**Mẫu decorator**  
- Ý nghĩa là mẫu có 1 lớp cha là lớp trừu tượng/ giao diện

- 2 lớp con kế thừa bao gồm :

- 1 lớp đại diện yêu cầu cần thực hiện  
- 1 lớp là lớp yêu cầu. Lớp này là lớp trừu tượng. Lớp có các lớp con và các lớp con là các yêu cầu của bài toán

Ví dụ Lắp 1 chiếc pc bao gồm: Ram, CPU, Màn hình

--Phân tích—

Lớp cha là lớp PC và gồm 2 lớp con

-- Lớp 1: là lớp Concreate: thực hiện yêu cầu lắp

-- Lớp 2: là lớp PcDecorater: là lớp yêu cầu gồm 3 lớp con bao gồm: RAM,CPU và màn hình

**Mẫu Chain:** là mẫu chuyển các công việc cho lớp có thể xử lý công việc cho lớp có thể xử lý được yêu cầu đó.

Mẫu gồm 3 lớp:   
- Lớp Cha – Lớp đại diện cho 2 lớp con và chứa các phương thức thực hiện yêu cầu của bài, 1 phương thức là Lớp Cha ketiep(LopCha v): Chịu trách nhiệm chuyển lên đối tượng có thể thực hiện yêu cầu.

- Lớp con 1: là Lớp dùng đẻ tạo các đối tượng cấp dưới và kế thừa các phương thức của lớp Cha.

Có 1 biến là Lớp cha: phụ trách chuyển đối tượng lên đối tượng cấp cao hơn có thể giải quyết yêu cầu.

- Lớp con 2: là Lớp dùng để tạo duy nhất 1 đối tượng là cấp cao nhất, kế thừa các phương thức của lớp Cha.

**Mẫu Observer  
- Subject:** là abstract/interface chịu trách nhiệm quản lý và thông báo đến các observer nào đã đăng ký khi có sự thay đổi.

Bao gồm 3 phương thức: Attach(), Detach(),Notify() và chứa 1 biến là đại diện observer(Lớp **Obsever)**.

Subject còn có các lớp con kế thừa là ConcreteSubject.

- **Observer:** là**:** là abstract/interface chịu trách nhiệm cập nhập thông tin thay đổi cho các **ConcreteObserver** và có phương thức Update()

- ConcreateObserver: sẽ kế thừa từ lớp Observer và có 1 biến là (Lớp **ConCreateSubject**) và có các phương thức khác như (Đăng ký, Hủy đăng ký).

**Mẫu Iterator**

Định nghĩa mẫu Iterator sẽ chịu trách nhiệm duyệt các phần tử trong tập hơp.

Tập có thể là 1 List, Mảng hoặc 1 tập hợp do người dùng quy định

Mẫu iterator được java hỗ trợ với class có tên là iterator bao gồm 2 phương thức chính:

hastNext() và Next

hastNext(): Chịu trách nhiệm duyệt các phần tử có trong mảng

Next(): Chịu trách nhiệm lấy giá trị của các phần tử đã được duyệt

--------------------

Các Phương pháp làm:

- Phương pháp 1: Duyệt 1 mảng có sẵn

- Phương pháp 2: Duyệt 1 tệp có sẵn.

- Phương pháp 3: Duyệt 1 tập hợp do người dùng định nghĩa. Cách này khó vì phải người dùng phải tự

các iterator.

**- Phương pháp 1: Làm trong hàm main**

- Tạo 1 List<Kiểu dữ liệu> = khởi tạo mảng mà add phần tử vào

- Tạo iterator trùng với kiểu dữ liệu mảng và add iterator cho list

- Thiết lập điều kiện dừng cho iterator và sử lý yêu cầu bên trong điều kiện dừng

**- Phương pháp 2:**

- Sau khi tạo xong hàm cấu trúc cho đối tượng thì chúng ta bắt đầu tạo 1 hàm để duyệt phần tử

- Hàm này phải là static và tham số truyền vào sẽ là 1 Iterator<Lớp cấu trúc>

- Trong điều kiện duyệt phần tử sẽ phải tạo 1 đối tượng của Lớp cấu trúc

và cho Iterator lấy dữ liệu trong đối tượng.

- Sau khi lấy được dữ liệu sẽ bắt đầu sử lý yêu cầu

- Sau đó sẽ viết hàm main giống với cách 1 nhưng chỉ khác là sẽ dùng hàm đã tạo để duyệt phần tử.