Padrões de Desenho

Padrões de desenho de *software*

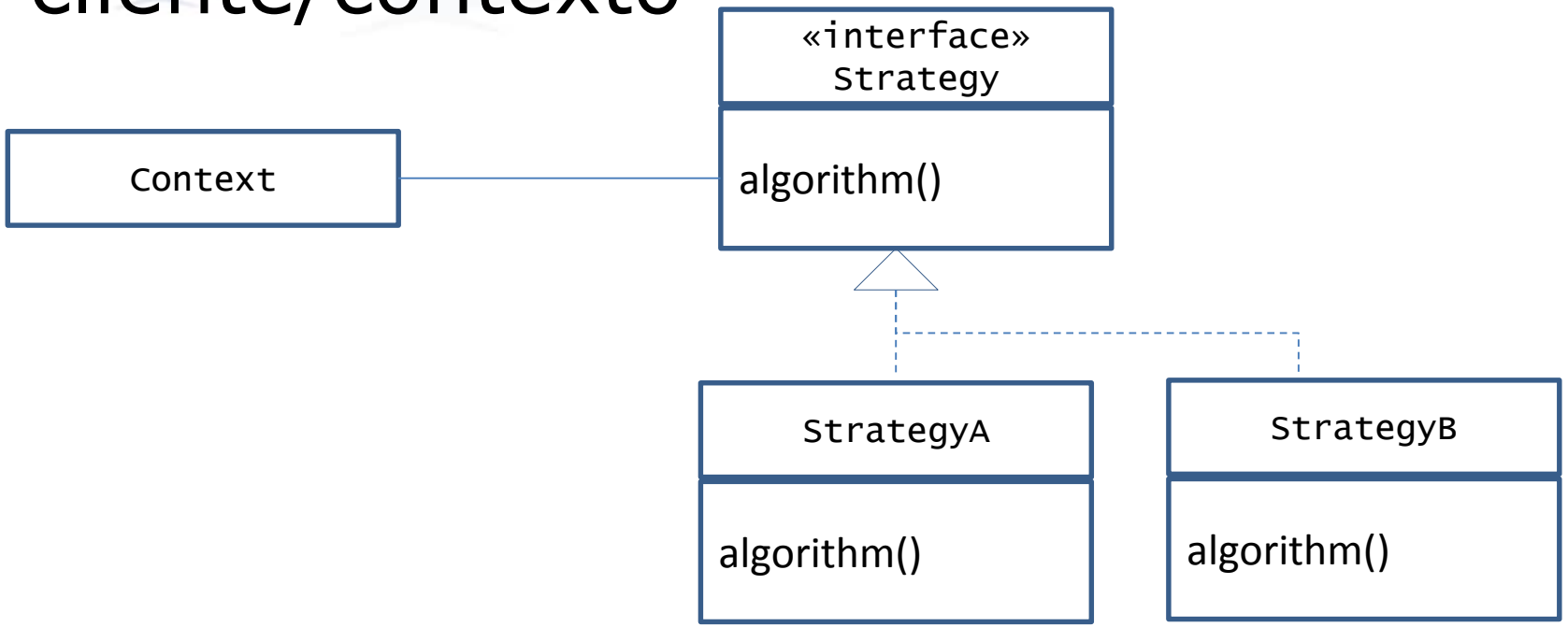
- Descreve um problema recorrente
- Captura a estrutura estática e dinâmica, assim como a colaboração entre os principais actores
- Categorias básicas:
  - "creational" -- "Simple" Factory, Factory Method, Abstract Factory, Singleton
  - "structural" -- Bridge, Composite, Proxy, ...
  - "behavioral" -- Command, Iterator, Strategy, Visitor, ...

Sumário

- Strategy
- Factory
- Command
- Template
- Singleton
- Proxy
- Bridge
- Flyweight
- Observer , Observed
- Façade
- Composite

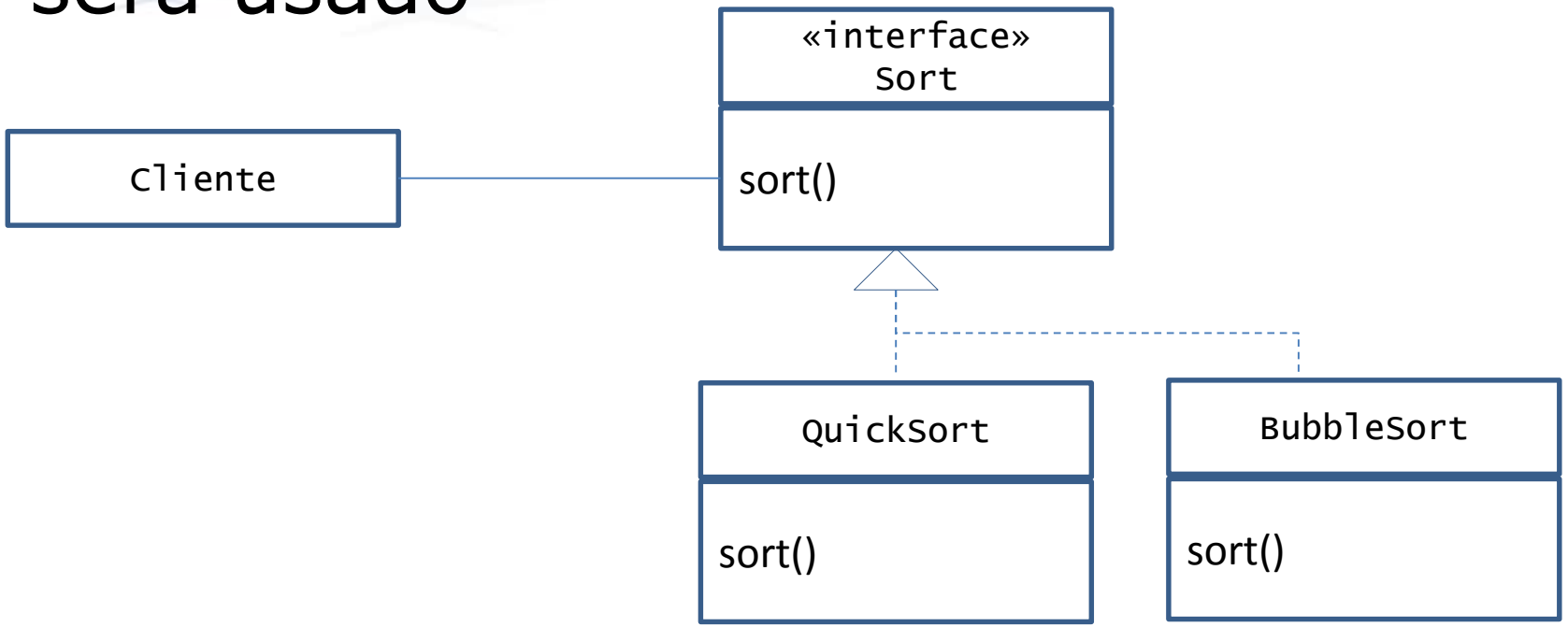
Estratégia (strategy/policy)

