

Engenharia de Software 2 - Padrões de Projeto

Alunos: Max Mohamed Freitas e Gabriel Venâncio de Carvalho

Padrão escolhido: Singleton.

O padrão Singleton é um design pattern que garante a existência de apenas uma instância de uma classe em toda a aplicação. Ele permite que a própria classe gerencie sua única instância, impedindo que qualquer outra classe crie novas instâncias.

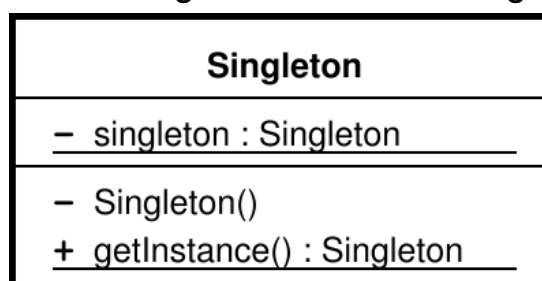
Este padrão oferece um ponto global de acesso à instância, se certificando que todas as partes do sistema utilizem sempre o mesmo objeto. Se a instância ainda não existe, a classe é responsável por criá-la e retorná-la.

O Singleton é ideal em situações no qual recursos compartilhados, como logs, configurações globais ou conexões com banco de dados, precisam ser controlados de forma unificada.

Resolve o problema de múltiplas instâncias que estão em conflito, que estão causando inconsistências ou desperdício de recursos. O padrão resolve estes problemas ao restringir a criação de novos objetos por meio de um construtor privado e ao fornecer um método público ou estático que retorna a instância única, criando-a apenas na primeira chamada.

Em resumo, o Singleton evita duplicação de recursos e assegura a consistência em sistemas no qual um único ponto de controle é necessário.

O diagrama de classe a seguir mostra a sua organização:



Fonte: <https://www.devmedia.com.br/padrao-de-projeto-singleton-em-java/26392>

O diagrama de classe acima mostra que ele tem um atributo chamado singleton, que é do mesmo tipo da classe e é estático. Isso

significa que essa variável armazena a única instância da classe e é compartilhada por todo o programa.

O construtor da classe Singleton é privado e ninguém pode criar uma instância diretamente, apenas a própria classe pode fazer isso através do método `getInstance()`.

O método `getInstance()` pode ser chamado de qualquer lugar sem precisar criar um objeto da classe. Esse método verifica se a instância já existe e se não existir, cria e retorna a instância.