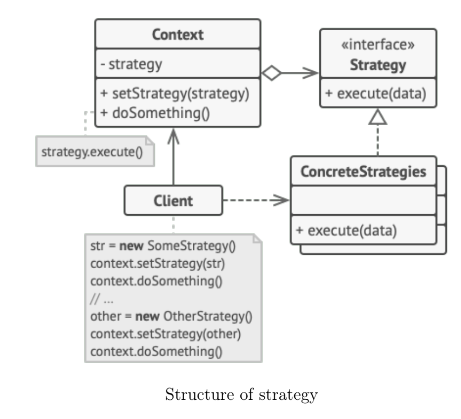
**Strategy pattern**

1. **Intent**

Strategy là mô hình hành vi (behavioral pattern), định nghĩa một loạt các thuật toán, cho từng các thuật toán vào các lớp riêng và cho phép các đối tượng của chúng hoán đổi cho nhau.

1. **Structure**



* Lớp Context duy trì một tham chiếu dến một trong các strategy cụ thể (concrete strategies) và chỉ giao tiếp với đối tượng này thông qua giao diện strategy.
* Giao diện Strategy là giao diện chung cho tất cả các strategy cụ thể (concrete strategies). Nó gọi một phương thức lớp context dùng để thực hiện một strategy.
* Concrete strategies áp dụng các biến thể khác nhau của một thuật toán mà lớp context sử dụng.
* Lớp context gọi phương thức thực hiện trên một đối tượng strategy tương mỗi khi nó cần thực thi thuật toán. Lớp context không biết nó làm việc với loại strategy gì hay cách mà thuật toán được thực thi.
* Lớp Client tạo ra một đối tượng strategy cụ thể và cho nó sang context. Lớp context hiện thị một setter mà cho phép client thay thế strategy tương ứng với context ở lúc đang chạy.

1. **When do we use it?**

* Use the Strategy pattern when you want to use different variants of an algorithm within an object and be able to switch from one algorithm to another during runtime.
* Use the Strategy when you have a lot of similar classes that only differ in the way they execute some behavior.
* Use the pattern to isolate the business logic of a class from the implementation details of algorithms that may not be as important in the context of that logic.
* Use the pattern when your class has a massive conditional statement that switches between different variants of the same algorithm.