

---

# Mẫu thiết kế

---

# Nội dung chính

---

- Tổng quan mẫu thiết kế
- Một số mẫu thiết kế
  - Mẫu thiết khởi tạo
  - Mẫu thiết kế cấu trúc
  - Mẫu thiết kế hành vi

# Tài liệu tham khảo

---

- Erich Gamma et al., *Design Patterns: Elements of Reusable Object-oriented Software*

# Mẫu thiết kế

---

- Mẫu thiết kế (design pattern) là một giải pháp tổng quát cho một vấn đề thường xảy ra trong quá trình thiết kế phần mềm
  - Lợi ích
    - Thúc đẩy sử dụng các giải pháp đã được chứng minh là đúng
    - Cung cấp các từ vựng chung cho người phát triển
-

# Các loại mẫu thiết kế

---

- Mẫu thiết kế khởi tạo
  - Singleton
  - Abstract Factory
  - Prototype
- Mẫu thiết kế cấu trúc
  - Adapter
  - Façade
  - Proxy
- Mẫu thiết kế hành vi
  - Observer
  - Template

# Mẫu Singleton

---

- Mẫu thiết kế này được dùng trong trường hợp chúng ta muốn chỉ khởi tạo duy nhất một đối tượng của một lớp và cho phép nhiều người dùng đối tượng đó
  - Ví dụ: có một lớp biểu diễn cho kết nối với cơ sở dữ liệu, trong toàn chương trình, chúng ta chỉ cần một kết nối (được sử dụng bởi nhiều phần khác nhau)
-

# Ví dụ

---

```
public class Connection{
    private static Connection instance = null;
    private Connection(){}
    public static Connection getInstance() {
        if (instance == null)
            instance = new Connection();
        return instance;
    }
}
```

