

C# 싱글턴패턴

싱글턴(Singleton) 패턴

- 인스턴스가 하나 뿐인 객체를 만들 수 있게 해주는 패턴
- 해당 클래스의 인스턴스가 하나만 생성 됨
- 어디서든 인스턴스에 접근 가능
- 스레드 풀, 캐시, 대화상자, 사용자 설정, 디바이스 드라이버 등 클래스의 객체가 어플리케이션에서 오직 하나만 생성되는 경우 사용

싱글턴 패턴의 구현

```
class OldSingleton
{
    private static OldSingleton _instance;
    protected OldSingleton()
    {
    }

    public static OldSingleton Instance()
    {
        if (_instance == null)
        {
            _instance = new OldSingleton();
        }

        return _instance;
    }
}
```

Singleton
-instance : Singleton
-Singleton() +Instance() : Singleton

문제

- 멀티 스레드로 동작하는 어플리케이션에서 서로 다른 곳에서 동시에 인스턴스를 호출할 경우?
- 이를 해결하기 위한 방법은?

싱글턴 패턴의 구현

- lock

- 특정 블록의 코드를 한번에 하나의 스레드만 실행할 수 있도록 해준다.

```
class NewSingleton
{
    private static NewSingleton _instance;
    private static object syncLock = new object();

    protected NewSingleton()
    {
    }

    public static NewSingleton Instance()
    {
        if (_instance == null)
        {
            lock (syncLock)
            {
                if (_instance == null)
                {
                    _instance = new NewSingleton();
                }
            }
        }

        return _instance;
    }
}
```

C# 최적화

- private 생성자
 - 클래스 내부에서만 생성자 호출 가능
- readonly
 - 읽기만 가능
- seal
 - 상속 봉인

```
sealed class OptimizeSingleton
{
    private static readonly OptimizeSingleton _instance = new OptimizeSingleton();

    private OptimizeSingleton()
    {
    }

    public static OptimizeSingleton Instance()
    {
        return _instance;
    }
}
```