**Đề bài:** Tìm hiểu về một design pattern (Observer). Nêu 3 ví dụ sử dụng, cài đặt 1 ví dụ. Chỉ ra các điểm cần lưu ý.

**Bài làm:**

1. **Tìm hiểu về Observer pattern**

Observer pattern được sử dụng trong trường hợp các đối tượng có mối quan hệ một - nhiều, khi một đối tượng thay đổi, các đối tượng phụ thuộc vào nó sẽ được thông báo tự động.

Mô hình chung của Observer pattern:



Quan hệ Subject – Observer là quan hệ một - nhiều. Khi đối tượng Subject thay đổi trạng thái, phương thức notifyAllObservers() được gọi, trong đó mỗi đối tượng Observer phụ thuộc sẽ thực hiện phương thức update().

1. **Nêu 3 ví dụ sử dụng Observer pattern**

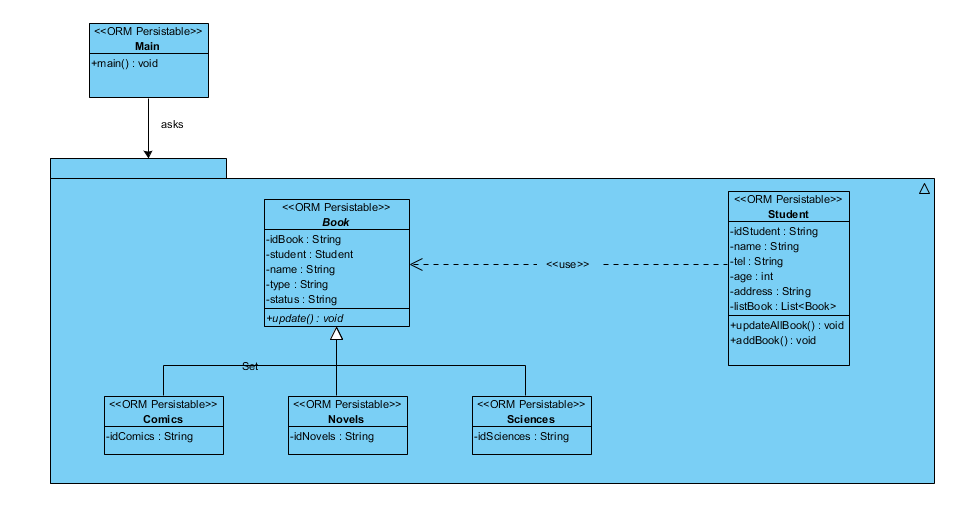
Ví dụ 1: Một kênh Youtube có thể có nhiều người đăng ký nhận thông báo. Khi kênh Youtube đó đăng video mới thì những người đăng ký sẽ nhận được thông báo tự động.

Ví dụ 2: Khi một ứng dụng trên store được nhà phát triển cập nhật, những người dùng đã tải ứng dụng đó sẽ được thông báo về cập nhật đó.

Ví dụ 3: Ví dụ về quản lý thư viện trong phần 3

1. **Ví dụ cài đặt**

Sơ đồ lớp của Observer pattern cho quản lý thư viện



Mô tả bài toán: Quản lý thư viện

- Tập thực thể Student có các thông tin về name, age, tel, address. Các thực

thể Student được phân biệt với nhau qua idStudent.

- Tập thực thể Book có các thông tin về name, type, status. Các thực thể

Book được phân biệt với nhau qua idBook.

- Trong Book thì có nhiều loại như Comics, Sciences, Novels.

- 1 Student có thể mượn được nhiều Book khác nhau, nhưng 1 Book chỉ

mượn được mượn bởi 1 Student (Giải thích: mỗi Book sẽ có id riêng (cùng

là comics về OP giống nhau nhưng mà id nó cũng khác nhau), nếu Book A

đã được mượn thì ko còn tồn tại Book A trong thư viện nữa).

- Student và Book sẽ có quan hệ với nhau là một nhiều => Trong class

Student sẽ phải có 1 List<Book> listBook và trong class Book sẽ có biến

kiểu Student student.

- Trong Book sẽ lưu lại thông tin của Student đã hoặc đang mượn Book đó.

- Trong Student sẽ có 1 danh sách các Book mà Student đó đã hoặc đang mượn.

- Khi mà delete Student thì dữ liệu ở Book mà có quan hệ với Student cũng sẽ được tự dộng delete.

**Các điểm lưu ý**

* Nên sử dụng trong trường hợp có nhiều đối tượng phụ thuộc vào trạng thái của một đối tượng.
* Được sử dụng trong các Event Listener trong các bộ công cụ GUI, truyền thông, nguồn cấp dữ liệu,…