## Lab 3. Sử dụng mẫu thiết kế cho Hệ thống điều khiển nhà thông minh (SHCS - Smart Home Control System)

### Mục tiêu

Thiết kế và cài đặt hệ thống điều khiển nhà thông minh, trong đó các thiết bị thông minh (đèn, điều hòa, camera, v.v.) có thể được điều khiển và gửi thông báo về bảng điều khiển (dashboard) khi trạng thái thay đổi.  
Sinh viên cần áp dụng các mẫu thiết kế Singleton, Abstract Factory, và Observer.

### Phát biểu bài toán

Một công ty muốn xây dựng hệ thống điều khiển nhà thông minh cho phép người dùng kết nối và điều khiển nhiều loại thiết bị khác nhau (đèn, điều hòa, camera giám sát,...).

Mỗi thiết bị có thể:

* Bật/tắt
* Báo cáo trạng thái (bật/tắt, nhiệt độ, pin, v.v.)

Có nhiều hãng thiết bị khác nhau (Philips, Xiaomi, Samsung,...), và hệ thống cần dễ dàng mở rộng để hỗ trợ các hãng mới.

Một bảng điều khiển sẽ hiển thị trạng thái của tất cả thiết bị và phải tự động cập nhật khi có thay đổi từ các thiết bị (ví dụ: đèn vừa bật/tắt, điều hòa vừa tăng/giảm nhiệt độ, …)

### Các lớp phân tích

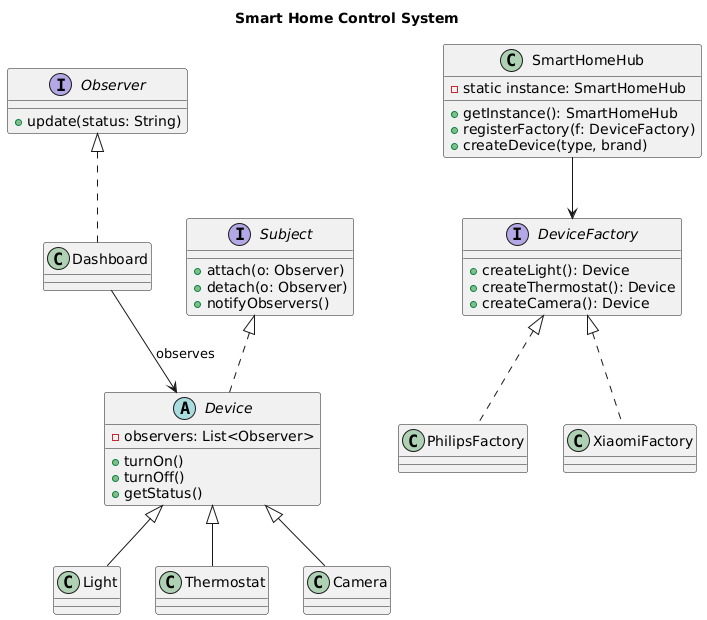
Sau khi phân tích yêu cầu bài toán, chúng ta có thể phát hiện được một số lớp phân tích sau:

* Device (abstract)
* Light (đèn)
* Thermostat (điều hòa)
* Camera (camera giám sát)
* Dashboard (bảng điều khiển)

### Mục tiêu thiết kế

1. Singleton: Lớp SmartHomeHub quản lý tất cả thiết bị, là trung tâm điều phối duy nhất.
2. Abstract Factory: Lớp DeviceFactory tạo các thiết bị theo từng hãng (PhilipsFactory, XiaomiFactory, ...).
3. Observer: Dashboard là observer, nhận thông báo khi trạng thái thiết bị thay đổi.

### Gợi ý sơ đồ thiết kế UML



Gợi ý cho sinh viên:

* Sử dụng Singleton cho SmartHomeHub.
* Áp dụng Abstract Factory để tạo thiết bị theo hãng.
* Áp dụng Observer giữa Device và Dashboard.
* Cho phép thêm hãng mới mà không cần sửa mã cũ.

### Kết quả mong đợi

* Dashboard tự động cập nhật khi thiết bị thay đổi trạng thái.
* Có thể thay đổi hãng sản xuất trong lúc chạy chương trình.
* Chỉ có một trung tâm điều khiển duy nhất (Singleton).