

BNF DA LINGUAGEM DE PROGRAMAÇÃO NOPL

```
1 <program> ::= FBE <id> <fbe_body> END_FBE
2
3 <fbe_body> ::= <include_block>
4 | <include_block> <fbe_body>
5 | <properties_block>
6 | <properties_block> <fbe_body>
7 | <instance>
8 | <instance> <fbe_body>
9 | <attribute>
10 | <attribute> <fbe_body>
11 | <vector_instance>
12 | <vector_instance> <fbe_body>
13 | <vector_attribute>
14 | <vector_attribute> >fbe_body>
15 | <method>
16 | <method> <fbe_body>
17 | <rule>
18 | <rule> <fbe_body>
19 | <formation_rule>
20 | <formation_rule> <fbe_body>
21 | <main_block>
22 | <main_block> <fbe_body>
23
24 <include_block> ::= INCLUDES <target> END_INCLUDES
25
26 <properties_block> ::= PROPERTIES <properties> END_PROPERTIES
27
28 <properties> ::= <property>
29 | <property> <properties>
30
31 <property> ::= TARGET <target>
32 | STRATEGY <strategy>
33
34 <target> ::= CODE_GENERATION_EXAMPLE
35 | NAMESPACES
36 | FRAMEWORK_CPP_2_0
37 | FRAMEWORK_CPP_3_0
38
39 <strategy> ::= NO_ONE
40 | BREADTH
41 | DEPTH
42 | PRIORITY
43
44 <instance> ::= <visibility> <id> <id>
45
46 <attribute> ::= <visibility> <type> <id> ASSIGN <factor>
47
48 <vector_instance> ::= <visibility> <id> LB INTEGER_VALUE RB <id>
49
50 <vector_attribute> ::= <visibility> <type> LB INTEGER_VALUE RB <id>
51
52 <vector_attribute> ::= <visibility> <type> LB INTEGER_VALUE RB <id>
53 | ASSIGN LC <array_factor> RC
54
55 <method> ::= <visibility> METHOD <id> <method_body>
56 END_METHOD
57
58 <method_body> ::= <code_blocks>
```

```

59         | <params_body> <attribution>
60         | <params_body> <code_blocks>
61         | <attribution>
62
63 <params_body> ::= PARAMS <params> END_PARAMS
64
65 <params> ::= <param>
66         | <param> <params>
67
68 <param> ::= <type> <id>
69
70 <code_blocks> ::= <code_block>
71         | <code_block> <code_blocks>
72
73 <code_block> ::= CODE <target> END_CODE
74
75 <attribution> ::= ATTRIBUTION <element> ASSIGN <factor>
76 END_ATTRIBUTION
77
78 <rule> ::= RULE <id> <rule_properties_block> <condition>
79         <action> END_RULE
80         | RULE <id> <condition> <action> END_RULE
81
82 <formation_rule> ::= FORMATION_RULE <id> <rule_indexes>
83         <rule_properties_block> <condition>
84         <action> END_FORMATION_RULE
85         | FORMATION_RULE <id> <rule_indexes> <condition>
86         <action> END_FORMATION_RULE
87
88 <rule_indexes> ::= <rule_index>
89         | <rule_index> <rule_indexes>
90
91 <rule_index> ::= INDEX <id> FROM INTEGER_VALUE TO INTEGER_VALUE
92
93 <rule_properties_block> ::= PROPERTIES <rule_properties> END_PROPERTIES
94
95 <rule_properties> ::= <rule_property>
96         | <rule_property> <rule_properties>
97
98 <rule_property> ::= PRIORITY INTEGER_VALUE
99         | KEEPER <boolean>
100
101 <condition> ::= CONDITION <subconditions> END_CONDITION
102         | CONDITION <id> <subconditions> END_CONDITION
103         | CONDITION <premises> END_CONDITION
104         | CONDITION <id> <premises> END_CONDITION
105
106 <subconditions> ::= <subcondition>
107         | <subcondition> <conjunction> <subconditions>
108
109 <subcondition> ::= SUBCONDITION <premises> END_SUBCONDITION
110         | SUBCONDITION <id> <premises> END_SUBCONDITION
111
112 <premises> ::= <premise>
113         | <premise> <conjunction> <premises>
114
115 <premise> ::= PREMISE <expression> END_PREMISE
116         | PREMISE <id> <expression> END_PREMISE
117         | PREMISE IMPERTINENT <expression> END_PREMISE
118         | PREMISE IMPERTINENT <id> <expression>
119 END_PREMISE

