(Optional) Coding Challenge: Date and Time Handling In Java

Travel Planner and Event Scheduler Applications

Estimated time needed: 20 minutes

This optional self-guided lab provides exercises that align with skills you learned in this module. Follow the instructions to write code using your knowledge to create the described apps. After you complete these lab exercises, you can validate that your code is correct with the solution provided.

Prerequisites (optional)

You must have completed all the labs in Module 1,2,3,4 and 5.

Clone the challenges repo

1. Run the following command to clone the repository with the challenges and solution.

   ```
   git clone https://github.com/ibm-developer-skills-network/flgsb-oop-additional-practice.git
   ```

2. Change to `OOPs/Module 5` directory.

   ```
   cd flgsb-oop-additional-practice/OOPs/Module\ 5
   ```

3. View the directories by running the `ls` command. Each directory contains one challenge.

   ```
   ls
   ```

Travel Planner

This challenge will help you apply your knowledge of Java date manipulation by creating a travel planner application. As a developer for a travel agency, you'll build functionality that helps users plan their trips by:
1. Calculating the duration of trips between departure and return dates
2. Validating that travel dates are valid and logical
3. Computing check-in and check-out dates based on arrival dates
4. Determining if trips overlap with holidays or special events

This practical challenge reinforces key concepts from the date manipulation lab including:

- Using LocalDate, Period, and ChronoUnit for date operations
- Validating user input dates
- Calculating differences between dates
- Adding or subtracting time periods from dates

Solve the challenge

1. In the terminal, change to `Read a text file`.

   ```
   cd Calculate\ the\ difference\ between\ two\ dates
   ```

2. Select the following button to open the java coding challenge.

Open **TravelPlanner.java** in IDE

3. Make changes in the file as per the specification.

4. Compile the java file.

5. Set the CLASSPATH.

6. Run the class `TravelPlanner` to ensure the class is created based on the instructions.

Click here for the solution

Event Scheduler Application

In this challenge, you'll apply your Java date and time knowledge to build a simple event scheduler application. This hands-on challenge will help you master essential date and time operations by creating a practical tool that manages events across different time zones.

You'll create a system that allows users to:

- Add events with date, time and timezone information
- Display events in various date formats
- Calculate event duration
- Convert event times between different time zones
- Find upcoming events within a specific timeframe

This challenge reinforces all the key concepts from the lab: Retrieving current date/time, formatting dates, parsing date inputs, adding and subtracting time units, and converting between time zones - all within a real-world application context.

Challenge Description

Create a Java application that manages a list of events. Each event should have a name, date, time, duration, and timezone. The application should provide the following functionality:

1. Create events: Allow users to add events with a name, date, time, duration, and timezone.
2. Display events: List all events, formatting the dates in user-specified formats.
3. Calculate timing: Find the time until an event starts from the current time.
4. Convert timezones: Display event times in different timezones.
5. Find upcoming events: List events that will occur within a specified number of days.

Solve the challenge

1. In the terminal, change to `Create your own class`.

   ```
   cd Format\ a\ given\ Date\ and\ Time
   ```

2. Select the following button to open `Event.java`

Open **Event.java** in IDE

3. Make changes in the file based on the specifications.

4. Compile the java file.

5. Set the CLASSPATH.

6. Select the following button to open `EventScheduler.java`

Open **EventScheduler.java** in IDE

7. Make changes in the file as per the specification.

8. Compile the java file.
9. Run the class `EventScheduler` to ensure the class is created based on the instructions.

Click here for the solution

Conclusion

After you complete these labs, you should be comfortably able to work with time and date operations in Java.

Author(s)

Lavanya

© IBM Corporation. All rights reserved.