SE 333 Software Testing

Assignment 3: Boundary Tests

Due Date: April 20, 2020, 11:59pm (in class and on-line students)

> Late penalties: 1% late penalty for each day of lateness, up to 7 days. No work later than 7 days will be accepted.

Objective

The objective of this assignment is to develop thorough boundary tests for a given scenario, using JUnit

Requirements

1. Download the starter project "SE333-week3-next-date".
2. Enhance the DateObj class it contains to correctly calculate the next date, given some other date. For example, the next date after April 5 2019 is April 6 2019.
3. It should correctly handle the following date calculation challenges:
   a. Handle the various month lengths
   b. Handle leap years
   c. Handle transition to next month and next year
4. The input must be a legitimate date. If it is not, throw an IllegalArgumentException
5. The result must be a legitimate date
6. It must work for dates in the past and in the future
7. You may not use Java Date and Calendar classes (or any other classes that deal with date and time concepts).

Tips

1. Do not use toString() to compare DateObj. Give DateObj a proper equals() method. If you are unfamiliar with the equals() function, read this: https://www.baeldung.com/java-equals-hashcode-contracts

2. This project needs 2 kinds of tests:
   a. Tests of the DateObj constructor. It should not ever create an invalid date. I have thought of 8 such tests but there might be more.

    b. Tests of the nextDate() function. It should produce a DateObj that is correct in all circumstances. I have thought of 6 of these but there are probably more.

3. A year is a Leap Year if:
    a. year is divisible by 4 but not by 100
    b. year is divisible by 400

Deliverables

1. A project zip file as produced by maven:

    a. The source code of the DateObj class.

    b. The source code of the JUnit test DateObjTest.

    c. The pom.xml

Produce the zip file by executing Maven’s clean and package commands. Upload the resulting zip file.