

Abaixo encontra-se a gramática CC-2022-2 na forma BNF. Ela é fortemente baseada na gramática X++ de Delamaro (veja bibliografia no plano de ensino). Os símbolos terminais de CC-2022-2 estão na cor amarela. Os terminais não-triviais são somente *ident*, *int_constant*, *float_constant* e *string_constant*. Os símbolos não-terminais de CC-2022-2 estão em letra de forma. Os demais símbolos (na cor azul) são símbolos da notação BNF. Consulte o livro de Delamaro para mais informações sobre a notação BNF (seção 2.3 - página 12).

Livro do Delamaro: <http://conteudo.icmc.usp.br/pessoas/delamaro/SlidesCompiladores/CompiladoresFinal.pdf>

<i>PROGRAM</i>	→ (<i>STATEMENT</i> <i>FUNCLIST</i>)?
<i>FUNCLIST</i>	→ <i>FUNCDEF</i> <i>FUNCLIST</i> <i>FUNCDEF</i>
<i>FUNCDEF</i>	→ <i>def ident</i> (<i>PARAMLIST</i>){ <i>STATELIST</i> }
<i>PARAMLIST</i>	→ ((<i>int</i> <i>float</i> <i>string</i>) <i>ident</i> , <i>PARAMLIST</i> (<i>int</i> <i>float</i> <i>string</i>) <i>ident</i>)?
<i>STATEMENT</i>	→ (<i>VARDECL</i> ; <i>ATRIBSTAT</i> ; <i>PRINTSTAT</i> ; <i>READSTAT</i> ; <i>RETURNSTAT</i> ; <i>IFSTAT</i> <i>FORSTAT</i> { <i>STATELIST</i> } <i>break</i> ; ;)
<i>VARDECL</i>	→ (<i>int</i> <i>float</i> <i>string</i>) <i>ident</i> ([<i>int_constant</i>])*
<i>ATRIBSTAT</i>	→ <i>LVALUE</i> = (<i>EXPRESSION</i> <i>ALLOCEXPRESSION</i> <i>FUNCCALL</i>)
<i>FUNCCALL</i>	→ <i>ident</i> (<i>PARAMLISTCALL</i>)
<i>PARAMLISTCALL</i>	→ (<i>ident</i> , <i>PARAMLISTCALL</i> <i>ident</i>)?
<i>PRINTSTAT</i>	→ <i>print</i> <i>EXPRESSION</i>
<i>READSTAT</i>	→ <i>read</i> <i>LVALUE</i>
<i>RETURNSTAT</i>	→ <i>return</i>
<i>IFSTAT</i>	→ <i>if</i> (<i>EXPRESSION</i>) <i>STATEMENT</i> (<i>else</i> <i>STATEMENT</i>)?
<i>FORSTAT</i>	→ <i>for</i> (<i>ATRIBSTAT</i> ; <i>EXPRESSION</i> ; <i>ATRIBSTAT</i>) <i>STATEMENT</i>
<i>STATELIST</i>	→ <i>STATEMENT</i> (<i>STATELIST</i>)?
<i>ALLOCEXPRESSION</i>	→ <i>new</i> (<i>int</i> <i>float</i> <i>string</i>) ([<i>NUMEXPRESSION</i>])+
<i>EXPRESSION</i>	→ <i>NUMEXPRESSION</i> ((< > <= >= == !=) <i>NUMEXPRESSION</i>)?
<i>NUMEXPRESSION</i>	→ <i>TERM</i> ((+ -) <i>TERM</i>)*
<i>TERM</i>	→ <i>UNARYEXPR</i> ((* / %) <i>UNARYEXPR</i>)*
<i>UNARYEXPR</i>	→ ((+ -)? <i>FACTOR</i>
<i>FACTOR</i>	→ (<i>int_constant</i> <i>float_constant</i> <i>string_constant</i> <i>null</i> <i>LVALUE</i> (<i>NUMEXPRESSION</i>))
<i>LVALUE</i>	→ <i>ident</i> ([<i>NUMEXPRESSION</i>])*