

# Phân tích thiết kế Hệ thống

Giảng viên: Nguyễn Bá Ngọc

Hà Nội-2021

# Mẫu thiết kế

# Khái niệm

Mỗi mẫu thiết kế mô tả 1 vấn đề thiết kế thường xuyên xuất hiện cùng với giải pháp khái quát cho vấn đề đó, để bạn có thể áp dụng nó trong nhiều tình huống.

# Các thành phần của mỗi mẫu thiết kế

- Tên - Ngắn gọn, gợi mở ý nghĩa, làm giàu bộ từ vựng thiết kế.
- Vấn đề - Khó khăn được tháo gỡ bởi mẫu thiết kế, có thể được mô tả trong tình huống cụ thể.
- Giải pháp - Mô tả các thành phần, mối quan hệ, các trách nhiệm và cộng tác.
- Hệ quả - Ưu và nhược điểm khi áp dụng mẫu thiết kế.

# Phân loại mẫu thiết kế

- Mẫu tạo / Creational patterns - Khái quát hóa tiến trình tạo đối tượng.
- Mẫu cấu trúc / Structural patterns - Khái quát hóa cách kết hợp các đối tượng để tạo các cấu trúc lớn hơn.
- Mẫu hành vi / Behavioral patterns - Khái quát hóa các giải thuật và cách phân chia trách nhiệm giữa các đối tượng.

# Nội dung

- Các mẫu tạo
- Các mẫu cấu trúc
- Các mẫu hành vi

# Nội dung

- Các mẫu tạo
- Các mẫu cấu trúc
- Các mẫu hành vi

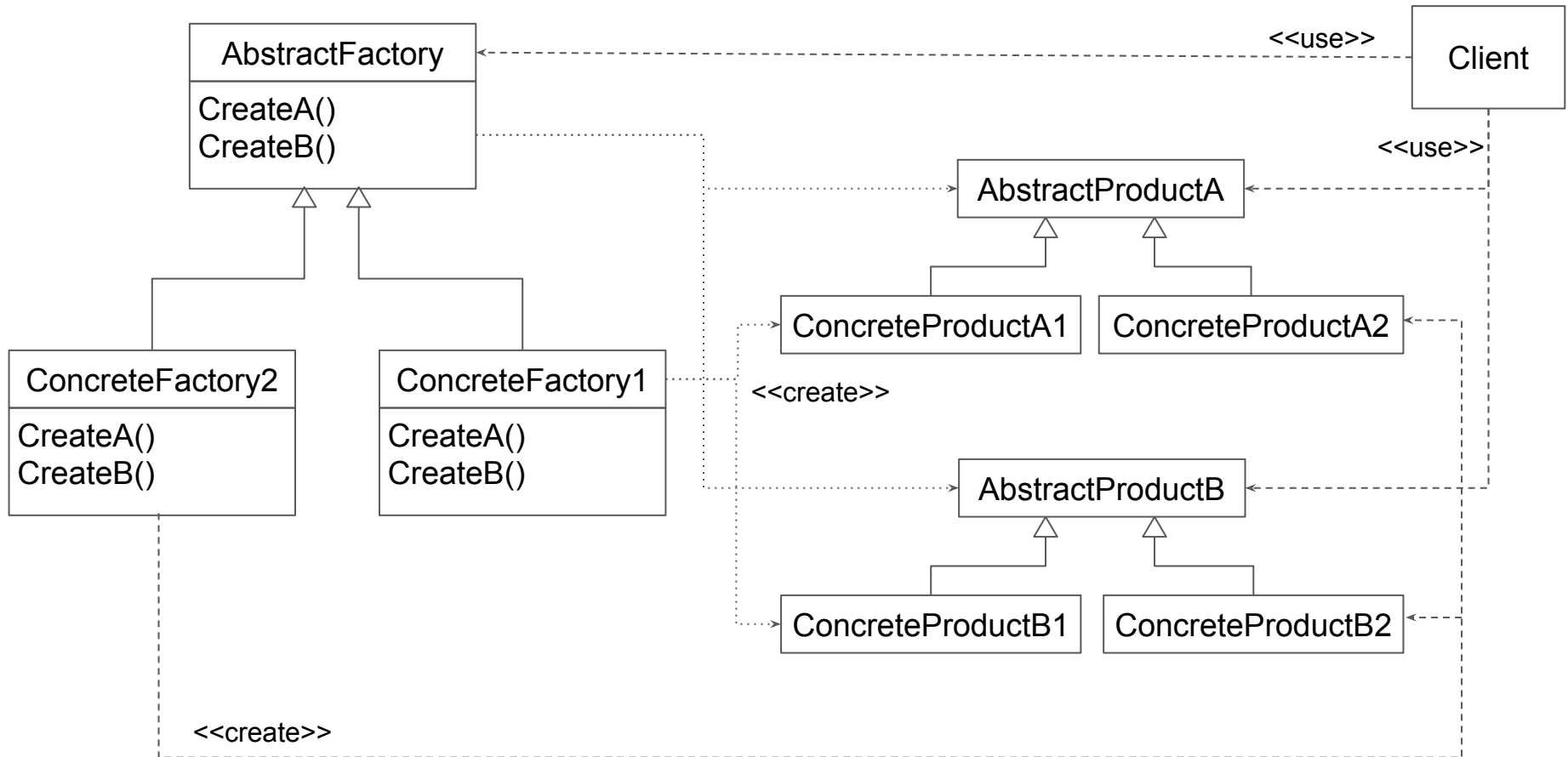
# Nhà máy ảo

## Abstract Factory

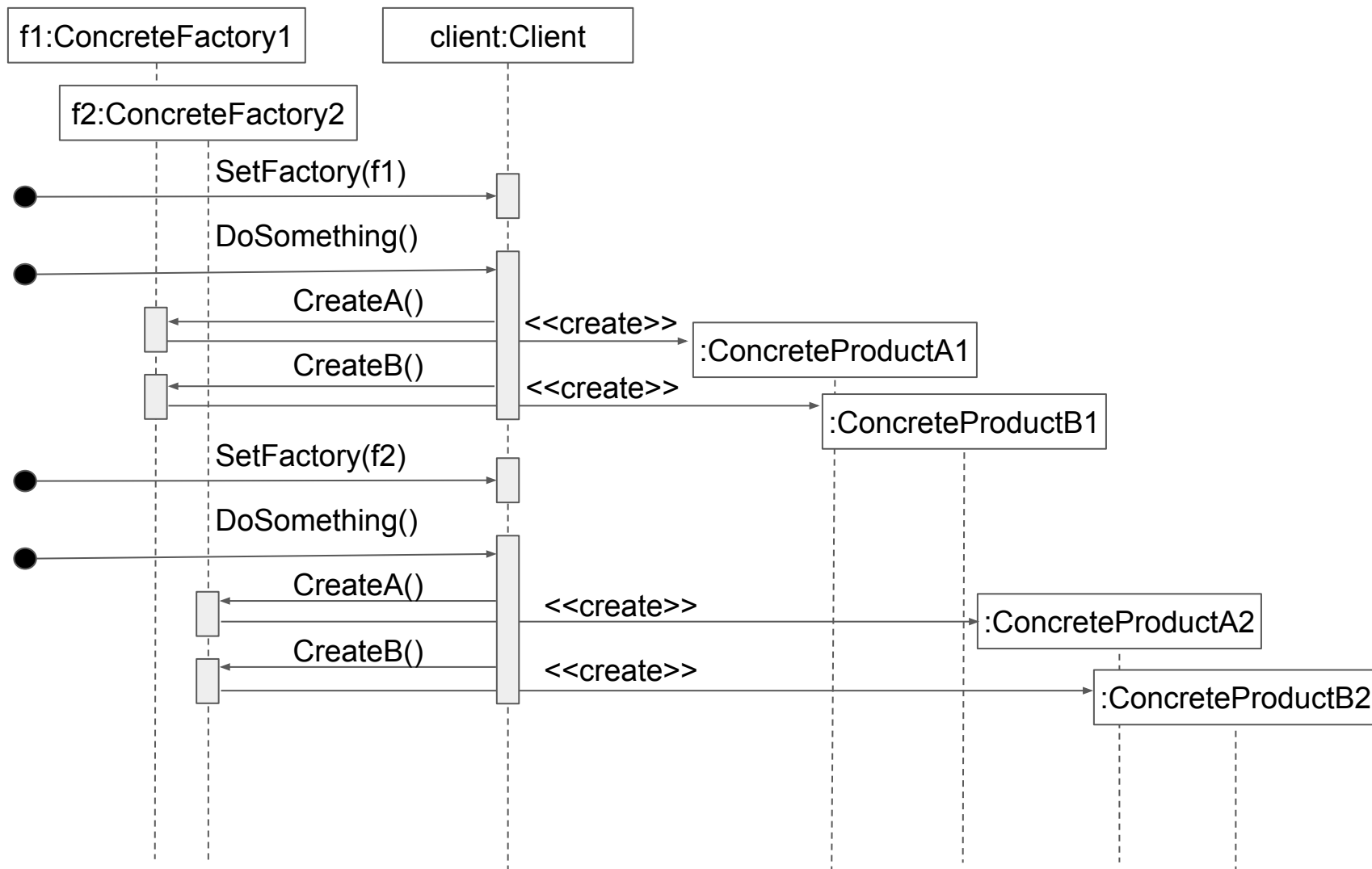
Cung cấp 1 giao diện để tạo 1 quần thể đối tượng có liên quan, hoặc phụ thuộc nhưng không sử dụng những tên lớp cụ thể.



