Final Project

Estimated Duration: 45 minutes

Skills Network

Learning Objectives

After completing this project, you will be able to:

- Create classes and objects with OOP concepts such as encapsulation and polymorphism
- Use collections to store objects in memory
- Create, store, retrieve and manipulate Date and Time objects
- Handle File I/O

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, and 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.

You are currently viewing this lab in a Cloud based Integrated Development Environment (Cloud IDE). It is a fully-online integrated development environment that is pre-installed with JDK 21, allowing you to code, develop, and learn in one location.

Mood class

1. Create a project directory by running the following command.

   ```
   mkdir final_proj
   ```

2. Run the following code to create the directory structure.

   ```
   mkdir -p final_proj/src
   mkdir -p final_proj/classes
   mkdir -p final_proj/test
   cd final_proj
   ```

3. Now, create a file named `Mood.java` inside the src directory.

   ```
   touch /home/project/final_proj/src/Mood.java
   ```

4. Select the following button to open the file for editing.

Open **Mood.java** in IDE

Start adding the code

1. Create a public class with the attributes, constructor, setters, getters, toString, and equals method. The `date` attribute should be set to default current date. The `time` attribute should be set to default `LocalTime.MIDNIGHT`. Overload the constructor to take the following parameters:

- Mood(String name)
- Mood(String name, LocalDate date)
- Mood(String name, LocalDate date, LocalTime time)
- Mood(String name, String notes)
- Mood(String name, LocalDate date, String notes)
- Mood(String name, LocalDate date, LocalTime time, String notes)

▶ Click here for sample

2. Compile `Mood.java` and verify the class is created within the `classes` directory.

3. Set the classpath. **Please ensure that this step is complete. Else your classes will not compile as expected**.

Create InvalidMoodException

1. Now create a file named `InvalidMoodException.java` inside the `src` directory.

   ```
   touch /home/project/final_proj/src/InvalidMoodException.java
   ```

2. Select the following button to open the file for editing.

Open **InvalidMoodException.java** in IDE

3. Add appropriate code in InvalidMoodException.java.

▶ Click here for sample

4. Compile `InvalidMoodException.java` and verify the class is created within the `classes` directory.

Create Mood Tracker app

1. Now create a file named `MoodTracker.java` inside the `src` directory.

```
touch /home/project/final_proj/src/MoodTracker.java
```

2. Select the following button to open the file for editing.

Open **MoodTracker.java** in IDE

3. Create a public class with a public main method.

   Note: Remember to add all the import statements as and when you are using any classes that are external.

▶ Click here for the sample code.

4. Create an ArrayList that can store `Mood` objects.

5. Import scanner from the `util` package and create an object of it to read from the console in the MoodTracker application.

Infinite loop with Menu Options

1. Select the following button to open the file for editing if it is not open already.

Open **MoodTracker.java** in IDE

2. Create an infinite loop that runs as long as the user wants it to. The code within the loop should present the user with options. The loop should exit when the user inputs `Exit`.

▶ Click here for sample code.

Add a mood

1. Get input for mood name from the user. Get date and time. You can allow user to press enter to skip entering the date and time.
2. Check mood validity. The mood on a particular date, at a given time cannot change. If a mood on the same date and time exists, throw an exception. If the mood is validated, store it.

▶ Click here for code sample.

Delete moods by date or delete a specific mood

1. For the delete option, provide the user with two further options.

- The first option is to delete the moods by date. Get the date as an input from user and delete all moods tracked for that day.
- The second option is to delete one mood, based on the name, date and time. Get the name, date, and time as an input from user and delete the mood.

▶ Click here for sample code.

Edit a specific mood

1. Get the mood, date and time as a user input and allow user to edit the notes.

▶ Click here for sample code.

Search moods by date or search a specific mood

1. For the search option, provide the user with two additional options.

- The first option is to get moods for a date. Get the date as an input from user and retrieve all moods tracked for that day.
- The second option is to get one mood, given the name, date and time. Get the name, date and time as an input from user and retrieve the mood that matches that.

▶ Click here for sample code

Get all the moods

1. Add code to get all moods that are stored in the tracker and print them on the console.

▶ Click here for sample code.

Write the moods to a file

1. Add code to write all the ArrayList moods to a file named `moodtracker.txt`.

▶ Click here for the sample code.

Make sure all the required packages are imported. Otherwise the compilation would fail.

▶ Click here for all the classes you need to import

Compile and run code

1. Set the classpath to point to `final_proj/classes`.

   ```
   export CLASSPATH=$CLASSPATH:/home/project/final_proj/classes
   ```

2. Compile the file `Mood.java`.

   ```
   javac -d final_proj/classes final_proj/src/Mood.java
   ```

3. Compile the file `MoodTracker.java`.

   ```
   javac -d final_proj/classes final_proj/src/MoodTracker.java
   ```

4. Run the java program.

   ```
   java MoodTracker
   ```

Author(s)

Lavanya