Simple Clock Application

# Overview

This Java program is a simple clock that shows the current time and date. It uses Java Threads to run two tasks at the same time: one to show the clock and another for a background task.

# Class Breakdown

## 1. TimeDisplay Class

This class is in charge of showing the time and date. It uses a loop to keep printing the current time every second. The time is shown in a readable format like HH:mm:ss dd-MM-yyyy. It also has a method to stop the loop when needed.

## 2. TimeUpdater Class

This class is a background task. It prints a message every 3 seconds. It simulates other work happening in the background while the clock runs.

## 3. TimeTrackerApp (Main Class)

This is the main class. It creates and starts the two threads: one for showing time, and one for background work. It also sets priorities, so the time thread has more importance. The program runs for 20 seconds, then stops the clock and ends.

# Key Concepts Used

|  |  |
| --- | --- |
| Concept | How It Is Used |
| Multithreading | The program runs two threads at the same time. |
| Thread Priority | The clock thread has higher priority for better timing. |
| Looping with Sleep | The threads use sleep to pause between updates. |
| Date Formatting | The time is shown in a clear format: HH:mm:ss dd-MM-yyyy. |
| Thread Control | The program can stop the time thread safely. |

# Why This Design?

The program separates tasks clearly: one for showing the time, and another for background work. This makes the clock smooth and accurate, even if something else is running in the background.

# Error Handling

The program uses try-catch blocks to handle errors when the threads are paused with sleep. This keeps the program from crashing if something unexpected happens.

# Summary

This Java clock program shows how threads can be used in real life. It runs two things at once, keeps the clock accurate with thread priority, and stops safely after some time.