
Bài 21

Creational Design Pattern

Module: ADVANCED PROGRAMMING WITH JAVA

- Trình bày được ý nghĩa của Design Pattern
- Trình bày được ý nghĩa của các Creational Design Pattern
- Triển khai được Singleton Pattern
- Triển khai được Factory Method Pattern
- Triển khai được Object Pool Pattern

- Design Pattern (DP - Mẫu Thiết kế) là các giải pháp tổng quát có thể tái sử dụng cho các trường hợp thường gặp khi thiết kế kiến trúc phần mềm.
- Design Pattern không phải là bản thiết kế hoàn chỉnh có thể dùng để chuyển hoá trực tiếp thành mã nguồn.
- Design Pattern là các khuôn mẫu (template) để giải quyết vấn đề trong các tình huống khác nhau.

Lợi ích của Design Pattern



- *Đẩy nhanh tốc độ thiết kế và phát triển phần mềm*
- *Chất lượng của giải pháp đã được minh chứng*
- *Ngăn ngừa các vấn đề phát sinh nếu thiết kế không tốt*
- *Có thể áp dụng cho rất nhiều tình huống khác nhau*
- *Dễ dàng cộng tác, chia sẻ thiết kế và mã nguồn giữa các bên*

Phân nhóm Design Pattern



- Creational Design Pattern:
 - Là nhóm các Design Pattern *được sử dụng để* giải quyết các vấn đề thường gặp đối với việc khởi tạo đối tượng.
- Structural Design Pattern:
 - Là nhóm các Design Pattern *được sử dụng để* thiết kế các thành phần của lớp và đối tượng giúp việc dễ dàng nhận ra mối quan hệ giữa các thực thể.
- Behavioral Design Pattern:
 - Là nhóm các Design Pattern *được sử dụng để* giải quyết các vấn đề phổ biến trong giao tiếp giữa các đối tượng.

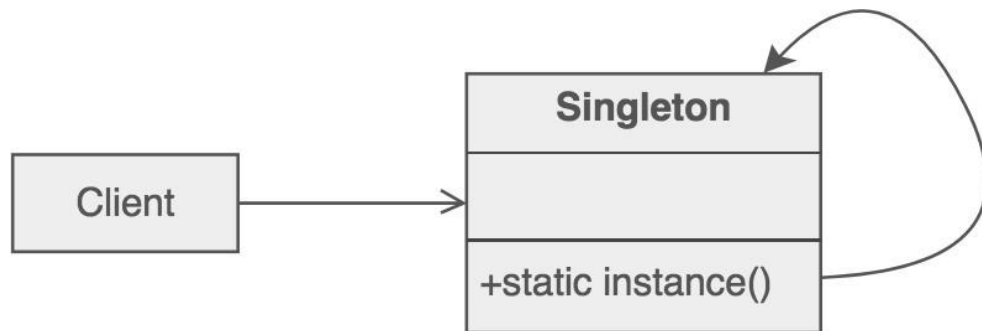
Các Creational Design Pattern thông dụng



- Singleton
 - Giải quyết vấn đề: Ứng dụng chỉ cần duy nhất đối tượng của một lớp nào đó.
- Factory Method
 - Giải quyết vấn đề: Các ứng dụng riêng lẻ xác định các đối tượng miền của riêng chúng và cung cấp sự khởi tạo những đối tượng này.
- Object Pool
 - Giải quyết vấn đề: Giới hạn số lượng các đối tượng và tái sử dụng chúng thay vì phải tạo ra quá nhiều đối tượng khiến tốn kém tài nguyên.

- Singleton là một mẫu thiết kế thuộc nhóm creational cho phép bạn *đảm bảo rằng một lớp chỉ có một đối tượng thể hiện và cung cấp truy cập đối tượng này với phạm vi toàn ứng dụng.*
- Singleton *đảm bảo chỉ duy nhất một thể hiện mới (new instance) được tạo ra và nó sẽ cung cấp cho bạn một phương thức để truy cập đến thực thể đó.*
- Một lớp thiết kế theo Singleton Pattern có các *đặc điểm*:
 - Phương thức khởi tạo private để ngăn cản việc tạo thể hiện của class từ các lớp khác.
 - Biến private static của class, nó là thể hiện duy nhất của lớp.
 - Có một phương thức public static để trả về thể hiện của lớp.