# Nhà máy ảo: Cấu trúc



# Nhà máy ảo: Kịch bản tương tác

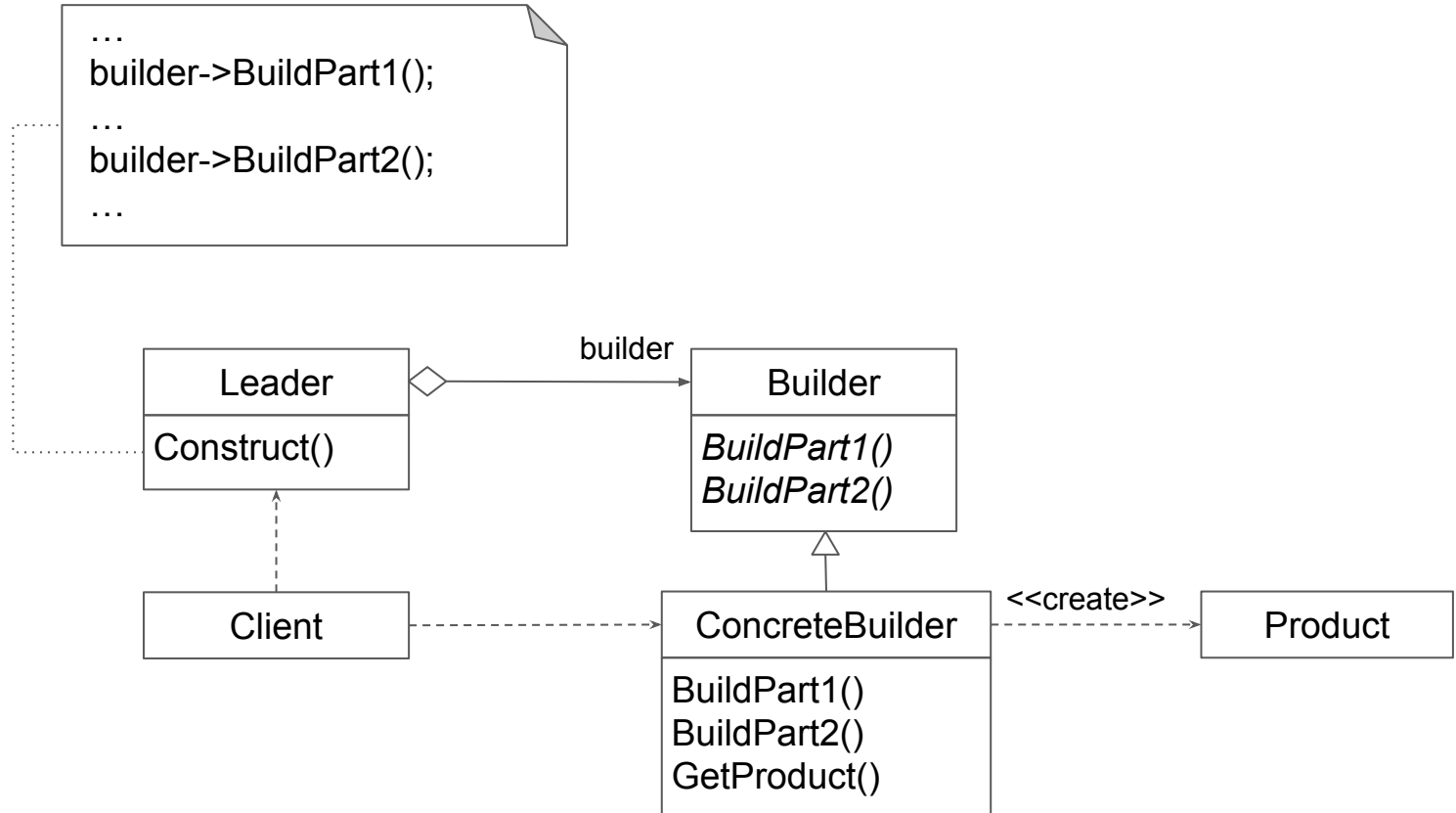


# Thợ xây

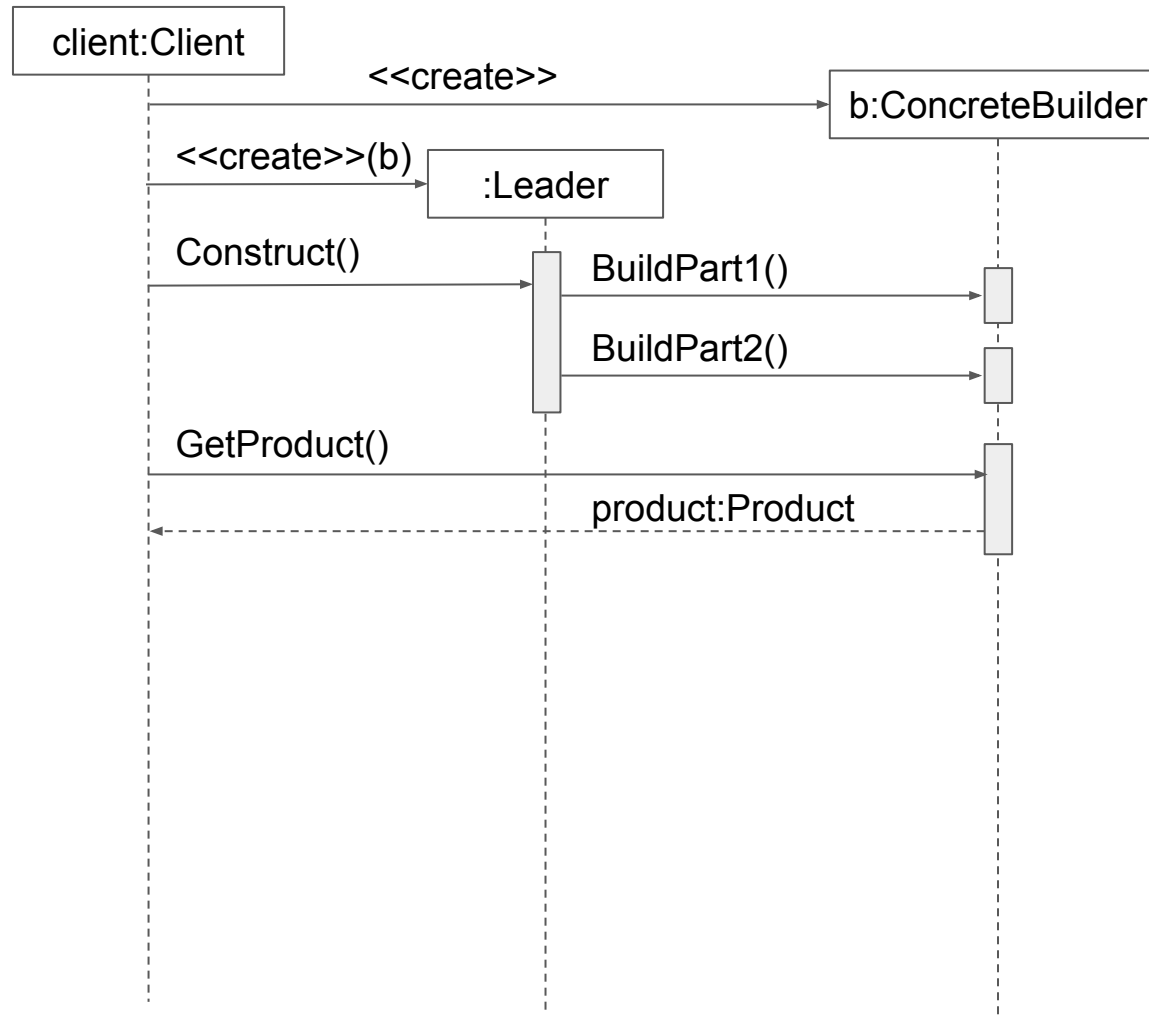
## Builder

Tách biệt việc khởi tạo 1 đối tượng phức tạp và biểu diễn của nó, nhờ đó cùng 1 tiến trình khởi tạo có thể tạo nhiều biểu diễn khác nhau.

