NGUYÊN LÝ SOLID

Nguyên lý SOLID có vai trò rất quan trọng khi phát triển ứng dụng và được sử dụng rất phổ biến trong thiết kế và lập trình hướng đối tượng.

Một ứng dụng không phải chỉ được code một lần và được dùng mãi mãi. Muốn ứng dụng được ưa chuộng hơn thì cùng với sự phát triển và cải tiến không ngừng của công nghệ, những ứng dụng phần mềm cũng phải không ngừng được nâng cấp, cải tiến cả về số lượng và chất lượng của chức năng phục vụ người dùng.

Thế nhưng không phải dự án phần mềm nào cũng có thể mở rộng và cải tiến một cách dễ dàng. Điều này một phần lớn phụ thuộc vào cách thiết kế trong lập trình.

Một dự án phần mềm được thiết kế tốt có thể giúp cho người phát triển phần mềm dễ dàng chỉnh sửa, bảo trì và mở rộng, cùng với đó là chi phí mở rộng, cải tiến phần mềm cũng được giảm thiểu đến mức tối đa. Ngược lại với một dự án phần mềm được thiết kế không tốt, chi phí để sửa chữa, mở rộng là vô cùng lớn, thậm chí có thể sẽ phải đập đi và xây lại hoàn toàn.

Để giúp các developer hiểu rõ và có thể vận dụng dư duy trong việc lập trình phần mềm, nguyên lý SOLID đã ra đời.

SOLID là một cách gọi tắt của bộ 5 nguyên lý sau đây:

* S: Single responsibility principle
* O: Open-Close principle
* L: Liskov substitution principle
* I: Interface segregation principle
* D: Dependence inversion principle

Nguyên lý S – Single responsibility principle

Nội dung: mỗi class chỉ nên chịu trách nhiệm cho một công việc duy nhất của phần mềm.