- Algoritmo concreto pode variar independentemente do cliente/contexto



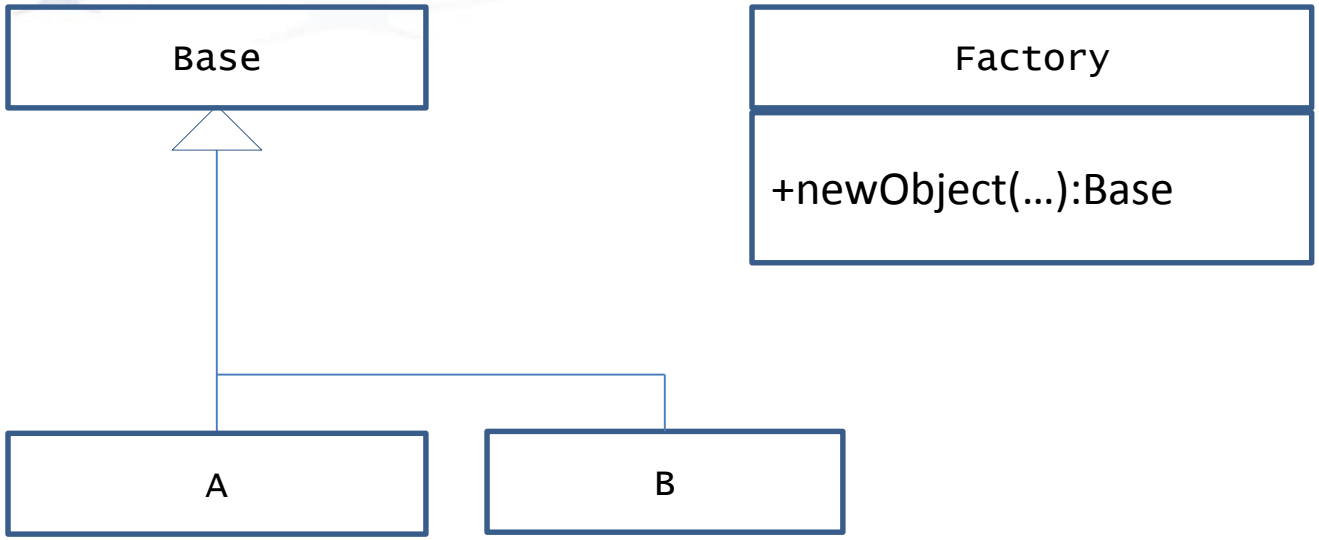
Exemplo: *strategy*

- O cliente chama sort() sem saber que tipo de algoritmo de ordenação será usado



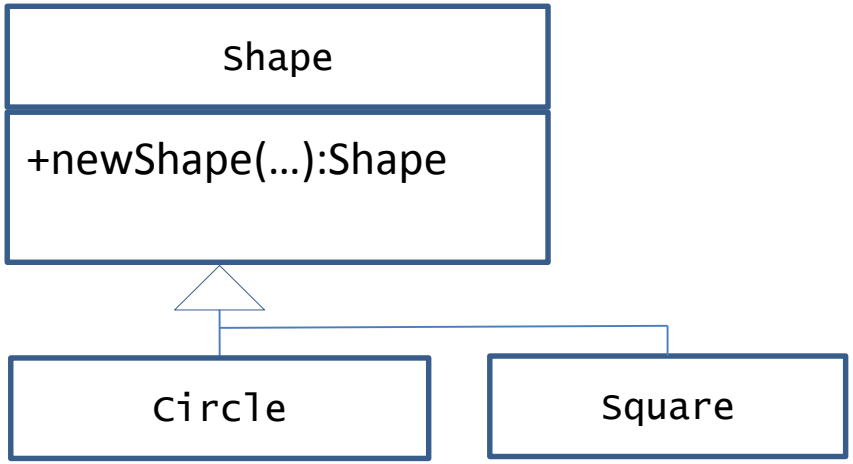