Singleton -2



```
public final class Singleton {
    private static volatile Singleton instance;

    private Singleton() {}

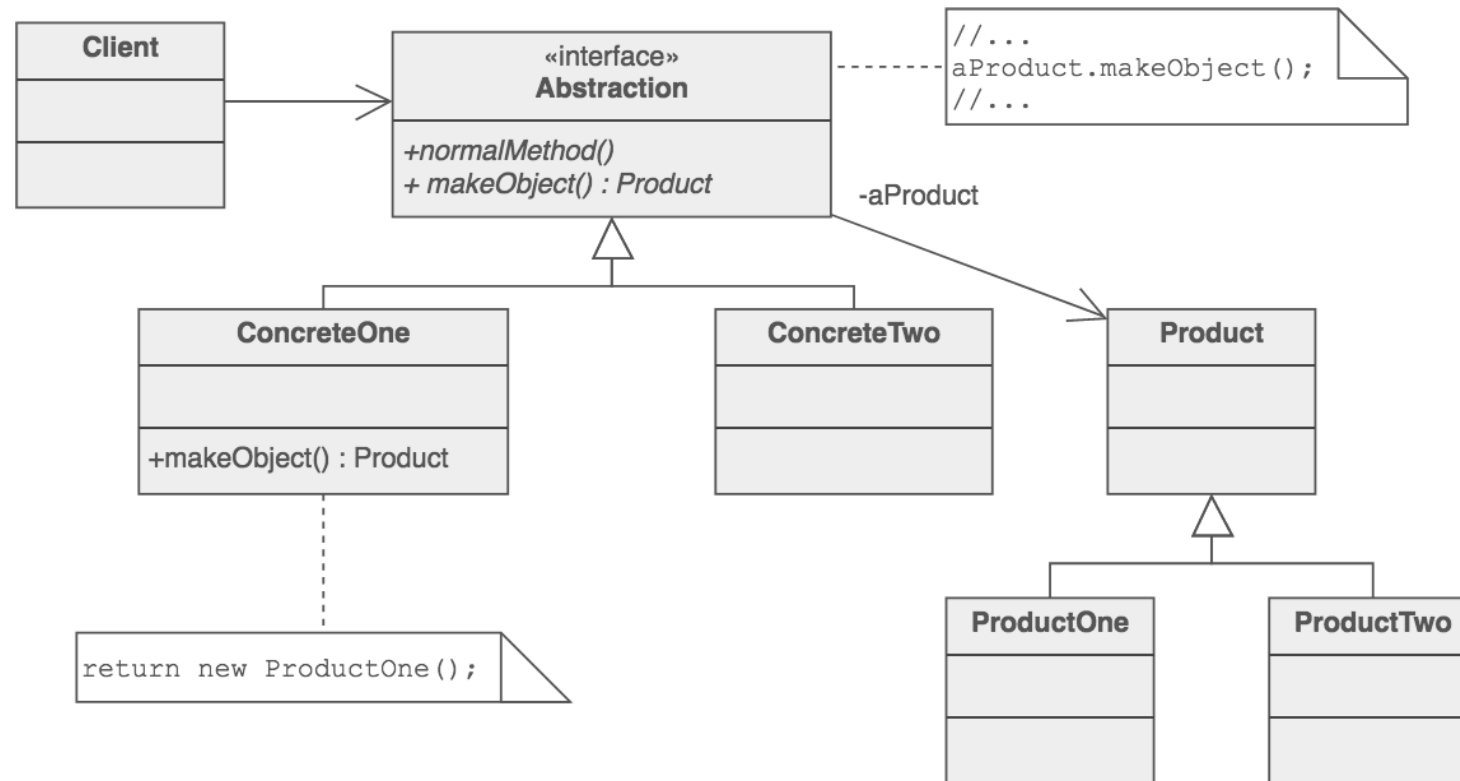
    public static Singleton getInstance(String value) {
        if (instance == null) {
            synchronized (Singleton.class) {
                if (instance == null) {
                    instance = new Singleton();
                }
            }
        }
        return instance;
    }
}
```