# Thợ xây: Cấu trúc



# Thợ xây: Hành vi

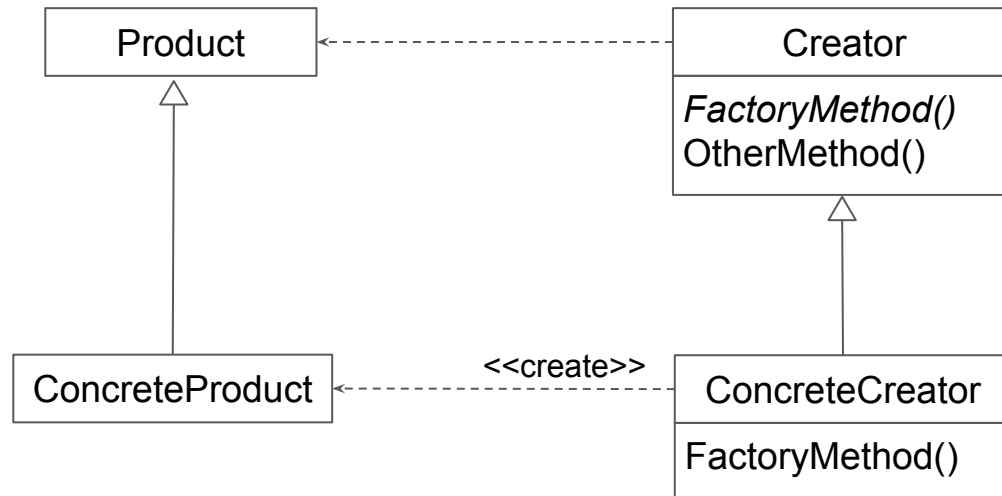


# Phương thức xưởng

## Factory Method

Định nghĩa 1 giao diện để tạo 1 đối tượng, nhưng để các lớp dưới quyết định tạo đối tượng thuộc lớp nào. Phương thức xưởng cho phép 1 lớp dời việc khởi tạo tới các lớp dưới.

# Phương thức xướng



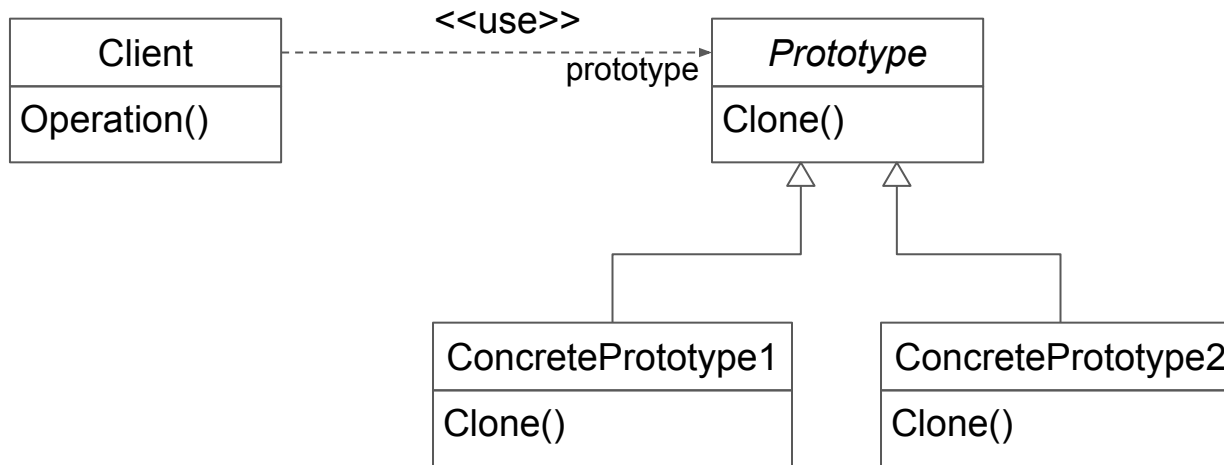
# Nguyên mẫu

## Prototype

Sử dụng 1 đối tượng mẫu để xác định loại đối tượng sẽ tạo, và tạo các đối tượng mới bằng cách sao chép đối tượng đó.



# Nguyên mẫu

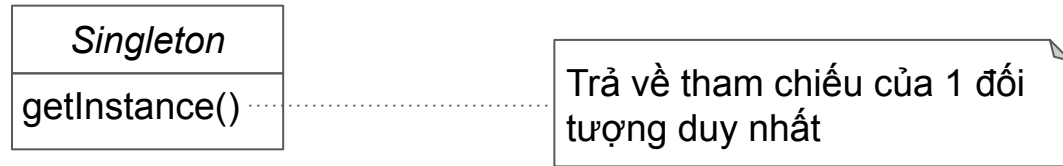


# Độc bản

## Singleton

Đảm bảo 1 lớp chỉ có đúng 1 đối tượng và cung cấp 1 điểm truy cập toàn cục cho nó.

# Độc bản



# Nội dung

- Các mẫu tạo
- Các mẫu cấu trúc
- Các mẫu hành vi

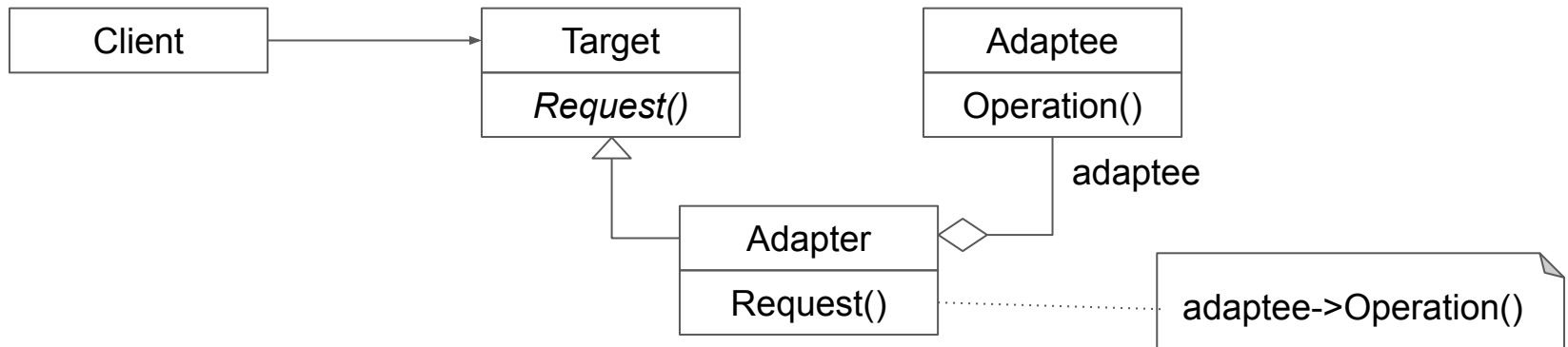
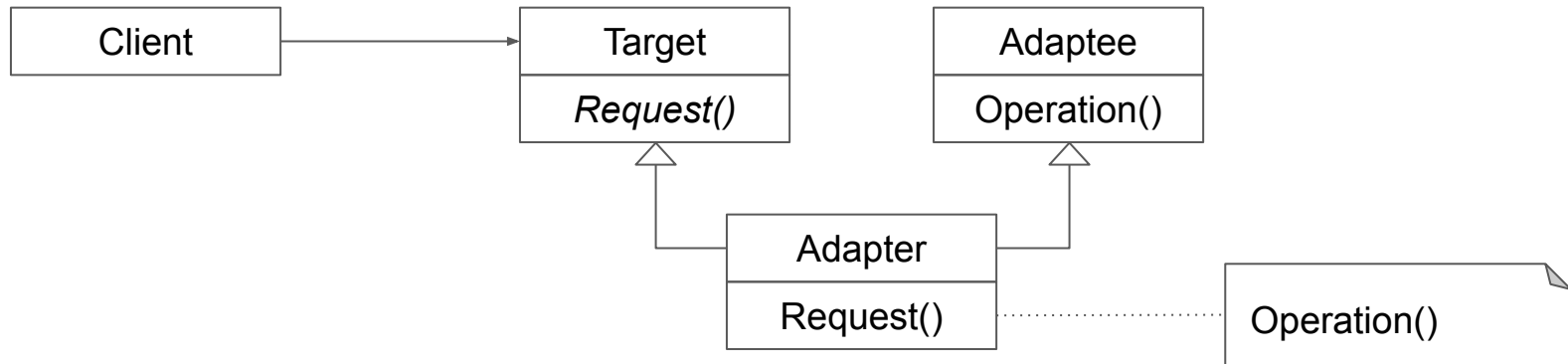


# Cục chuyển

## Adapter

Biến đổi giao diện của 1 lớp thành giao diện được yêu cầu bởi phía khách. Cục chuyển biến 1 lớp với giao diện không tương thích thành 1 lớp với giao diện tương thích mà không thay đổi lớp đó.

