Smart Task Scheduler with Priority Queues

Introduction

A lightweight desktop task manager that prioritizes tasks by urgency using a priority queue, provides a simple Swing UI, reminder alerts, and local JSON persistence.

Abstract

Tasks are modeled with priority and deadlines, ordered by urgency in a PriorityQueue, managed via a Swing UI with filters, reminders via Timer, and saved/loaded from a JSON file; packaged as an executable JAR.

Tools Used

- Language and Runtime: Java (JDK 17)
- UI: Java Swing (JTable, dialogs, listeners)
- Data Structures: PriorityQueue for urgency ordering; HashMap for fast ID lookups
- Scheduling: java.util.Timer + TimerTask (with SwingUtilities for UI thread)
- Persistence: File I/O with a custom minimal JSON serializer/deserializer
- Build/Packaging: Maven compiler (release 17) and assembly (fat JAR) or IntelliJ Artifacts

Steps Involved in Building the Project

- Requirements and Design
  - Defined core features: CRUD, priority-driven ordering, reminders, persistence, filters/search, and a responsive desktop UI.
  - Urgency rule: higher priority first, earlier deadline next, then earlier creation time for tie-breaking.
- Project Setup
  - Created a Java project targeting JDK 17; set main class as com.example.taskscheduler.Main.
  - Configured build for an executable JAR: either Maven with assembly (jar-with-dependencies) and Main-Class in manifest, or IntelliJ “JAR from modules with dependencies.”
- Core Implementation
  - Model: Priority enum (HIGH, MEDIUM, LOW with weights) and Task class (UUID id, title, description, priority, deadline, createdAt, completed, reminderMinutesBefore).
  - Ordering: Implemented urgencyCompareTo combining priority weight, deadline proximity, and creation time.

  - Manager: TaskManager encapsulating a PriorityQueue and an ID index; supports add, update, delete, list, and predicate-based filtering.
- Reminder Service
  - Timer-based scheduling per task; triggers a Swing dialog at deadline minus reminderMinutesBefore.
  - Immediate reminder behavior if the trigger time has already passed and the task is still pending.
  - Rescheduling on app load and after edits; cancellation on completion or deletion.
- UI (Swing)
  - MainFrame: JTable listing tasks with columns for Title, Priority, Deadline, Created, Completed, Reminder(min).
  - Controls: Filter (All, Today, High Priority, Upcoming 7 days, Overdue, Completed), Sort (Urgency/Deadline/Priority), and title search.
  - Actions: Add/Edit/Delete/Mark Complete/Refresh; validation for title, datetime format (yyyy-MM-dd HH:mm), and non-negative reminder minutes.

Conclusion

A portable, offline desktop scheduler that elevates urgent tasks via priority-queue ordering, with a clean Swing UI, dependable reminders, and simple JSON persistence—ready as an executable JAR and extensible for future features (recurring tasks, richer storage, and tray notifications).