Factory

- Cria objeto de uma sub-classe

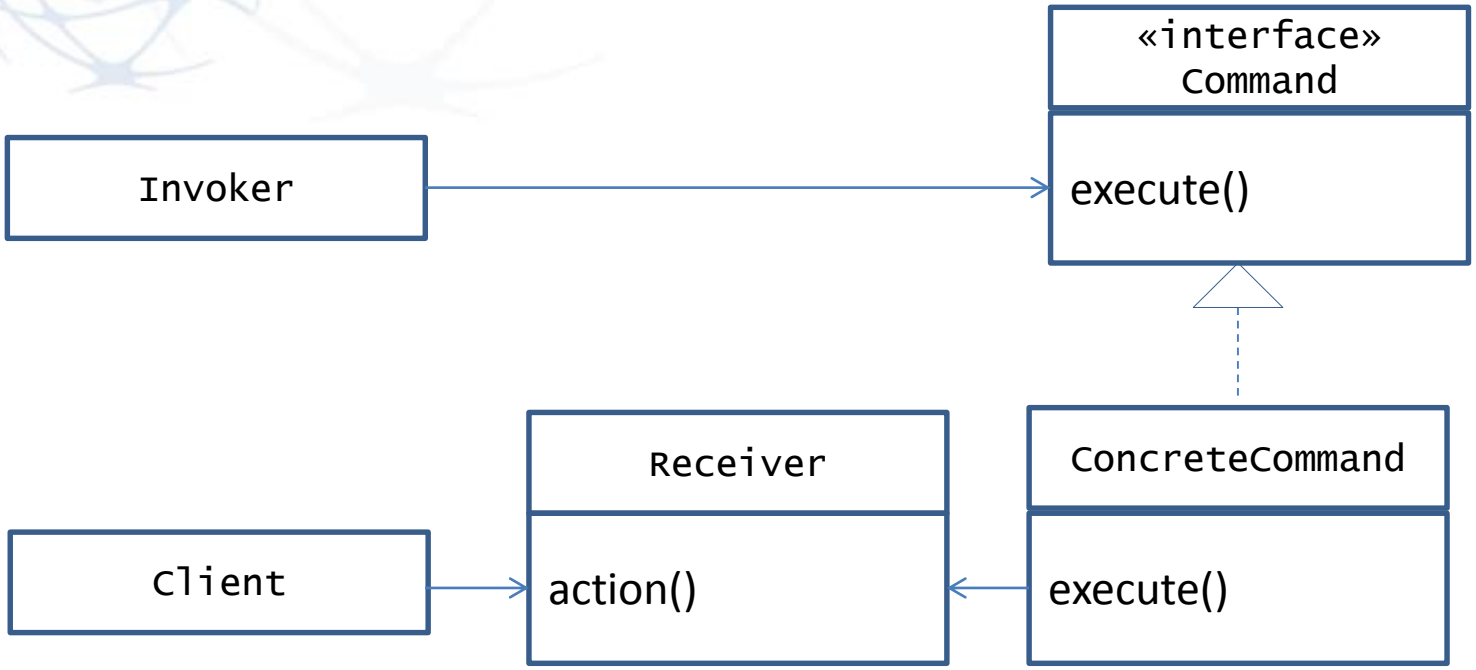


Factory

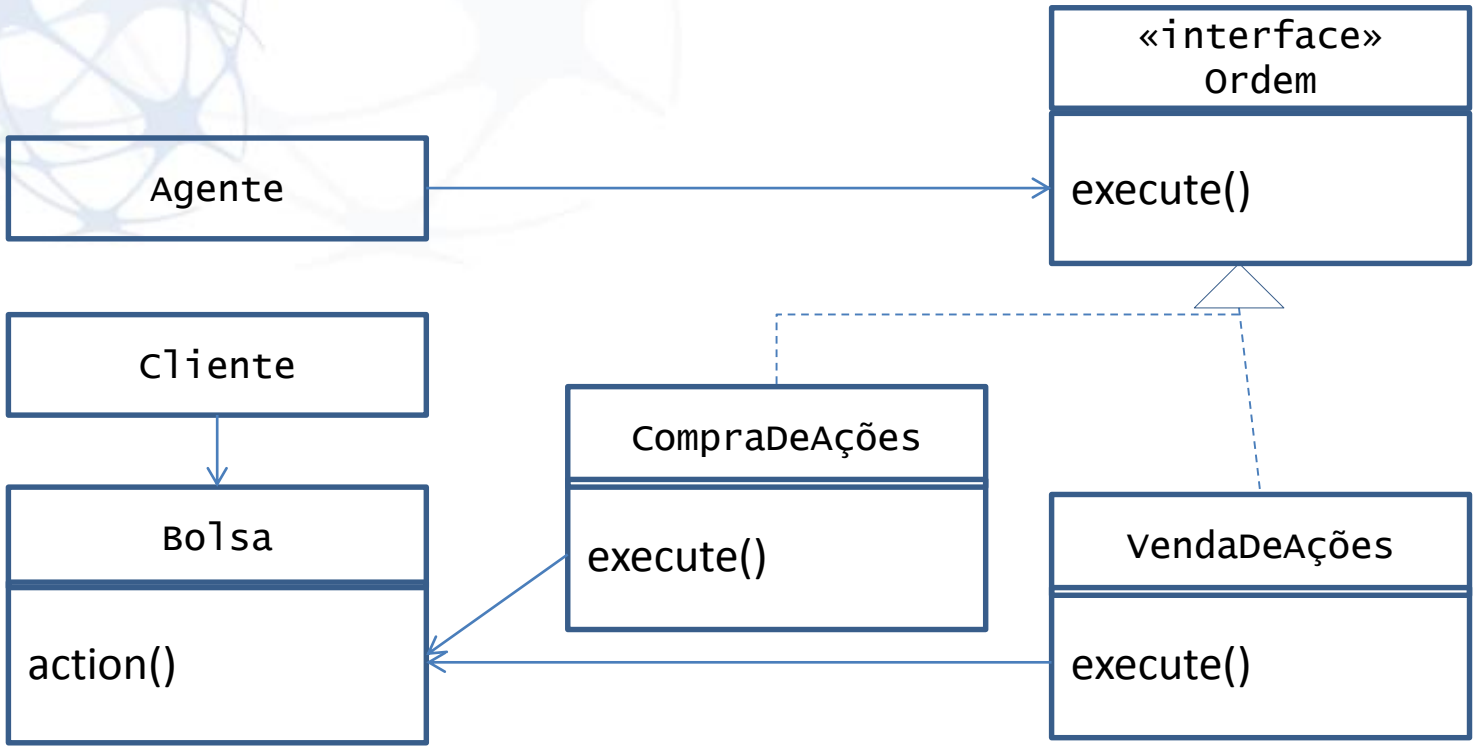
```
public static Shape newShape(String shapeName) {
  if (shapeName.equals("Square"))
     return new Square();

  …

}
```



