

# Phân tích thiết kế Hệ thống

Giảng viên: Nguyễn Bá Ngọc

Chủ đề mở: Mẫu thiết kế

Hà Nội-2021

Chủ đề nâng cao

# Các khái niệm

- Một mẫu thiết kế
  - Mô tả một giải pháp tiêu chuẩn cho
  - Một vấn đề thiết kế tiêu chuẩn trong
  - Một ngữ cảnh cụ thể
- Mục đích: Tái sử dụng các thiết kế
  - Mẫu thiết kế không "mới"
- Mẫu thiết kế được quan tâm trong nhiều lĩnh vực kỹ nghệ
  - Kiến trúc
  - Xây dựng
  - Cơ khí

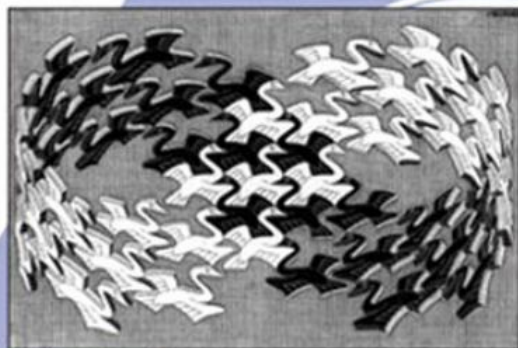
# Cấu trúc tài liệu mẫu thiết kế (biểu mẫu GOF)

- Tên
- Mục đích
- Ứng dụng
- Giải pháp
  - Cấu trúc
  - Các thành phần
  - Tương tác
  - Ưu & nhược điểm
  - Triển khai
  - Ví dụ mã nguồn
- Các trường hợp được biết
- Các mẫu thiết kế liên quan

# Design Patterns

## Elements of Reusable Object-Oriented Software

Erich Gamma  
Richard Helm  
Ralph Johnson  
John Vlissides



Cover art © 1994 M.C. Escher / Cordon Art - Baarn - Holland. All rights reserved.

Foreword by Grady Booch

ADDISON-WESLEY PROFESSIONAL COMPUTING SERIES



# Mục đích của mẫu thiết kế

- Hỗ trợ giao tiếp trong nhóm
  - Giữa khách hàng và lập trình viên
  - Người thiết kế - triển khai - kiểm thử
  - Hiểu các ý tưởng thiết kế hay
- Các mẫu thiết kế có vai trò như một bộ từ vựng chuẩn trong thiết kế phần mềm
  - Sử dụng trong thảo luận và phát triển giải pháp
  - Tránh việc tái phát minh lại giải pháp đã có
- Mẫu thiết kế mô tả ý tưởng thiết kế khái quát
  - Giúp tạo lập cấu trúc mã nguồn, nâng cao chất lượng mã nguồn

Một số mẫu thiết kế đơn giản để thay thế linh hoạt

# Khả năng thay thế linh hoạt

- Khả năng điều chỉnh thay thế
  - Biểu diễn các thành phần chung và các thành phần riêng (khác nhau)
  - Chúng ta cố định phần chung và tham số hóa phần khác biệt
- Hiểu các mẫu thiết kế Phương thức khung và lớp khung