# Cục chuyển

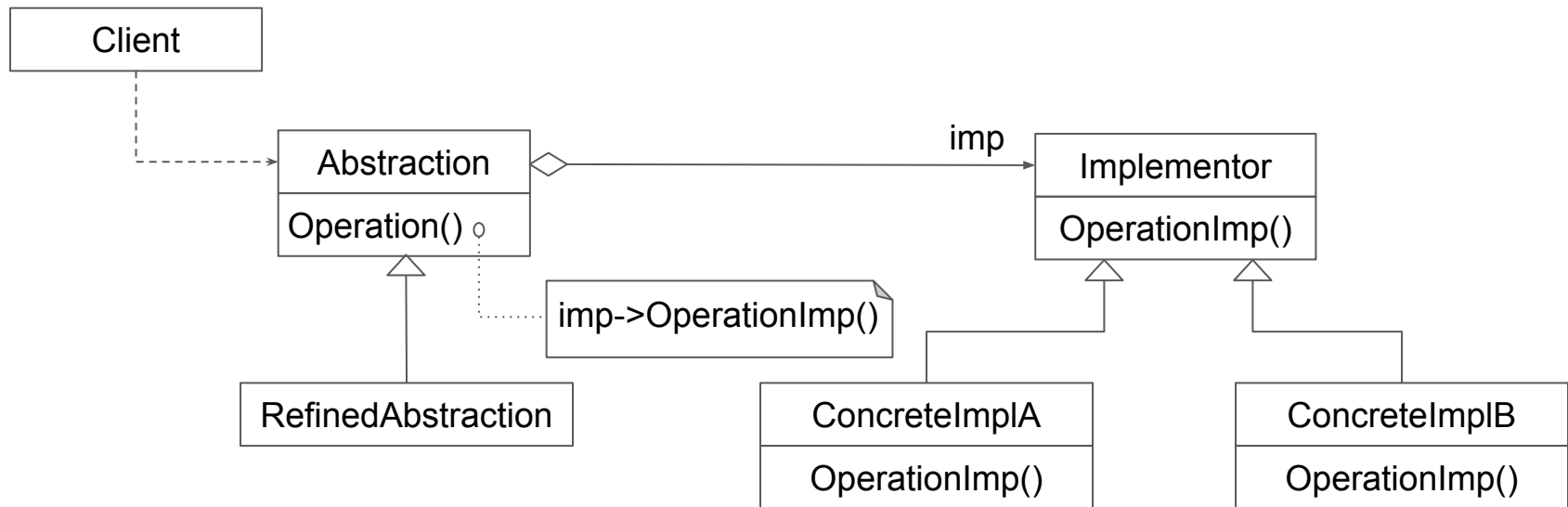


# Cầu nối

## Bridge

Tách biệt giao diện và biểu diễn để cả 2 có thể thay đổi độc lập.

# Cầu nối



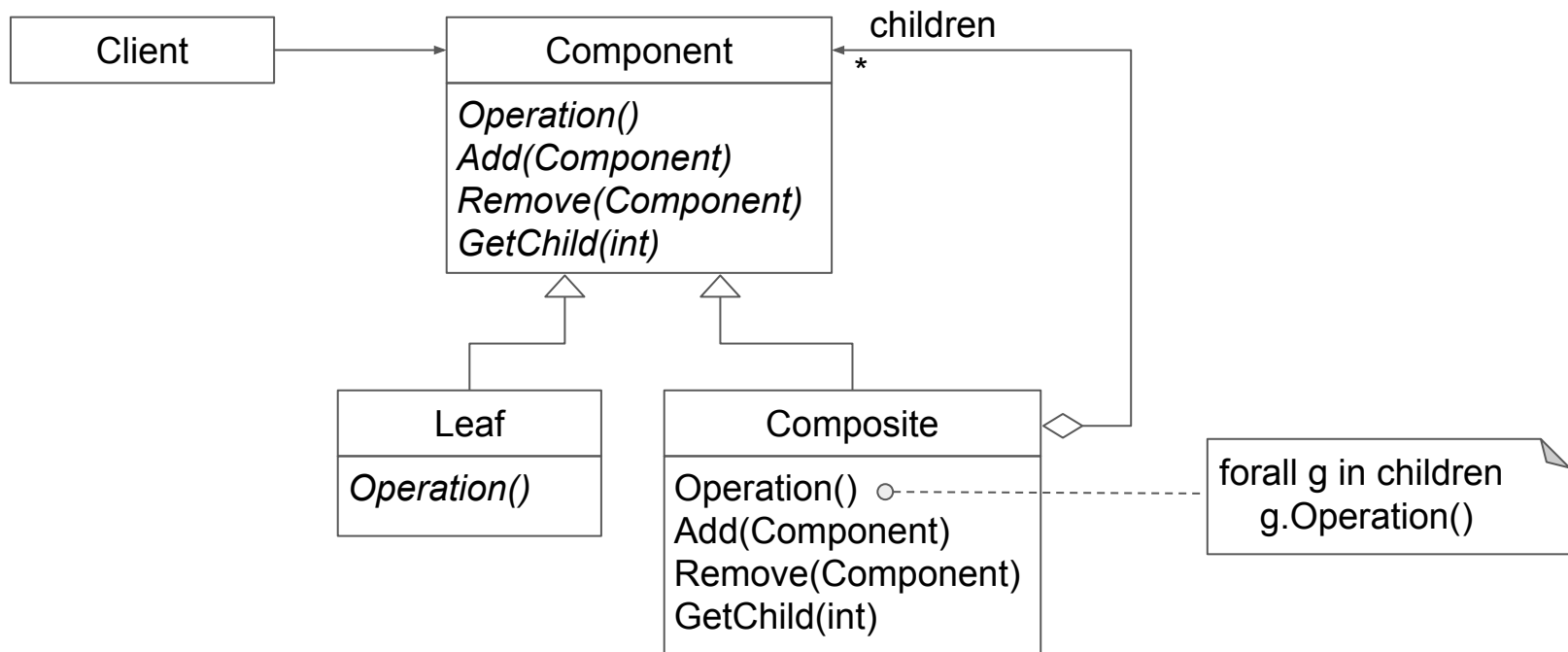


# Hợp chất

## Composite

Tổng hợp các đối tượng thành cây để biểu diễn cây bộ phận-tổng thể. Cơ chế hợp chất cho phép phía khách xử lý các đối tượng riêng lẻ và tổ hợp của nhiều đối tượng theo cùng 1 cách.

# Hợp chất

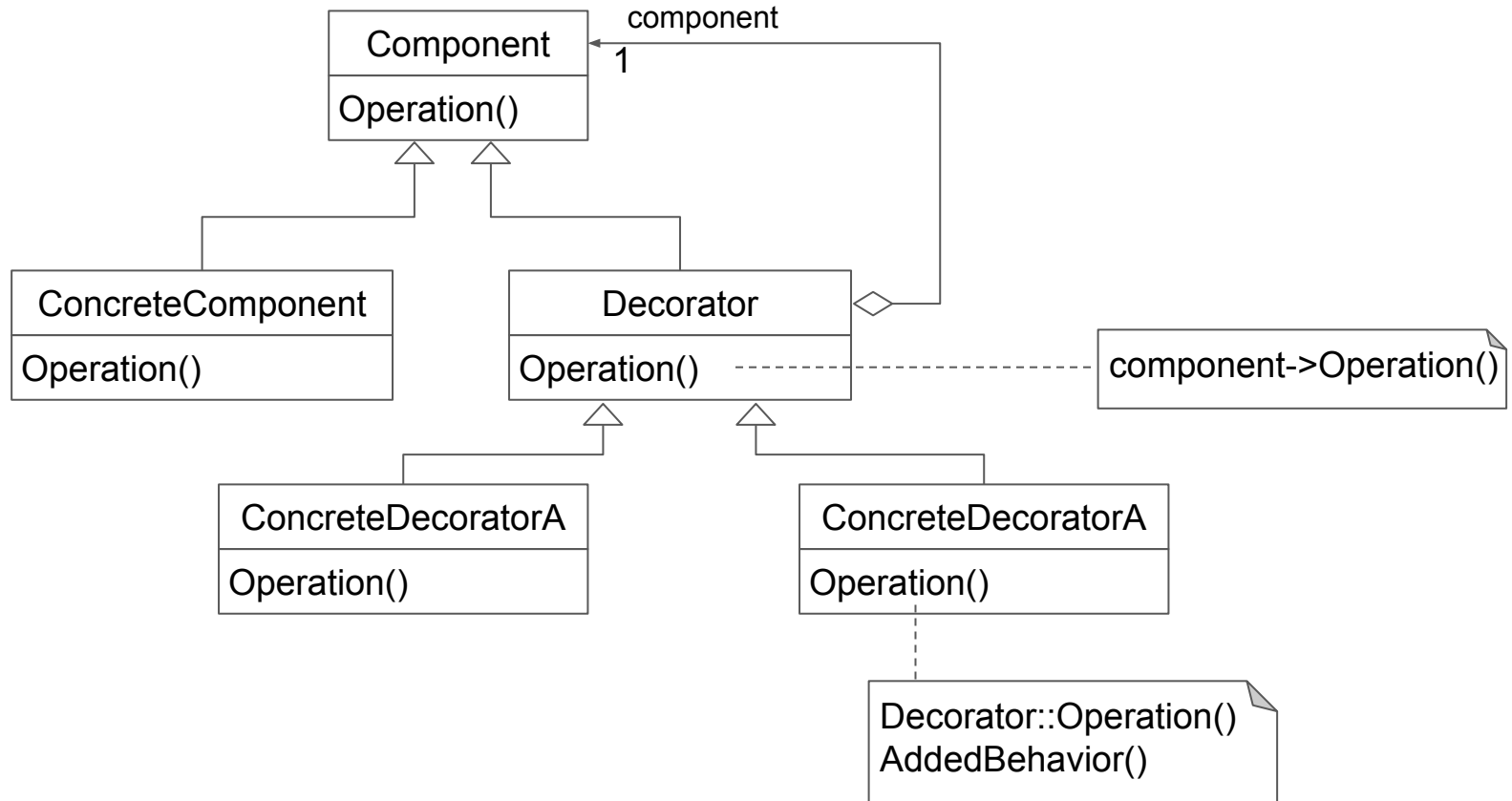


# Tô điểm

## Decorator

Gán động, thêm các trách nhiệm cho đối tượng ở giai đoạn thực thi. Cơ chế tô điểm cung cấp 1 cách khác, linh động hơn so với kế thừa để mở rộng tính năng.