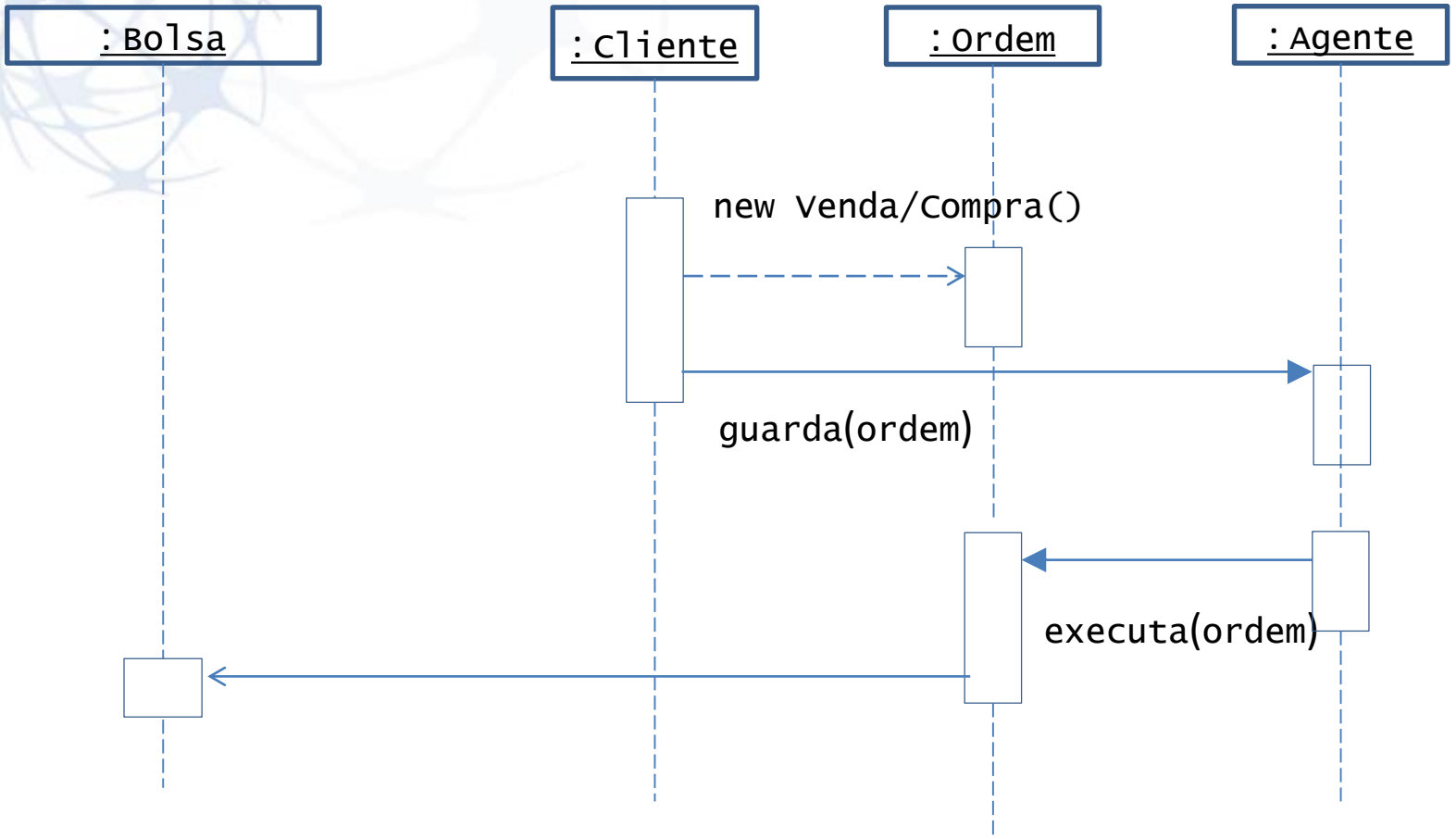
Comando (command)



