Assignment 3: Interpreter 101

1 Implementation

You are to implement an interpreter for a simple operator language in Java. The language is a subset of the Java expression language with integers, floats and booleans for primitive types. The interpreter is to evaluate the expressions in exactly the same way as a Java would.

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 prints the value to which the parsed expression evaluates to standard output using `System.out.println`.

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

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

2 Remarks

The given grammar accepts expressions like "++5" or "–8". You are not expected to handle these, you can print "5", "7", "8" or even throw an exception – the testcases will simply not cover these special cases. Note that in a language which does support prefix and postfix increments and/or decrements, the grammar would usually contain special productions for these constructs (to make it easier for the compiler or interpreter to detect these patterns).

Use the provided driver script to test your implementation. Run the provided testcases, implement additional testcases and make sure that the driver validates your implementation against the testcases.

Your project will be graded based on which fraction of our (partially secret) testcases your code passes. Make sure you understand and use the visitor pattern and write reasonably clean code. While your assignment will not be graded on coding style or documentation, you will have to use it as the basis for assignments P5, P6 and P7. For a reasonable implementation of this project, you can expect to submit between 500 and 1500 lines of Java code (excluding the code generated by JTB/JavaCC).

3 Submission

You must submit the implementations to your subversion repository to the directory `comp3351/f2007/$USER/P3/`. Include only the provided grammar, the Interpreter implementation and the provided build script. The files must be called

- `expressions.jj`
- `Makefile`
- `src/edu/du/cs/comp3351/p3/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.

You will not get any points if your submission does not compile without modifications, fails to run with the provided testing script or if your submission is in the wrong directory. Make sure to use "P3" for the directory name and **not** "p3". Furthermore, make absolutely sure to use the correct package ("edu.du.cs.comp3351.p3") for your `Interpreter` class. Also make sure that your `main` method is in `Interpreter` and not in `Parser`.