Homework Assignment #8

CS5004 – Object-Oriented Design
Northeastern University – Silicon Valley
Summer 2019

Due Sunday 07/28 at 11:00pm PDT

Grading: Each programming problem is graded as follows

- A submission which does not compile gets 0.
- A submission which compiles but does something completely irrelevant gets 0.
- A submission which works (partially) correctly, gets (up to) %80 of the total credit.
- %20 is reserved for the coding style. Follow the coding style described in the book.

Formatting: Each class must reside in its own file. If you need to instantiate objects of class `A` in your class `B`, your `B.java` file must import `A.java` first. The names you choose for your classes should match the ones specified in each problem. Also, for each problem i, you must have a `Problem_i.java` file containing (only) the class which has the `main` method. Finally, you must put all files related to problem i into a folder named `problem_i`. The files *must be placed in the right sub-folder*. Make sure that your code complies. See the note in Problem 1 for an example.
Submission: For each problem i submit one `problem_i.zip` file to Blackboard. The zip file should maintain the structure of the sub-folders. See the note in Problem 1 for an example.

Problem 1 [30pts]. Write a program that converts dates from numerical month/day/year format to normal "month day, year" format (for example, 12/25/2000 corresponds to December 25, 2000). You will define three exception classes, one called `MonthException`, another called `DayException`, and a third called `YearException`. If the user enters anything other than a legal month number (integers from 1 to 12), your program will throw and catch a `MonthException` and ask the user to reenter the month. Similarly, if the user enters anything other than a valid day number (integers from 1 to either 28, 29, 30, or 31, depending on the month and year), then your program will throw and catch a `DayException` and ask the user to reenter the day. If the user enters a year that is not in the range 1000 to 3000 (inclusive), then your program will throw and catch a `YearException` and ask the user to reenter the year.

Problem 2 [60pts]. Write a program that can serve as 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 cycle allows the user to repeatedly add, subtract, multiply, or divide by a second number. The result of one of these operations becomes the new value of result. The calculation ends when the user enters the letter R for "result" (either in upper- or lowercase). The user is allowed to do another calculation from the beginning as often as desired.
The input format is shown in the following sample dialogue. If the user enters any operator symbol other than +, −, *, or /, then an `UnknownOperatorException` is thrown and the user is asked to reenter that line of input. Defining the class `UnknownOperatorException` is part of this project.

```
Calculator is on.
result = 0.0
+5
result + 5.0 = 5.0
new result = 5.0
* 2.2
result * 2.2 = 11.0
updated result = 11.0
% 10
% is an unknown operation.
Reenter, your last line:
* 0.1
result * 0.1 = 1.1
updated result = 1.1
r
Final result = 1.1
Again? (y/n)
yes
result = 0.0
+10
result + 10.0 = 10.0
new result = 10.0
/2
result / 2.0 = 5.0
updated result = 5.0
r
Final result = 5.0
Again? (y/n)
N
End of Program
```