Exemplo: *command*

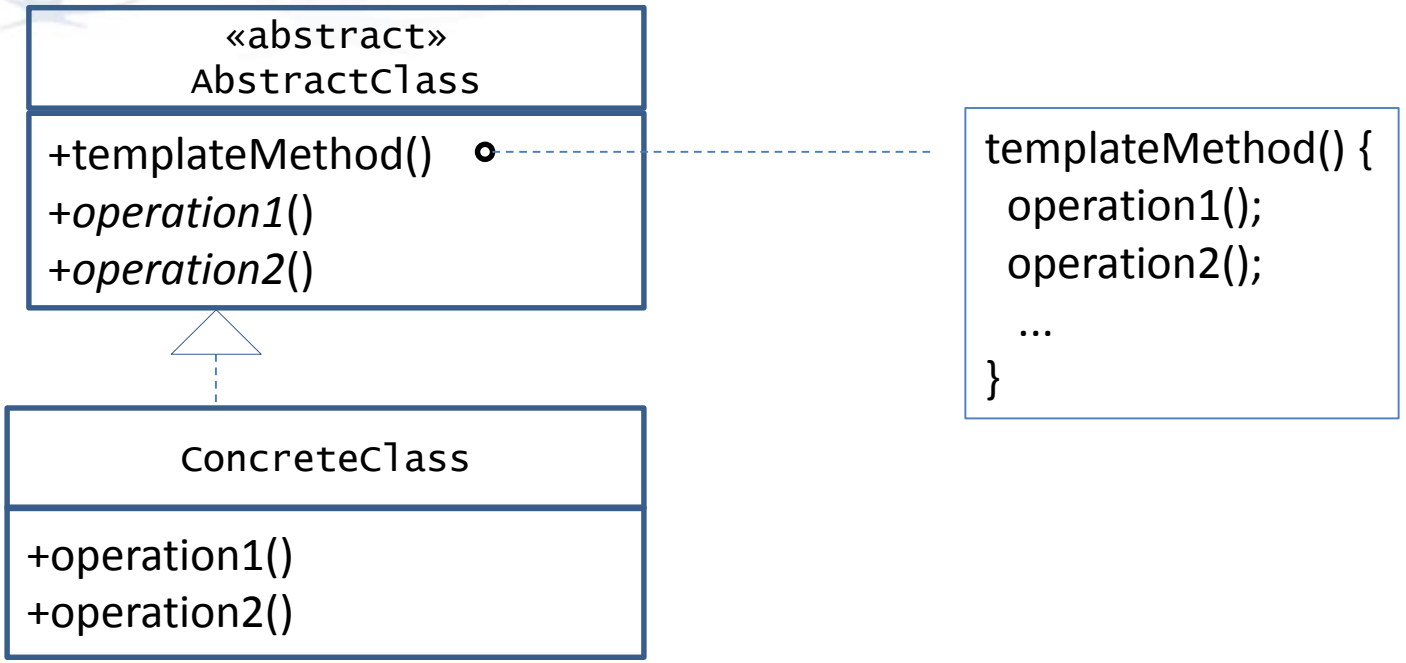


Exemplo: *command*



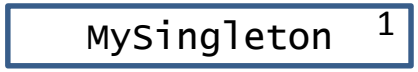
Método modelo (Template method)

- Define o esqueleto do algoritmo sem definir os detalhes



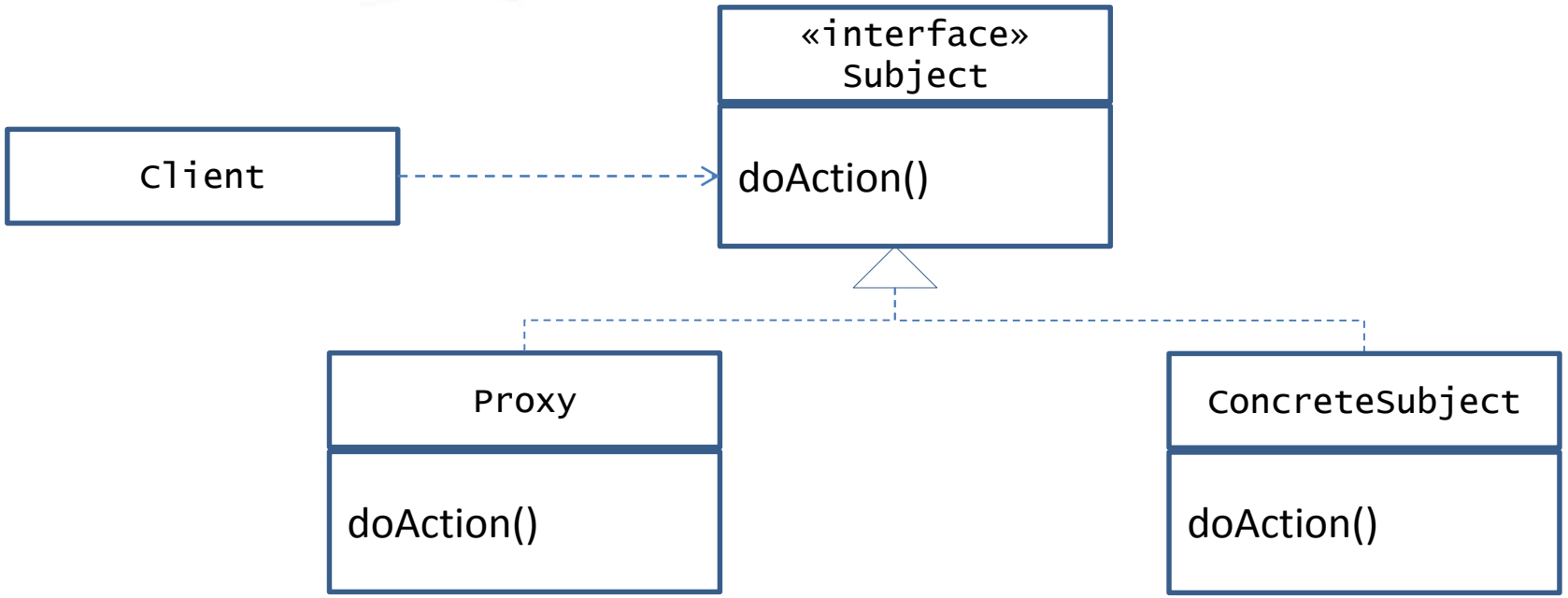
Solitão (*singleton*)

```
package mypackage;
public final class MySingleton {
    private static final MySingleton INSTANCE =
        new MySingleton();
    private MySingleton() {
        assert INSTANCE == null : …;
    }
    public static MySingleton getInstance() {
        return INSTANCE;
    }
}
```