# Vấn đề



Ảnh sưu tầm

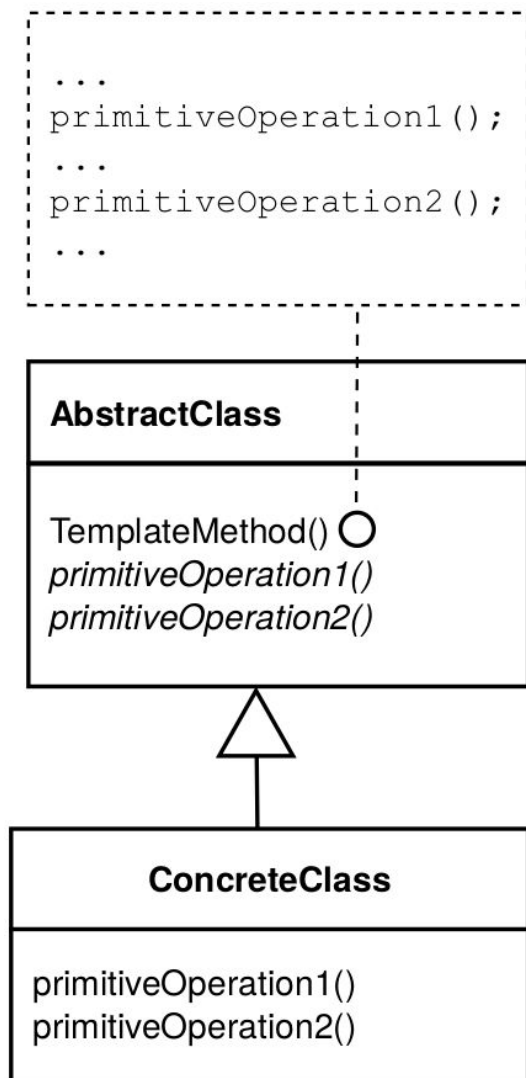


Phần ghép nối



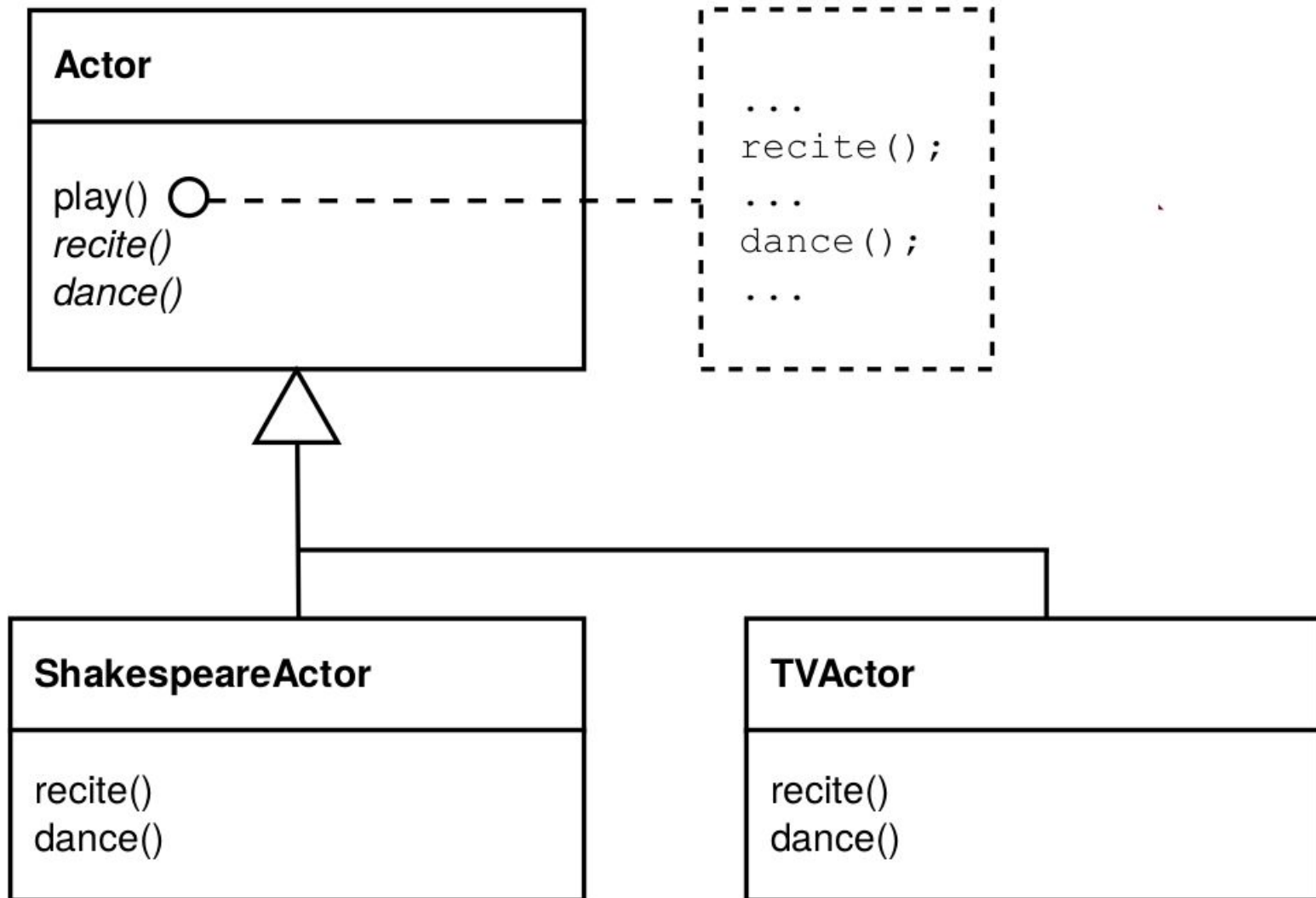
Khung

# Mẫu thiết kế Phương thức mẫu (TemplateMethod)



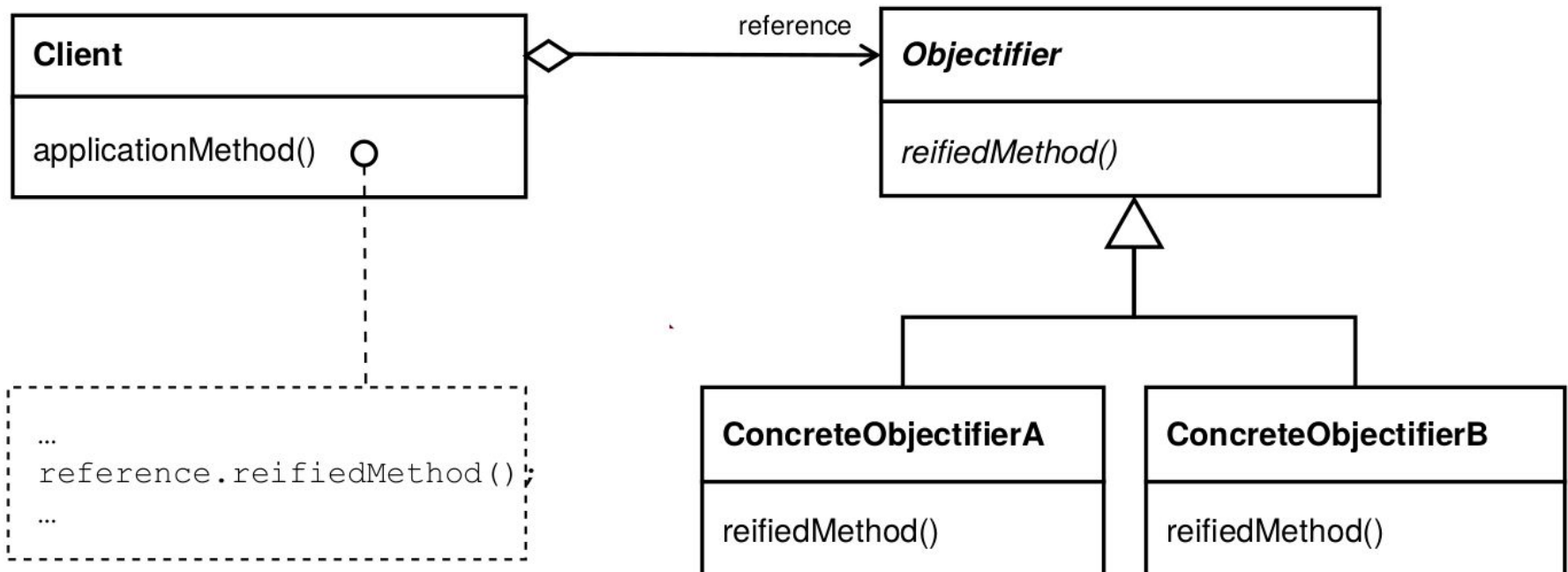
- Phương thức mẫu thiết lập phần khung của một giải thuật
- Chứa các phương thức ghép nối (hook methods), được hoàn thiện trong các lớp con
  - Yêu cầu kế thừa
- Cả khung và phần ghép nối là các phương thức khác nhau trong một lớp

# Ví dụ: Phương thức khung



# Mẫu đối tượng hàm

- Đối tượng hàm/Đóng gói hàm thành đối tượng - Objectifier
- Triển khai với các đối tượng
  - Kết hợp với một lớp trừu tượng và phương thức trừu tượng
  - Mã khách gọi các đối tượng đa hình

