Gramatika pro jazyk IFJ24

# LL-Gramatika

1. **<PROGRAM>** **➜** **<PROLOG>** **<FUN\_DEF\_LIST> EOF**
2. **<PROLOG>** **➜** **const ifj = @import ( [prec\_expr] ) ;**
3. **<FUN\_DEF\_LIST>** **➜** **<FUN\_DEF>** **<FUN\_DEF\_LIST>**
4. **<FUN\_DEF\_LIST>** **➜** **ε**
5. **<FUN\_DEF>** **➜** **pub fn id ( <PARAMETERS> )** **<RETURN\_TYPE>** **<SEQUENCE>**
6. **<PARAMETERS>** **➜** **<PARAM\_LIST>**
7. **<PARAMETERS>** **➜** **ε**
8. **<PARAM\_LIST>** **➜** **<PARAM> <PARAM\_LIST\_REST>**
9. **<PARAM\_LIST\_REST>** **➜ , <PARAM\_LIST>**
10. **<PARAM\_LIST\_REST>** **➜ ε**
11. **<PARAM>** **➜** **id : <DATA\_TYPE>**
12. **<RETURN\_TYPE>** **➜** **<DATA\_TYPE>**
13. **<RETURN\_TYPE>** **➜** **void**
14. **<DATA\_TYPE>** **➜ i32**
15. **<DATA\_TYPE>** **➜ ?i32**
16. **<DATA\_TYPE>** **➜** **f64**
17. **<DATA\_TYPE>** **➜** **?f64**
18. **<DATA\_TYPE>** **➜** **[]u8**
19. **<DATA\_TYPE>** **➜** **?[]u8**
20. **<STATEMENT\_LIST>** **➜** **<STATEMENT>** **<STATEMENT\_LIST>**
21. **<STATEMENT\_LIST>** **➜** **ε**
22. **<STATEMENT>** **➜** **<VAR\_DEF>** **;**
23. **<STATEMENT>** **➜** **id <STATEMENT\_REST>** **;**
24. **<STATEMENT>** **➜** **\_ = <THROW\_AWAY>** **;**
25. **<STATEMENT>** **➜** **<IF>**
26. **<STATEMENT>** **➜** **<WHILE>**
27. **<STATEMENT>** **➜** **<RETURN>** **;**
28. **<STATEMENT>** **➜ ifj . id ( <ARGUMENTS> ) ;**
29. **<VAR\_DEF>** **➜** **<MODIFIABLE>** **id** **<POSSIBLE\_TYPE>** **=** **[prec\_expr]**
30. **<MODIFIABLE>** **➜** **var**
31. **<MODIFIABLE>** **➜** **const**
32. **<POSSIBLE\_TYPE>** **➜** **:** **<DATA\_TYPE>**
33. **<POSSIBLE\_TYPE>** **➜** **ε**
34. **<STATEMENT\_REST>** **➜ =** **[prec\_expr]**
35. **<STATEMENT\_REST>** **➜ ( <ARGUMENTS> )**
36. **<THROW\_AWAY>** **➜ [prec\_expr]**
37. **<IF>** **➜** **if ( [prec\_expr] )** **<NULL\_COND>** **<SEQUENCE>** **else** **<SEQUENCE>**
38. **<NULL\_COND>** **➜** **| id |**
39. **<NULL\_COND>** **➜** **ε**
40. **<SEQUENCE>** **➜** **{** **<STATEMENT\_LIST>** **}**
41. **<WHILE>** **➜** **while ( [prec\_expr]** **) <NULL\_COND>** **<SEQUENCE>**
42. **<RETURN>** **➜** **return** **<RETURN\_REST>**
43. **<RETURN\_REST>** **➜** **[prec\_expr]**
44. **<RETURN\_REST>** **➜** **ε**
45. **<ARGUMENTS>** **➜** **[prec\_expr]**

**Poznámky:**

**červené ... terminály**

**[prec\_expr] ... spec. terminál (předání kontroly precedenčnímu parseru)**

**<modré> ... NEterminály**

**ε ... epsilon (prázdný řetězec)**

# Precedenční syntaktická analýza

**Pravidla kombinující LL a precendenční syntaktickou analýzu:**

1. **<VAR\_DEF>** **➜** **<MODIFIABLE>** **id** **<POSSIBLE\_TYPE>** **=** **[prec\_expr]**
2. **<STATEMENT\_REST>** **➜ =** **[prec\_expr]**
3. **<THROW\_AWAY>** **➜** **[prec\_expr]**
4. **<IF>** **➜** **if ( [prec\_expr] )** **<NULL\_COND>** **<SEQUENCE>** **else** **<SEQUENCE>**
5. **<WHILE>** **➜** **while ( [prec\_expr]** **) <NULL\_COND>** **<SEQUENCE>**
6. **<RETURN\_REST>** **➜** **[prec\_expr]**
7. **<ARGUMENTS>** **➜** **[prec\_expr]**

**TERMINÁLY vystupující v precedenční analýze**

1. **Ostatní:**

* **ifj**
* **.**
* **(**
* **)**
* **$** (obecný konec vstupu)

1. **Operandy:**

* **id**
* **i32\_literal**
* **f64\_literal**
* **u8\_literal**
* **null\_literal**

