**TỔNG QUAN VỀ NGUYÊN LÝ SOLID TRONG THIẾT KẾ HƯỚNG ĐỐI TƯỢNG**

1. **Giới thiệu chung**
2. **Ưu nhược điểm của SOLID trong thiết kế**
3. **Cách sử dụng của nguyên lý SOLID**

**Nguyên lý 1 SRP: The Single Responsibility Principle**



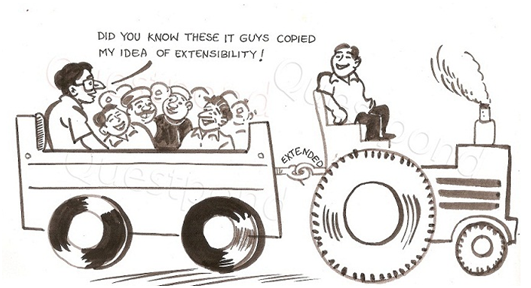
*Hình mô tả nguyên lý Đơn chức năng*

**Slogan:**

Một class chỉ nên giữ 1 trách nhiệm duy nhất, chỉ có thể sửa đổi class với 1 lý do duy nhất.

A class should have one and only one reason to change, meaning that a class should have only one job

**Nguyên lý 2 OCP: The Open-Closed Principle**



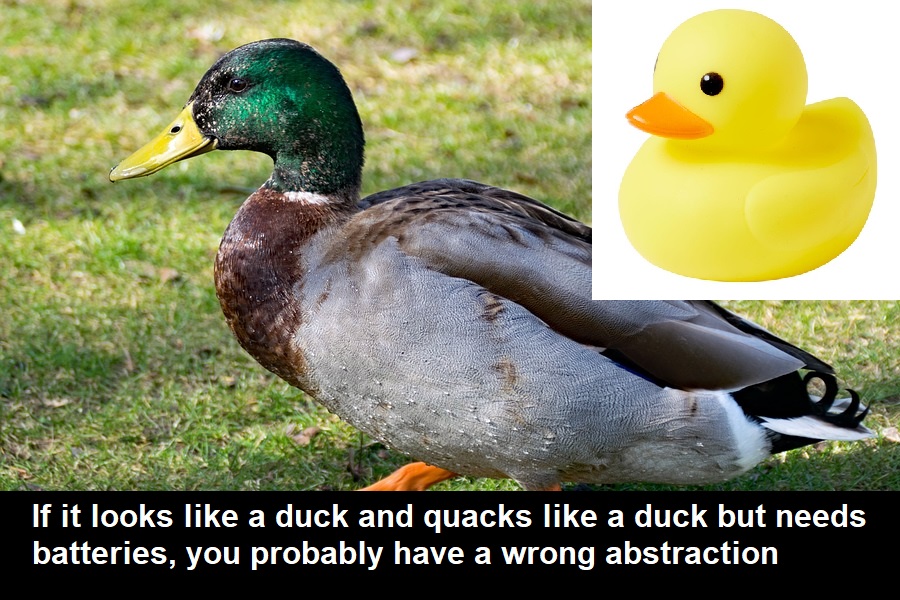
*Hình mô tả nguyên lý đóng mở trong thiết kế*

**Slogan:**

Có thể thoải mái mở rộng 1 class, nhưng không được sửa đổi bên trong class đó.