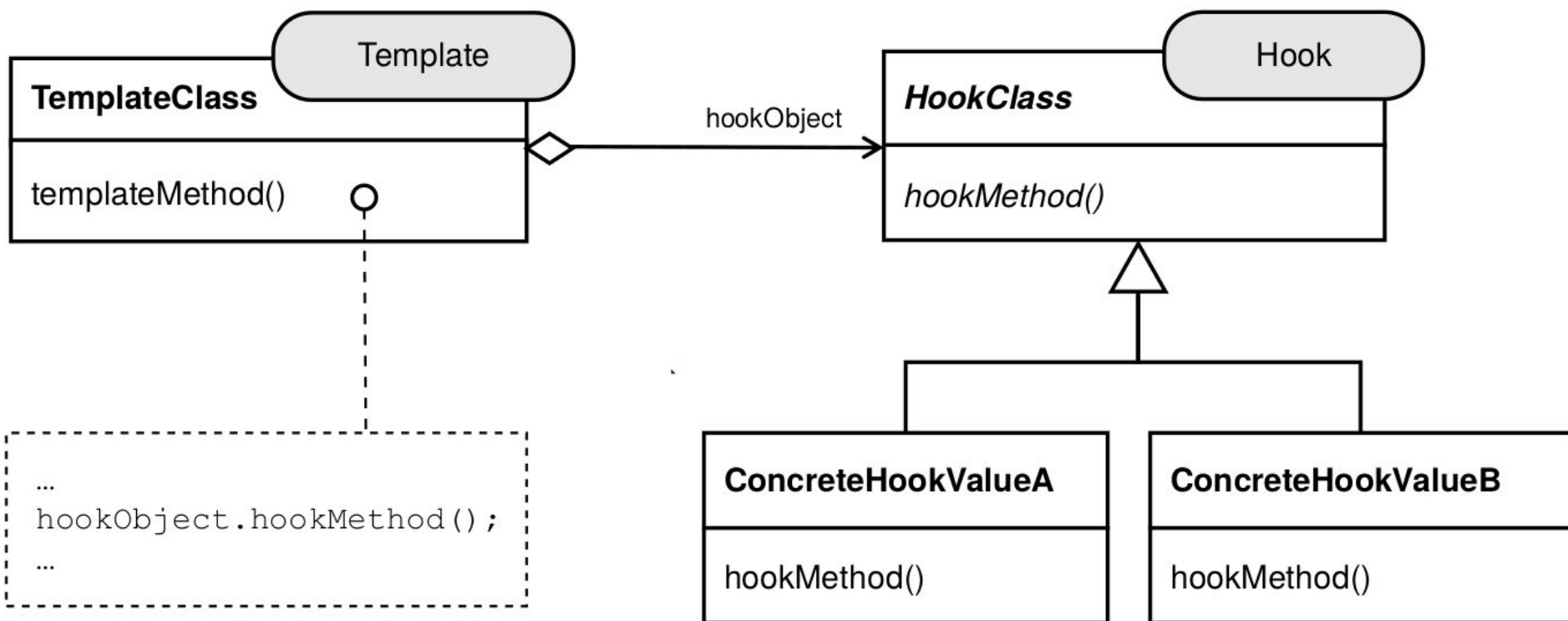


# Khung và phần ghép nối ở mức lớp

- Với đối tượng hàm chúng ta có thể triển khai các lớp khung và các lớp ghép nối
  - Các vai trò mới cho các lớp
    - Vai trò khung
    - Vai trò ghép nối
- Các mẫu thu được
  - Lớp khung
  - Lớp khung khái quát
  - ...

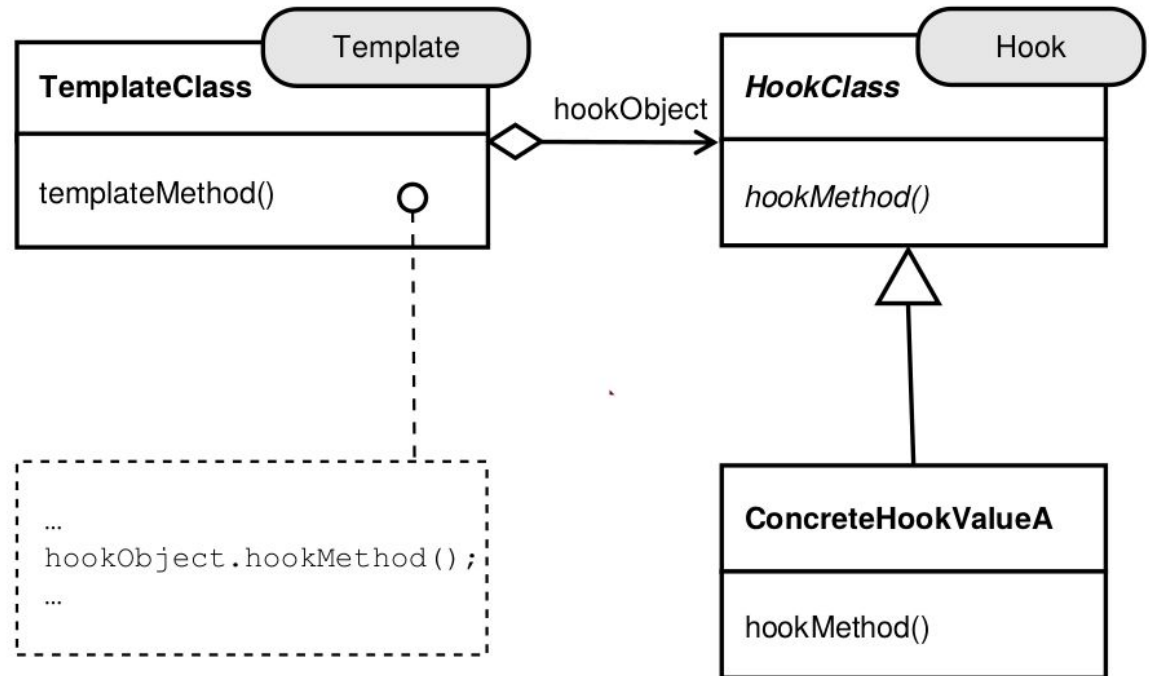
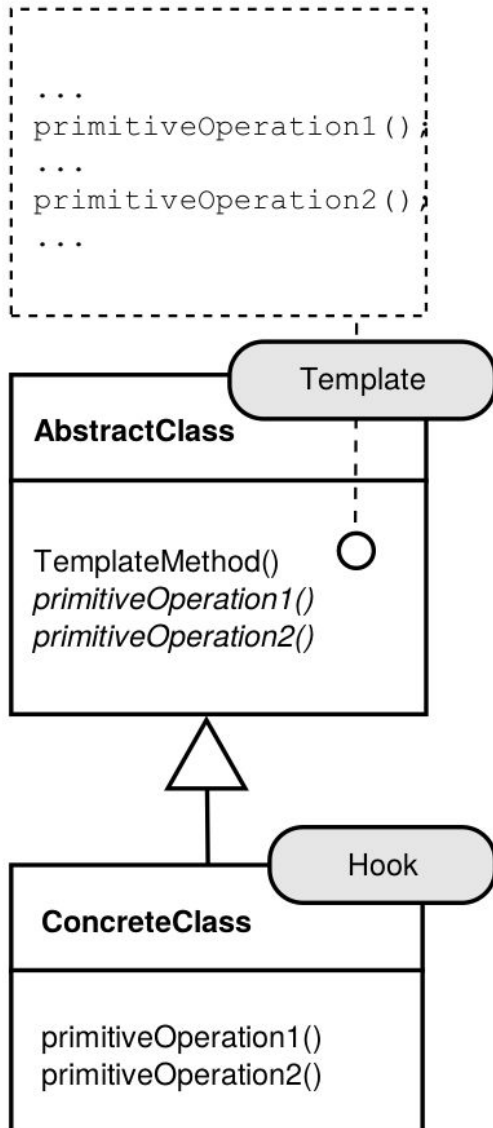
# Lớp khung (TemplateClass)

