# Padrões de design estruturais

### *Facade*

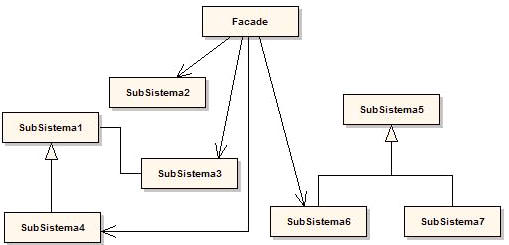
**Sobre Facade**

* O objetivo deste patrão é simplificar o sistema para o usuário, fornecendo uma interface unificada para vários subsistemas.
* Procura reduzir ao minimo a comunicação e as dependências entre os subsistemas.
* É por isso que é chamado de "fachada".

**Aplicações de Facade**

* É recomendável utilizar o Facade quando:
* Você quer fornecer uma interface simples para um subsistema complexo.
* Você deseja desacoplar um subsistema de seus usuários e de outros subsistemas, tornando-o mais independente e portátil.
* Você quer dividir os sistemas em níveis: as fachadas seriam o ponto de entrada para cada nível. Fachada pode ser usada no nível do aplicativo

**Diagrama UML de Facade**

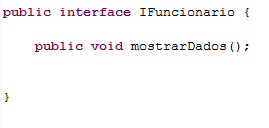


* Facade: num pedido ele conhece quais são os subsistemas responsáveis. Delega as petições dos clientes nos objetos do subsistema.
* Subsistema: Não tem nenhum conhecimento do Facade e só controla o trabalho dado pelo objeto Facade.

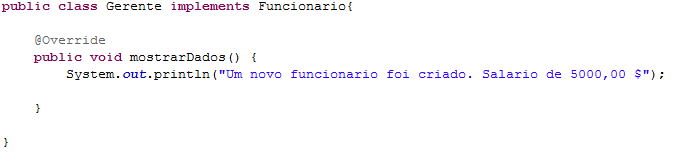
**Exemplo de Facade**

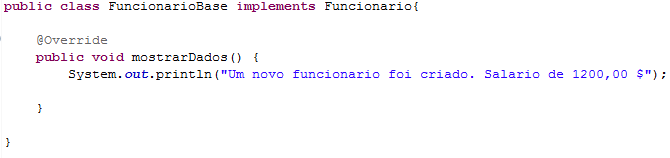
* Somos uma empresa que está desenvolvendo o software de contratações de uma sucursal de um banco. O banco está contratando porque tem funcionários que vão se aposentar e há vacantes.

Criamos as classes Funcionario, FuncionarioBase, Gerente.

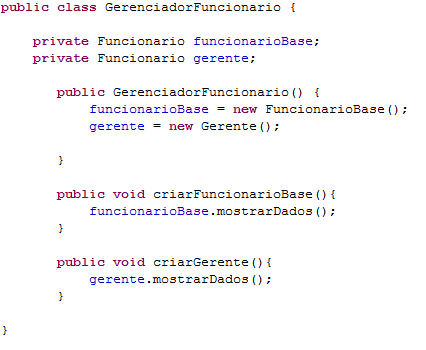


* Exemplo de métodos principais das diversas classes que tem o sistema:





* Agora faremos uma clase chamada GerenciadorFuncionario que será nosso Facade:



* Tudo bem, vamos ver 2 tipos de clientes. O primeiro não chamará ao Facade e o segundo sim. Veremos como o primeiro está obrigado a conhecer muitos detalhes dos subsistemas e o segundo não para o mesmo resultado no console.

