**SOLID**

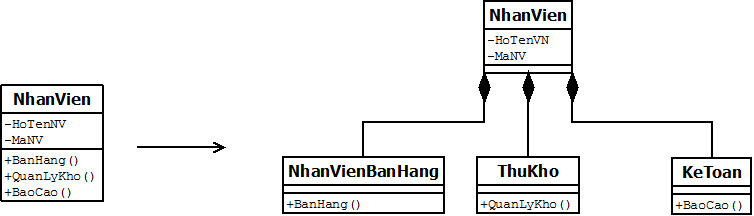
Là tập hơp 5 nguyên tắc

1. **Single responsibility principle (Nguyên tắc đơn nhiệm) (S)**

Nội dung: ***1 class chỉ nên giữ 1 trách nhiệm duy nhất.***

* Nên tách 1 class thành nhiều class nếu class đó có nhiều chức năng. Như vậy sẽ dễ dàng chỉnh sửa hơn và ít gây ra bug

Một nhân viên không thể đảm nhận quá nhiều trách nhiệm, nên tách thành nhiều loại nhân viên, mỗi loại sẽ đảm nhận một trách nhiệm khác nhau

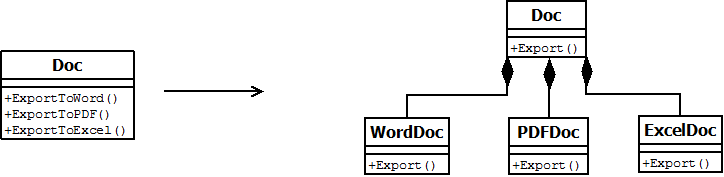


1. **Open/closed principle (Nguyên tắc đóng/mở) (O)**

Nội dung: ***Có thể thoải mái mở rộng 1 class, nhưng không được thay đổi bên trong class đó.***

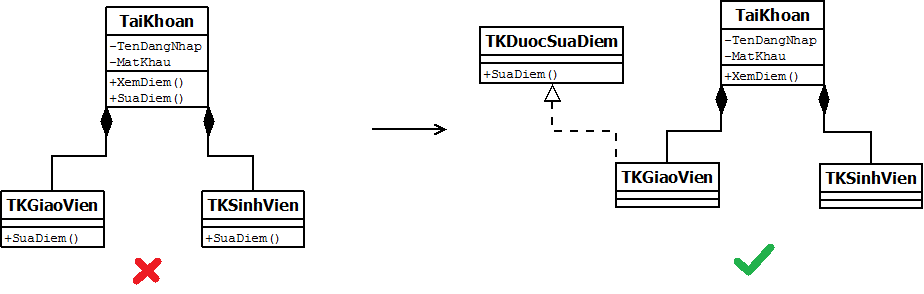
* Nên tạo class mới để mở rộng class cũ, tuyệt đối không sửa đổi class cũ

Khi định nghĩa các hàm xuất file trong 1 class Doc, nếu muốn xuất một định dạng khác ta phải chỉnh sửa lại class đó. Thay vào đó, ta tạo các class con kế thừa từ class Doc, mỗi class con sẽ xuất 1 định dạng khác nhau. Từ đó mở rộng class Doc mà không ảnh hưởng đến class.



1. **Liskov Substitution Principle (Nguyên tắc thay đổi Liskov) (L)**

Nội dung: ***Trong 1 chương trình, class con có thể thay thế class cha mà không ảnh hưởng đến tính đúng đắn của trương trình.***



1. **Interface Segregation Principle (Nguyên tắc phân tán interface) (I)**

Nội dung: ***Thay vì dùng 1 interface lớn, ta nên tách thành nhiều interface nhỏ với các chức năng khác nhau.***

Tương tự nguyên tắc 1 (nguyên tắc S)

1. **Dependency inversion principle (Nguyên tắc phụ thuộc ngược) (D)**

Nội dung:

1. ***Các module cấp cao không nên phụ thuộc vào các module cấp thấp. Cả 2 nên phụ thuộc vào abstraction.***
2. ***Interface (abstraction) không nên phụ thuộc vào chi tiết, mà phải ngược lại. (Các class giao tiếp với nhau thông qua interface chứ không phải implementation.)***

Ta nên thiết kế các class ít phụ thuộc nhau nhất có thể

**Cơ chế của nguyên tắc: Có 2 loại đối tượng. Một loại là cung cấp service (dependency) và một loại sử dụng service đó. Đối tượng dependency sẽ cung cấp service từ bên ngoài dối tượng sử dụng service.**

