Java Lab 12

In this lab, you will practice with exceptions.

Create a project named Lab12. Download the file BadWeekday.java into the src directory. Create a new class named Lab12Driver with a main program. Add a public static Scanner object that wraps System.in. Read BadWeekday to see what it does.

Each time you call one of the methods described below from main( ), first print "Problem 1" or whatever problem it is (just for identification) and print a blank line at the end of each section for spacing. Run each new method with both good and bad data.

1. Create a public static void method named problem1( ) that does the following: prompt the user to enter a day of the week, a String. If their entry is invalid, print the error message "Bad day entered"; otherwise, print their entry and "Nice job!". To tell if the entry is invalid, use this: `!BadWeekday.WEEKDAYS.contains(entry).`

Call problem1( ) from main.

2. Create a public static void method named problem2( ) that does essentially the same thing as problem1 with these changes: if the entry is invalid, throw a new BadWeekday exception with the message "Bad day entered" – don't print an error message; if it's valid, print "Nice job". Add "throws BadWeekday" to the method signature.

Call problem2( ) from main. – when prompted, add a try-catch block around the call (note: it's not the default choice when IntelliJ complains about it). Don't print the stack trace, print e.getMessage( ) instead.

3. Create a public static void method named problem3( ) that calls problem2 – when prompted, add a try-catch block around the call. Print e.getMessage( ), not the stack trace.

Call problem3( ) from main.

4. Create a public static void method named problem4( ) that calls problem2 – this time, take the default option when it complains ("Add exception to method signature").

Call problem4( ) from main – when prompted, add a try-catch block around the call; print e.getMessage( ).

5. Create a public static void method named problem5( ) that calls problem2 – take the try-catch block option. In the catch, print "Caught " + e.getMessage( ), then re-throw the exception.

Call problem5( ) from main – when prompted, take the try-catch block option. Then surround the call to problem5 with a try-catch.

6. Create a public static void method named problem6( ) that calls problem2 – take the default option.

Call problem6( ) from main – take the default option. This should tag main as "throws BadWeekday". How is this different?