v první podmínce ve for*
  - i. `<for_init> -> [INT, DOUBLE, STRING] ID ASSIGN <expr>`

#### 4. Nápořád

- a. Terminals
  - i. ID: identifikátor
  - ii. K\_WORD: klíčové slovo
  - iii. PLUS: +
  - iv. MINUS: -
  - v. TIMES: \*
  - vi. DIV: /
  - vii. L\_EQ: <=
  - viii. G\_EQ: >=
  - ix. LESS: <
  - x. GREAT: >
  - xi. N\_EQ: ≠
  - xii. L\_VINCL: (
  - xiii. R\_VINCL: )
  - xiv. LEFT\_BRAC: {
  - xv. RIGHT\_BRAC: }
  - xvi. ASSIGN: := (přiřazení)
  - xvii. STRING\_LITERAL: str(ing)
  - xviii. INT\_LITERAL: int(eger)
  - xix. DOUBLE\_LITERAL: double
  - xx. PACKAGE\_MAIN: "prolog"
  - xxi. Q\_MARK: " "
  - xxii. F\_MAIN: main (povinná funkce)
  - xxiii. EOF: end of file

- xxiv. EOL: end of line
- b. Non-terminals
  - i. <prog>: program
  - ii. <func>: funkce
  - iii. <body>: tělo
  - iv. <expr>: výraz
  - v. <type>: datový typ
  - vi. <param>: parametr
  - vii. <param\_n>: další parametr
  - viii. <retval>: návratová hodnota v definici funkce
  - ix. <retvals>: návratové hodnoty v definici funkce
  - x. <f\_retval>: návratová hodnota funkce
  - xi. <f\_retvals>: návratové hodnoty funkce
  - xii. <func\_call>: volání funkce
  - xiii. <call-params>: parametry při volání funkce