Final Project Overview

Estimated Duration: 3 minutes

Learning Objectives

After completing this project, you will be able to comfortably work with:

- Creating classes and objects with object oriented programming (OOP) concepts
- Using Collections to store objects in memory
- Create, store, retrieve and manipulate date and time objects
- File I/O Handling

About the course project

Welcome to the final project for `Object Oriented programming in Java`. In this project, you will apply the knowledge and skills learned in this course to a simulated scenario.

In this task, you will create a console application for a mood tracker app, which will allow you to create a `Mood` object with attributes, `name`, `date`, `time`, `notes`. The tasks in this hands-on project correspond to the activities performed by a Java Developer who is creating a stand-alone console application.

This final project, which will take about 30 minutes to complete, comprises eight tasks.

- **Task 1: Create a `Mood` class**
  Create a Mood class, with attributes such as `name`, `date`,`time` and `notes`. The class should have appropriate constructor, setters, getters and toString method implements.

- **Task 2: Create Mood Tracker app**
  Create `MoodTracker.java` with main method and all necessary inputs.

- **Task 3: Import and create a Scanner object**
  Import and create a scanner object and create an object of it to read from the console.

- **Task 4: Create an ArrayList**
  Create an ArrayList to contain Mood objects within the main method.

- **Task 5: Create for infinite loop for user input**
  Create an infinite loop to handle user input as long as the user wishes. The loop should exit when the user inputs `Exit` (ignoring the case).

- **Task 6: Create a menu for the user to handle moods**
  Create a menu for the user to handle add moods, delete mood, edit mood notes and search for mood.

- **Task 7: Create `switch-case` statement to handle user input**
  Create a switch case statement to handle the user input. For example, `a` for add, `d` for delete, `e` for edit, `s` for search, `w` for writing on to a file.

- **Task 8: Add mood with validation**
  Get input for mood name from the user. Get date and time. Check the mood validity, which is mood on a date at a given time that cannot change. If a mood on the same date and time exists, throw an exception. If the mood is validated, store it.

- **Task 9: Edit notes**
  Get the mood, date and time and allow user to edit the notes.

- **Task 10: Delete by date**
  Get the date as an input from user and delete all moods tracked for that day.

- **Task 11: Delete by all details**
  Get the name, date and time as an input from user and delete the mood that matches that.

- **Task 12: Search by date**
  Get the date as an input from user and retrieve all moods tracked for that day.

- **Task 13: Search by all details**
  Get the name, date and time as an input from user and return the mood that matches that.

- **Task 14: Get all the moods**
  Get all the moods from mood tracker.

- **Task 15: Write mood on to a file**
  Write all the ArrayList moods to a file named moodtracker.txt

While performing the tasks, ensure you closely adhere to the instructions. You will be provided with a Cloud IDE environment for project execution.

To summarize:

- Read and follow the instructions carefully to complete the project.

Let's get started!

Author(s)

Lavanya

Skills Network