1. **Operátory:**
2. **Relační operátory:**

* **==**
* **!=**
* **<**
* **>**
* **<=**
* **>=**

1. **Aditiviní operátory:**

* **+**
* **–**

1. **Multiplikativní operátory:**

* **\***
* **/**

**Množnina FOLLOW aka komunikační symboly mezi parsery**

**FOLOW\_VAR\_DEF = { ; }**

**FOLOW\_STATEMENT\_REST = { ; }**

**FOLOW\_THROW\_AWAY = { ; }**

**FOLOW\_IF = { ) }**

**FOLOW\_WHILE = { ) }**

**FOLOW\_RETURN\_REST = { ; }**

**FOLOW\_ARGUMENTS = { ) }**

**Redukční pravidla pro precedenční syntaktickou analýzu:**

1. **<EXPR>** ➜ **id**
2. **<EXPR>** ➜ **i32\_literal**
3. **<EXPR>** ➜ **f64\_literal**
4. **<EXPR>** ➜ **u8\_literal**
5. **<EXPR>** ➜ **null\_literal**
6. **<EXPR>** ➜ **<EXPR>** **+** **<EXPR>**
7. **<EXPR>** ➜ **<EXPR>** **–** **<EXPR>**
8. **<EXPR>** ➜ **<EXPR>** **\*** **<EXPR>**
9. **<EXPR>** ➜ **<EXPR>** **/** **<EXPR>**
10. **<EXPR>** ➜ **<EXPR>** **==** **<EXPR>**
11. **<EXPR>** ➜ **<EXPR>** **!=** **<EXPR>**
12. **<EXPR>** ➜ **<EXPR>** **<** **<EXPR>**
13. **<EXPR>** ➜ **<EXPR>** **>** **<EXPR>**
14. **<EXPR>** ➜ **<EXPR>** **<=** **<EXPR>**
15. **<EXPR>** ➜ **<EXPR>** **>=** **<EXPR>**
16. **<EXPR>** ➜ **(** **<EXPR>** **)**
17. **<EXPR>** ➜ **id (** **<ARG\_LIST>** **)**
18. **<EXPR>** ➜ **ifj . id (** **<ARG\_LIST>** **)**
19. **<ARG\_LIST>** ➜ **<EXPR> <ARG\_REST>**
20. **<ARG\_LIST>** **➜** **ε**
21. **<ARG\_REST>** **➜** **,** **<EXPR> <ARG\_REST>**
22. **<ARG\_REST>** **➜** **,**
23. **<ARG\_REST>** **➜** **ε**

**WARNING:  
TABULKA DOLE JE NEAKTUÁLNÍ**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **IN**  **STACK** | **id** | **lit** | **ifj** | **.** | **(** | **)** | **+** | **–** | **\*** | **/** | **==** | **!=** | **<** | **>** | **<=** | **>=** | **$** |
| **id** |  |  |  | **<** | **<** | **>** | **>** | **>** | **>** | **>** | **>** | **>** | **>** | **>** | **>** | **>** | **>** |
| **lit** |  |  |  |  |  | **>** | **>** | **>** | **>** | **>** | **>** | **>** | **>** | **>** | **>** | **>** | **>** |
| **ifj** |  |  |  | **<** |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **.** | **=** |  | **=** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **(** | **<** | **<** | **<** | **<** | **<** | **=** | **<** | **<** | **<** | **<** | **<** | **<** | **<** | **<** | **<** | **<** |  |
| **)** |  |  |  |  |  | **>** | **>** | **>** | **>** | **>** | **>** | **>** | **>** | **>** | **>** | **>** | **>** |
| **+** | **<** | **<** | **<** | **<** | **<** | **>** | **>** | **>** | **<** | **<** | **>** | **>** | **>** | **>** | **>** | **>** | **>** |
| **–** | **<** | **<** | **<** | **<** | **<** | **>** | **>** | **>** | **<** | **<** | **>** | **>** | **>** | **>** | **>** | **>** | **>** |
| **\*** | **<** | **<** | **<** | **<** | **<** | **>** | **>** | **>** | **>** | **>** | **>** | **>** | **>** | **>** | **>** | **>** | **>** |
| **/** | **<** | **<** | **<** | **<** | **<** | **>** | **>** | **>** | **>** | **>** | **>** | **>** | **>** | **>** | **>** | **>** | **>** |
| **==** | **<** | **<** | **<** | **<** | **<** | **>** | **<** | **<** | **<** | **<** |  |  |  |  |  |  | **>** |
| **!=** | **<** | **<** | **<** | **<** | **<** | **>** | **<** | **<** | **<** | **<** |  |  |  |  |  |  | **>** |
| **<** | **<** | **<** | **<** | **<** | **<** | **>** | **<** | **<** | **<** | **<** |  |  |  |  |  |  | **>** |
| **>** | **<** | **<** | **<** | **<** | **<** | **>** | **<** | **<** | **<** | **<** |  |  |  |  |  |  | **>** |
| **<=** | **<** | **<** | **<** | **<** | **<** | **>** | **<** | **<** | **<** | **<** |  |  |  |  |  |  | **>** |
| **>=** | **<** | **<** | **<** | **<** | **<** | **>** | **<** | **<** | **<** | **<** |  |  |  |  |  |  | **>** |
| **$** | **<** | **<** | **<** | **<** | **<** |  | **<** | **<** | **<** | **<** | **<** | **<** | **<** | **<** | **<** | **<** |  |