- Được kết hợp từ Phương thức khung + Hàm đối tượng
  - Các phương thức khung và phương thức ghép nối thuộc các lớp khác nhau





# Phương thức khung vs. Lớp khung



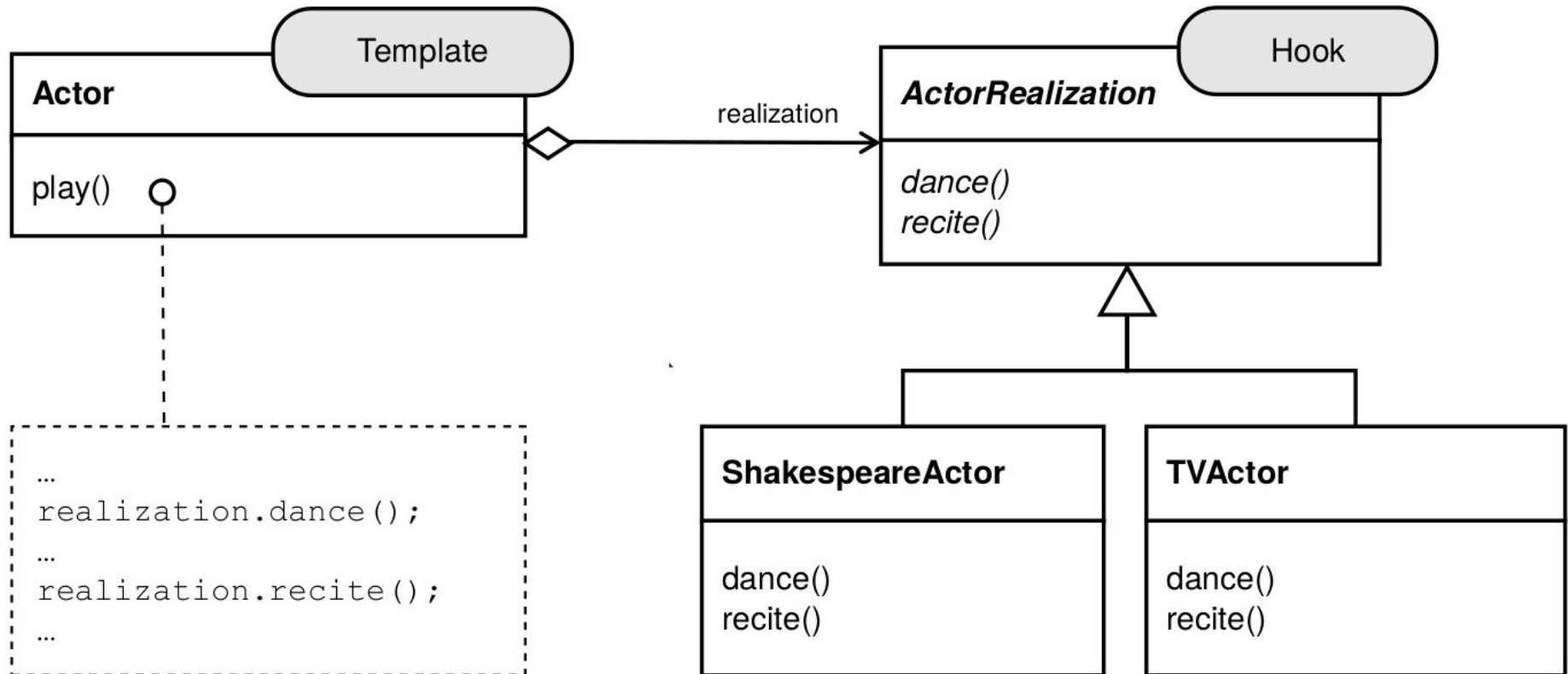
*Có thể thay đổi phần ghép nối ở thời gian thực thi chương trình.*



