

## Disciplina de Projeto Orientado a Objetos

**Nome:** Ana Paula Lourenço

**Curso:** Tecnologia em Desenvolvimento de Sistemas

**Atividade 01:** Pesquise pelo menos 2 padrões de projeto e fale um pouco sobre cada um.

Os padrões de projetos são soluções desenvolvidas em nível de design para os engenheiros de software resolverem problemas no seu dia a dia de trabalho. É considerada uma solução reutilizável para problemas comuns dentro do desenvolvimento, sobretudo, em sistemas orientados a objetos.

Existem três tipos de Padrões de Projeto:

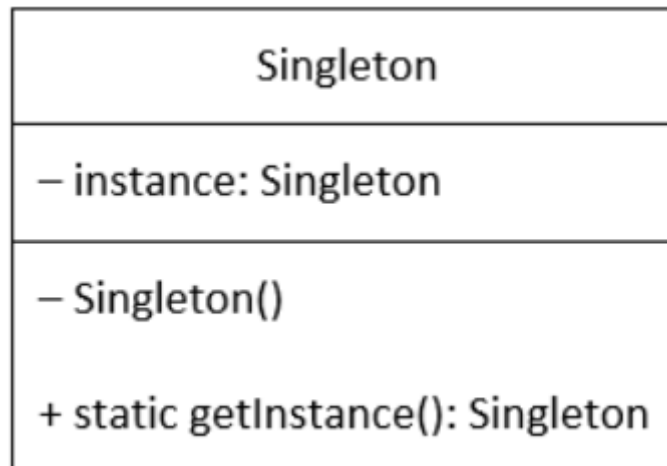
1. **Padrões de Criação:** padrões criados para instanciação de classes. Eles podem tanto ser padrões de criação de classes como de criação de objetos
2. **Padrões Estruturais:** foram criados levando em conta a estrutura de uma classe e sua composição. O objetivo principal da maioria desses padrões é aumentar a funcionalidade das classes envolvidas, sem alterar muito sua composição.
3. **Padrões Comportamentais:** levam em conta a forma como uma classe se comunica com outras.

Nesta atividade focaremos, de forma sintética, em dois exemplos de Padrões de Projetos em Java, são eles: *Decorator* (estrutural) e *Singleton* (criacional).

### Singleton

O Singleton é um padrão de criação clássico e seu objetivo é criar apenas uma instância de uma classe e fornecer apenas um ponto global de acesso àquele objeto. Para que isso aconteça um projeto Singleton é composto por: uma variável estática, que contém uma instância da classe; um método construtor privado, para que não seja possível instanciá-la de outro lugar e um método estático público para retornar a instância da única classe existente.

### Exemplo de Diagrama de Classe Singleton



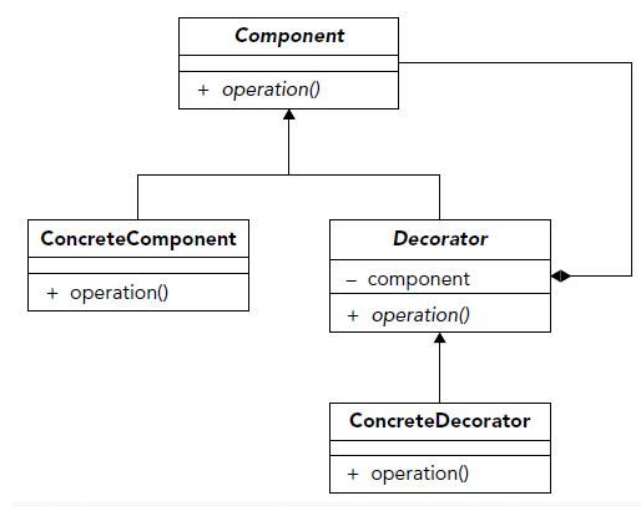
Fonte: FreeCodeCamp, 2024.

O padrão Singleton é utilizado quando necessitamos de uma única instância de uma classe para controlar ações. Fazer conexões com o banco de dados é uma delas.

### Decorator

É um Padrão de Projeto que permite os desenvolvedores acoplarem novos comportamentos para objetos dinamicamente, ou seja, usando o padrão *decorator* é possível agrupar inúmeras vezes, pois objetos de destino e os decoradores seguem a mesma interface.

### Exemplo de Diagrama de Classe Decorator



Fonte: DevMedia, 2024.

O Decorator geralmente é utilizado quando é necessário adicionar funcionalidades a objetos de maneira flexível e independente. Ele é composto de duas partes principais:

1. **Componente:** é a interface ou classe base que define as operações que os objetos decorados e decoradores devem experimentar.
2. **Decorador Concreto:** são classes que implementam ou herdam a interface de Componente, encapsulando e adicionando funcionalidade ao objeto original.

Um exemplo clássico de projeto Decorator é o da cafeteria, que vende diferentes tipos de cafés e cada bebida pode ter adicionais, como leite, açúcar, adoçante etc.

Por exemplo, sem o padrão Decorator o desenvolvedor poderia criar várias subclasses para cada combinação de bebida e adicionais, mas isso teria como resultado muitas classes para cada possível combinação. Com o padrão Decorator é possível criar uma classe de bebida e decorá-la com os adicionais de forma dinâmica.

## REFERÊNCIAS

Padrão de Projeto Singleton em Java. Disponível em:  
<<https://www.devmedia.com.br/padrao-de-projeto-singleton-em-java/26392>>.  
Acesso em: 08 nov. 2024.

MEDEIROS, H. **Como implementar o padrão Decorator no Java EE**. Disponível em:  
<<https://www.devmedia.com.br/como-implementar-o-padrao-decorator-no-java-ee/32834>>. Acesso em: 8 nov. 2024.

**Padrão de projeto Singleton – como ele funciona em JavaScript, com código de exemplo.** Disponível em:  
<<https://www.freecodecamp.org/portuguese/news/padrao-de-projeto-singleton-como-ele-funciona-em-javascript-com-codigo-de-exemplo/>>. Acesso em: 8 nov. 2024.