# Tô điểm

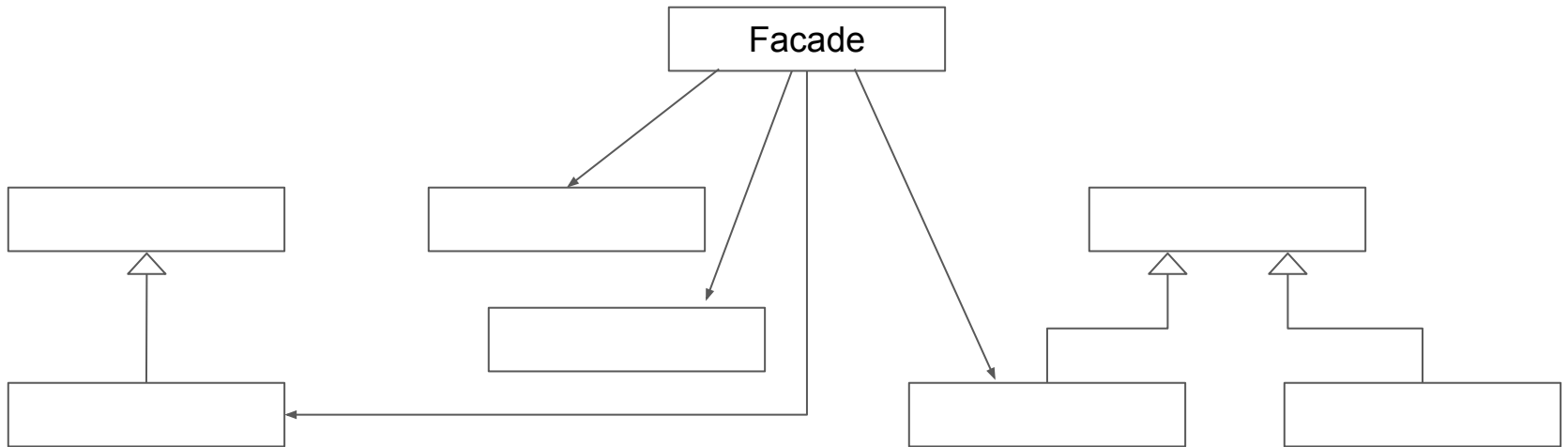


# Mặt tiền

## Facade

Cung cấp 1 giao diện hợp nhất cho 1 tập giao diện trong 1 hệ thống con. Mặt tiền định nghĩa 1 giao diện bậc cao hơn để có thể sử dụng hệ thống con dễ dàng hơn.

# Mặt tiền

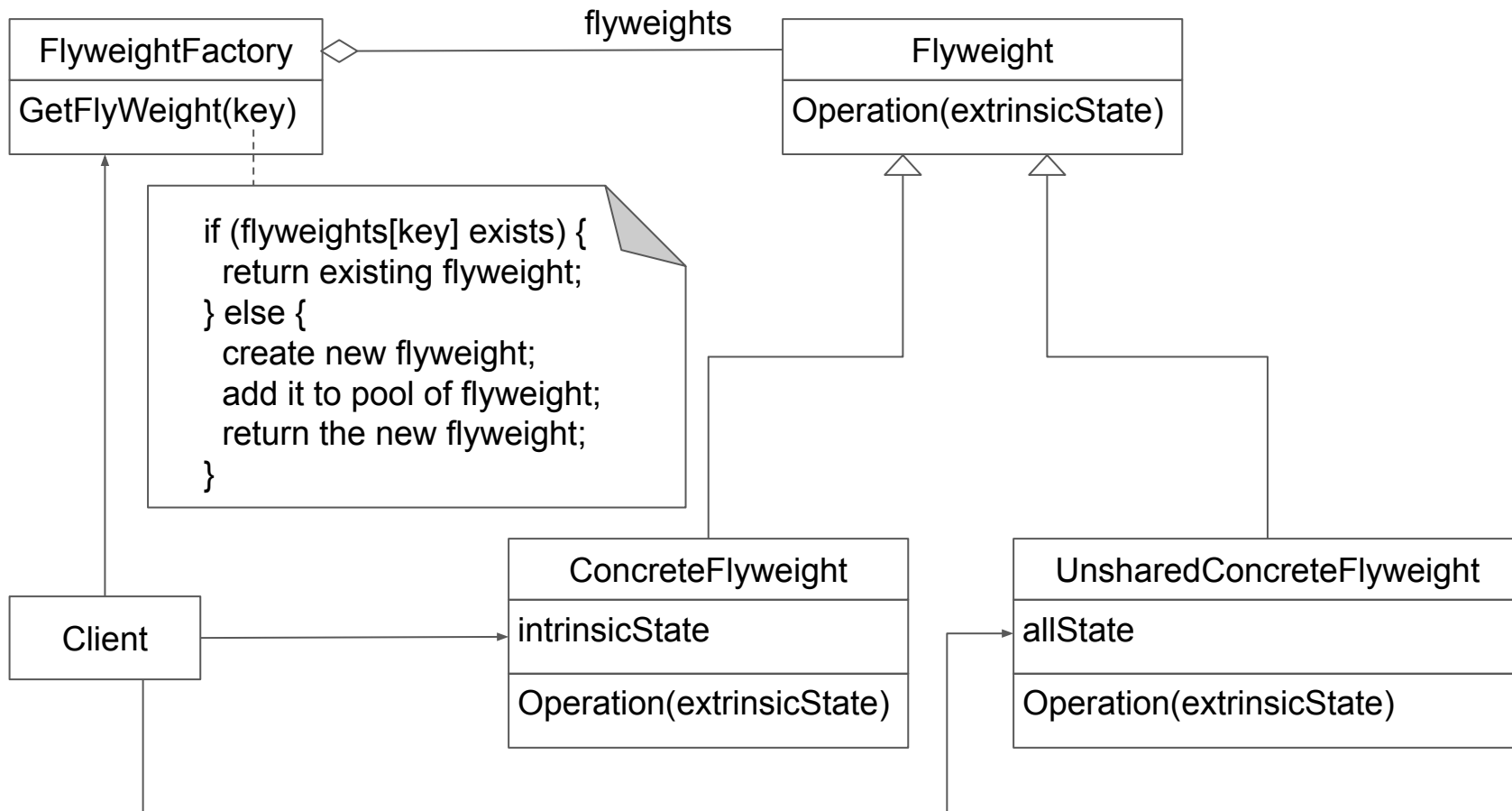


# Trọng lượng bay

## Flyweight

Sử dụng cơ chế chia sẻ để hỗ trợ hiệu quả 1 lượng lớn đối tượng nhỏ.

# Trọng lượng bay



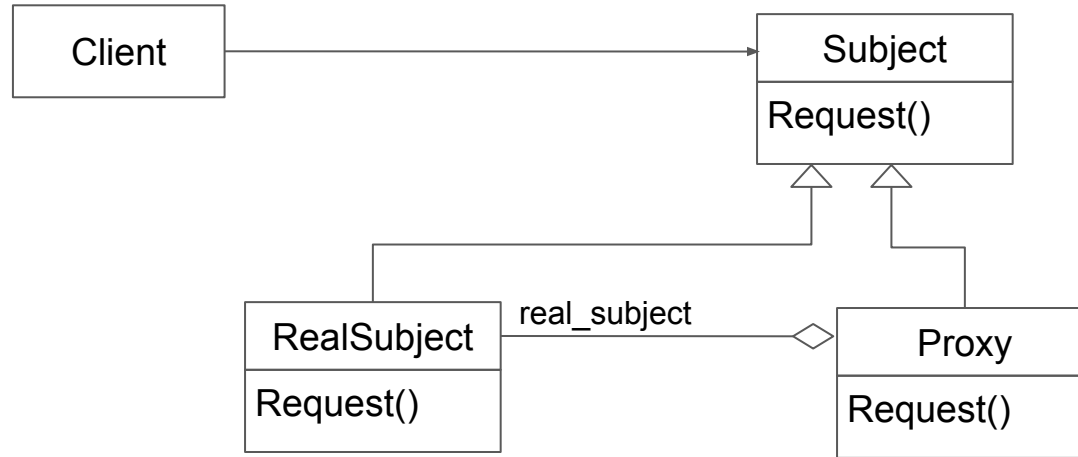


# Đại diện

## Proxy

Cung cấp 1 đối tượng đại diện của 1 đối tượng khác để kiểm soát truy cập tới đối tượng đó.