# Lớp khung

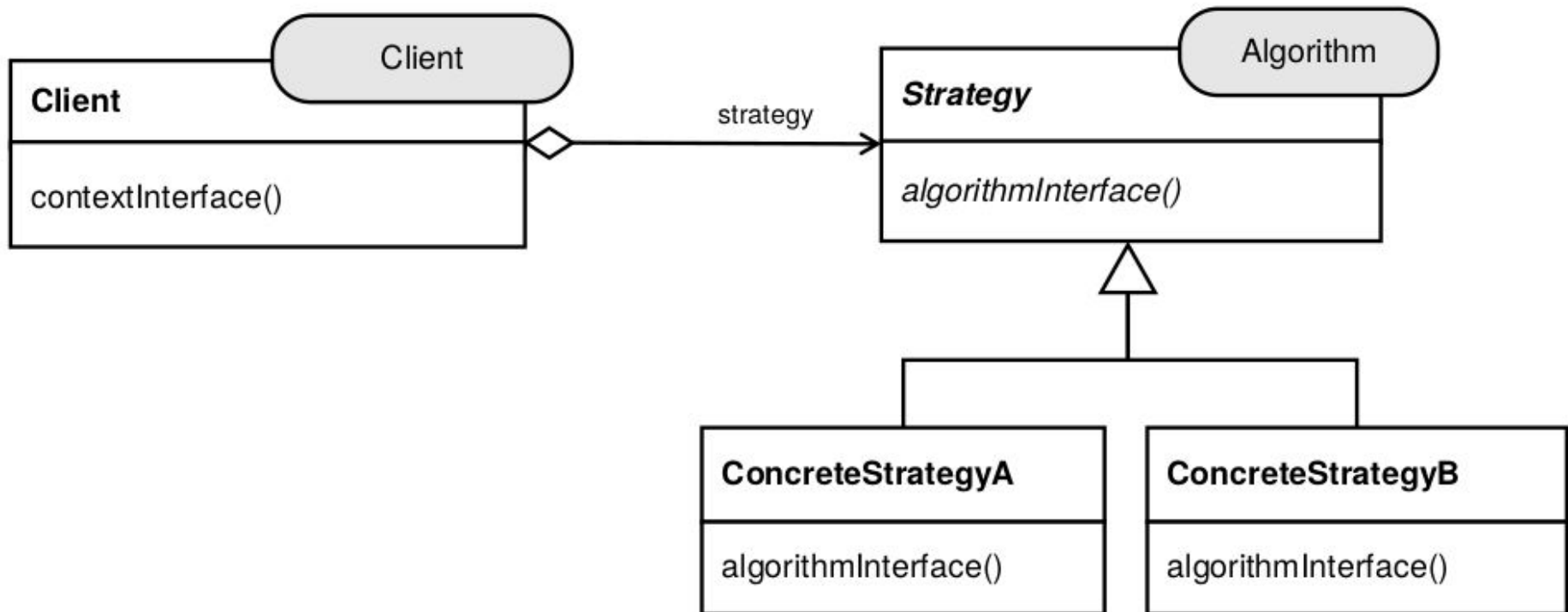
- Là cơ sở cho nhiều mẫu thiết kế khác:
  - Bridge
  - Builder
  - Command
  - Iterator
  - Observer
  - Prototype
  - State
  - Strategy
  - Visitor

