

Gramática MiniPython

Program -> 'class' ID ':' inicioBloque <field_decl>* finBloque <method_call>*

field_decl -> <assign>

method_decl -> 'def' ID ('(' ID (',' ID)* ')')? ':' <block>

block -> <inicioBloque><statement>*<finBloque>

<statement> -> ID <Statementtp>
| <methodcall>
| 'if' <expr> ':' ('elif' <expr> ':' <block>)* ('else' ':' <block>)
| 'while' <expr> ':' <block>
| 'for' ID 'in' <range> ':' <block>
| 'return' <expr>
| 'break'

<Statementtp> -> 'ID'
| '=' <assignP>
| '[' <assignP>
| '(' methodcall2

<assign> -> <lvalue> '=' <expr>

<assignP> -> '[' <expr> ']'
| '=' <expr>

<method_call> -> ID '(' (<expr> (',' <expr>)*)?
| 'read' <lvalue>
| 'print' (<expr> (',' <expr>)*)?

<lvalue> -> ID
 | ID '[' <expr> ']

<expr> -> Relacional
 | ' - ' <expr>
 | '~' <expr>
 | '[' <expr> (' , ' <expr>)* ']'
 | '(' <expr> ')'

<Relacional> -> | AritmeticoSumaResta (!= AritmeticoSumaResta) *
 | AritmeticoSumaResta (<= AritmeticoSumaResta) *
 | AritmeticoSumaResta (>= AritmeticoSumaResta) *
 | AritmeticoSumaResta (== AritmeticoSumaResta) *
 | AritmeticoSumaResta (< AritmeticoSumaResta) *
 | AritmeticoSumaResta (> AritmeticoSumaResta) *

< AritmeticoSumaResta > -> Produccion ('+' Produccion) *
 | Produccion ('-' Produccion) *
 | Produccion ('or' Produccion) *

<Produccion> -> Shift ('/' Shift) *
 | Shift ('*' Shift) *
 | Shift ('%' Shift) *
 | Shift (' and ' Shift) *

<Shift> -> Term ('>>' Term) *
 | Term ('<<' Term) *

<Term>	->	<constant>
		exprP
		method_call
<inicioBloque>	->	INDENT
<finBloque>	->	DEDENT
<range>	->	<expr> '...' <expr>
<op_bin>	->	<arith_op>
		<rel_op>
		<eq_op>
		<cond_op>
<arith_op>	->	'+' '-' '*' '/' '<<' '>>' '%'
<rel_op>	->	'<' '>' '<=' '>='
<eq_op>	->	'==' '!='
<cond_op>	->	'and' 'or' 'not'
<constant>	->	'NUMBER' 'CHARCONSTANT' <bool_const>
<bool_const>	->	'TRUE' 'FALSE'