# Đại diện



# Nội dung

- Các mẫu tạo
- Các mẫu cấu trúc
- Các mẫu hành vi

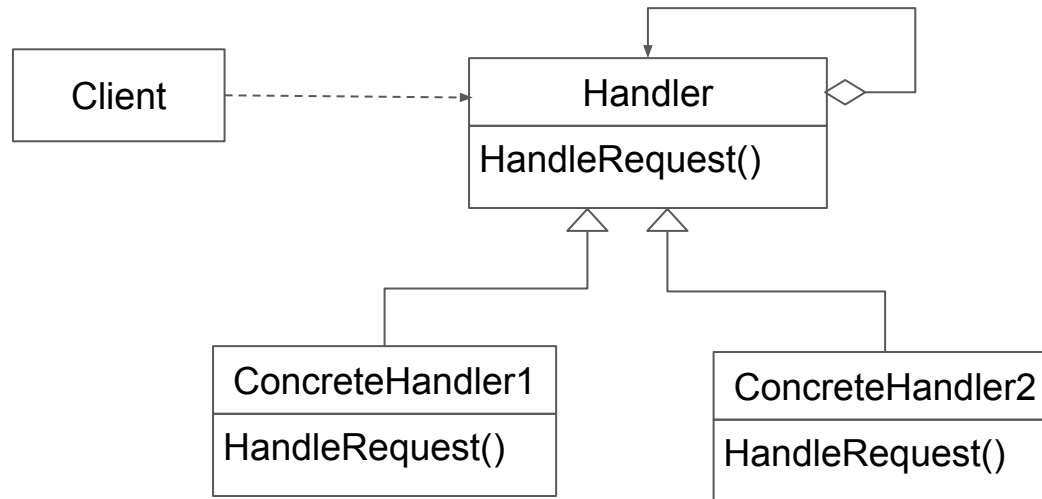


# Chuỗi trách nhiệm

## Chain of Responsibility

Giảm sự gắn kết của đối tượng gửi yêu cầu với đối tượng tiếp nhận nó bằng cách cho nhiều đối tượng cơ hội xử lý yêu cầu. Xâu chuỗi các đối tượng tiếp nhận và truyền yêu cầu trên chuỗi cho tới khi có đối tượng xử lý nó.

# Chuỗi trách nhiệm

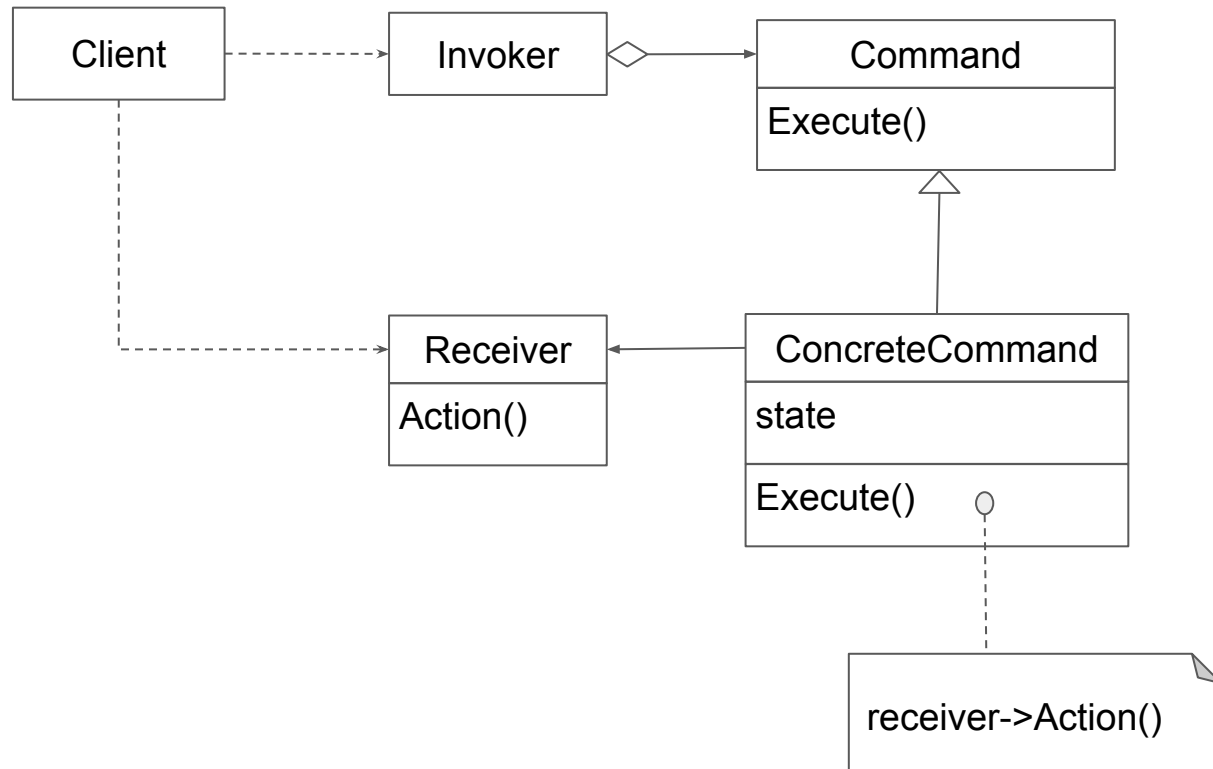


# Lệnh

## Command

Đóng gói 1 yêu cầu như 1 đối tượng, qua đó có thể tùy chỉnh phía khách với nhiều yêu cầu khác nhau, thiết lập hàng đợi hoặc ghi lịch sử yêu cầu, và hỗ trợ hoàn tác.

# Lệnh



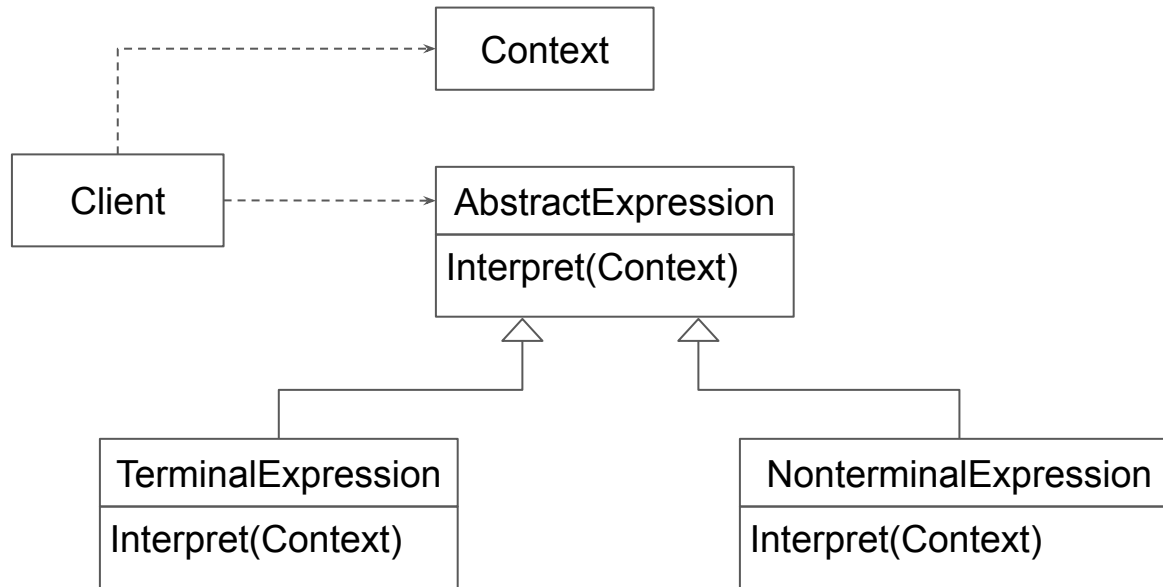
# Diễn dịch

## Interpreter

Đưa ra 1 ngôn ngữ, định nghĩa 1 biểu diễn cho ngữ pháp của nó cùng với 1 trình diễn dịch sử dụng biểu diễn đó để trình diễn các câu trong ngôn ngữ.



