# Báo cáo Refactor Code – Loại bỏ vi phạm nguyên tắc YAGNI

## 1. Mô tả thay đổi chính

Tên phương thức cũ: addNewTaskWithViolations  
Tên phương thức mới: addNewTask  
  
Lý do thực hiện refactor:  
- Vi phạm nguyên tắc YAGNI (You Aren’t Gonna Need It):  
 + Trong phiên bản cũ có sử dụng các thuộc tính như isRecurring và recurrencePattern nhưng hoàn toàn không có logic xử lý nào kèm theo.  
 + Việc sinh ID bằng UUID là dư thừa và không cần thiết với phạm vi sử dụng hiện tại.  
- Vi phạm nguyên tắc DRY (Don’t Repeat Yourself):  
 + Logic xử lý file (đọc, ghi, cập nhật) không được tối ưu, trùng lặp ở nhiều nơi.

## 2. So sánh code trước và sau khi refactor

|  |  |  |
| --- | --- | --- |
| Tiêu chí | Code Trước (Vi phạm) | Code Sau (Refactor) |
| Cách sinh ID | UUID.randomUUID().toString() – phức tạp, không cần thiết | AtomicInteger – đơn giản, tăng dần theo số thứ tự |
| Tính năng không cần thiết (YAGNI) | Có is\_recurring và recurrence\_pattern không dùng đến | Đã loại bỏ hoàn toàn |
| Xử lý file | loadTasksFromDb() không cập nhật ID | Cập nhật currentId để đảm bảo ID không trùng |
| Thông báo và kiểm tra lỗi | Thông báo chung, không rõ ràng | Tách rõ từng bước kiểm tra và thông báo cụ thể |

## 3. Hiệu quả sau khi refactor

- Mã nguồn gọn gàng, dễ đọc hơn.  
- Không còn tồn tại mã dư thừa không cần thiết (loại bỏ vi phạm YAGNI).  
- Tối ưu logic xử lý ID và file dữ liệu.  
- Tuân thủ nguyên tắc Clean Code: rõ ràng, đơn nhiệm, dễ bảo trì.