

Homework 7
CMPU102 – Spring 2017
Due: Thursday, April 27th

Exercise on Matching Openers and Closers

For this homework, you will implement a program that determines whether (, [, {, and < (*openers*) are nested properly and whether they match),], }, and > (*closers*) properly on each input line. Your file should clean out all characters that are not openers or closers. It should then go through the String of openers and closers and do the following:

1. When an opener is read, push it on a stack;
2. When a closer is read, peek at the top element on the stack.
 - If the top element on the stack is not a matching opener or if the stack is empty, print the cleaned string "is not balanced".
 - If the top element on the stack is a matching opener, pop the top element and keep processing the cleaned string. When the stack is empty and no more characters are in the cleaned string, print the cleaned string followed by "is balanced".

Download the zipped folder `Matcher` and unzip to find starter file `Matcher.java`. The downloaded folder should also contain `CharStack.java`, `acm.jar`, and text file `braces.txt`. The `Matcher.java` file extends the `DialogProgram` class of the `acm.program` package (note the import statements at the top of the file). This means that the main method must create an object of its own type and then call `start()` on that object. This creates a callback to the `run` method.

When you download this folder, the programs should compile and, if you type `braces.txt` in when asked for a filename, it should read each

line and report that each line is balanced (which is obviously not true for some of the input lines).

There are 2 tasks given in the starter file. After you are done implementing both methods, your program should accurately report which lines in the text file have balanced openers and closers.

The output from running the starter file in the interactions window should look like this:

```
Clean String is (()(){}  
Clean String is (2 + ((3*4) - 4))  
Clean String is (  
Clean String is ()<>{}  
Clean String is )))))))  
Clean String is ((((((  
Clean String is public static void main(String[] args)  
Clean String is (  
Clean String is (  
Clean String is <<<{>>>
```

Note that your program should produce "clean strings" composed solely of openers and closers as shown below:

```
Clean String is (()(){}  
Clean String is ((())  
Clean String is (  
Clean String is ()<>{}  
Clean String is )))))))  
Clean String is ((((((  
Clean String is (  
Clean String is (  
Clean String is (  
Clean String is <<<{>>>
```

Finally, your program should use `println` to print each cleaned string and the phrase "is balanced" or "is not balanced". These should be displayed as `JOptionPane` because the class extends `DialogProgram`.

Submit your zipped folder on our Moodle page when you are finished.