Lab 14 - Exception Handling

1. Download the files `DaysPerMonth.java` and `lab14Data.txt`. This class is designed to determine the number of days in a month if provided a file with months (as a number) and corresponding years.

2. Modify the program so that the filepaths for the input and output files are read in as command line arguments. The output filename is `lab14Output.txt`.

3. If no command line arguments are provided, the program will prompt the user to input filepaths for both the input and output files.

4. Once the program has obtained the filepaths, create File objects from those filepaths, then call **processFile()**, passing the created File objects as arguments.

5. In **processFile()**, write code to handle a bad input or output filename. If you receive a bad filename, the code will print "**Bad File Name**" to the console and the program will terminate gracefully (end without an error).

6. After verifying the filepaths are correct, **processFile()** will read in the data from the inputFile, which is separated by commas, one row at a time using a while loop. For each line, it will take the month and year values, pass them as arguments to the **getDays()** method, then write the following String to the output file: "**There are *x* days in this month**" where x is replaced with the number of days in the given month.

7. Once all statements are written to the output file, you will close the file and end the program.

    - At this point, your program should be able to read `lab14Data.txt` without error.

8. Modify **processFile()** so that it fully validates the input and outputs the results to the output file.

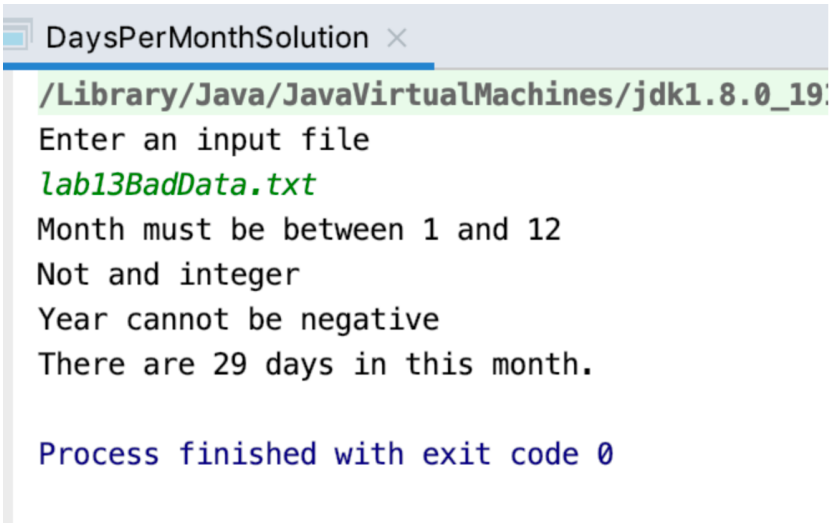
    - If the month or year isn't an integer, the program must write the message "**`Not an integer`**" to the output file. It will not pass these values to the **getDays()** method.

    - If the month isn't between 1 and 12, the program must write the message "**`Month must be between 1 and 12`**" to the output file. It will not pass these values to the **getDays()** method.

    - If the year is less than 0, the program must write the message "**`Year cannot be negative`**" to the output file. It will not pass these values to the **getDays()** method.

    - In each case the program must continue to process the inputs. Please do not overuse exceptions. If statements can handle some of this.

    You can use the file lab14BadData.txt to test your error handling. Here is what your code should generate with this file:



9. Test your code with the Lab14Test.java JUnit test.

10. Once all tests cases pass, upload your file, `DayPerMonth.java` to Gradescope and your screenshot of the test cases passing to Canvas to earn points for this lab.