CS 1400-03 Introduction to Programming and Problem Solving
Project #8
(Due: 11:59 PM, Monday, 5/10/2021)

Write a program that simulates a simple calculator. This calculator keeps track of a single number (of type double) that is called `result` and that starts out as 0.0. Each run allows the user to repeatedly add, subtract, multiply, or divide by a second number. The result of each of these operations becomes the new value of result. The calculation ends when the user enters the letter R for “result” (either in uppercase or lowercase).

If the user enters any operator symbol other than `+, -, *, or /`, then an `UnknownOperatorException` is thrown and the user is allowed to reenter that line of input. Define the class `UnknownOperatorException` as a subclass of the `Exception` class. Your program should also handle `NumberFormatException` if the user enters non-numeric data for the operand.

The following is a sample interaction. The user’s input is shown in bold.

```
fcsang@fluffy ~/cs1400/project $ java CalculatorTest
Calculator is on.
result = 0.0
   +3
updated result = 3.0
-    5
updated result = -2.0
  *   10
updated result = -20.0
/-2
updated result = 10.0
%3
% is an unknown operation.
# quit
# is an unknown operation.

Must specify an operation.
+-*/
Invalid non-numeric operand.
+a
Invalid non-numeric operand.
* daisy
Invalid non-numeric operand.
/ 1.2.3
Invalid non-numeric operand.
   -8
updated result = 2.0
  r
final result = 2.0
```

Here is the UML diagram of `Calculator`.

Calculator
- result: double
+ Calculator() + getResult(): double + doCalculation(expression: String): void

The no-argument constructor will set `result` to 0.0. The method `doCalculation` has one parameter `expression` of type `String`. The method `doCalculation` will parse and verify the user entered expression. If the expression is syntactically correct, the method will calculate the new result. Otherwise, it throws appropriate exceptions, and these exceptions will be handled by the caller `CalculatorTest.java`.

In your `cs1400/project` directory, create three programs named `UnknownOperatorException.java`, `Calculator.java`, and `CalculatorTest.java`. Your Java programs must begin with the comments below and follow the naming and coding conventions posted on Bb.

```
// your name
// CS1400, section 03
// Project 8 – Calculator with Exception Handling
// date
```

Generate a `script` file `pj8.txt` with appropriate time stamps and the following steps visible:

1. a `pwd` to show the current working directory
2. a `ls -l` to show in long format the files in your `cs1400/project` directory
3. display `UnknownOperatorException.java`, `Calculator.java`, and `CalculatorTest.java`
4. compile `CalculatorTest.java`
5. run `CalculatorTest` with different test cases to demonstrate all the capabilities and features described above and to show the correctness of the program

Submit `pj8.txt` on Gradescope.