“Coordinated” Interpreter

A custom (fun) programming language I developed and wrote an interpreter for (in Java). This language was developed for the purpose to be used in a contest. As a result, it is not necessarily practical; instead, it presents logic challenges for performing simple tasks.

In the "Coordinated" system, variables are represented as moveable boxes on a 2x3 grid.

The grid:

(0, 0)	COM1
PRINT	EXIT
FOR	COM2 (1, 2)

The Commands:

PUT <variable name>

Places a box at (0, 0) with the given variable name with the value of the next input integer.

PRINT

Outputs the value of the box on the PRINT square.

MOVE <variable name> <UP/RIGHT/LEFT/DOWN>

Moves the box with the given variable name in the specified direction.

EXIT

Removes and de-allocated the variable of the box on the exit square.

COMPARE <variable name>

Places a box at (0, 0) with the given variable name and the value of COM2 - COM1. If one of these squares contains no variable/box, zero is used instead.

LOOP

When this called, if the box stored in loop is > 0, its value is decremented by one and the contents of the loop are ran. Otherwise, the contents are skipped over. It should be noted that the box inside LOOP can be moved around and changed freely during the loop process.

ENDLOOP

Signifies the end of the loop. Calls the previous LOOP again if the box stored in that loop is > 0.

DEBUG

Prints the current state of the board.

The specifics:

- If a box is moved off of the grid, an error is caused
- If a box B is moved/placed onto box A, A + B is stored into A and B is removed.
- Only one command per line.
- Tab indents are perfectly ok! They are ignored by the compiler.