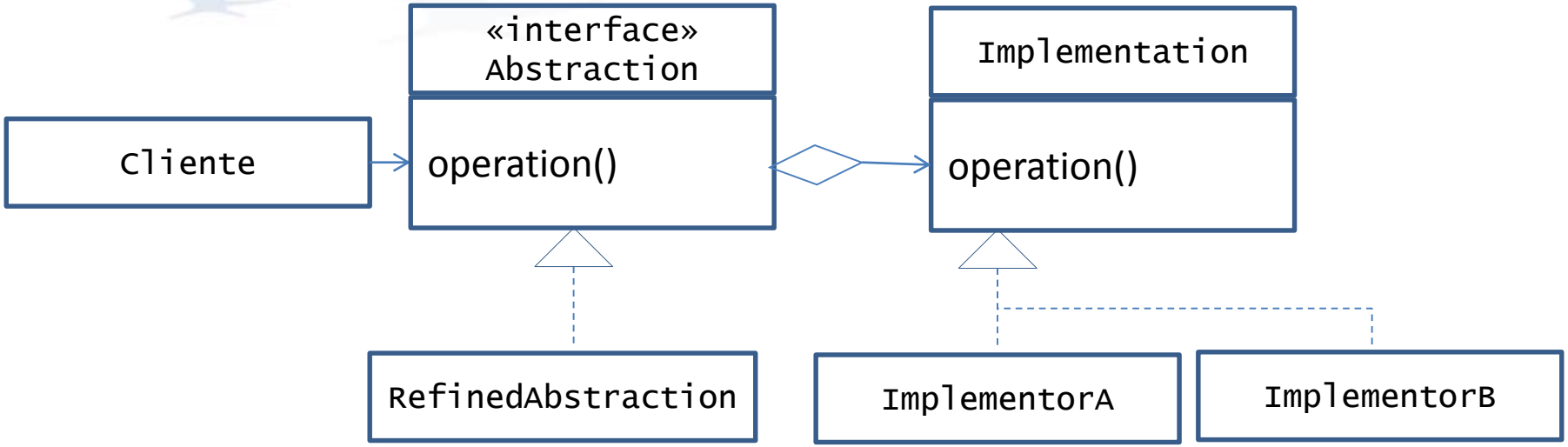
Representante (proxy)

- Uma classe funciona como representante de outra, passando os pedidos e devolvendo as respostas em seu nome, eventualmente limitando acesso a esta



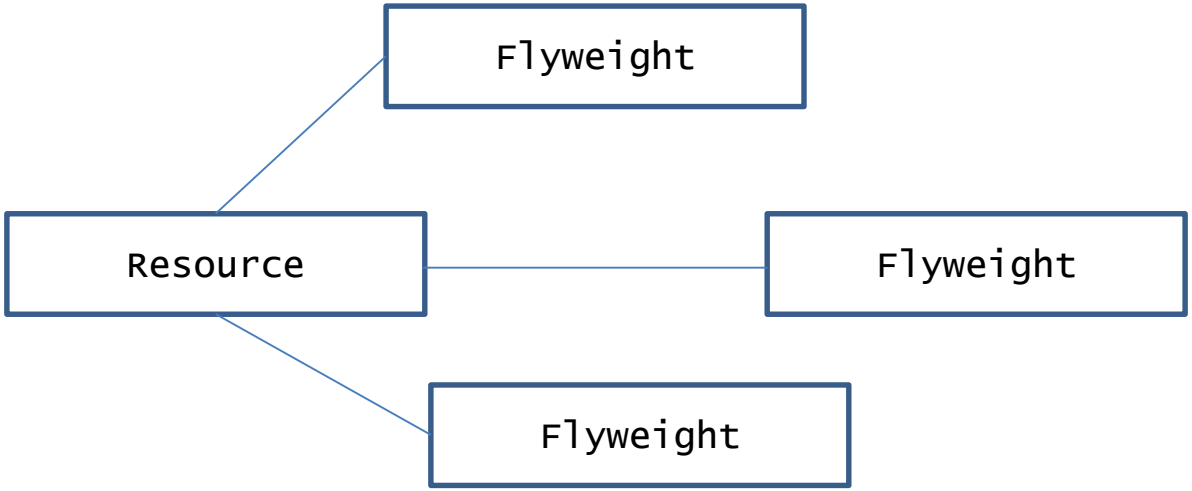
Ponte (bridge)

