Langara
THE COLLEGE OF HIGHER LEARNING.

Department of Computing Science & Information Systems
CPSC 1181
Lab#9
Nov 6, 2024

Objectives:

Study GUI, Event handling, and Exception

Preparation:

Study chapters layout management, Event Handling, and Exception Handling

Due date:

Due Date: 11:00 PM on Tuesday Nov 12, 2024

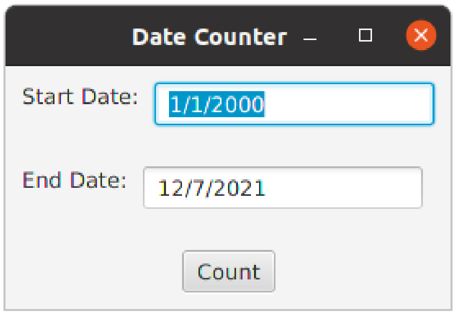
Where to upload:

zip folder to yourStudentID.zip and upload it to Lab9 in D2L.

Part 1: Tutorail (due date: Nov 6, 8:00 PM)

Part2: GUI Application

Write a Java Graphical User Interface (GUI) application program to obtain from a user two strings that should represent two valid dates. The dates are verified both for their correct format and for their validity in a calendar. Your program should have a GUI with two labels, and a button. Here is an example:

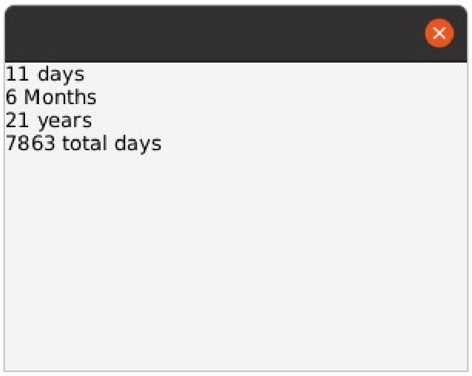


Date Counter

Start Date: 1/1/2000

End Date: 12/7/2021

The default start date should be 1/1/2000, and the default ending date must be today's date, i.e. the day when the program is being run by your marker. If the dates entered are correct, a popup window gives the number of days (months, years) between the dates; however, if there is an error in the input, a detailed error about the date is given in a popup window as shown below:



```
11 days
6 Months
21 years
7863 total days
```

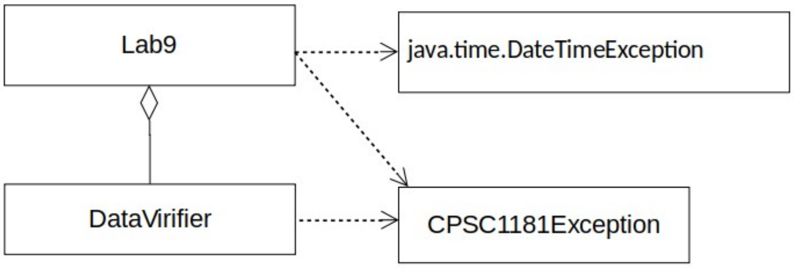
The dialog does not mean should match exactly as shown above. As long as there are appropriate margins and the count button is aligned in the middle it is acceptable.

Possible errors to detect are missing parts of the date, no date given, an impossible date, a negative date, a date with characters and so on. Your program should handle errors gracefully and it should report the errors with an informative message.

The date should be in the format: DD/MM/YYYY, where D, M, and Y are all digits.

Allowed, as well, are D/M/YYYY and D/MM/YYYY and DD/M/YYYY. Note that in this assignment you may **not** assume that the input is numeric. In addition, DD should be less than or equal to 28, 30 or 31 depending on the month, and MM should be less than or equal to 12. The range of years allowed is 1000 to 3000.

The rough UML diagram of the implementation shown below:



Class Lab9 handles all exceptions.

DataVerifier checks for the format of the input dates for validation and throws exceptions. Note that DataVerifier class throws exceptions and Lab9 class catches and handles them.

CPSC1181Exception is a user defined unchecked exception.

java.time.DateTimeException is thrown by Java builtin methods. Your program should handle this type of exception too.

Your program should provide the same messages by throwing and catching exceptions.

Sample Errors:

Start date	End date	Message
1/-10/2010	30/10/2023	Start Date: incorrect “month digits” -1
11/8/2001	-5/10/2023	End Date: incorrect “day digits” of -5
11/584/2000	8/6/2020	Start Date: not the right number of symbols. Use the following format DD/MM/YYYY
11/HI/2000	8/6/2020	Start Date: only digits are allowed D, M, and Y are digits in DD/MM/YYYY
8/6/2020	5/10/23	End Date: incorrect “year digits” of 23
11/2000	5/10/23	Start Date: missing or too many fields. The correct format is DD/MM/YYYY
11/11/1000	5/10/5000	End Date: maximum year is 3000

And similar messages for wrong input

Notes:

You must Throw exceptions when errors are detected and implement the appropriate error handlers. Do not just use if statements.

Study the following classes:

java.time.LocalDate
java.time.LocalDateTime
java.time.temporal.ChronoUnit
java.time.Period

Use the classes and the sub-classes (and interfaces) in the package **java.time** e.g. **java.time.LocalDate** and **java.time.temporal.ChronoUnit.**

Do not use the Java **deprecated** classes such as java.util.Calendar nor java.util.Date nor java.util.Gregorian nor any time classes from JDK 5, 6, 7 (i.e. you can only run your code with JDK 1.8).

You can use "between" method of ChronoUnit class to find total days between two instance of LocalDate instances :

```
 long days =ChronoUnit.DAYS.between(startLocalDate, endLocalDate);
```

Download SampleJavaTime.java, study, compile, and run it as a demo.

The exceptions that you probably are going to deal with are NumberFormatException, DateTimeException, and your own exception CPSC1181Exceptrion classes depends on your implementation. Refer to Java API for the methods you are using from Java Library and study the exceptions they may throw.

How to create a popup window in JavaFX:

Study the sample program PopupWindowFX.java provided that shows how to create popup windows in JavaFX, and use it to display your message.

What to upload

Create one page UML diagram of your design, and save it either as PDF or as word 2003 (JPEG Inserted into word document).

Zip your source files and UML diagram into *your StudentID.zip* file and then upload it to lab9 in D2L Dropbox.

TOTAL MARK: 50