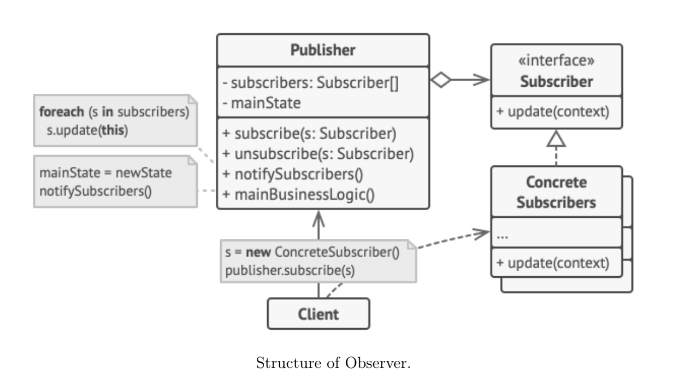
**Observer pattern**

1. **Intent**

Observer là một mô hình design hành vi (behavioral pattern), cho phép định nghĩa một cơ chế đăng ký (subscription) để thống báo đến nhiều đối tượng về bất cứ sự kiện nào xảy ra tới đối tượng mà chúng đang theo dõi (observe).

1. **Structure**



* Lớp **Publisher** cho các sự kiện đáng quan tâm (events of interests) đến các đối tượng khác. Các sự kiện này xảy ra khi lóp publisher thay đổi trạng thái cả nó hay thực hiện một hành vi nào đó. Các lớp Publisher chứa một subscription infrastructure cho phép các subscriber mới tham gia và các subscriber hiện tại rời khỏi danh sách.
* Khi một sự kiện mới diễn ra, lớp publisher xem qua danh sách subscription và gọi phương thức notification được xác định ở giao diện subscriber ở mỗi đối tượng subscribe.
* Giao diện **Subscriber** gọi giao diện notification. Ở đa số trường hợp, nó chứa đụng một phương thức update duy nhất. Phương thức này có thể có một vài thông số mà cho phép lớp publisher thông qua một số chi tiết của sự kiện cùng với đó là update (cập nhật).
* Lớp **Concrete Subscribers** thực hiện một số hành động phản ứng lại các thông báo (notification) từ lớp publisher. Tất cả các lớp này phải áp dụng một giao diện chung, để lớp publisher không bị lẫn vào các lớp concrete.
* Lớp **Client** tạo ra các đối tượng publisher và subscriber riêng rẽ và rồi đăng ký các đối tượng subscriber cho các cập nhật của publisher.

1. **When do you use it?**

* Use the Observer pattern when changes to the state of one object may require chang-ing other objects, and the actual set of objects is unknown beforehand or changes dynamically.
* Use the pattern when some objects in your app must observe others, but only for a limited time or in specific cases.