

1. <prog> -> HEADER <prog_con>
2. <prog_con> -> func FID (<arg>) <types> <prog_con>
3. <prog_con> -> <variant> ID <option> <prog_con>
4. <prog_con> -> <id_type> <prog_con>
5. <prog_con> -> if <expression> {<statement>} <else_statement> <prog_con>
6. <prog_con> -> while <expression> {<statement>} <prog_con>
7. <prog_con> -> ϵ EOF
8. <arg> -> PARAM_NAME ID : <type_spec> <args_more>
9. <arg> -> ϵ
10. <args_more> -> , PARAM_NAME ID : <type_spec> <args_more>
11. <args_more> -> ϵ
12. <types> -> <type_spec> {<statement_with_return>} // return so spravnym dat typom exp
13. <types> -> {<statement_with_return>} //return bez expression OR ziadny return
14. <type_spec> -> Double
15. <type_spec> -> Int
16. <type_spec> -> String
17. <type_spec> -> Double?
18. <type_spec> -> Int?
19. <type_spec> -> String?
20. <variant> -> let
21. <variant> -> var
22. <option> -> : <type_spec> <assigns>
23. <option> -> <assign>
24. <assigns> -> <assign>
25. <assigns> -> ϵ
26. <assign> -> = <expression>
27. <assign> -> = <builtin>
28. <assign> -> = ID <preopt>
29. <id_type> -> <builtin_extra>
30. <id_type> -> ID <opt>
31. <builtin_extra> -> write(<write_params>)
32. <builtin_extra> -> <builtin>
33. <write_params> -> <term> <write_params_more>
34. <write_params> -> ϵ
35. <write_params_more> -> , <term> <write_params_more>
36. <write_params_more> -> ϵ
37. <builtin> -> readString ()
38. <builtin> -> readInt ()
39. <builtin> -> readDouble ()
40. <builtin> -> Int2Double (<chr_t>)
41. <builtin> -> Double2Int (<double_t>)
42. <builtin> -> length (<length_t>)
43. <builtin> -> substring (of <length_t>, startingAt <chr_t>, endingBefore <chr_t>)
44. <builtin> -> ord (<length_t>)
45. <builtin> -> chr (<chr_t>)
46. <preopt> -> (<params>)
47. <preopt> -> ϵ EOL
48. <opt> -> (<params>)
49. <opt> -> = ID EOL
50. <opt> -> = ID (<params>)
51. <opt> -> = <builtin> EOL
52. <opt> -> = <expression> EOL
53. <params> -> PARAM_NAME <params_more>
54. <params> -> PARAM_NAME : <term> <params_more>
55. <params> -> ϵ
56. <params_more> -> ,PARAM_NAME <params_more>
57. <params_more> -> ,PARAM_NAME : <term> <params_more>
58. <params_more> -> ϵ
59. <length_t> -> "STRINGLIT"
60. <length_t> -> ID //type STRING
61. <chr_t> -> INTLIT
62. <chr_t> -> ID //type INT
63. <double_t> -> DOUBLELIT
64. <double_t> -> ID //type DOUBLE
65. <statement_with_return> -> <variant> ID <option> <statement_with_return>
66. <statement_with_return> -> <id_type> <statement_with_return>
67. <statement_with_return> -> if <expression> {<statement_with_return>} else {<statement_with_return>} <statement_with_return>
68. <statement_with_return> -> while <expression> {<statement_with_return>} <statement_with_return>
69. <statement_with_return> -> return <return_value> <statement_with_return>
70. <statement_with_return> -> ϵ
71. <statement> -> <variant> ID <option> <statement>
72. <statement> -> <id_type> <statement>
73. <statement> -> if <expression> {<statement>} else {<statement>} <statement>
74. <statement> -> while <expression> {<statement>} <statement>
75. <statement> -> ϵ
76. <return_value> -> <expression>

77. <term> -> „STRINGLIT“
78. <term> -> INTLIT
79. <term> -> DOUBLELIT
80. <term> -> ID
81. <expression> -> <term> <sign> <term> <expression_more>
82. <expression_more> -> <sign> <term> <expression_more>
83. <expression_more> -> ϵ
84. <sign> -> +
85. <sign> -> -
86. <sign> -> *
87. <sign> -> /
88. <sign> -> ==
89. <sign> -> !=
90. <sign> -> <
91. <sign> -> >
92. <sign> -> <=
93. <sign> -> >=
94. <sign> -> ??