# Facade pattern [fa sa:d]

<https://github.com/doctor-blue/design-patterns/tree/master/Structural/Facade>

* Mỗi một hàm chỉ làm một việc duy nhất. Nguyên tắc S trong solid ( Single Responsibility Principles)
* Gói một subsystem vào một class và cung cấp interface để client sử dụng đơn giản hơn
* Trong trường hợp bạn có nhiều hệ thống con và mỗi lần gọi tới những hệ thống này lại quá sức phức tạp và tốn nhiều thời gian. Vậy **Facade** sẽ giúp đỡ bạn trong việc này. Cũng giống như [**Adapter**](https://github.com/doctor-blue/design-patterns/tree/master/Structural/Adapter)**Facade** cung cấp cho chúng ta một lớp để làm chung gian giữa hai thành phần, nhưng khác nhau ở chỗ Adapter thì giúp chúng ta làm việc giữa hai mã nguồn, còn đối với Facade thì thì sẽ là thành phần bao bọc che giấu đi các hệ thống con để giúp giao tiếp với thành phần khác dễ dàng hơn.
* Như vậy ta có thể hiệu Facade sẽ cung cấp một giao diện cho việc giao tiếp giữa các hệ thống con, giúp người dùng có thể giao tiếp với chúng một cách dễ dàng. Facade giúp che dấu đi xử lý logic phức tạp bên trong hệ thống con.

# Strategy pattern

* Để giải quyết một việc (tùy thuộc arguments) thì sẽ cho một kết quả trong một lần thực thi ( Switch case )

1. a