Sinleton Pattern

Garante a criação de uma única instância de uma classe através de toda a aplicação, mantendo um ponto global de acesso ao seu objeto (Um padrão totalmente fechado)

**Principais utilizações desse padrão**

* Única conexão com banco de dados
* Acesso ao sistema de arquivos de um sistema
* Sistema de logs

**Problemas de sua utilização**

* Lazy construction (Só instancia se for chamado)
* Não há parâmetros para sua construção
* Acesso de forma global

**Estrutura principal**

* Método construtor é privado
* Método”. instance” responsável por criar o projeto
* Ele verifica se o objeto existe, caso contrário cria
* Possui outros métodos como uma classe qualquer

**Singleton: not thredSafe vs thread safe**

* Padrão “original” singleton não é thread safe. Isso é grave
* Existem formas de contornar o problema através de variações desse padrão

**Consequências: Singleton como um anti-pattern**

* Alto acoplamento
* Acesso global de um objeto
* Não é threadsafe
* Dificuldade para trabalhar com teste
* Quebra o SOLID
* SRP (ela quebra a responsabilidade de se gerenciar)
* Pode ser facilmente substituída através de injeção de dependências (DI)