LL-Gramatika

1. <program> -> <?php EOL declare(strict\_types=1); < body >
2. < body > -> < stmt > < body >
3. < body > -> <end>
4. <end> -> ?> EOF
5. <end> -> EOF
6. < func > -> function func\_ID (<args>) <ret\_type> { <stmt\_list> }
7. < stmt \_list > -> < stmt > <stmt\_list >
8. <stmt\_list > -> ε
9. < stmt > -> < func >
10. < stmt > -> [expresion];
11. < stmt > -> $var\_ID = [expresion];
12. < stmt > -> while ([expresion]) { <stmt\_list> }
13. < stmt > -> if ([expresion]) { stmt\_list } <else\_stmt>
14. < stmt > -> return [expresion];
15. < args > -> <data\_type> $var\_ID <arg\_def> <arg\_list>
16. < args > -> ε
17. < arg\_list > -> , <data\_type> $var\_ID> <arg\_list>
18. <arg\_list > -> ε
19. < ret\_type > -> : <data\_type>
20. < ret\_type > -> ε
21. < else\_stmt > -> else { <stmt\_list > }
22. < else\_stmt > -> ε
23. < data\_type > -> null
24. < data\_type > -> int
25. < data\_type > -> float
26. < data\_type > -> string
27. < arg\_def > -> = literal
28. < arg\_def > -> ε

GRAMATIKA VYRAZOV

EXPR → EXPR + EXPR

EXPR → EXPR \* EXPR

EXPR → EXPR – EXPR

EXPR → EXPR / EXPR

EXPR → EXPR . EXPR

EXPR → EXPR < EXPR

EXPR → EXPR > EXPR

EXPR → EXPR <= EXPR

EXPR → EXPR >= EXPR

EXPR → EXPR === EXPR

EXPR → EXPR !== EXPR

EXPR → ! EXPR

EXPR → ( EXPR )

EXPR → id ( )

EXPR → id ( EXPR )

EXPR → id ( EXPR , EXPR )

EXPR → id ( EXPR , EXPR , EXPR )

EXPR → literal