- Desacoplar a abstração da implementação



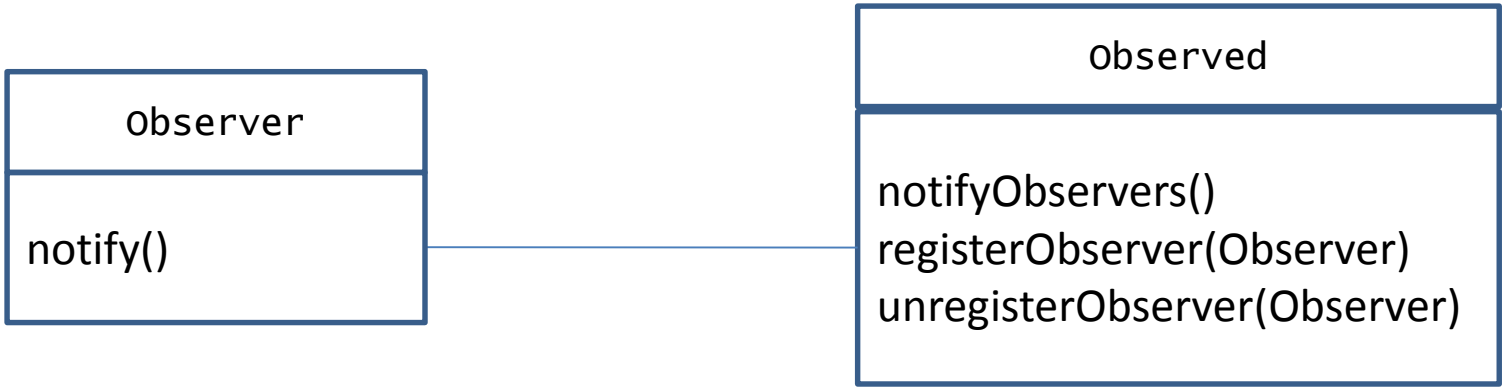
Peso-pluma (flyweight)

- Objeto que minimiza o consumo de memória partilhando recursos com outros da mesma classe

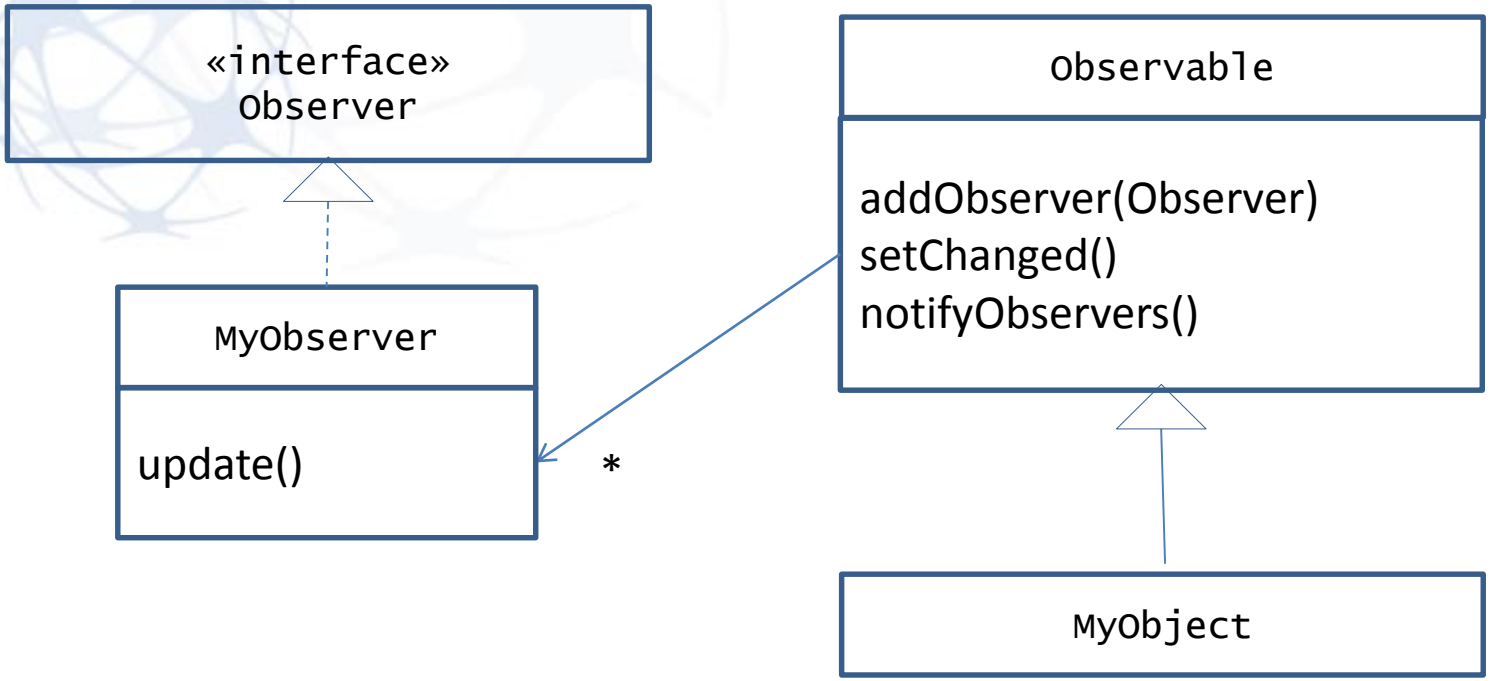


Observador - Observado (Observer - Observed)

