**Mẫu Flyweight**

1. **Thông tin mẫu**

* Tên đầy đủ: Flyweight Pattern
* Phân loại: Mẫu kiến trúc

1. **Mục đích, ý định**

* Sử dụng việc chia sẻ để giúp cho việc xử lý các đối tượng lớn một cách hiệu quả.
* Một mẫu flyweight là một đối tượng chia sẻ mà có thể sử dụng trong đồng thời nhiều ngữ cảnh. Mẫu flyweight hoạt động như một đối tượng độc lập trong mỗi thời điểm.

1. **Động lực sử dụng**

* Thay vì phải làm việc với nhiều đối tượng độc lập, to lớn, bạn giảm bớt kích thước chúng bằng việc tạo một tập hợp các đối tượng dùng chung nhỏ hơn, gọi là flyweight mà bạn có thể cài đặt vào lúc thực thi chương trình để chúng trông giống như những đối tượng lớn hơn.
* Mỗi đối tượng to lớn có thể tiêu tốn nhiều tài nguyên hệ thống, bằng cách tách những điểm giống nhau của các đối tượng này, và dựa trên việc cấu hình thời gian thực để mô phỏng lại các đối tượng lớn, bạn đã làm giảm bớt gánh nặng lên tài nguyên hệ thống.

1. **Khi nào sử dụng**

* Khi ứng dụng sử dụng một số lượng lớn các đối tượng.
* Chi phí lưu trữ cao bởi vì dung lượng của mỗi đối tượng.
* Đa số các trạng thái của đối tượng thể hiện là extrinsic state (trạng thái bên ngoài).
* Nhiều nhóm các đối tượng có thể được thay thế bằng một số các đối tượng có liên quan được chia sẻ một khi các extrinsic state bị gỡ bỏ.

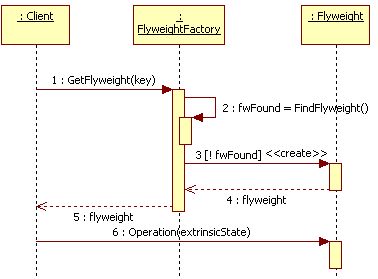
1. **Cấu trúc**



1. **Các thành viên - Mối quan hệ:**

* **Flyweight:**
  + Khai báo một interface mà thông qua nó các flyweight có thể nhận và xử lý trên các trạng thái extrinsic.
* **ConcreteFlyweight**:
  + Thực thi giao diện Flyweight và thêm vào phần lưu trữ cho trạng thái nội (intrinsic state) nếu có.
  + Một đối tượng ConcreteFlyweight phải có khả năng chia sẻ được. Bất kì trạng thái nào mà nó lưu trữ phải là trạng thái nội; nghĩa là nó phải độc lập với bối cảnh của đối tượng ConcreteFlyweight.
* **UnsharedConcreteFlyweight**:
  + Không phải tất cả các lớp con đều cần được chia sẻ.
  + Flyweight interface cho phép chia sẻ, nhưng nó không ép buộc.
* **Client:**
  + Giữ tham chiếu đến một (nhiều) đối tượng flyweight.
  + Tính toán hoặc lưu trữ các trạng thái ngoại của flyweight.
* **Flyweight Factory:**
  + Tạo và quản lý các đối tượng flyweight
  + Đảm bảo rằng các flyweight được chia sẻ một cách đúng đắn. Khi một client yêu cầu một flyweight, FlyweightFactory trả về một flyweight hoặc tạo mới nếu không có.

1. **Sự cộng tác**



1. **Các hệ quả**
2. **Lưu ý cài đặt**
3. **Hệ thống sử dụng**
4. **Mẫu liên quan**

* Mẫu Flyweight thường được kết hợp với mẫu Composite để hiện thực cấu trúc cây phân cấp logic.
* Các đối tượng State và Strategy được hiện thực như các flyweight thì tốt hơn.

1. **Demo**