Factory Method



- Đây là mẫu thiết kế có thể xem là được sử dụng phổ biến nhất, với mục đích quản lý và khởi tạo các đối tượng một cách linh hoạt.
- Factory Method định nghĩa một interface dành cho việc tạo ra đối tượng, nhưng để lớp con quyết định lớp nào sẽ được dùng để khởi tạo đối tượng.
- Factory Method tương tự Abstract Factory nhưng khác với Factory Method, Abstract Factory là factory của các factory.

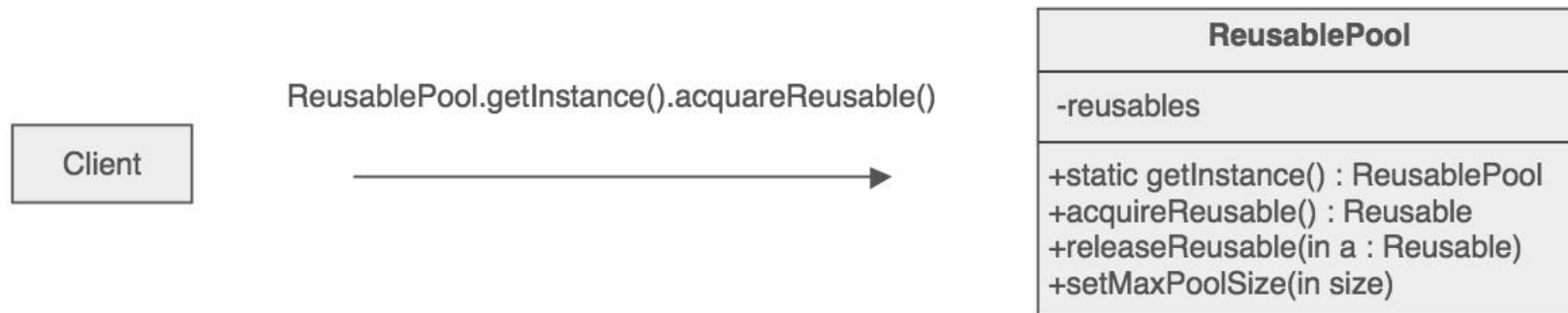
Factory Method - 2



Object Pool



- Object Pool Pattern quản lý một tập hợp các *đối tượng* sẽ được tái sử dụng trong chương trình.
- Các *đối tượng* được chứa chung ở một nơi gọi là pool (bể chứa). Khi cần dung, *đối tượng* được gọi ra từ pool, sử dụng trong một khoảng thời gian nhất định rồi trả về pool. Không thành phần nào có thể sử dụng tận được *đối tượng* được lấy ra khỏi pool cho tới khi nó được quay trả về.
- Object Pool Pattern thích hợp với tình huống khi có nhiều hơn một *đối tượng* và số *đối tượng* được khởi tạo là hạn chế.



[Thực hành] Triển khai Singleton Pattern



[Thực hành] Triển khai Factory Method



[Thực hành] Triển khai Object Pool



[Bài tập] Áp dụng Factory Method



- Design Pattern (Mẫu Thiết kế) là các giải pháp tổng quát có thể tái sử dụng được trong các trường hợp thường gặp khi thiết kế kiến trúc phần mềm
- Singleton chỉ khởi tạo một đối tượng duy nhất của một lớp
- Factory Method định nghĩa một Interface để khởi tạo đối tượng, nhưng để cho các lớp con quyết định sẽ khởi tạo đối tượng của lớp nào.
- Object Pool nâng cao hiệu năng của ứng dụng thông qua việc tạo ra một tập hạn chế các đối tượng và tái sử dụng chúng thay vì phải tạo ra quá nhiều đối tượng

Hướng dẫn

- Hướng dẫn làm bài thực hành và bài tập
- Chuẩn bị bài tiếp: Structural Design Pattern