

[Bài đọc] Tổng quan về khái niệm Clean code



1. Clean code?

- "Clean Code" (Mã sạch) là một khái niệm trong lập trình chỉ những đoạn mã:
 - Dễ đọc
 - Dễ hiểu
 - Dễ bảo trì
 - Ít lỗi
- Thuật ngữ này được phổ biến bởi cuốn sách nổi tiếng *"Clean Code: A Handbook of Agile Software Craftsmanship"* của **Robert C. Martin (Uncle Bob)**
- "Code is clean if it can be understood easily - by everyone on the team." - Robert C. Martin

2. Tại sao cần Clean code

- Giúp dễ đọc - dễ hiểu cho người khác và cả chính bạn trong tương lai
- Giảm sai sót và lỗi logic
- Dễ dàng bảo trì, mở rộng
- Là nền tảng để làm việc nhóm hiệu quả

3. Các nguyên tắc chính trong Clean code

- Đặt tên rõ nghĩa
 - Tốt: `getUserName()`, `calculateSalary()`
 - Kém: `getData()`, `doStuff()`
 - Tên biến, hàm, class nên cho biết rõ mục đích sử dụng
- Ngắn gọn, đúng mục tiêu
 - Một hàm nên chỉ thực hiện một nhiệm vụ duy nhất (Single Responsibility)

```
// ❌ Không sạch
public void processUser() {
    validate();
    saveToDatabase();
    sendEmail();
}

// ✅ Sạch hơn
public void validateUser() { ... }
public void saveUserToDatabase() { ... }
public void sendUserEmail() { ... }
```

- **Tránh lặp lại (Don't Repeat Yourself – DRY)**
 - Viết các đoạn code tái sử dụng thay vì sao chép dán lại nhiều nơi
- **Comment đúng lúc**
 - Code tốt nên **ít cần comment**, vì chính code đã tự giải thích. Tuy nhiên, với đoạn phức tạp thì comment rõ ràng sẽ rất hữu ích
- **Format mã sạch sẽ**
 - Dùng **indentation hợp lý** (thụt lề)
 - Cách dòng rõ ràng giữa các block code
 - Nhóm logic cùng loại lại với nhau

4. Một số ví dụ Clean Code trong Java

- Tên biến rõ nghĩa:

```
int age; // Tốt
int a;   // Không rõ ràng
```

- Hàm ngắn, đúng mục đích:

```
public double calculateTotalPrice(List<Item> items) {  
    double total = 0;  
    for (Item item : items) {  
        total += item.getPrice();  
    }  
    return total;  
}
```

Tài liệu đọc thêm: <https://www.freecodecamp.org/news/clean-coding-for-beginners>

Danh sách các bài học

