Singleton Design Pattern

1. Mục lục
2. Định nghĩa Singleton DP là gì ?
3. Cách một class trở thành Singleton
4. Những trường hợp cần sử dụng Singleton DP.

2. Nội dung

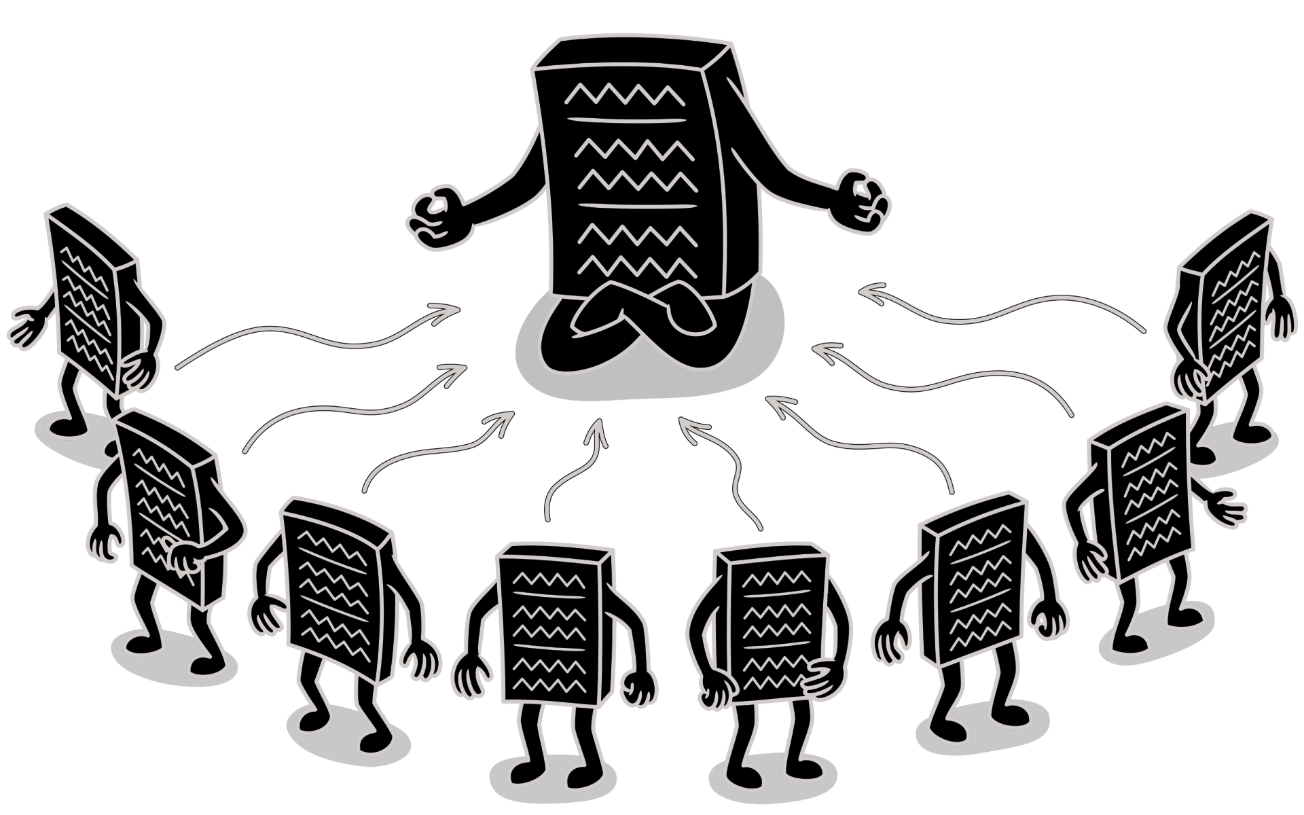
1.Định nghĩa

Singleton DP là một design pattern hỗ trợ người lập trình :

\_ Giups người lập trình đảm bảo được một class đã chọn

chỉ có một thể hiện (instance) duy nhất.

\_ Đảm bảo truy cập được mọi lúc mọi nơi (toàn cục)

 \_ Khởi tạo just-in-time hoặc khởi tạo trong lần gọi đầu tiên.

Chúng ta có thể kể đến một số lợi ích của nó như sau:

\_ Do chỉ có một thể hiện duy nhất nên việc quản lý tốt hơn

,ngoài ra chúng ta có thể quản lý số lượng thể hiện của nó, cũng

không nhất thiết chỉ có một thể hiện.

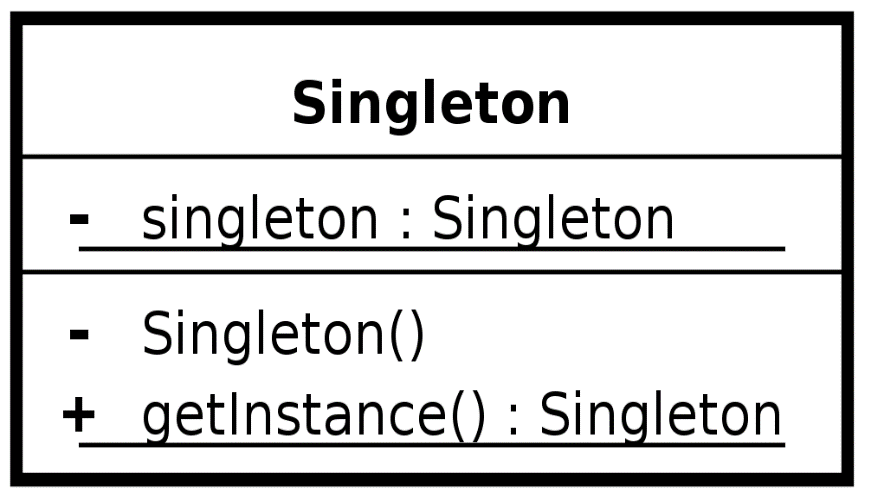
\_ Do pattern có thể kế thừa và tùy biến lại thông qua thể

hiện của lớp con nên dĩ nhiên, có thể cải tiến lại các thể hiện.

\_ Được truy cập từ mọi nơi (global)

Ví dụ avatar trong một game thời trang, một tài khoản chỉ có một avatar, thích hợp cho singleton.

1. Cách để một class trở thành Singleton

\_ Định nghĩa thuộc tính là private static và nó sẽ là thể hiện của class  
\_ Định nghĩa phương thức là public static để khởi tạo và thực thi lazy-init ( còn gọi là accessor)

\_ Bắt buộc các constructor là private hoặc protected để ngăn chặn việc tạo thêm thể hiện.

MINH HỌA UML

1. Những trường hợp không nên dùng Singleton

\_ Biến toàn cục: Việc sử dụng Singleton sẽ làm cho chương trình rất khó hiểu, debug vì dùng nhiều. Hơn nữa biến toàn cục làm cho nó không còn “bảo mật” nữa.

\_ Thực thể của Singleton sẽ luôn tồn tại trong suốt vòng đời của nó, và nó được truy cập từ mọi nơi trong chương trình, đi ngược lại với phong cách lập trình hướng đối tượng, do đó những trường hợp sử dụng Singleton đôi khi nên xem xét có thể thay thế bởi Dependency Injection