Department of Computer Science and Software Engineering
Xi'an Jiaotong-Liverpool University

Lab 12 – LocalDateTime

Aim

This short Lab is intended to familiarise yourself with using the Java LocalDate, LocalTime, LocalDateTime, and Period functionality, and reinforce your existing knowledge.

Resources

All Java files you need are found on ICE.

Tips:

1. This sheet is intended to take less than the full 2 hours. You should use free time to work on your CW3 assignment.

Creating Events

Java 1.8 has implemented improved approaches to deal with date and time. We are going to use it to store events, and then display and search these events.

- Create a new java project and add the Event.java class to your project. Create a new main class and give it a constructor, and a global arraylist of events.

- Think of 5 different events that have happened to you in the last year, 2 of them should be of less than one day. Examples:
  - Party with friends On 25 July from 18:00 to 24:00
  - Summer holiday from 11 August to 1 September
  - Spring Semester from February 1 to June 14

- Create a method in your main class (called from the constructor that will add five memories). Use the class and week 12 lecture notes to work out how to add events

- Use System.outs to check that this is working

Listing Events

- Create a method that will iterate through your list of events, and use the methods in the Events class to display information

- If the event is longer than one day, then show the start and end dates

- If the event is less than one day, then show the start date, and the start and end times, as shown below

```
Output - Lab12-Time (run)
run:
Event: visiting Scotland was on the dates 2017-03-07 to 2017-04-18
    ***
Event: visiting Germany was on the dates 2017-06-07 to 2017-06-28
    ***
Event: Summer Festival was on 2017-08-07 from 12:00 till 18:00
    ***
BUILD SUCCESSFUL (total time: 0 seconds)
```

Searching for an Event

Now you want to add functionality that will check whether any of your events are happening on a chosen date. It's a form of checking your calendar!

- Create a method that will receive a LocalDateTime object, iterate through the loop, and if an event is found, will display the matching event.
- Look at the Event class to see if there is a suitable method you can use

```
Output - Lab12-Time (run)
run:
Event: visiting Scotland was on the dates 2017-03-07 to 2017-04-18
    ***
Event: visiting Germany was on the dates 2017-06-07 to 2017-06-28
    ***
Event: Summer Festival was on 2017-08-07 from 12:00 till 18:00
    ***
Looking for event that happens on 2017-03-15
visiting Scotland was on the chosen date 2017-03-15
BUILD SUCCESSFUL (total time: 0 seconds)
```

- If no events are found, then display an appropriate message

```
Output - Lab12-Time (run)
run:
Event: visiting Scotland was on the dates 2017-03-07 to 2017-04-18
    ***
Event: visiting Germany was on the dates 2017-06-07 to 2017-06-28
    ***
Event: Summer Festival was on 2017-08-07 from 12:00 till 18:00
    ***
Looking for event that happens on 2017-12-15
No events on the chosen date 2017-12-15
BUILD SUCCESSFUL (total time: 0 seconds)
```

Checkpoint

Show a TA your completed scheduling check, with Events, Iterators, and an overlap check.Answer any questions they have.

This will count towards your final grade.