```

```

120
121 <expression> ::= <factor> <symbol> <factor>
122
123 <symbol> ::= EQ
124 | NE
125 | LT
126 | GT
127 | LE
128 | GE
129
130 <action> ::= ACTION <execution> <instigations> END_ACTION
131 | ACTION <id> <execution> <instigations>
132 END_ACTION
133 | ACTION <instigations> END_ACTION
134 | ACTION <id> <instigations> END_ACTION
135
136 <instigations> ::= <instigation>
137 | <instigation> <instigations>
138
139 <instigation> ::= INSTIGATION <execution> <calls> END_INSTIGATION
140 | INSTIGATION <id> <execution> <calls>
141 END_INSTIGATION
142 | INSTIGATION <calls> END_INSTIGATION
143 | INSTIGATION <id> <calls> END_INSTIGATION
144
145 <execution> ::= SEQUENTIAL
146 | PARALLEL
147
148 <calls> ::= <call>
149 | <call> <calls>
150
151 <call> ::= CALL <elementcall>
152
153 <elementcall> ::= <id> POINT <id> LP RP
154 | THIS POINT <id> LP RP
155 | <id> POINT <id> LP <arguments> RP
156 | THIS POINT <id> LP <arguments> RP
157
158 <arguments> ::= <argument>
159 | <argument> COMMA <arguments>
160
161 <argument> ::= <factor>
162
163 <type> ::= <basictype>
164 | <supertype>
165
166 <supertype> ::= <basictype> LT INTEGER_VALUE GT
167
168 <basictype> ::= BOOLEAN
169 | INTEGER
170 | DOUBLE
171 | STRING
172 | CHAR
173
174 <visibility> ::= PRIVATE
175 | PUBLIC
176
177 <conjunction> ::= AND
178 | OR
179
180 <array_factor> ::= <array_factor> COMMA <factor>

```

```

181         | <factor>
182
183 <factor> ::= <element>
184         | <boolean>
185         | INTEGER_VALUE
186         | DOUBLE_VALUE
187         | CHAR_VALUE
188         | STRING_VALUE
189
190 <boolean> ::= TRUE
191         | FALSE
192
193 <element> ::= <ID_or_VectElem> POINT <ID_or_VectElem>
194         | THIS POINT <ID_or_VectElem>
195         | <ID_or_VectElem>
196         | <ID_or_VectElem> POINT <VectElem>
197         | <VectElem> POINT <ID_or_VectElem>
198         | <VectElem> POINT <VectElem>
199         | THIS POINT <VectElem>
200
201 <ID_or_VectElem> ::= <id> LB INTEGER_VALUE RB
202         | <id>
203
204 <VectElem> ::= <id> LB <vector_operation> RB
205         | <id> LB <id> RB
206
207 <id> ::= ID
208
209 <vector_operation> ::= <id> <operation> INTEGER_VALUE
210         | INTEGER_VALUE <operation> <id>
211
212 <operation> ::= PLUS
213         | MINUS
214
215 <main_block> ::= MAIN <main_attributions> END_MAIN
216         | MAIN END_MAIN
217
218 <main_attributions> ::= <main_attribution>
219         | <main_attribution> <main_attributions>
220
221 <main_attribution> ::= <element> ASSIGN <factor>

```

Código 1: BNF da linguagem de programação NOPL
Fonte: Ronszcka, 2019