**Bài soạn Clean code vs Dirty code - *Đỗ Văn Hoàng CGHN C0624G1***

|  |  |
| --- | --- |
| Từ khóa | Ghi chú |
|  | |  |  |  | | --- | --- | --- | | *Tiêu chí* | Clean code | Dirty code | | *Độ rõ ràng và dễ hiểu* | Mã được viết rõ ràng, dễ đọc và dễ hiểu. Các biến, hàm và lớp đều được đặt tên rõ nghĩa và mô tả chính xác chức năng của chúng. | Mã thường khó đọc và khó hiểu. Các biến, hàm và lớp có thể được đặt tên không rõ ràng hoặc không có ý nghĩa, gây khó khăn trong việc hiểu chức năng của mã. | | *Cấu trúc và tổ chức* | Mã được tổ chức tốt với cấu trúc hợp lý. Các phương thức và lớp được chia nhỏ theo chức năng, tuân theo các nguyên tắc thiết kế. | Mã thiếu cấu trúc và tổ chức, các phương thức và lớp thường dài và phức tạp, thực hiện nhiều chức năng, vi phạm các nguyên tắc thiết kế. | | *Khả năng mở rộng và bảo trì* | Dễ dàng mở rộng và bảo trì. Mã được thiết kế để có thể thêm các tính năng mới hoặc sửa đổi một cách dễ dàng mà không ảnh hưởng đến các phần khác của mã. | Khó mở rộng và bảo trì. Việc thêm tính năng mới hoặc sửa đổi mã có thể làm phát sinh nhiều lỗi và ảnh hưởng đến các phần khác của mã. | | *Trùng lặp mã* | Tránh sự trùng lặp mã. Các đoạn mã giống nhau được trích xuất ra thành các phương thức hoặc lớp chung để tái sử dụng. | Có nhiều mã trùng lặp, các đoạn mã giống nhau xuất hiện ở nhiều nơi trong chương trình, làm tăng độ phức tạp và khó khăn trong việc duy trì. | | *Sử dụng giá trị "magic"* | Tránh sử dụng các giá trị "magic". Sử dụng các biến hằng số hoặc cấu trúc dữ liệu có ý nghĩa để thay thế. | Sử dụng nhiều giá trị "magic", các giá trị số hoặc chuỗi xuất hiện trong mã mà không có ý nghĩa rõ ràng hoặc không được giải thích. | | *Đặt tên* | Tên biến, hàm và lớp được đặt rõ ràng và có ý nghĩa, giúp người đọc dễ dàng hiểu được chức năng của chúng. | Tên biến, hàm và lớp không rõ ràng, không liên quan hoặc quá ngắn, làm cho người đọc khó hiểu được chức năng của chúng. | | *Kiểm thử* | Có unit test và acceptance test đầy đủ, đảm bảo mã hoạt động như mong đợi và giúp phát hiện lỗi nhanh chóng. | Thiếu unit test hoặc kiểm thử không đầy đủ, dẫn đến việc khó phát hiện lỗi và đảm bảo chất lượng mã. | | *Bình luận trong mã* | Sử dụng bình luận đúng cách để giải thích các logic phức tạp hoặc các quyết định thiết kế đặc biệt. | Lạm dụng bình luận để giải thích mã. | | *Sự phụ thuộc* | Có ít sự phụ thuộc giữa các thành phần, tăng tính linh hoạt và dễ bảo trì. | Có nhiều mối quan hệ phụ thuộc giữa các thành phần, làm giảm tính linh hoạt và khó khăn trong việc bảo trì. | | *Phản ánh thiết kế* | Mã thể hiện được ý tưởng và nguyên tắc của thiết kế, giúp mọi người hiểu rõ hơn về giải pháp tổng thể. | Mã không phản ánh rõ ý tưởng thiết kế, làm cho người đọc khó hiểu về giải pháp và ý định của người viết mã. | |
|  | Kết luận |