ChatRoom

Wednesday, November 5, 2025 7:52 PM

🗒️ Example 6 — Event Logger (Multi-Sink)

Story

Your app produces events (INFO/WARN/ERROR with a message). An **EventBus** broadcasts each event to multiple **sinks**: console, file (simulate), and memory buffer. New sinks can be added without changing the bus.

Entities

- **Subject:** EventBus
- **Observers (sinks):**
  - ConsoleSink — prints formatted lines.
  - FileSink — simulates append to a file (use an internal std::vector<std::string> as "file").
  - RingBufferSink — keeps only the last N events in memory.
  - (Optional) MetricsSink — counts events by level.

Data Model (push)

- Event: struct Event { enum Level { Info, Warn, Error }; Level level; std::string message; }
- The bus pushes the **whole event** to sinks: onEvent(const Event&).

Subject API (suggested)

- attach(Observer*), detach(Observer*), removeAll()
- publish(const Event&) → **notify** all observers
- **Duplicate attach guard** + **snapshot notify** (you already know these)

Behavioral Rules

1. Every publish notifies all attached sinks.
2. Sinks must not modify the event (pass by const&).
3. RingBufferSink holds only the last **N** events (drops oldest).
4. FileSink stores lines like: "ERROR: disk full".
5. ConsoleSink prints: [WARN] Low battery.
6. MetricsSink (if you add it) keeps counters per level and can print a summary on demand.
7. removeAll() clears all sinks; further publishes do nothing.

Acceptance Test (what to simulate)

1. Create EventBus; attach ConsoleSink, FileSink, RingBufferSink(N=3).
2. Publish sequence:
   - INFO "start"
   - WARN "low battery"
   - ERROR "disk full"
   - INFO "retrying"
3. Expected behavior:
   - Console prints all 4 lines.
   - FileSink contains 4 lines in order.
   - RingBuffer contains the **last 3** events only: ["low battery", "disk full", "retrying"].
4. Detach ConsoleSink; publish ERROR "still failing" → file + ring buffer update; **no console**.
5. removeAll(); publish INFO "silent" → no sink receives it.