Gramatika pro jazyk IFJ24

# LL-Gramatika – Česky

1. **<program>** **➜** **<prolog>** **<definiceFunkcí> EOL**
2. **<prolog>** **➜** **const ifj = @import ( [prec\_výraz] ) ;**
3. **<definiceFunkcí>** **➜** **<deklaraceFunkce>** **<definiceFunkcí>**
4. **<definiceFunkcí>** **➜** **ε**
5. **<deklaraceFunkce>** **➜** **pub fn id ( <parametry> )** **<návratovýTyp>** **<blok>**
6. **<parametry>** **➜** **<seznamParametrů>**
7. **<parametry>** **➜** **ε**
8. **<seznamParametrů>** **➜** **<parametr> <seznamParamZbytek>**
9. **<seznamParamZbytek>** **➜ , <seznamParametrů>**
10. **<seznamParamZbytek>** **➜ ε**
11. **<parametr>** **➜** **id : <datovýTyp>**
12. **<návratovýTyp>** **➜** **<datovýTyp>**
13. **<návratovýTyp>** **➜** **void**
14. **<datovýTyp>** **➜ i32**
15. **<datovýTyp>** **➜ ?i32**
16. **<datovýTyp>** **➜** **f64**
17. **<datovýTyp>** **➜** **?f64**
18. **<datovýTyp>** **➜** **[]u8**
19. **<datovýTyp>** **➜** **?[]u8**
20. **<příkazy>** **➜** **<příkaz>** **<příkazy>**
21. **<příkazy>** **➜** **ε**
22. **<příkaz>** **➜** **<definiceProměnné >** **;**
23. **<příkaz>** **➜** **id <příkazZbytek>** **;**
24. **<příkaz>** **➜** **\_ = <zahodit>** **;**
25. **<příkaz>** **➜** **<if>**
26. **<příkaz>** **➜** **<while>**
27. **<příkaz>** **➜** **<return>** **;**
28. **<definiceProměnné>** **➜** **<modifikovatelnost>** **id** **<volitelněTyp>** **=** **[prec\_výraz]**
29. **<modifikovatelnost>** **➜** **var**
30. **<modifikovatelnost>** **➜** **const**
31. **<volitelněTyp>** **➜** **:** **<datovýTyp>**
32. **<volitelněTyp >** **➜** **ε**
33. **<příkazZbytek>** **➜ =** **[prec\_výraz]**
34. **<příkazZbytek>** **➜ ( <argumenty> )**
35. **<příkazZbytek>** **➜ ifj . id ( <argumenty> )**
36. **<zahodit>** **➜ [prec\_výraz]**
37. **<zahodit>** **➜ ( <argumenty> )**
38. **<zahodit>** **➜ ifj . id ( <argumenty> )**
39. **<if>** **➜** **if ( [prec\_výraz] )** **<nullPodmínka>** **<blok>** **else** **<blok>**
40. **<nullPodmínka>** **➜** **| id |**
41. **<nullPodmínka>** **➜** **ε**
42. **<blok>** **➜** **{** **<příkazy>** **}**
43. **<cyklus>** **➜** **while ( [prec\_výraz]** **) <nullPodmínka>** **<blok>**
44. **<return>** **➜** **return** **<returnZbytek>**
45. **<returnZbytek>** **➜** **[prec\_výraz]**
46. **<returnZbytek>** **➜** **ε**
47. **<argumenty>** **➜** **<seznamArgumentů>**
48. **<argumenty>** **➜** **ε**
49. **<seznamArgumentů>** **➜** **[prec\_výraz]** **<dalšíArgument>**
50. **<dalšíArgument>** **➜** **,** **[prec\_výraz]** **<dalšíArgument>**
51. **<dalšíArgument>** **➜** **ε**

# LL-Gramatika – Anglicky

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. **<VAR\_DEF>** **➜** **<MODIFIABLE>** **id** **<POSSIBLE\_TYPE>** **=** **[prec\_expr]**
29. **<MODIFIABLE>** **➜** **var**
30. **<MODIFIABLE>** **➜** **const**
31. **<POSSIBLE\_TYPE>** **➜** **:** **<DATA\_TYPE>**
32. **<POSSIBLE\_TYPE>** **➜** **ε**
33. **<STATEMENT\_REST>** **➜ =** **[prec\_expr]**
34. **<STATEMENT\_REST>** **➜ ( <ARGUMENTS> )**
35. **<STATEMENT\_REST>** **➜ ifj . id ( <ARGUMENTS> )**
36. **<THROW\_AWAY>** **➜ [prec\_expr]**
37. **<THROW\_AWAY>** **➜ ( <ARGUMENTS> )**
38. **<THROW\_AWAY>** **➜ ifj . id ( <ARGUMENTS> )**
39. **<IF>** **➜** **if ( [prec\_expr] )** **<NULL\_COND>** **<SEQUENCE>** **else** **<SEQUENCE>**
40. **<NULL\_COND>** **➜** **| id |**
41. **<NULL\_COND>** **➜** **ε**
42. **<SEQUENCE>** **➜** **{** **<STATEMENT\_LIST>** **}**
43. **<WHILE>** **➜** **while ( [prec\_expr]** **) <NULL\_COND>** **<SEQUENCE>**
44. **<RETURN>** **➜** **return** **<RETURN\_REST>**
45. **<RETURN\_REST>** **➜** **[prec\_expr]**
46. **<RETURN\_REST>** **➜** **ε**
47. **<ARGUMENTS>** **➜** **<ARG\_LIST>**
48. **<ARGUMENTS>** **➜** **ε**
49. **<ARG\_LIST>** **➜** **[prec\_expr]** **<ARG>**
50. **<ARG>** **➜** **,** **[prec\_expr]** **<ARG>**
51. **<ARG>** **➜** **ε**

**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. **<ARG\_LIST>** **➜** **[prec\_expr]** **<ARG>**
8. **<ARG>** **➜** **,** **[prec\_expr]** **<ARG>**

**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 = { ; }**

**FOLOW\_ARG\_LIST= { , , ) }**

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