COMP285/COMP220 Lab coursework (Ant and Junit) Date 19 April 2023 2.00pm

Name Sun, Jihan Student ID 201501353

University of Liverpool Computer Science

Please note, each student's test is slightly different, if you submit someone else's work, you will lose marks.

For this test you must produce a build directory and ant build file.

You must also produce a Junit java test file call DateCalculatorTest.java with tests to test the code.

A stub form of this test file is provided for you.

There are 3 files, DateCalculator.java, DateCalculatorTest.java and BadDateException.java

NOTE The file DateCalculator.java you have been given contains no bugs. So all tests should pass,

when testing this file.

DateCalculator.java is the file you will be testing and it should be copied into the src directory.

IMPORTANT The package name for all the files in this project needs to be labtest, so please copy the files

into the correct source directories. So DateCalculator.java needs

to to be in src\labtest

Here is the specification of the code (note each student's specification is different, do not use another

student's coursework specification)

There are 2 methods you should use from the class.

Method 1
public int calculateDaysTill(int day, int month,int year)

This method will return the number of days until the date given, at takes three arguments.

day integer representation of day of the month (range depends on month)
month integer representation of the month (range 1 to 12)
year integer representation of the year (range > 0)

For the 19th of April 2023, day=19, month=4, year=2023

Method 2
public void setCurrentDate(int day, int month,int year)
This is used to help test the code, this sets the current date for the code.

The arguments are same as for, public int calculateDaysTill(int day, int month,int year)

Example..

If the currentDate is set to 19th April 2023

calculateDaysTill for the 25th April 2023, will return 6, i.e. 6 days later.

If the date is in the past, calculateDaysTill will return a negative number.

So calculateDaysTill for 17th April 2023, will return -2.

You must use setCurrentDate when you do your testing.

dates that are invalid it will throw a BadDateException.

The following are the rules the date must follow.

The day must be valid, greater than zero and less than or equal to days in month.

The month must be valid, >=1 and <=12.

The year must not be negative >=0.

Leap year rules
If the year is divisible by 4, then the year IS a leap year, except

If the year is divisible by 100, then the year IS NOT a leap year, except

If the year is divisible by 400, then the year IS a leap year.

In leap years, an extra day is added to February.

The test source files for this lab test should be stored in test\src

There should be a directory called build\classes where the classes are stored.

Your ant file should compile all the Java files and then

leave the result in build\classes.

Your ant file should run the JUnit test case and produce reports in XML and HTML.

Submission and deliverables

You must submit your work electronically via Canvas

Your submission must consist of the following:

A zipped up file which is the whole build directory, the name of this file should be named

LabtestXXXXXX.zip where XXXXXXX is your long University id number.

This zipped file must contain:

A build.xml file which compiles makes output directories if needed, builds the target code,

builds the test code and runs the JUnit tests and also produces reports about the tests in XML and HTML.

The directory structure should have the following directories

build\classes Containing all classes for the application (test and target classes)

test\reports Stores HTML and XML reports

test\src Test code containing the code DateCalculatorTest.java

src Containing the file DateCalculator.java and BadDateException.java in your submission do not change these files, keep them as they were passed to you.

lib Any libraries you may need e.g. JUnit

Marking

The marks are assigned as follows:
Quality of the Junit file 60% broken down as:
Appropriate structure of JUnit file and readability 10%
Ability to reveal bugs 50%

Quality of Ant file 40% broken down as
Production of correct final outputs 20%
Formatting, readability, good use of properties 20%

Note for the ability to reveal bugs part of your assessment you tests must do the following:

1. Run with no failure if the code has no bugs, if your tests fail for the bug free code,

then you will get Zero marks for this component.

2. Reveal bugs if the code has bugs in it, your tests will be tested against various

versions of the code with bugs that have been added on purpose

TIPS Make each assertion test case test only 1 issue at a time. Check all boundaries, edge cases.

Plagiarism/collusion is described here

https://www.liverpool.ac.uk/media/livacuk/tqsd/code-of-practice-on-assessment/appendix_L_cop_assess.pdf