

## Problem 2 - Symbol Balance - 30 points

Define a class called *SymbolBalance* in the provided empty *SymbolBalance.java* file.

Your *SymbolBalance* class will read through a Java file and check for simple syntactical errors. You should write two methods, as specified by the *SymbolBalanceInterface* which you must implement for full credit.

The first method, *setFile*, should take in a *String* representing the path to the file that should be checked.

The second method, *checkFile*, should read in the file character by character and check to make sure that all `{ }`'s, `( )`'s, `[ ]`'s, `" "`'s, and `/* */`'s are properly balanced. Make sure to ignore characters within literal strings (`" "`) and comment blocks (`/* */`). Process the file by iterating through it one character at a time. During iteration, the symbol currently pointed to in the loop will be referred to as `<Current Symbol>` henceforth.

You **do not need** to handle single line comments (those that start with `//`), literal characters (things in single quotes), or the diamond operator(`<>`).

There are three types of errors that can be encountered:

- The file ends with one or more opening symbols missing their corresponding closing symbols.
- There is a closing symbol without an opening symbol.
- There is a mismatch between closing and opening symbols (for example: `{ [ } ]`).

Once you encounter an error, return a *BalanceError* object containing error information. Each error type has its own class that descends from *BalanceError* and each has its own required parameters:

- Symbol mismatch after popping stack: `return MismatchError(int lineNumber, char currentSymbol, char symbolPopped)`
- Empty stack popped: `EmptyStackError(int lineNumber)`
- Non-empty stack after parsing entire file: `NonEmptyStackError(char topElement, int sizeOfStack)`

If no error is found, return `null`.

Only push and pop the `*` character to the stack when handling multi-line comments. Do **not** push the `/` character or the string `\*`.

You must use your `MyStack` from Problem 1 in this problem.

We have provided you with a number of test inputs in the sub-folder `TestFiles`. We will use our own test files to grade your performance on all conditions - those files will be released **after** the assignment is due.