---

# Mẫu Abstract Factory

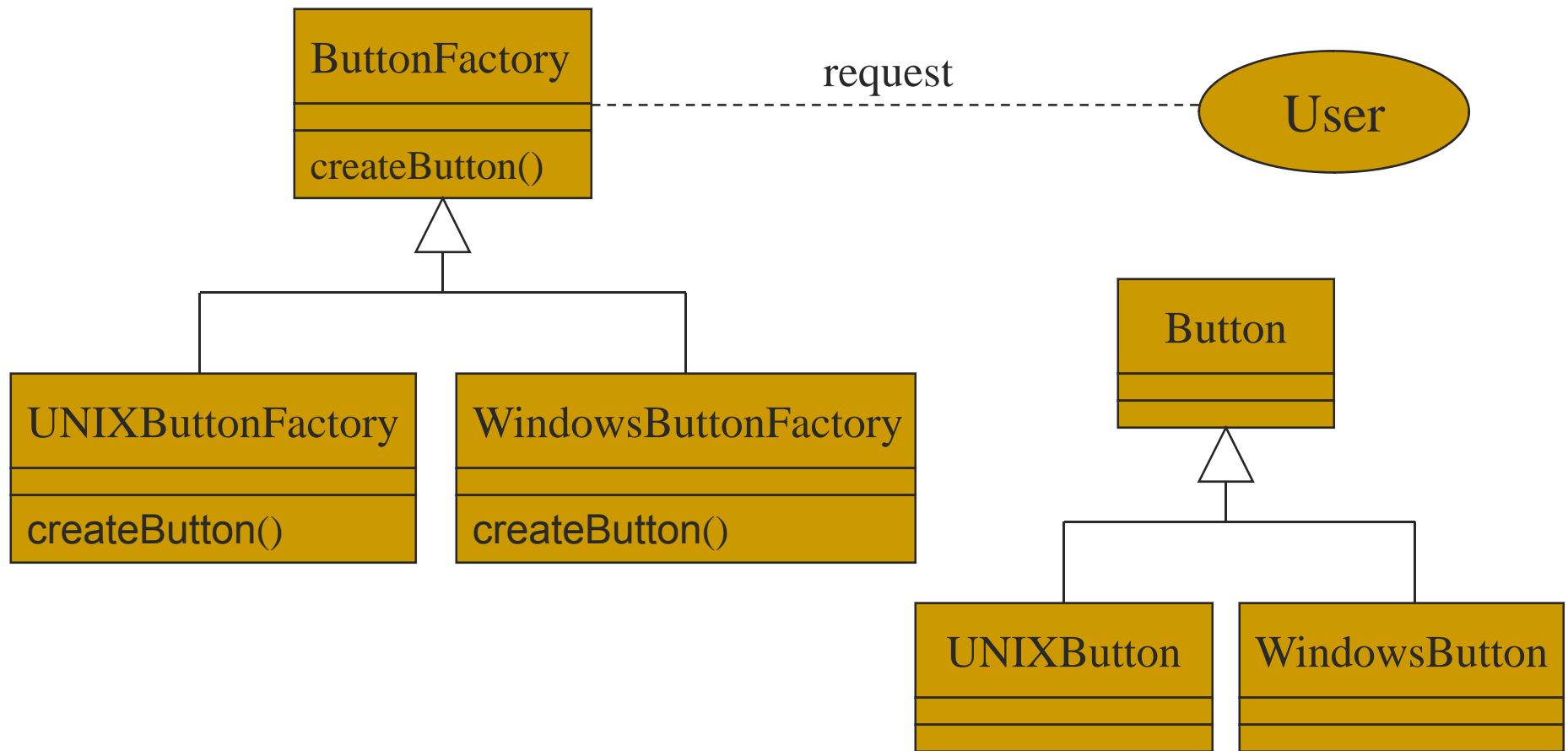
---

- Dùng trong trường hợp chúng ta muốn tạo một tập các đối tượng có liên quan mà không cần phải chỉ định rõ lớp
  - Ví dụ: tạo các cửa sổ theo phong cách Unix hoặc là Windows
-



# Ví dụ

---



# Mẫu Adapter

---

- Làm cho một giao diện có thể hoạt động như một giao diện khác
    - Cụ thể hơn, đối tượng O1 có thể dùng các đối tượng có giao diện I1; O2 có giao diện I2, để O1 dùng O2, chúng ta cần chuyển giao diện của O2 thành I1.
  - Được dùng khi
    - Muốn sử dụng một lớp có sẵn, nhưng giao diện không như mong muốn
    - Tạo mới một lớp có thể hoạt động với các lớp khác (chưa được phát triển)
-

