Singleton là 1 nguyên tắc xây dựng class trong java. Các lớp khởi tạo bằng singleton chứa 1 trường final field, contructor của singleton khai báo ở dạng private.

Kế thừa ?

Bản chất của kế thừa java là khởi tạo 1 constructor không biến. không nên kế thừa một lớp singleton. Bản chất của singleton là khởi tạo 1 object duy nhất trong toàn hệ thống, hỗ trợ cấu hình toàn cục và lưu trữ các giá trị duy nhất. khi kế thừa 1 singleton, ta cố gắng cấu hình các lớp con và mang bản chất của lớp cha, điều này có thể dẫn đến các vấn đề phức tạp khác.

Việc nhiều lớp con kế thừa 1 lớp cha singleton cũng làm mất đi bản chất của singleton, khi mà 1 hệ thống chỉ nên có 1 object cấu hình theo singleton (có thể có nhiều object khác cấu hình theo Singleton nhưng khác nhau về chức năng và bản chất ? giá trị lưu trữ ? … CHỈ NÊN CÓ 1) – còn các lớp con cố gắng cấu hình theo hướng cá nhân của nó (không chắc nữa, nhưng cấu hình để có thể tối ưu hóa khả năng sử dụng của nó). ĐIỀU NÀY VI PHẠM TÍNH ĐÓNG GÓI

Giải pháp ở đâu là nhúng. Nhúng đối tượng singleton vào lớp như 1 thành viên của lớp đó để có thể sử dụng instance của nó.

Còn nếu singleton kế thừa các lớp con thì sao ? KHÔNG NÊN. Việc 1 lớp singleton kế thừa 1 lớp con bình thường khác sẽ làm cho mất bản chất của singleton. Khi mà lớp cha sinh ra hàm mới thì singleton cũng phải kế thừa các hàm đó, mất đi tính duy nhất của singleton.

NHƯNG, có thể kế thừa các lớp trừu tượng khác. Việc kế thừa 1 lớp trừu tượng vẫn đảm bảo tính duy nhất của singleton.

Nếu có 2 object cùng khai báo 1 kiểu dữ liệu singleton, việc khai báo không làm ảnh hưởng tới value của singleton. Nhưng việc truy xuất dữ liệu theo thứ tự không mong muốn có thể khiến kết quả của singleton class khác đi so với dự tính ban đầu. từ đó, ta chỉ nên sử dụng 1 singleton class với 1 đối tượng, trong 1 lớp cụ thể (controller, service, …)

Nếu có 2 đối tượng trở lên cần kế thừa 1 object singleton thì cần phải dùng @Qualifier cho nó, hoặc sử dụng @Primary