

Grammar for the Exp programming language

Version 6

program \rightarrow main

main \rightarrow (statement)⁺

statement \rightarrow st_print | st_attrib | st_if | st_while | st_break | st_continue

st_print \rightarrow **print** (expression (, expression)^{*})

st_attrib \rightarrow **name** = expression

st_if \rightarrow **if** comparison { (statement)⁺ (**}** **else** { (statement)⁺ **}**)[?] }

st_while \rightarrow **while** comparison { (statement)⁺ }

st_break \rightarrow **break**

st_continue \rightarrow **continue**

comparison \rightarrow expression (== | != | < | <= | > | >=) expression

expression \rightarrow term ((+ | -) term)^{*}

term \rightarrow factor ((* | / | %) factor)^{*}

factor \rightarrow **number** | (expression) | **name** | **read_int** ()

tokens: + - * / % () = , { } == != < <= > >=

print number name read_int if while break continue else

bytecode instructions:

+1 **ldc** *value*

-1 **iadd**

-1 **isub**

-1 **imul**

-1 **idiv**

-1 **irem**

-1 **istore** *index*

+1	iload	<i>index</i>	
+1	getstatic		
-2	invokevirtual	.../print(I)V	
-1	invokevirtual	.../println()V	
+1	invokestatic	.../readInt()I	
-2	if_icmp??	<i>label</i>	?? → eq ne lt le gt ge
0	goto	<i>label</i>	