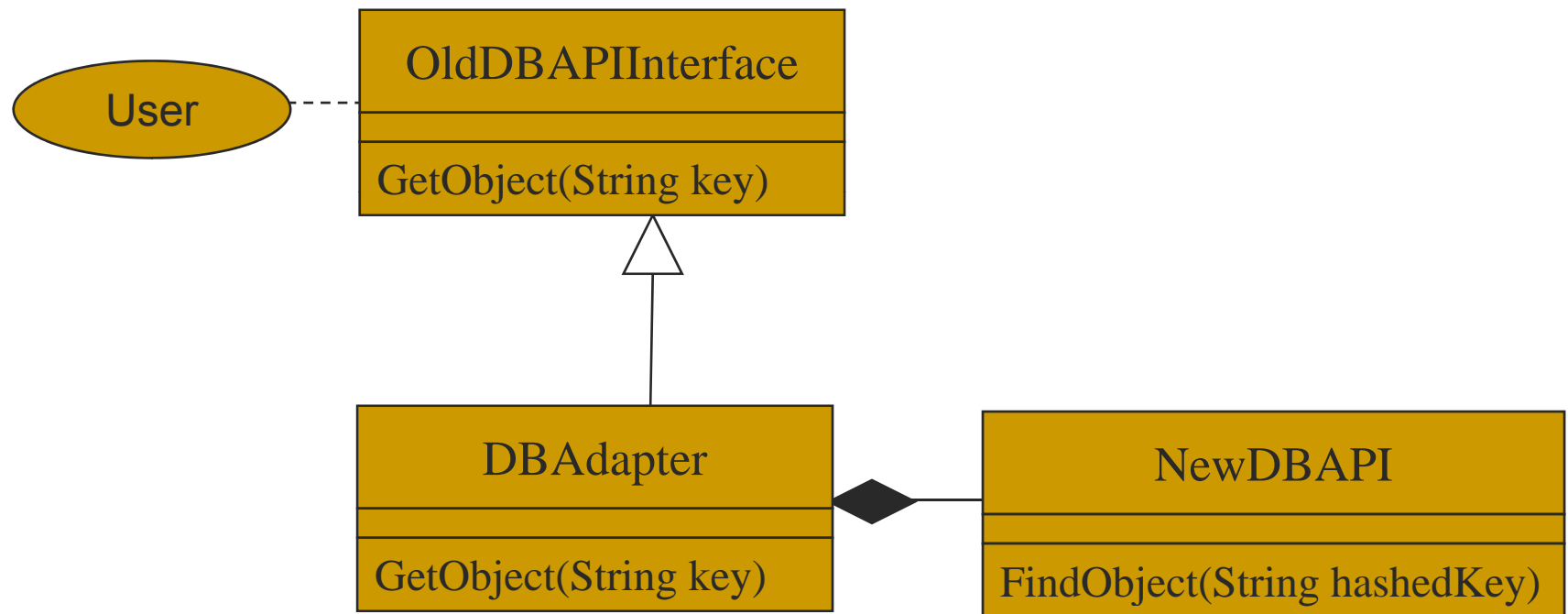
# Ví dụ

---

- Một công ty thay đổi hệ quản trị cơ sở dữ liệu
    - Hệ thống đang sử dụng CSDL thông qua các API
    - Hệ quản trị CSDL mới có các API khác, nhưng chức năng tương tự
-

