1. BLOCO::= void main { DCLVAR DCLFUNC CORPO }
2. DCLVAR::= nomevariavel 100 REPIDENT : TIPO 107; LDVAR
3. DCLVAR::= î
4. REPIDENT::= î
5. REPIDENT::= , nomevariavel 100 REPIDENT
6. TIPO::= integer
7. TIPO::= float
8. TIPO::= string
9. TIPO::= char
10. LDVAR::= î
11. LDVAR::= LID : TIPO ; LDVAR
12. LID::= nomevariavel REPIDENT
13. DCLFUNC::= TIPO\_RETORNO nomevariavel 101 DEFPAR{ DCLVAR DCLFUNC CORPO return (VALORRETORNO)} DCLFUNC
14. TIPO\_RETORNO::= integer
15. TIPO\_RETORNO::= void
16. TIPO\_RETORNO::= char
17. TIPO\_RETORNO::= float
18. TIPO\_RETORNO::= string
19. DCLFUNC::= î
20. VALORRETORNO::= numerointeiro
21. VALORRETORNO::= numerofloat
22. VALORRETORNO::= nomevariavel
23. VALORRETORNO::= nomedochar
24. VALORRETORNO::= string
25. VALORRETORNO::= î
26. DEFPAR::= î
27. DEFPAR::= ( PARAM )
28. PARAM::= TIPO LPARAM
29. LPARAM::= ; TIPO LPARAM
30. LPARAM::= î
31. CORPO::= inicio COMANDO ; REPCOMANDO fim
32. REPCOMANDO ::= î
33. REPCOMANDO::= COMANDO ; REPCOMANDO
34. COMANDO::= nomevariavel 100 = EXPRESSAO 103
35. COMANDO::= nomedastring = EXPRESSAO
36. COMANDO::= nomedochar = EXPRESSAO
37. COMANDO::= EXPRESSAO = EXPRESSAO
38. COMANDO::= î
39. COMANDO::= callfuncao nomevariavel PARAMETROS
40. PARAMETROS::= î
41. PARAMETROS::= ( TPARAM REPPAR )
42. REPPAR::= î
43. REPPAR::= , TPARAM REPPAR
44. TPARAM::= numerointeiro
45. TPARAM::= nomedastring
46. TPARAM::= numerofloat
47. TPARAM::= nomedochar
48. TPARAM::= nomevariavel
49. COMANDO::= if (nomevariavel 102 COMPARACAO) { COMANDO ; REPCOMANDO } ELSEPARTE
50. ELSEPARTE::= else { COMANDO ; REPCOMANDO }
51. ELSEPARTE::= î
52. COMANDO::= while ( nomevariavel 102 COMPARACAO) { COMANDO ; REPCOMANDO }
53. COMPARACAO::= == CONTCOMPARACAO
54. COMPARACAO::= != CONTCOMPARACAO
55. COMPARACAO::= > CONTCOMPARACAO
56. COMPARACAO::= >= CONTCOMPARACAO
57. COMPARACAO::= < CONTCOMPARACAO
58. COMPARACAO::= <= CONTCOMPARACAO
59. CONTCOMPARACAO::= numerointeiro
60. CONTCOMPARACAO::= numerofloat
61. CONTCOMPARACAO::= nomedastring
62. CONTCOMPARACAO::= nomedochar
63. CONTCOMPARACAO::= nomevariavel 102
64. COMANDO::= for (nomevariavel 102 = CONTCOMPARACAO; nomevariavel 102 COMPARACAO; INCREMENTO) { COMANDO ; REPCOMANDO }
65. INCREMENTO::= ++ numerointeiro
66. INCREMENTO::= -- numerointeiro
67. COMANDO::= do { COMANDO ; REPCOMANDO } while ( nomevariavel COMPARACAO)
68. COMANDO::= cin >> nomevariavel 102
69. COMANDO::= cout << literal SEQCOUT
70. SEQCOUT::= î
71. SEQCOUT::= << nomevariavel 102 SEQUENCIA SEQCOUT
72. SEQCOUT::= << literal SEQCOUT
73. SEQUENCIA::= î
74. SEQUENCIA::= , nomevariavel 102 SEQUENCIA
75. EXPRESSAO::= TERMO REPEXP
76. EXPRESSAO::= callfuncao nomevariavel PARAMETROS
77. REPEXP::= + TERMO REPEXP
78. REPEXP::= - TERMO REPEXP
79. REPEXP::= î
80. TERMO::= FATOR REPTERMO
81. REPTERMO::= î
82. REPTERMO::= \* FATOR REPTERMO
83. REPTERMO::= / FATOR REPTERMO
84. FATOR::= numerointeiro
85. FATOR::= numerofloat
86. FATOR::= nomevariavel 102
87. FATOR::= nomedastring
88. FATOR::= nomedochar
89. FATOR::= ( EXPRESSAO )

Erros semanticos

- Nomes de variáveis repetidas (100)

- Nomes de funções repetidas (101)

- Variáveis usadas que não foram declaradas (102)

- Variáveis de um determinado tipo que recebe um outro valor (103)

- Definir uma função com um determinado número de parâmetros e ao invocar, colocar mais do que foi definido. (104)

- A operação entre um String e um integer (105)

- Declarar uma função com uma ordem especifica dos parâmetros e ao invocar inverter a ordem dos tipos dos parâmetros (106)

100 e 101 – Também terão a função de resgatar a categoria

107 – Identificar o tipo da variável

OBS: A edição da gramatica está incompleta. edição em andamento