**Mẫu Template Method**

1. **Thông tin mẫu**

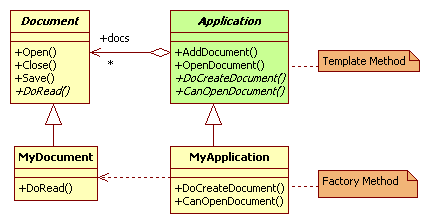
* Tên đầy đủ: Template Method Pattern
* Phân loại: Mẫu hành vi

1. **Mục đích, ý định**

* Định nghĩa một bộ khung của một thuật toán trong một chức năng, chuyển giao việc thực hiện nó cho các lớp con.
* Mẫu Template Method cho phép lớp con định nghĩa lại cách thực hiện của một thuật toán, mà không phải thay đổi cấu trúc thuật toán.

1. **Động lực sử dụng**

* Đôi khi chúng ta muốn xác định thứ tự thực hiện của các hành động được sử dụng bên trong một phương thức, nhưng lại muốn các lớp con quyết định cách thực thi cho một hay nhiều các hành động này.



1. **Khi nào ứng dụng**

* Khi định nghĩa một thuật toán, ta muốn hiện thực một phần của nó, sau đó để cho các lớp con hiện thực phần còn lại.
* Khi các lớp con có một hành vi chung, cần được đưa vào một lớp riêng để tái sử dụng và tránh trùng lặp code.
  + Trước tiên ta xác định các sự khác biệt trong các đoạn code có sẵn, sau đó tách các phần khác nhau vào một operation mới.
  + Sau đó, thay các đoạn mã bên trên bằng một template method sẽ gọi các operation mới được định nghĩa.
* Khi ta cần kiểm soát việc mở rộng các lớp con.
  + Ta có thể định nghĩa một template method, nó sẽ gọi các “hook” operation tại một số nơi của chương trình, nhờ đó việc mở rộng chỉ được cho phép tại các điểm đó mà thôi.

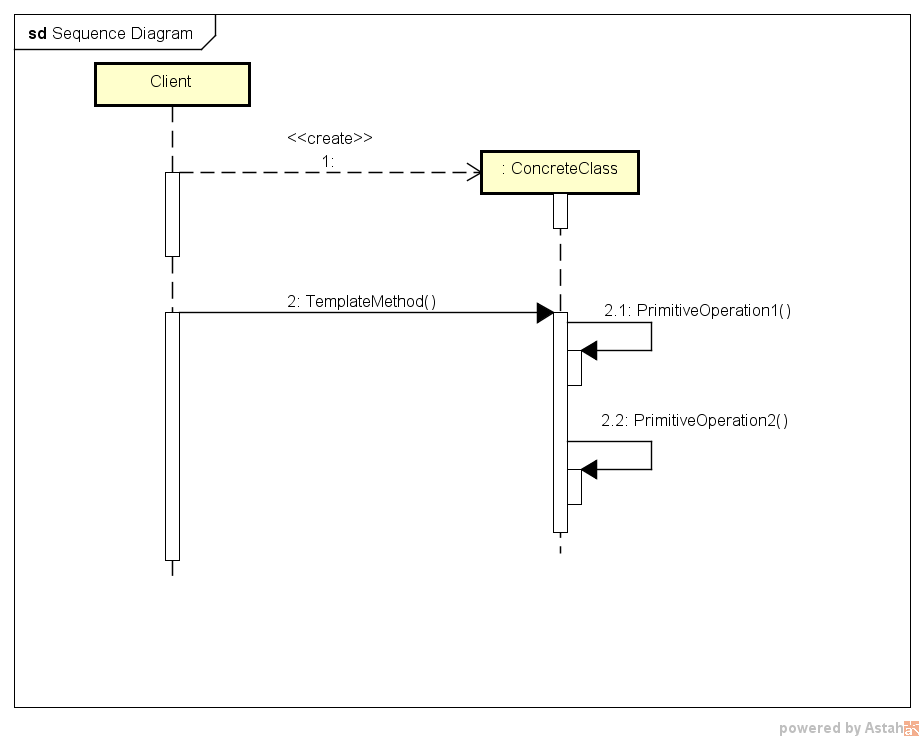
1. **Cấu trúc**



1. **Mối quan hệ - Các thành viên**

* **AbstractClass:**
  + Định nghĩa các hoạt động ảo nguyên thủy (abstract primitive operation) cấu thành nên một thuật giải thuật.
  + Hiện thực một Template method định nghĩa khung sườn của một giải thuật. Template method này sẽ gọi các hàm ảo nguyên thủy được định nghĩa trong AbstractClass.
* **ConcreteClass**:
  + Cài đặt các hàm ảo nguyên thủy được định nghĩa trong **AbstracClass.**
  + Thuật toán cài đặt trong Abstract Class sẽ thay đổi tùy theo cách hiện thực các hàm nguyên thủy trong các lớp con.

1. **Sự cộng tác**

****

* Khi một ConcreteClass được gọi, code của template method sẽ được thực thi từ lớp cơ sở trong khi các phương thức bên trong template method sẽ gọi các hiện thực của chúng từ các lớp con.

1. **Các hệ quả**

* Kỹ thuật cơ bản cho việc tái sử dụng code:
  + Đặc biệt quan trọng trong các lớp thư viện, chúng là phương tiện dể chúng ta tìm ra các hành vi chung trong các lớp.

1. **Lưu ý cài đặt**
2. **Hệ thống sử dụng**
3. **Mẫu liên quan**

* Factory Method:
  + Thường được gọi bởi các template method.
* Strategy:
  + Template method sử dụng kế thừa để thay đổi một phần của một giải thuât, Stategy sử dụng ủy nhiệm để thay đổi cả một giải thuật.

1. **Demo**