

Assignment 6: Imperative Interpreter

1 Implementation

You are to implement an interpreter for an extended version of the statement language featuring static function calls. The language is no longer a subset of the Java language. In contrast to Java, arguments can declare the desired argument passing style, and the interpreter must respect the particular choice.

You will implement a Java class `Interpreter` with a main method that reads a file in the given language from standard input (`System.in`) and starts to execute the `static main` method of the class which takes no arguments. Execution should continue until the program reaches the end of the main method or fails.

If the execution fails, the `Interpreter` should print the line and column number of the failing operator (`beginLine` and `beginColumn` fields of `NodeToken`) with a descriptive error message to standard error. The format of the message should be “LINE: COLUMN MESSAGE”.

If the file does not parse, the program should print “Parse error.” to standard error (`System.err`).

2 Remarks

Details on the semantics of the different calling conventions will be discussed in class. You do not need to consider overloading in your implementation – all methods within the same class are guaranteed to have different names.

3 Submission

You must submit the implementations to your subversion repository to the directory `3351/$USER/P6/`. Include only the provided grammar, the `Interpreter` implementation and the provided build script. The files must be called

- `functions.jj`
- `Makefile`

- `src/edu/du/cs/comp3351/p6/Interpreter.java`

You must check that the submitted code compiles by invoking `make`. Verify that the output of your program matches the expected output using your own testcases.