**Singleton Design Partern là gì?**

* Là 1 Creational Design Partern.
* Đảm bảo rằng 1 class chỉ được khởi tạo 1 lần duy nhất (chỉ có 1 instance duy nhất trong toàn bộ ứng dụng), đảm bảo giá trị các thuộc tính (Property) của class không bị thay đổi sau khi class được khởi tạo.
* Cung cấp 1 Global Entrypoint để các class khác truy cập.

**Ưu điểm:**

* Đảm bảo class chỉ có duy nhất 1 instance.
* Có thể truy cập ở bất cứ đâu trong ứng dụng. Đến đây bạn có thể thấy khá giống với biến toàn cục (global variable) nhưng Singleton Class khác với biến toàn cục ở chỗ không bị ghi đè giá trị.
* Chỉ được khởi tạo khi có yêu cầu.

Nhược điểm:

* Vi phạm quy tắc Single Responsibility Principle (1 đối tượng chỉ phục vụ 1 mục đích duy nhất) trong SOLID. Vì thế có thể gọi ISngleton là 1 Aniti Partern.
* Phải rất cẩn thận khi sử dụng đa luồng (Multi-Thread) vì có thể nhiều Thread cùng truy cập vào 1 object trong 1 thời điểm.
* Khó khan trong Unit Test

**Khi nào sử dụng Singleton?**

* Quản lý cấu hình (Configuration Management).
* Quản lý kết nối cơ sở dữ liệu (Database Connection).
* Logging (Ghi nhật ký).
* Trình điều khiển (Driver).
* Cache.
* Thread Pool (Bộ xử lý luồng).
* File System Manager (Quản lý hệ thống tệp.

**Cấu trúc của Singleton Partern**

* Private Constructor: Đảm bảo các class sử dụng Singleton Class không thể khởi tạo instance mới.
* GetInstance Method: Kiểm tra xem class đã được khởi tạo chưa? Nếu chưa thì tạo mới instance ngược lại trả về instance hiện tại.

**Demo**

* Trong loạt bài này mình sử dụng ngôn ngữ C# để Demo các Design Partern.
* Đầu tiên mình tạo ra class Singleton

A screenshot of a computer

Description automatically generated

* Sau đó trong Program.cs, mình tạo ra 2 biến s1, s2 để tạo các instance của class Singleton

A screenshot of a computer

Description automatically generated

- OK! Mọi thứ hoạt động tốt khi chỉ trả lại 1 instance duy nhất của class Singleton.

Bây giờ mình them một Property InitialValue và được gián giá trị trọng constructor của Singleton và hàm Speak để in giá trị của InitialValue ra Console:

A computer screen shot of a program

Description automatically generated

* Trong Program.cs mình tạo ra 2 process ở 2 thread và gọi GetInstance trong đó với 2 giá trị truyền vào khác nhau lần lượt là “Singleton 1” và “Singleton 2”, sau đó chạy process:

A screenshot of a computer

Description automatically generated Có gì đó sai sai. InitialValue khác nhau ở 2 Thread như vậy Singleton Design Partern đã sai khi tạo ra 2 instance khác nhau. Như mình đã nói trong phần nhược điểm của Singleton Design Partern thì phải rất cần thận khi dung Multi-Thread. OK vậy sau đây mình sẽ giải quyết vấn đề này:

A computer screen with text on it

Description automatically generated

* Mình them Property \_lock và sử dụng từ khóa lock trong hàm GetInstance để khóa chỉ cho 1 luồng duy nhất truy cập trong 1 thời điểm:

A computer screen shot of a program

Description automatically generated

* OK! Kết quả trả về 2 dòng giống nhau chứng tỏ luôn trả về duy nhất 1 instance của class Singleton.

Các bạn có thể tham khảo repo git của mình: [Design-Partern/2. Singleton at main · haitc21/Design-Partern (github.com)](https://github.com/haitc21/Design-Partern/tree/main/2.%20Singleton)

Cảm ơn các bạn đã theo dõi. Hãy để lại bình luận để mình phát triển bài viết tốt hơn. Đừng quên Insight, Share để mình có them động lực viết them nhiều bài viết bổ ích. Xin chào! Hẹn các bạn ở bài viết tiếp theo trong seri Design Partern.