# Ví dụ: Lớp khung



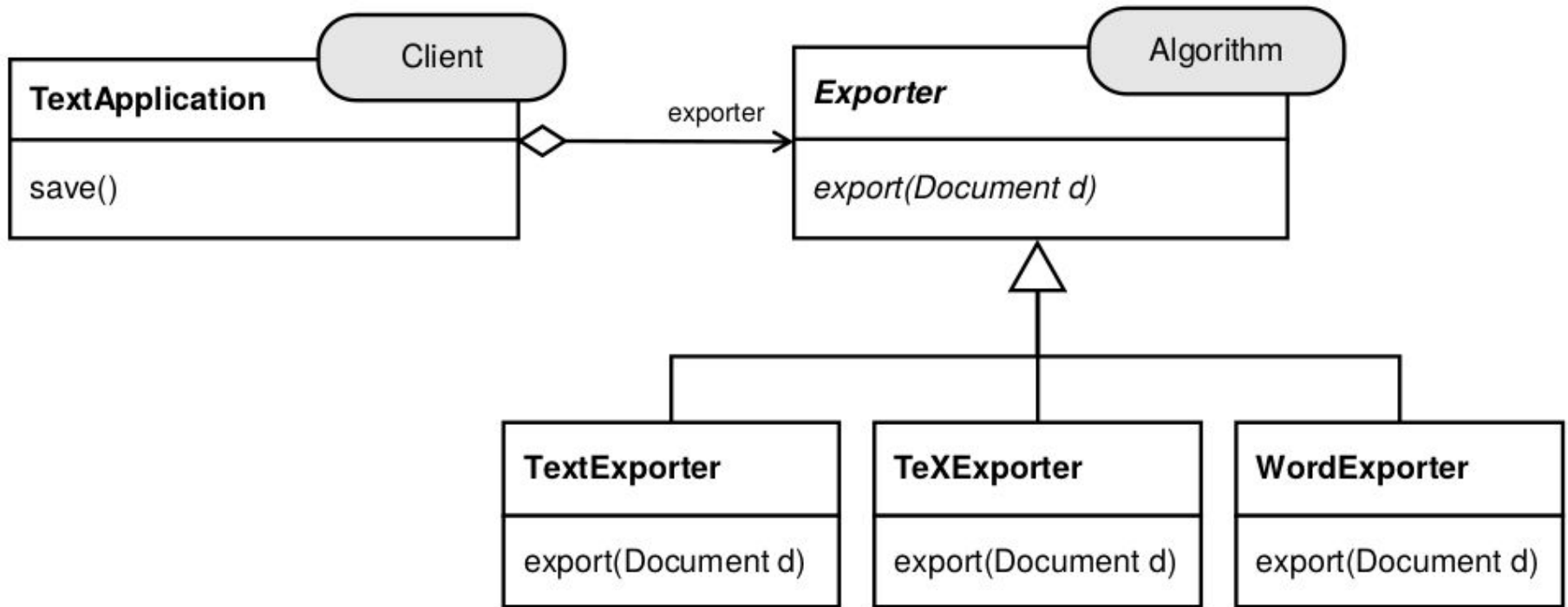
# Mẫu Chiến Thuật (GOF) (Strategy)