# Adapter: Example

---

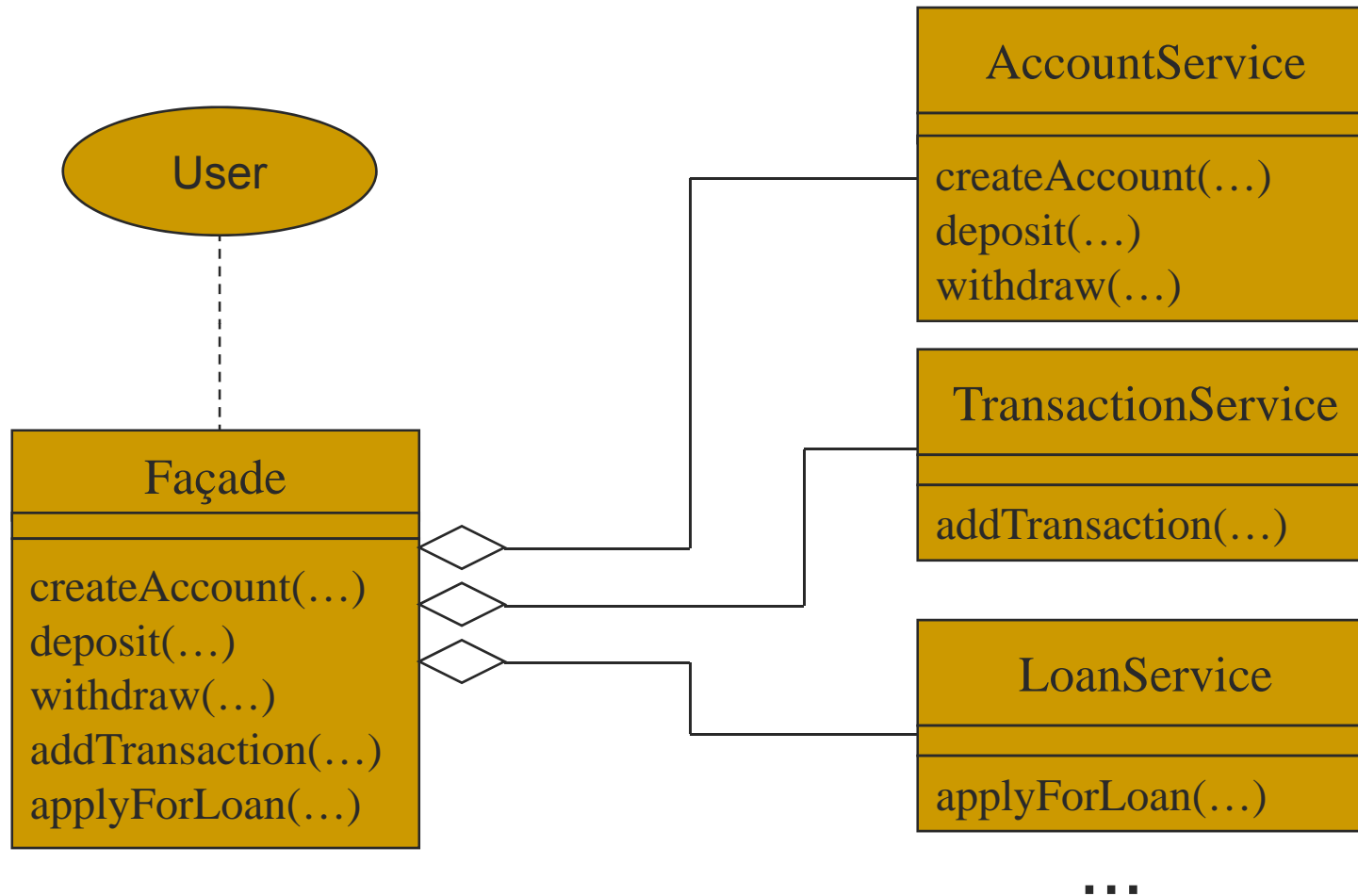


# Mẫu Façade

---

- Đưa ra một giao diện thay vì nhiều giao diện chi tiết của các lớp trong phân hệ
  - Sử dụng khi
    - Muốn cung cấp một giao diện đơn giản thay vì nhiều giao diện phức tạp
    - Giảm sự phụ thuộc của client với giao diện của các lớp trong phân hệ
    - Tạo nên kiến trúc tầng (layer)
-

