**Tính tương thích động – Dependency Inversion principle (DIP)**

## **Nguyên tắc thứ 5 của SOLID nói gì?**

*Mỗi thành phần hệ thống (class, module, …) chỉ nên phụ thuộc vào các abstractions, không nên phụ thuộc vào các concretions hoặc implementations cụ thể.*

Nguyên tắc cũng khá đơn giản về mặt mô hình hoá, bạn hãy tưởng tượng một việc trong thực tế như sau: một hệ thống máy tính sẽ có mainboard là thành phần chính, bộ phận này kết nối các thành phần khác trong hệ thống (như CPU, Ram, ổ cứng, …) lại với nhau để tạo nên một hệ thống hoạt động hoàn chỉnh và thống nhất. Như bạn đã biết, một mainboard có khả năng kết nối nhiều loại Ram, nhiều loại ổ cứng, … Dù những nhà sản xuất Ram hay mainboard là độc lập nhau hoàn toàn, và cũng có rất nhiều loại mainboard và ổ cứng khác nhau, nhưng các bộ phận này được kết nối với nhau rất dễ dàng. Làm thế nào để các nhà sản xuất có thể làm được điều này?

Câu trả lời thực ra rất đơn giản: mọi linh kiện máy tính **dù cho có cấu tạo chi tiết khác nhau** (implement khác nhau), **nhưng luôn giao tiếp với nhau thông qua các chuẩn đã định sẵn** (abstraction), cụ thể ở đây là mainboard giao tiếp với ổ đĩa cứng thông qua chuẩn kết nối chung SATA.