- Tương tự Lớp Khung



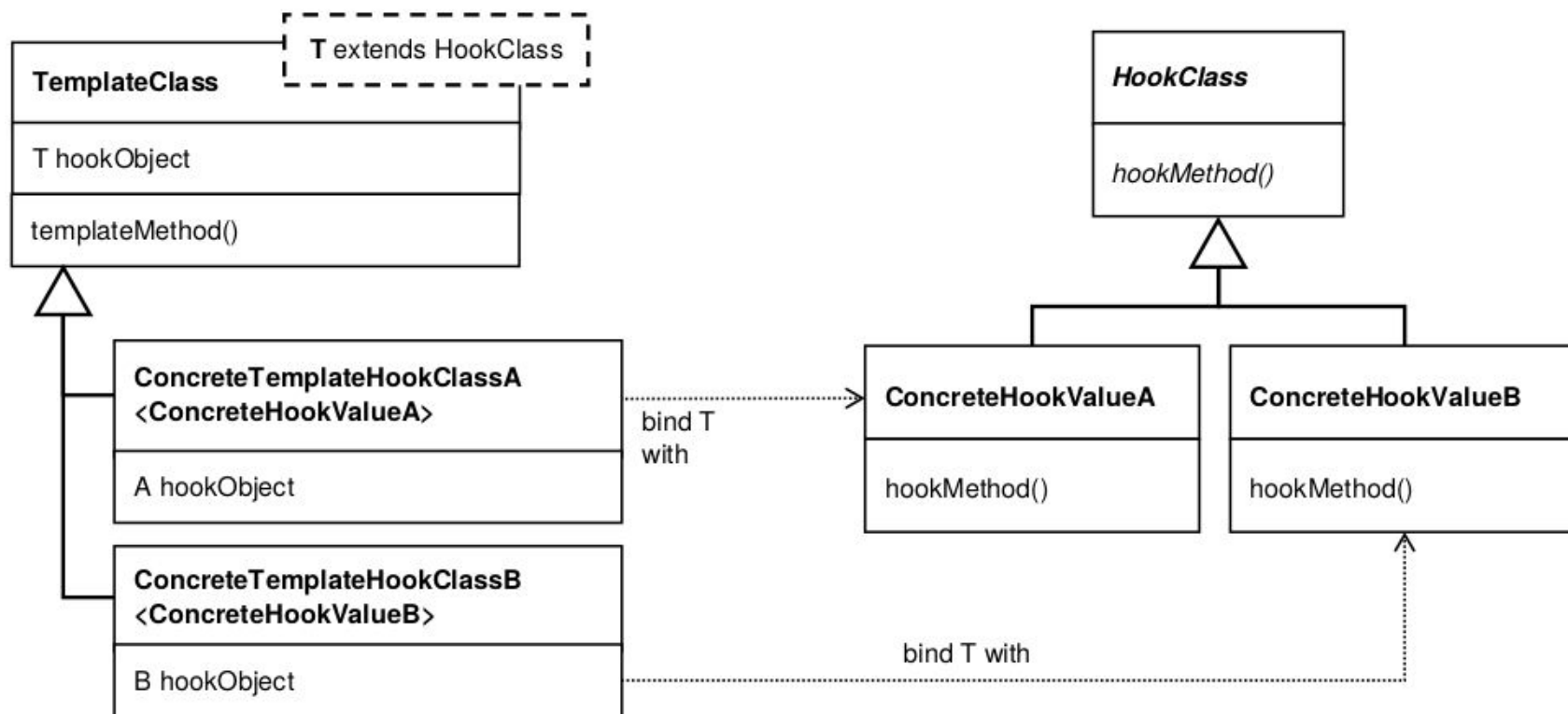
Tuy nhiên: Khác nhau về động cơ

# Ví dụ: Mẫu thiết kế Chiến Thuật

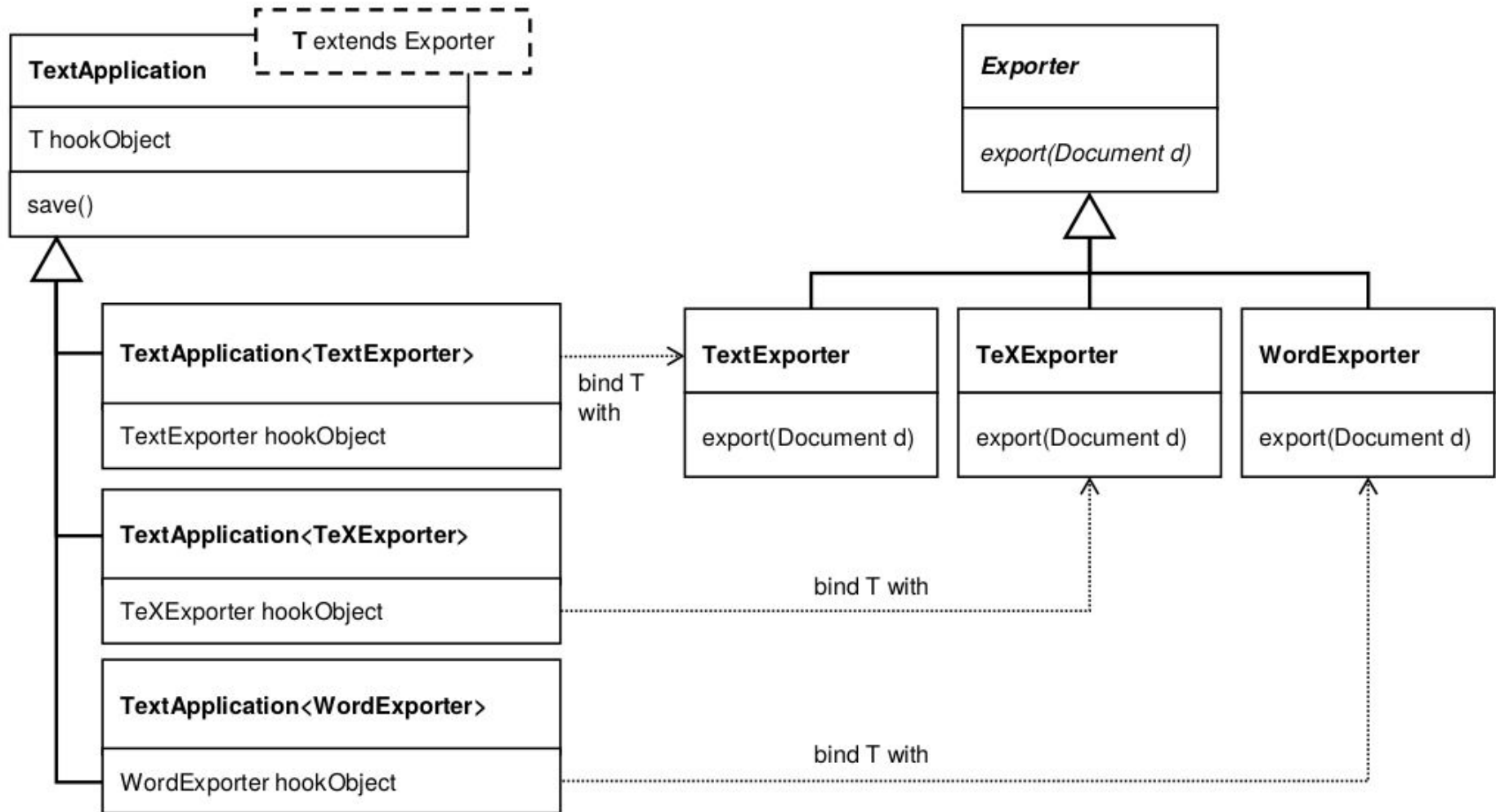


# Lớp khung khái quát

- Có thể triển khai các lớp khung khái quát bằng các ngôn ngữ có hỗ trợ lớp khái quát (ví dụ C++, Java)
- An toàn kiểu, giảm thiểu các gọi hàm



# Ví dụ: Lớp xuất văn bản khái quát



# Câu hỏi, Bài tập & Thảo luận