# Ví dụ



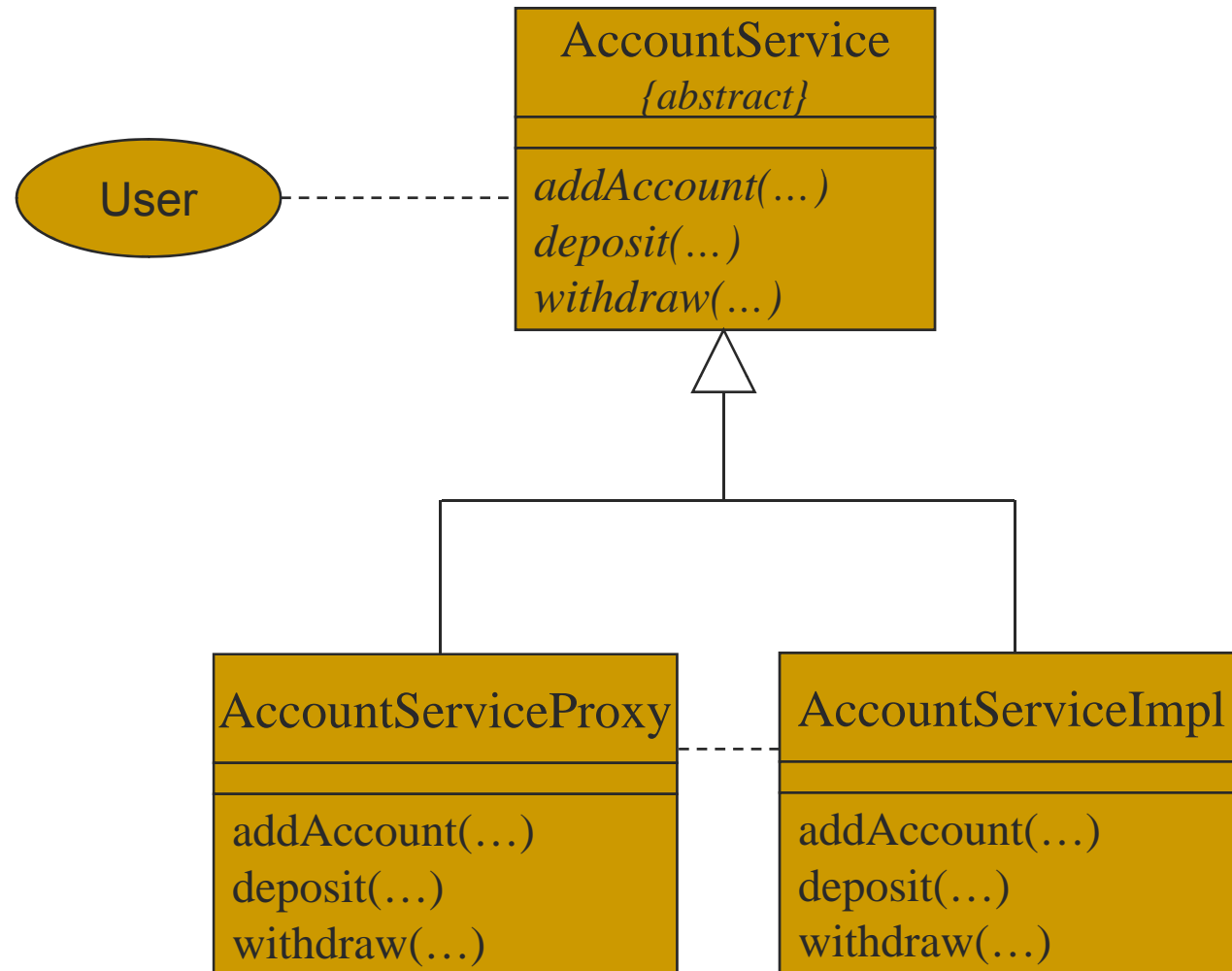
# Mẫu Proxy

---

- Ý tưởng chính: Một đối tượng proxy sẽ đại diện cho một đối tượng khác
  - Sử dụng khi
    - Điều khiển truy cập vào tài nguyên
    - Tạo đối tượng ảo khi việc tạo đối tượng thật sẽ có chi phí cao
    - Truy cập đối tượng ở máy khác
    - Thêm các chức năng khác như thu gom rác, đồng bộ,...
-

# Ví dụ

---





# Mẫu Observer

---

- Tạo mối quan hệ một-nhiều giữa các đối tượng sao cho khi một đối tượng thay đổi trạng thái, những đối tượng khác có thể biết
  - Sử dụng
    - Khi một đối tượng thay đổi, những đối tượng khác cần được thay đổi theo; số lượng đối tượng được thay đổi là không biết trước
-

# Mẫu Template Method

---

- Cung cấp phương tiện để thay đổi hành vi thông qua việc tạo các lớp con
    - Các lớp Template mô tả giải thuật, nhưng sẽ để một số phương thức dưới dạng trừu tượng
    - Các lớp con sẽ định nghĩa các phương thức trừu tượng theo cách mong muốn
  - Sử dụng khi
    - Một giải thuật gốc có thể dùng cho nhiều đối tượng
-

# Mẫu Template Method

---

```
class FileReader{
    abstract protected void openFile();
    public void open(){
        checkPermission();
        openFile();
    }
}
class TextReader extends FileReader{
    protected void openFile(){
        ...
    }
    ...
}
```

# Tóm tắt

---

- Mẫu thiết kế (design pattern) là một giải pháp tổng quát cho một vấn đề thường xảy ra trong quá trình thiết kế phần mềm
- Mẫu thiết kế có thể chia làm 3 loại
  - Mẫu thiết khởi tạo
  - Mẫu thiết kế cấu trúc
  - Mẫu thiết kế hành vi