Objects or entities should be open for extension, but closed for modification

**Nguyên lý 3 LSP\*: Liskov Substitution Principle**



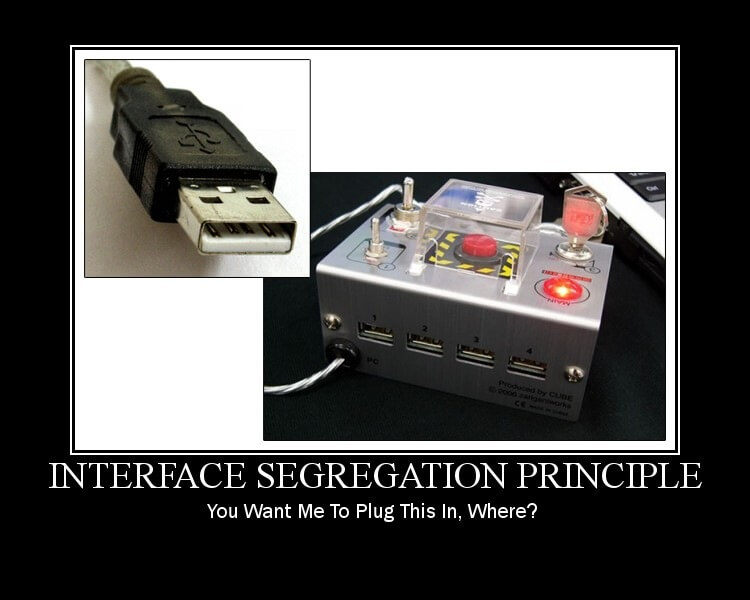
*Hình: Mô tả nguyên lý thay thế trong thiết kế hướng đối tượng*

**Slogan:**

Trong một chương trình, các object của class con có thể thay thế class cha mà không làm thay đổi tính đúng đắn của chương trình.

Objects in a program should be replaceable with instances of their subtypes without altering the correctness of that program.

**Nguyên lý 4 ISP: The Interface Segregation Principle**

**

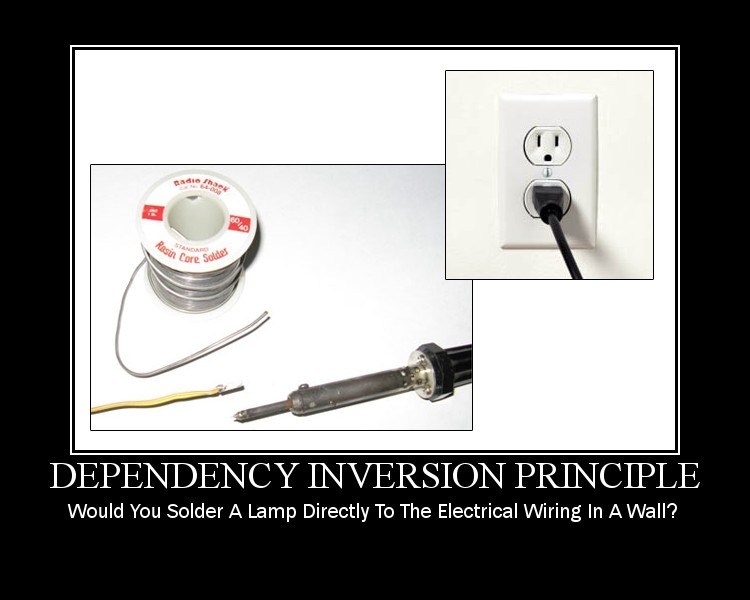
*Hình mô tả nguyên lý phân tách*

**Slogan:**

Thay vì dùng 1 interface lớn, ta nên tách thành nhiều interface nhỏ, với nhiều mục đích cụ thể.

Many client-specific interfaces are better than one general-purpose interface.

**Nguyên lý 5 DIP: The Dependency Inversion Principle**



*Hình mô tả nguyên lý đảo ngược phụ thuộc*

**Slogan:**

1. Các module cấp cao không nên phụ thuộc vào các modules cấp thấp. Cả 2 nên phụ thuộc vào abstraction.
2. Interface (abstraction) không nên phụ thuộc vào chi tiết, mà ngược lại. ( Các class giao tiếp với nhau thông qua interface, không phải thông qua implementation.)
3. *High-level modules should not depend on low-level modules. Both should depend on abstractions.*
4. *Abstractions should not depend upon details. Details should depend upon abstractions.*
5. **Các vấn đề chính**
6. **Dấu hiệu**
7. **Kết luận**
8. **Tài liệu tham khảo**

[1] Tổng quan về nguyên lý SOLID

<https://gpcoder.com/4200-cac-nguyen-ly-thiet-ke-huong-doi-tuong/>

[2] Nguyên lý đảo ngược phụ thuộc

<https://tuanfadbg.com/dependency-inversion-principle-nguyen-li-nghich-dao-phu-thuoc/>

[3] Nguyên lỹ SOLID trong lập trình hướng đối tượng

<https://tuanfadbg.com/series/nguyen-li-solid-nang-tam-developer/>