**Lab 4: Tìm hiểu và cài đặt nhóm mẫu Behavioral (5 tiết)**

**Structural Patterns:**

1. [**Command**](http://www.dofactory.com/Patterns/PatternCommand.aspx)

Command Pattern trong ứng dụng quản lý văn bản

Diagram

Description automatically generatedCommand Pattern trong ứng dụng mở tài khoản ngân hàng

Diagram

Description automatically generated

* **Account**: là một request class.
* **Command**: là một interface của Command Pattern, cung cấp phương thức execute().
* **OpenAccount**, **CloseAccount** : là các **ConcreteCommand**, cài đặt các phương thức của Command, sẽ thực hiện các xử lý thực tế.
* **BankApp**: là một class, hoạt động như **Invoker**, gọi **execute()** của **ConcreteCommand** để thực thi request.
* **Client**: tiếp nhận request từ phía người dùng, đóng gói request thành ConcreteCommand thích hợp và gọi thực thi các Command.

1. [**Iterator**](http://www.dofactory.com/Patterns/PatternProxy.aspx)

Diagram

Description automatically generated

1. **Observer**

Diagram

Description automatically generated

* **Subject**: cung cấp các phương thức để thêm, loại bỏ, thông báo observer.
* **AccountService**: đóng vai trò là ConcreteSubject, sẽ thông báo tới tất cả các observers bất cứ khi nào có thao tác của người dùng liên quan đến đăng nhập, tài khoản hết hạn.
* **Observer**: định nghĩa một phương thức update() cho các đối tượng sẽ được subject thông báo đến khi có sự thay đổi trạng thái. Phương thức này chấp nhận đối số là SubjectState, cho phép các ConcreteObserver sử dụng dữ liệu của nó.
* **Logger**, **Mailer**và **Protector** là các ConcreteObserver. Sau khi nhận được thông báo rằng có thao tác với user và gọi tới phương thức update(), các ConcreteObserver sẽ sử dụng dữ liệu SubjectState để xử lý

Diagram

Description automatically generated