# Diễn dịch

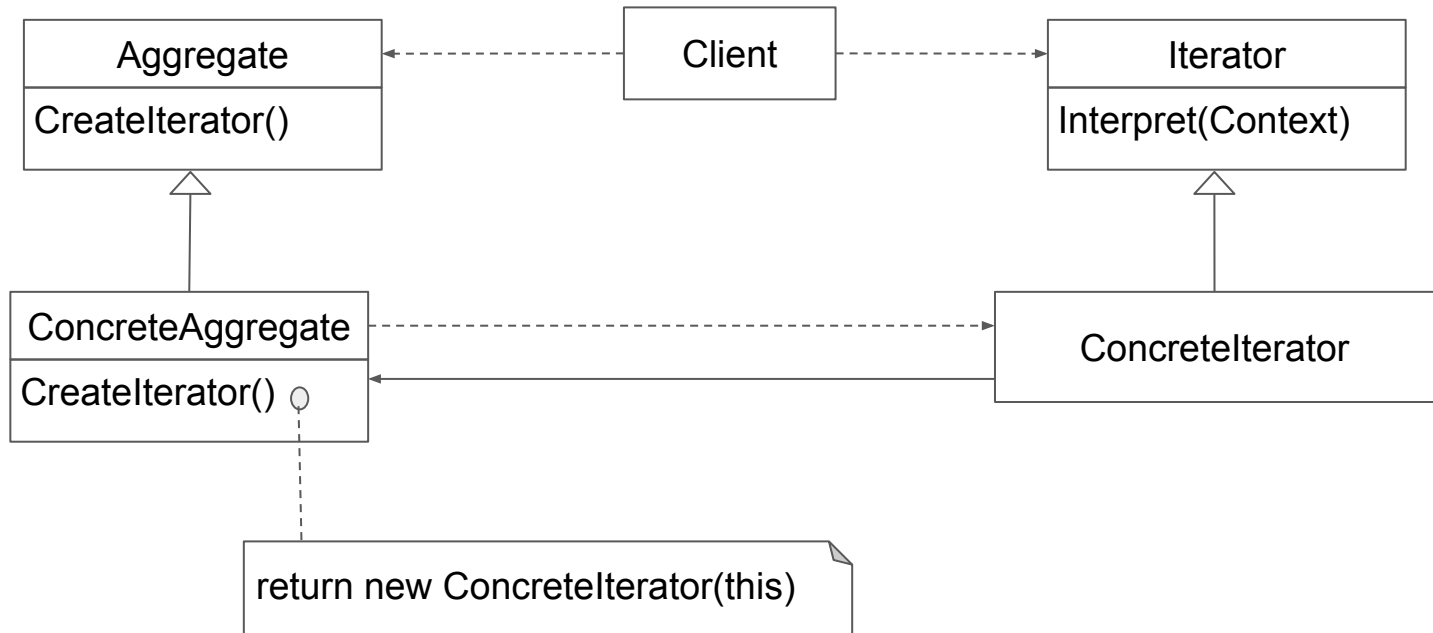


# Biến duyệt

## Iterator

Cung cấp 1 cách truy cập tuần tự các phần tử của 1 đối tượng tổng hợp mà không mở biểu diễn của nó.

# Biến duyệt

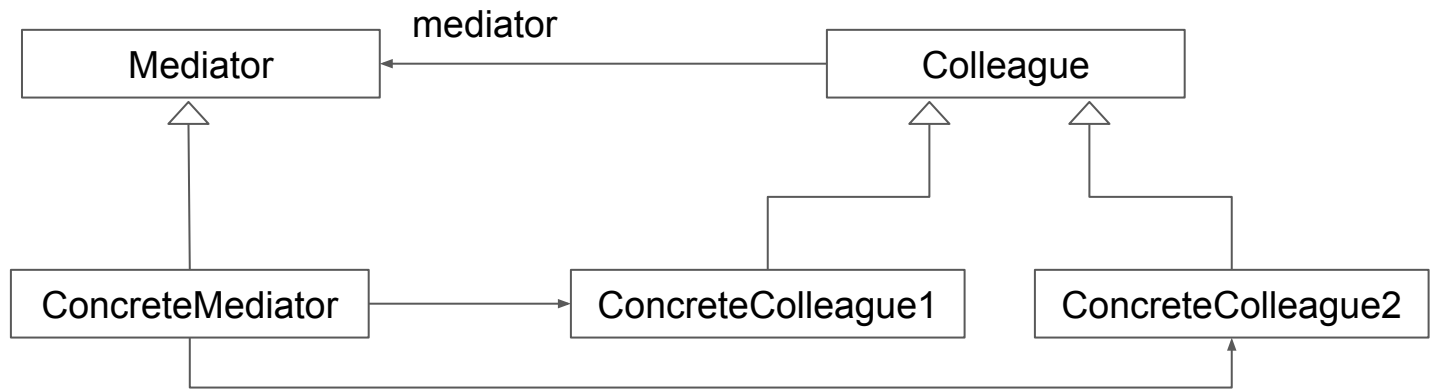


# Điều phối

## Mediator

Định nghĩa 1 đối tượng đóng gói cách 1 tập đối tượng tương tác. Mẫu điều phối hỗ trợ liên kết lỏng bằng cách hạn chế tương tác trực tiếp giữa các đối tượng, và cho phép thay đổi tương tác của chúng 1 cách độc lập.

# Điều phối

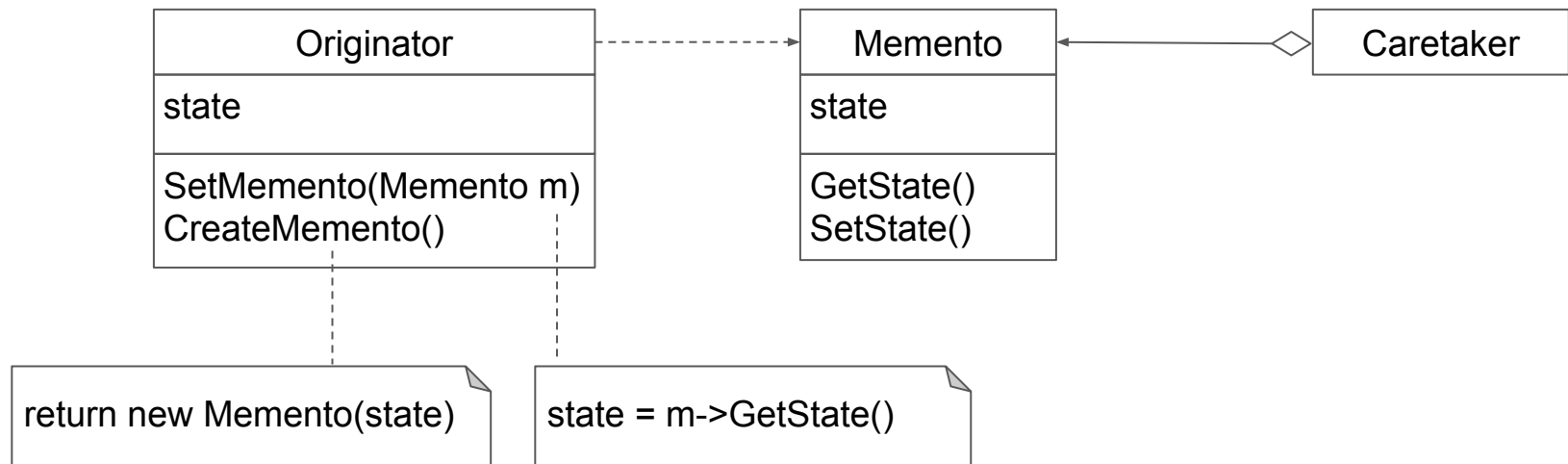


# Bản sao

## Memento

Thâu tóm và mở trạng thái của đối tượng cho bên ngoài sao cho đối tượng có thể khôi phục về trạng thái đó sau đó, trong khi không phá vỡ quy tắc đóng gói.

# Bản sao



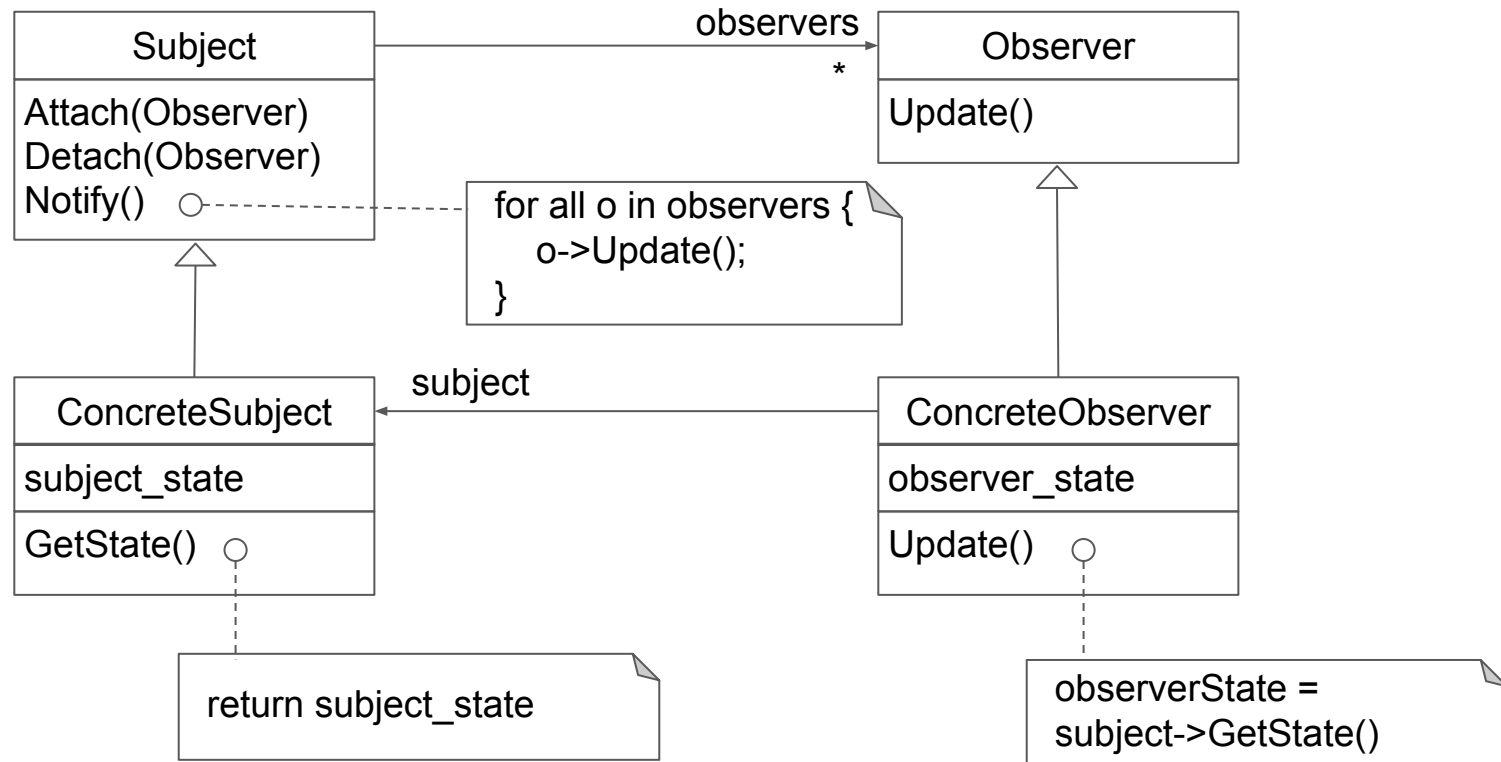
# Quan sát

## Observer

Thiết lập quan hệ một-nhiều giữa các đối tượng sao cho khi một đối tượng thay đổi trạng thái, tất cả các đối tượng phụ thuộc vào nó được thông báo và cập nhật tự động.



