Aluno: Erivaldo Barros

Escolha e descreva um padrão de projetos.

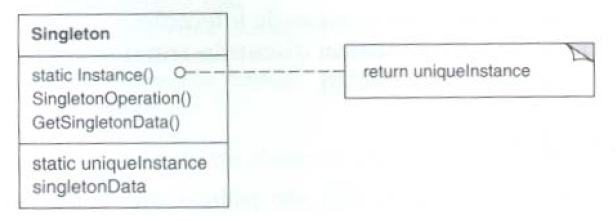
Padrões de Criação -> ***Singleton***

Função -> Garantir que uma classe tenha somente uma instância e fornecer um ponto global de acesso para a mesma.

O padrão *singleton* deve ser utilizado quando:

* Haver apenas uma instância de uma classe, e essa instância deve dar acesso aos clientes através de um ponto bem conhecido;
* Quando a única instância tiver de ser extensível através de subclasses, possibilitando aos clientes usarem uma instância estendida sem alterar o seu código.

**Estrutura**



**Benefícios**

1. **Acesso controlado à instância única**. Porque a classe *Singleton* encapsula a sua única instância, pode ter um controle total sobre como e quando os clientes a acessam.
2. **Espaço de nomes reduzidos**. O padrão *Singleton* representa uma melhoria em relação ao uso de variáveis globais. Ele evita a poluição do espaço de nomes com variáveis globais que armazenam instâncias únicas.
3. **Permite um refinamento de operações e da representação**. A classe *Singleton* pode ter subclasses e é fácil configurar uma aplicação com uma instância desta classe estendida. Você pode configurar a aplicação com uma instância da classe de que necessita em tempo de execução.
4. **Permite um número variável de instâncias.** O padrão torna fácil mudar de ideia, permitindo mais de uma instância da classe *Singleton*. Além disso, você pode usar a mesma abordagem para controlar o número de instâncias que a aplicação utiliza. Somente a operação que permite acesso à instância de *Singleton* necessita ser mudada.
5. **Mais flexível do que operações de classe**. Uma outra maneira de empacotar a funcionalidade de um *Singleton* é usando operações de classe. Porém, isso torna difícil mudar um projeto para permitir mais que uma instância de uma classe.