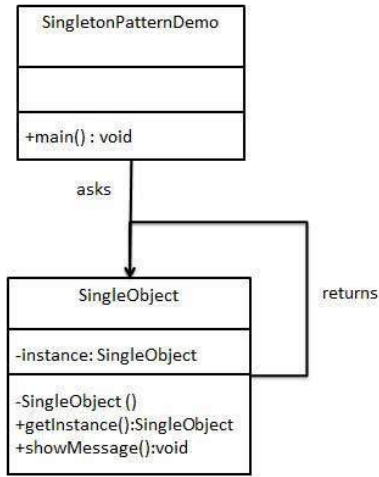


UM POUCO DE PADRÕES DE PROJETO

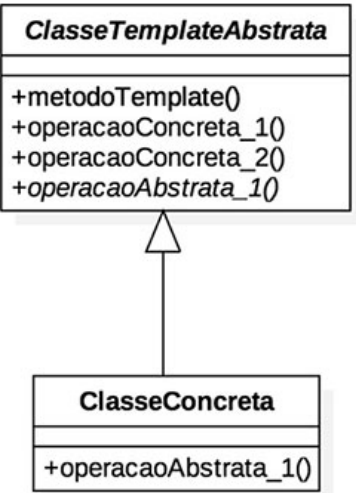
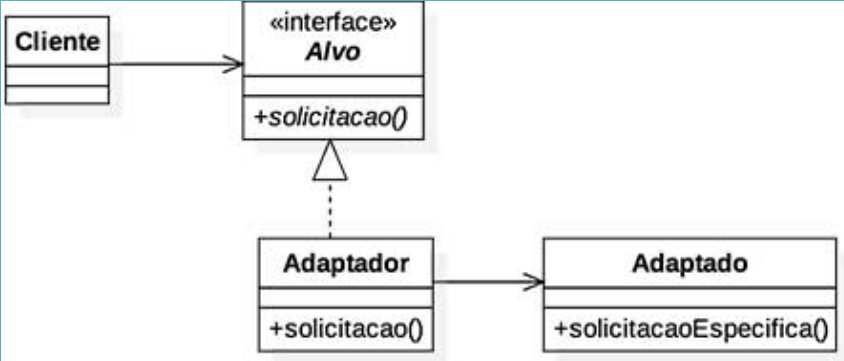


PATTERN SINGLETON

Garantir que um objeto terá apenas uma única instância, isto é, que uma classe irá gerar apenas um objeto e que este estará disponível de forma única para todo o escopo de uma aplicação. Algumas aplicações têm a necessidade de controlar o número de instâncias criadas de algumas classes, seja pela necessidade da própria lógica ou por motivos de performance e economia de recursos.

PATTERN ADAPTER

Converter a interface de uma classe por outra esperada pelos clientes . O que possibilita que classes com interfaces incompatíveis trabalhem em conjunto – ou que, de outra forma, seria impossível.Também conhecido comoWrapper(adaptador).

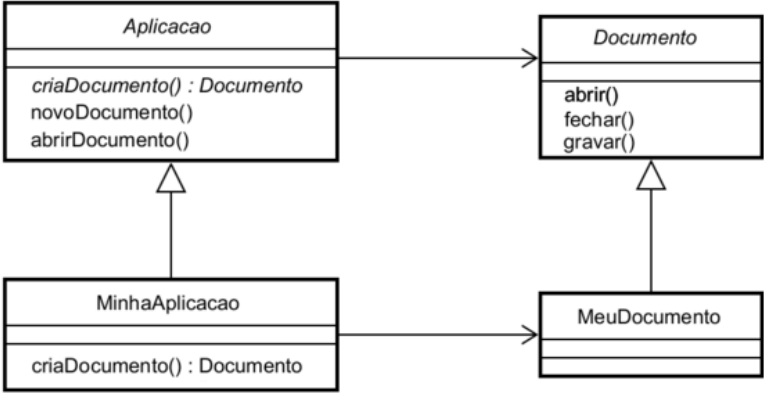
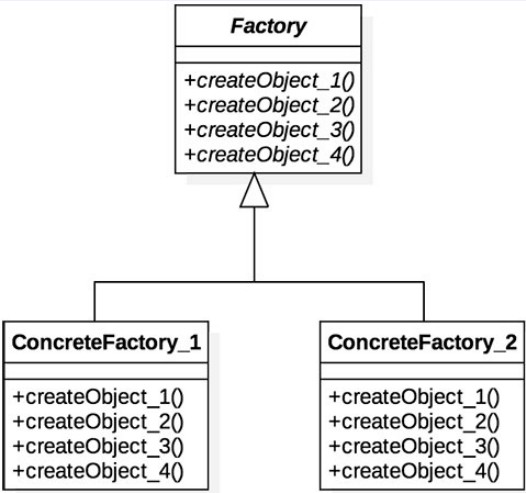


PATTERN TEMPLATE METHOD

Definir o esqueleto de um algoritmo em uma operação, postergando(deferring) alguns passos para subclasses. Template Method (gabarito de método) permite que subclasses redefinam certos passos de um algoritmo sem mudar a estrutura do mesmo.

PATTERN ABSTRACT FACTORY

Este padrão deve ser aplicado quando se deseja isolar a aplicação da implementação da classe concreta, que poderia ser um componente e ou framework específico no qual a aplicação conheceria apenas uma interface e a implementação concreta seria conhecida apenas em tempo de execução ou compilação.



PATTERN FACTORY METHOD

Definir uma interface para criar objetos, mas deixar que as subclasses decidem que classe instanciar. O Factory Method, também conhecido como construtor virtual, possibilita adiar a criação do objeto a subclasses .