# Quan sát

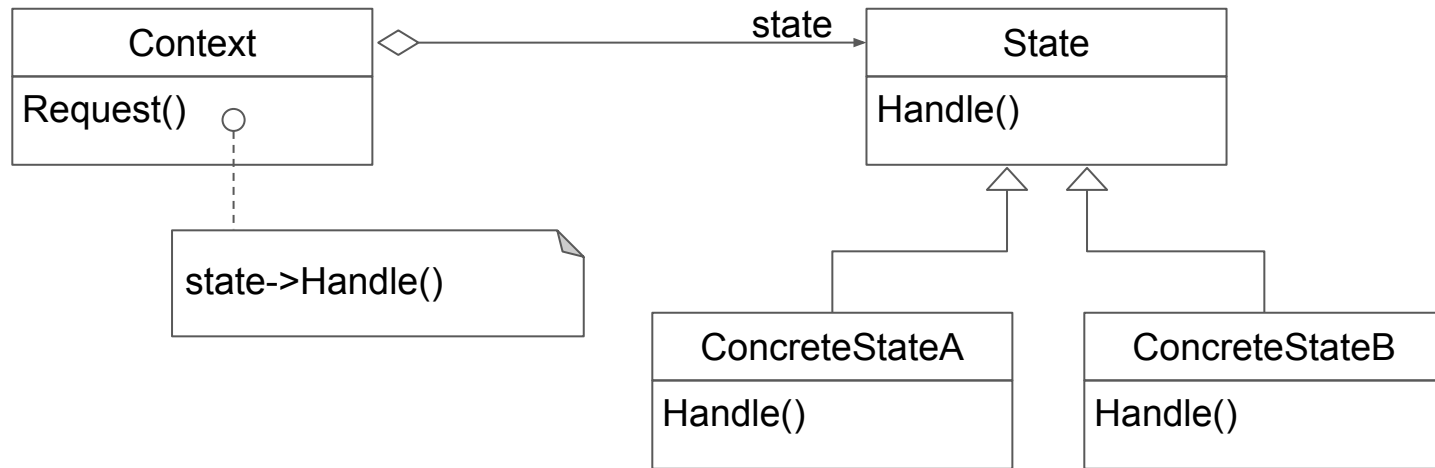


# Trạng thái

## State

Cho phép 1 đối tượng thay đổi hành vi khi thay đổi trạng thái bên trong của nó. Đối tượng sẽ giống như thay đổi lớp của nó.

# Trạng thái

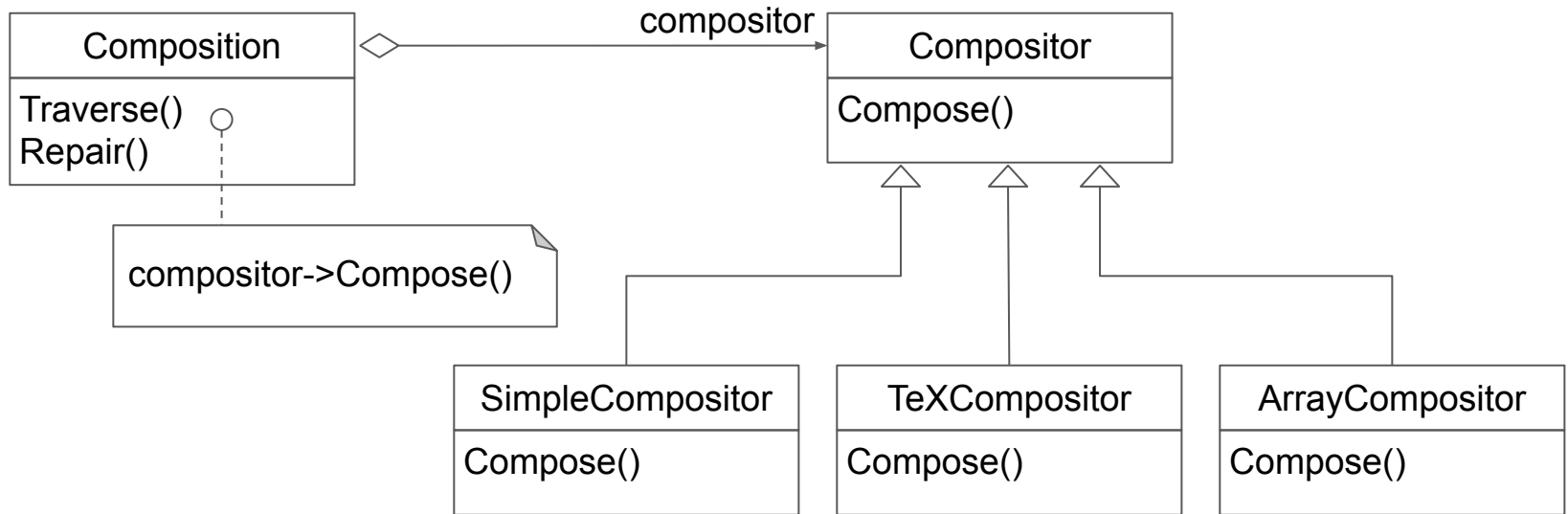


# Chiến thuật

## Strategy

Định nghĩa 1 họ giải thuật, đóng gói mỗi cái, và khiến chúng khả thay. Chiến thuật cho phép giải thuật thay đổi độc lập so với phía khách sử dụng nó.

# Chiến thuật

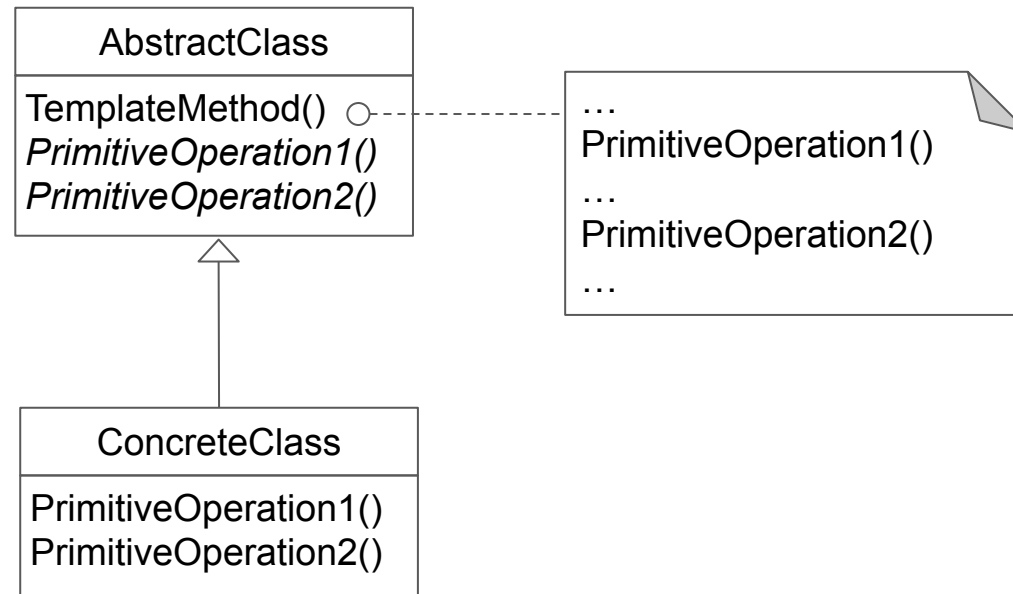


# Phương thức khung

## Template Method

Thiết lập khung của 1 giải thuật trong 1 thao tác, dời một số bước tới các lớp con. Phương thức mẫu cho phép các lớp con định nghĩa lại các bước cụ thể của 1 giải thuật mà không thay đổi cấu trúc giải thuật.

# Phương thức khung



# Khách thăm

## Visitor

Biểu diễn một thao tác cần được thực hiện trên các thành phần của 1 đối tượng cấu trúc. Khách thăm cho phép định nghĩa 1 thao tác mới mà không thay đổi các lớp của các phần tử mà nó thao tác.



# Khách thăm

