To do: Make a submission

Opened: Thursday, 21 March 2024, 12:05 AM
Due: Thursday, 28 March 2024, 11:55 PM

Assignment Title: Weather Information App

This programming assignment involves the creation of a Weather Information App in Java with a graphical user interface (GUI) to provide real-time weather updates.

This assignment aims to cover API integration and data retrieval in Java, GUI application development using either JavaFX or Java Swing, as well as user input validation and error handling.

Assignment Instructions

Scenario: You are tasked with developing a Weather Information App in Java that provides real-time weather updates to users. The application should have a graphical user interface (GUI) to display weather details for a specific location.

Requirements:

1. **API Integration:**

   a. Utilize a weather API (such as OpenWeatherMap, Weatherstack, or any of your choice) to fetch real-time weather data.

2. **GUI design:**

   a. Design a user-friendly GUI using JavaFX or Java Swing.

   b. Include components for users to input the location (e.g., city name or coordinates).

3. **Display Weather Information:**

   a. Display relevant weather information such as temperature, humidity, wind speed, and conditions for the specified location.

4. **Icon Representation:**

   a. Use appropriate icons or images to visually represent weather conditions (e.g., sun for clear sky, clouds for cloudy weather).

5. **Forecast Display:**

   a. Implement a section to display a short-term weather forecast for the chosen location.

6. **Unit Conversion:**

   a. Include an option to switch between different units for temperature (Celsius, Fahrenheit) and wind speed.

7. **Error Handling:**

   a. Implement proper error handling for scenarios like invalid location input or failed API requests.

8. **History Tracking:**

a. Allow users to view a history of their recent weather searches with timestamps.

9. **Dynamic Backgrounds:**

    a. Implement dynamic background changes based on the time of day (e.g., a sunset background in the evening).

Guidelines

- Submit well-commented Java source code.
- Include a README file explaining how to use the Weather Information App and providing details about your implementation.

Grading Criteria

Your assignment will be evaluated based on the following criteria:

1. API Integration: Students need to utilize a weather API for real-time weather data. Successful integration and retrieval of relevant weather information. Appropriate use of API key and adherence to API usage guidelines.
2. GUI Design: User-friendly design with clear navigation. Uses JavaFX or Java Swing for graphical interface. Components for user input (city name or coordinates) are well-implemented.
3. Logic and Computation
4. Program Flow and Structure
5. Output
6. Code Style and Readability

Submission Instructions

1. Read the rubric on how you are going to be graded before you start to work on this assignment.
2. Remember to use appropriate variable names and follow best practices of coding. Please provide a screenshot of the outputs. Submit the assignment in MS Word or PDF file.

This assignment will be assessed by your instructor using the rubric below.

Submission status

Attempt number	This is attempt 1.
Submission status	No submissions have been made yet
Grading status	Not graded
Time remaining	Assignment is overdue by: 15 days 21 hours
Last modified	-
Submission comments	▸ Comments (0)

Grading criteria

API Integration	AccuratelyOpenWeatherMap, Weatherstack) to fetch real-time weather data. Fetches relevant weather information such as temperature, humidity, wind speed, and conditions. ***20 points***	Accurately integrates a weather API (e.g., OpenWeatherMap, Weatherstack) to fetch real-time weather data. However, they did not fetch relevant weather information such as temperature, humidity, wind speed, and conditions. ***16 points***	Students have issues with integration of weather API to fetch real-time weather data. But some of the data is fetched. However, all of the weather information has not been fetched. ***10 points***	Incorrect API integration. ***0 points***
GUI Design	Designed a user-friendly GUI using JavaFX or Java Swing. Includes components for users to input the location (e.g., city name or coordinates). ***20 points***	Designed interface is not user friendly but includes components for users to input the location (e.g., city name or coordinates). ***10 points***	The GUI designed is not user friendly and does not include components for users to input the location. ***0 points***	
Logic and Computation	The program accurately demonstrated most of the operations of Weather App. ***20 points***	The program accurately demonstrated some of the operations of the Weather App. ***10 points***	The program does not correctly demonstrate the operations of Weather App. ***0 points***	
Program Flow and Structure	The program follows a logical flow and is well-structured. Proper variable declaration and initialization are done. Meaningful variable names and appropriate data types are used. ***20 points***	The program follows a logical flow and is structured. Variable names and data types used are not correct. ***10 points***	Variables are not declared, no proper logical flow and inappropriate data types declared. ***0 points***	
Output	Provided screen shot of a GUI user interface. Submit well-commented Java source code. Included a README file explaining how to use the Weather Information App and providing details about implementation. ***10 points***	The program does not provide a screenshot of the program's GUI based user interface and README file. ***0 points***		
Code Style and Readability	The code follows consistent indentation and formatting conventions. The code is easy to read and understand. The program does not contain any unnecessary or redundant code. ***10 points***	The code does not have proper indentation and formatting. ***8 points***	Unnecessary code and no proper indentation are followed. ***5 points***	Redundant code. ***0 points***