

Documentation for Lab 1.

In our code we have one main class and 2 normal classes.

The main class we have called MainLogoProgram. In this class we have the main function and the instructions to execute a program.

First, we have the class instruction, where we have as attributes code and parameter. The code is the command that needs to be executed, for example, "FWD" which mean forward, and the parameter to store in this particular case the length. In this class we have various methods. Setters and getters, a method that checks if the current instruction is "REP" which will help us with another method in class Program, which is the second class we have created. isCorrect, a method that checks if the commands are correct. ErrorCode, method that prints in the screen if the command is incorrect, and info, method that prints the current command on screen.

Next, we have the second class, which takes as attributes linked list of instructions, currentLine (we store here the current line of instuction), loopIteration(we store here the parameter after the command REP, in order to save the number of repetitions needed), programName(to identify the program) and lastly we have decided to add repeatStartLine to save the current line, before we have repetition, so that after the repetitions are done, with this variable we can return to the code faster(basically we use it to implement goToStartLoop easier.) As methods we have getters and setters for name, addInstruction method, that adds an instruction. We have getNextInstruction, here we create an aux variable to hold the current instruction and we go to the next. If aux is the instruction REP (which mean repeat) we check the parameter after REP and put it in loopIteration, afterwards we save the current line in repeatStartLine so that after the loop we can return to where we started. We also check if we have the command END, (This happens only if previously we have encountered command repeat) if so, we decrement the loopIteration variable so that we can loop as

many times as needed and then we call `goToStartLoop` which is a method that returns us to the line that we have saved in `repeatStartLine`.

Basically, we managed to run the code, we did not have time to finish every function, for example some of the error and print methods are not done. We found it very difficult to implement the code at first, since it's our first-time using Object Oriented programming language, also we did not manage our time properly, we expected to be much easier, however using [hackerrank](#) and some tutorials in the web we managed to do this lab.