✔ **Done:** Make a submission

Opened: Thursday, 15 February 2024, 12:05 AM
Due: Thursday, 22 February 2024, 11:55 PM

Assignment Title: Simple Clock Application

Through this assignment, you will gain knowledge and skills in understanding the basics of the Java Thread model. You will also be able to gain your skills in the implementation of multithreading concepts and the usage of thread priorities for task prioritization in a real-life clock application.

Assignment Instructions

Scenario: You are tasked with developing a simple clock application that utilizes Java threads to display the current time and date concurrently. This project aims to explore the Java Thread model and its basics while illustrating the use of threads and their priorities in a straightforward real-life scenario.

Requirements:

1. **Clock Class:**
    a. Create a **Clock** class responsible for displaying the current time and date.
    b. Implement a method to continuously update and print the current time.
2. **Thread Implementation:**
    a. Utilize Java threads to ensure that the clock continuously updates its time in the background.
    b. Implement a separate thread for printing the time to the console.
3. **Thread Priorities:**
    a. Introduce thread priorities for better timekeeping precision.
    b. The clock display thread should have a higher priority than the background updating thread.
4. **Simulation Output:**
    a. Display the current time and date in a readable format, e.g., "HH:mm:ss dd-MM-yyyy".
    b. Ensure that the clock continuously updates the time.

Guidelines

- Use meaningful variable and method names.
- Implement proper error handling where necessary.
- Ensure that your code is well-organized and follows Java coding standards.
- Provide comments to explain the purpose of classes, methods, and any complex logic.

Deliverables

1. **Java Program Source Code:**
    a. Includes the **Clock** class and necessary threads.
    b. Demonstrates the use of thread priorities for better precision.

?

2. **Output Screenshot:**
    a. Provide a screenshot of the program's output, showcasing the continuously updating clock with different thread priorities.

Grading Criteria

Your assignment will be evaluated based on the following criteria:

- **Clock Class:** The Clock class should accurately display the current time and date. The method responsible for updating and printing the time should work as expected. Use of meaningful variable and method names, proper error handling, adherence to Java coding standards, and well-organized code.
- **Thread Implementation:** Threads should be appropriately used to ensure the clock continuously updates its time in the background. There should be a separate thread for printing the time to the console. Ensure proper synchronization and handling of concurrency issues. Threads should work seamlessly without conflicts.
- **Thread Priorities:** Thread priorities should be introduced to achieve better timekeeping precision. The clock display thread should have a higher priority than the background updating thread.
- **Readability and Continuity:** The displayed time and date should be in a readable format, and the clock should continuously update.
- **Screenshot:** Provide a screenshot of the program's output, showcasing the continuously updating clock with different thread priorities.

Submission Instructions

- Read the rubric on how you are going to be graded before you start to work on this assignment.
- 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	Submitted for grading
Grading status	Graded
Time remaining	Assignment was submitted 2 days 6 hours early
Last modified	Tuesday, 20 February 2024, 5:52 PM
File submissions	Programming Assignment Unit 3.pdf 20 February 2024, 5:52 PM
Submission comments	Comments (0)

Grading criteria

Thread Implementation	The Java Threads have been used to ensure that the clock continuously updates time. Clock class has also been implemented with a method to continuously update and print the current time. ***20 points***	The concept of Java Threads has been used to correctly ensure that the clock continuously updates the time. But the Clock class has not been implemented with an appropriate method to continuously update and print the current time. ***16 points***	The concept of Java Threads has not been used correctly to update the clock time. However, the Clock class has been implemented with a method to continuously update and print the current time. ***10 points***	The Java Threads have not been used to ensure that the clock continuously updates time. Also, the Clock class has not been implemented with a method to continuously update and print the current time. ***0 points***
Thread Priorities	Thread priorities have been used for timekeeping precision. The clock display thread has a higher priority than the background updating thread. ***20 points***	Thread priorities have been used for timekeeping precision. The clock display thread does not have a higher priority than the background updating thread. ***10 points***	Thread priorities have not been used for timekeeping precision. The clock display thread does not have a higher priority than the background updating thread. ***0 points***	
Logic and Computation	The program accurately displays the current time and date in a readable format and ensures that the clock continuously updates the time. ***20 points***	Although the program does not accurately display the current time, the clock continuously updates the time. ***16 points***	Although the program accurately displays the current time and date, it is not in a readable format and the clock is not continuously updating the time. ***10 points***	Neither the program accurately displays the correct time and date in a readable format nor does the clock continuously update the time. ***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	Provide a screenshot of the program's output, showcasing the continuously updating clock with different thread priorities. ***10 points***	The program does not provide a screenshot of the program's output, showcasing the continuously updating clock with different thread priorities. ***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***

Feedback

Grade	10.00 / 10.00
Graded on	Wednesday, 21 February 2024, 9:10 PM
Graded by	 Vikas Thada (Instructor)
Feedback comments	Dear Godfrey Good work ! Regards

Grade breakdown

Thread Implementation	The Java Threads have been used to ensure that the clock continuously updates time. Clock class has also been implemented with a method to continuously update and print the current time. ***20 points***	The concept of Java Threads has been used to correctly ensure that the clock continuously updates the time. But the Clock class has not been implemented with an appropriate method to continuously update and print the current time. ***16 points***	The concept of Java Threads has not been used correctly to update the clock time. However, the Clock class has been implemented with a method to continuously update and print the current time. ***10 points***	The Java Threads have not been used to ensure that the clock continuously updates time. Also, the Clock class has not been implemented with a method to continuously update and print the current time. ***0 points***
Thread Priorities	Thread priorities have been used for timekeeping precision. The clock display thread has a higher priority than the background updating thread. ***20 points***	Thread priorities have been used for timekeeping precision. The clock display thread does not have a higher priority than the background updating thread. ***10 points***	Thread priorities have not been used for timekeeping precision. The clock display thread does not have a higher priority than the background updating thread. ***0 points***	

Logic and Computation	The program accurately displays the current time and date in a readable format and ensures that the clock continuously updates the time. ***20 points***	Although the program does not accurately display the current time, the clock continuously updates the time. ***16 points***	Although the program accurately displays the current time and date, it is not in a readable format and the clock is not continuously updating the time. ***10 points***	Neither the program accurately displays the correct time and date in a readable format nor does the clock continuously update the time. ***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	Provide a screenshot of the program's output, showcasing the continuously updating clock with different thread priorities. ***10 points***	The program does not provide a screenshot of the program's output, showcasing the continuously updating clock with different thread priorities. ***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***