Gramatika

1. <program> → package main EOL <def\_func\_o> EOF

# Tělo funkce

1. <body> → <return> <body>
2. <body> → <if> EOL <body>
3. <body> → <for> EOL <body>
4. <body> → <id> <after\_id> EOL <body>
5. <body> → \_ <ids\_lo> = <assign\_r> EOL <body>
6. <body> → ε
7. <after\_id> → <ids\_lo> = <assign\_r>
8. <after\_id> → <func>
9. <after\_id> → <def\_var>

# For cyklus

1. <for> → for <def> ; <expr> ; <assign> { EOL <body> }
2. <assign> → <id> <ids\_lo> = <assign\_r>
3. <assign> → \_ <ids\_lo> = <assign\_r>
4. <assign> → ε
5. <def> → <id> <def\_var>
6. <def> → ε

# Definice proměnné

1. <def\_var> → := <id\_expr>

# Identifikátory levé strany přiřazení

1. <ids\_lo> → , <ids\_l> <ids\_lo>
2. <ids\_lo> → ε
3. <ids\_l> → \_
4. <ids\_l> → id

Pravá strana přiřazení

1. <assign\_r> → <expr\_literal> <ids\_exprs\_o>
2. **<assign\_r> → id <func\_ids\_exprs\_o>**
3. **<ids\_exprs\_o> →** **, <id\_expr> <ids\_exprs\_o>**
4. **<id\_expr> → id**
5. **<id\_expr> → <literal>**
6. **<id\_expr> → <expr>**
7. **<func\_ids\_exprs\_o> → <func>**
8. **<func\_ids\_exprs\_o> → <ids\_exprs\_o>**

Volání funkce

1. <func> → ( <params> )

# Parametry funkce

1. <params> → ε
2. **<params> → <id\_literal> <params\_o>**
3. **<params\_o>** **→ , <id\_literal> <params\_o>**
4. <params\_o> → ε

# Návrat z funkce

1. <return> → return <return\_v> EOL
2. **<return\_v> → <id\_expr> <ids\_exprs\_o>**
3. <return\_v> → ε

# Definice funkce

1. <def\_func\_o> → ε
2. <def\_func\_o> → <def\_func> EOL <def\_func\_o>
3. <def\_func> → func <id> ( <f\_params> ) <f\_types> { EOL <body> }

Formální parametry funkce

1. <f\_params> → ε
2. <f\_params> → <id> <type> <f\_params\_o>
3. <f\_params\_o> → ε
4. <f\_params\_o> → , <id> <type> <f\_params\_o>

# Návratové typy funkce

1. <f\_types> → ε
2. <f\_types> → ( <types> )
3. <types> → <type> <types\_o>
4. <types> → ε
5. <types\_o> → , <type> <types\_o>
6. <types\_o> → ε

# If/Else

1. <if> → if <expr> { EOL <body> } <else>
2. <else> → else { EOL <body> }
3. <else> → ε

# Datové typy

1. <type> → int
2. <type> → float
3. <type> → string
4. <id\_literal> → id
5. <id\_literal> → literal
6. <expr\_literal> → <expr>
7. <expr\_literal> → literal
8. <literal> → integer\_value
9. <literal> → float\_value
10. <literal> → string\_value

LL tabulka

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | package | main | eol | eof | id | \_ | = | for | ; | { | } | := | , | ( | ) | return | func | if | else | int | float | string | integer\_value | float\_value | string\_value |
| <program> | 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| <def\_func\_o> |  |  |  | 38 |  |  |  |  |  |  |  |  |  |  |  |  | 39 |  |  |  |  |  |  |  |  |
| <body> |  |  |  |  | 5 | 6 |  | 4 |  |  | 7 |  |  |  |  | 2 |  | 3 |  |  |  |  |  |  |  |
| <return> |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 35 |  |  |  |  |  |  |  |  |  |
| <if> |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 51 |  |  |  |  |  |  |  |
| <for> |  |  |  |  |  |  |  | 11 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| <after\_id> |  |  |  |  |  |  | 8 |  |  |  |  | 10 | 8 | 9 |  |  |  |  |  |  |  |  |  |  |  |
| <ids\_lo> |  |  |  |  |  |  | 19 |  |  |  |  |  | 18 |  |  |  |  |  |  |  |  |  |  |  |  |
| <assign\_r> |  |  |  |  | 23 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 22 | 22 | 22 |
| <func> |  |  |  |  |  |  |  |  |  |  |  |  |  | 30 |  |  |  |  |  |  |  |  |  |  |  |
| <def\_var> |  |  |  |  |  |  |  |  |  |  |  | 17 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| <def> |  |  |  |  | 15 |  |  |  | 16 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| <assign> |  |  |  |  | 12 | 13 |  |  |  | 14 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| <ids\_l> |  |  |  |  | 21 | 20 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| <ids\_exprs\_o> |  |  |  |  |  |  |  |  |  |  |  |  | 24 |  |  |  |  |  |  |  |  |  |  |  |  |
| <id\_literal> |  |  |  |  | 57 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 58 | 58 | 58 |
| <func\_ids\_exprs\_o> |  |  |  |  |  |  |  |  |  |  |  |  | 29 | 28 |  |  |  |  |  |  |  |  |  |  |  |
| <id\_expr> |  |  |  |  | 25 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 26 | 26 | 26 |
| <params> |  |  |  |  | 32 |  |  |  |  |  |  |  |  |  | 31 |  |  |  |  |  |  |  | 32 | 32 | 32 |
| <params\_o> |  |  |  |  |  |  |  |  |  |  |  |  | 33 |  | 34 |  |  |  |  |  |  |  |  |  |  |
| <return\_v> |  |  | 37 |  | 36 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 36 | 36 | 36 |
| <def\_func> |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 40 |  |  |  |  |  |  |  |  |
| <f\_params> |  |  |  |  | 42 |  |  |  |  |  |  |  |  |  | 41 |  |  |  |  |  |  |  |  |  |  |
| <f\_types> |  |  |  |  |  |  |  |  |  | 45 |  |  |  | 46 |  |  |  |  |  |  |  |  |  |  |  |
| <type> |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 54 | 55 | 56 |  |  |  |
| <f\_params\_o> |  |  |  |  |  |  |  |  |  |  |  |  | 44 |  | 43 |  |  |  |  |  |  |  |  |  |  |
| <types> |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 48 |  |  |  |  | 47 | 47 | 47 |  |  |  |
| <types\_o> |  |  |  |  |  |  |  |  |  |  |  |  | 49 |  | 50 |  |  |  |  |  |  |  |  |  |  |
| <else> |  |  | 53 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 52 |  |  |  |  |  |  |
| <expr\_literal> |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 60 | 60 | 60 |
| <literal> |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 61 | 62 | 63 |