- Observado mantém uma lista de observadores que avisa quando há uma atualização

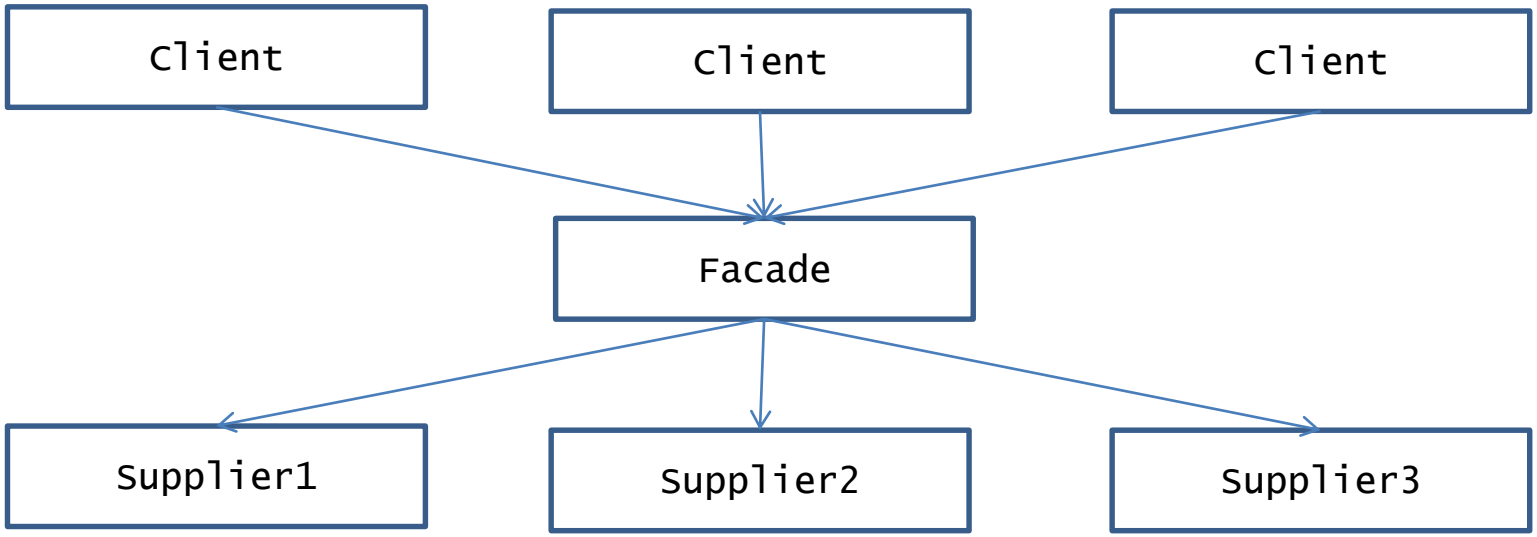


Observador – Observado (Java)



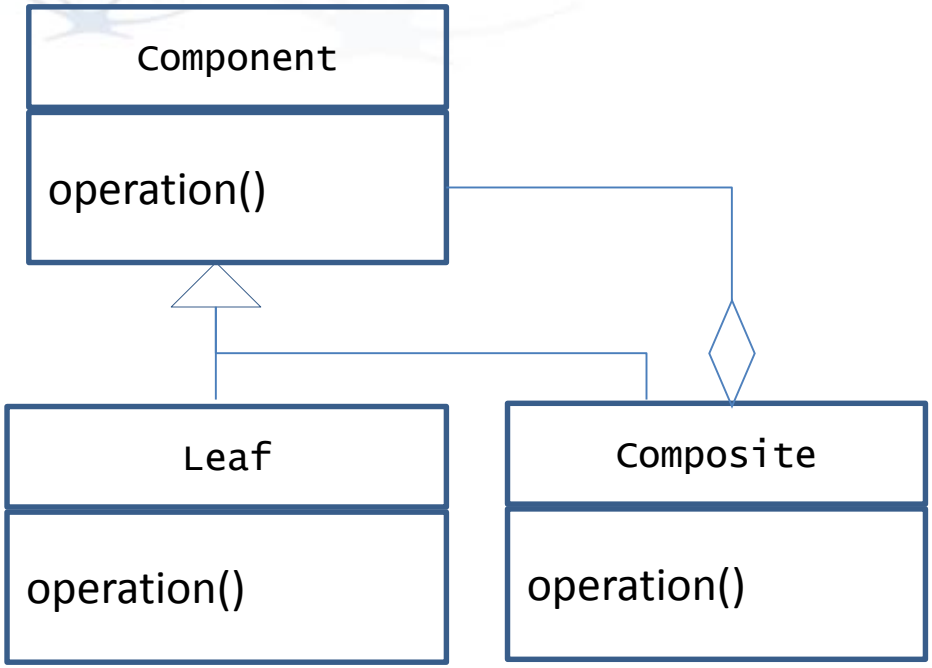
Fachada (Façade)

- Oferece um interface simplificado para um conjunto de código



Composto (Composite)

- Grupo de objetos que podem ser tratados como uma única instância de um objeto



Mais informação / Referências

- Y. Daniel Liang, *Introduction to Java Programming*, 7.ª edição, Prentice-Hall, 2008.
- Gamma, Helm, Johnson & Vlissides, *Design Patterns*. Addison-Wesley. ISBN 0-201-63361-2, 1994.
- Eric Freeman, Elisabeth Robson, Bert Bates, Kathy Sierra, *Head First Design Patterns*, O'Reilly Media, October 2004