State Pattern là gì?

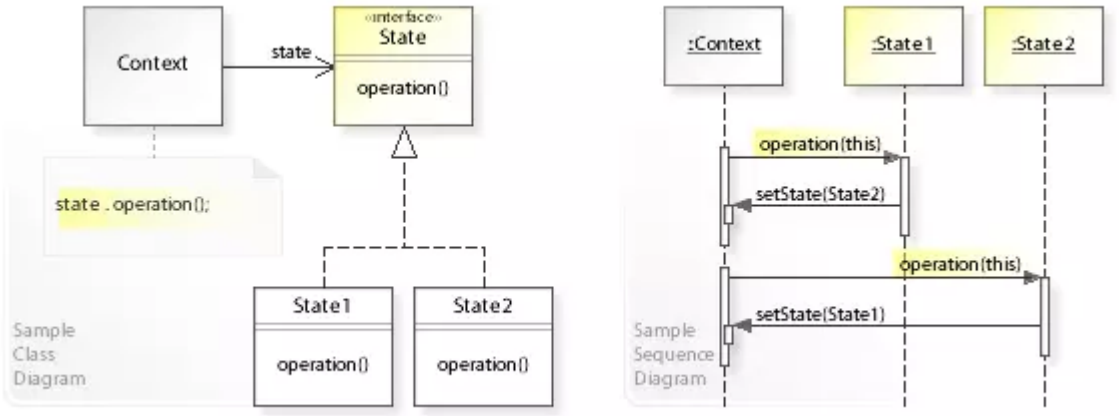
State Pattern là một trong những mẫu thiết kế thuộc nhóm behavioral cho phép một object có thể biến đổi hành vi của nó khi có những sự thay đổi trạng thái nội bộ. Nó có thể chuyển đổi các chiến lược thông qua các phương thức được định nghĩa trong interface.

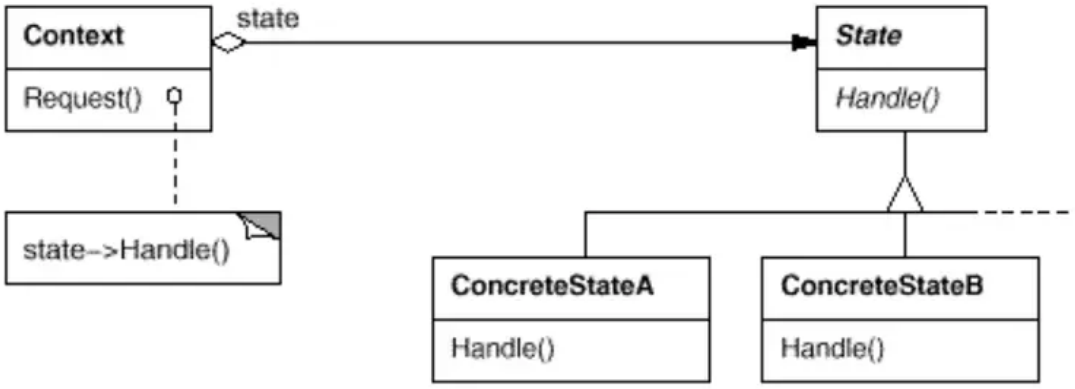
Ứng dụng của State Pattern

Chúng ta sẽ áp dụng State Pattern trong các trường hợp như sau:

- Hành vi của một đối tượng phụ thuộc vào trạng thái của nó. Tại thời điểm runtime, khi đối tượng thực hiện hành vi, trạng thái của nó sẽ thay đổi theo.
- Đối tượng có nhiều trường hợp sử dụng với các hành vi của nó, nhiều hành vi phụ thuộc vào trạng thái của đối tượng. Hay nói cách khác, đối tượng có nhiều trạng thái, mỗi trạng thái có những hành vi khác nhau.

Cấu trúc của State Pattern





Thành phần chính

Context

- Định nghĩa giao diện chính để giao tiếp với clients.
- Chứa một thể hiện của ConcreateState tương ứng với trạng thái hiện tại của đối tượng.

State

- Định nghĩa giao diện để đóng gói những hành vi giao tiếp với từng trạng thái của ConText

ConcreateState subclasses

- Mỗi một class con implements một hành vi giao tiếp với một trạng thái của ConText

Luồng hoạt động

- ConText sẽ định nghĩa những hành vi có thể giao tiếp với Clients, vì thế clients sẽ yêu cầu các hành vi thông qua Context
- ConText sẽ chứa một thể hiện của State, State ban đầu có thể được cài đặt từ Client, nhưng khi đã cài đặt rồi thì clients không được sửa đổi nó nữa.
- ConText có thể gửi chính nó như một arguments tới State, vì thế State có thể truy cập vào ConText để thay đổi trạng thái nếu cần thiết.

- Khi ConText thực hiện hành vi, nó sẽ gọi State hiện tại để thực hiện hành vi đó, thực hiện xong,State có thể sẽ thay đổi trạng thái từ Context nếu cần

Kết quả

- Phân vùng hành vi của đối tượng với những trạng thái khác nhau.
- Đối tượng được thay đổi trạng thái một cách rõ ràng.
- Trạng thái của những đối tượng có thể chia sẻ lẫn nhau.

Tham khảo

[1] Erich Gamma, Richard Helm, Ralph Johnson, John M. Vlissides (1995). Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley. ISBN 0-201-63361-2.

[2] http://www.w3sdesign.com/GoF_Design_Patterns_Reference0100.pdf