

Grammar for the Exp programming language

Version 8

program \rightarrow main

main \rightarrow (statement)⁺

statement \rightarrow st_print | st_attrib | st_if | st_while | st_break | st_continue
| st_array_new | st_array_push | st_array_set

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**

st_array_new \rightarrow **name** = []

st_array_push \rightarrow **name** . **push** (expression)

st_array_set \rightarrow **name** [expression] = expression

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

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

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

factor \rightarrow **number** | **string** | (expression) | **name** | **read_int** () | **read_str** ()
| **name** . **length** | **name** [expression]

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

number **string** **name** **read_int** **read_str**
print **if** **else** **while** **break** **continue** **push** **length**

bytecode instructions:

```
+1  ldc   integer or "string"
-1  iadd
-1  isub
-1  imul
-1  idiv
-1  irem
-1  istore index
+1  iload index
-1  astore index
+1  aload index
+1  getstatic
-2  invokevirtual .../print(I)V
-2  invokevirtual .../print(Ljava/lang/String;)V
-1  invokevirtual .../println()V
+1  invokestatic .../readInt()I
-2  if_icmp?? label           ?? → eq | ne | lt | le | gt | ge
  0  goto    label
+1  new Array
+1  dup
-1  invokespecial Array/<init>()V
-2  invokevirtual Array/push(I)V
  0  invokevirtual Array/length()I
  0  invokevirtual Array/string()Ljava/lang/String;
-3  invokevirtual Array/set(II)V
-1  invokevirtual Array/get(I)I
```