

Padrões de Projeto

PADRÕES GOF

Dr^a. ALANA MORAIS

2019

A solid orange horizontal bar spanning the entire width of the slide at the bottom.

Aula Passada

- Bridge
- Façade

Aula Passada

- Fizeram o exercício sobre o Front Controller?



Padrões GoF – Padrões de Estrutura

		Propósito		
		1. Criação	2. Estrutura	3. Comportamento
Escopo	Classe	Factory Method	Class Adapter	Interpreter Template Method
	Objeto	Abstract Factory Builder Prototype Singleton	Object Adapter Bridge Composite Decorator Facade Flyweight Proxy	Chain of Responsibility Command Iterator Mediator Memento Observer State Strategy Visitor

Flyweight

USAR COMPARTILHAMENTO PARA SUPORTAR GRANDES QUANTIDADES DE OBJETOS REFINADOS EFICIENTEMENTE.

Flyweight

Implantar compartilhamento de objetos de granularidade muito pequena para dar suporte ao uso eficiente de grande quantidade deles.

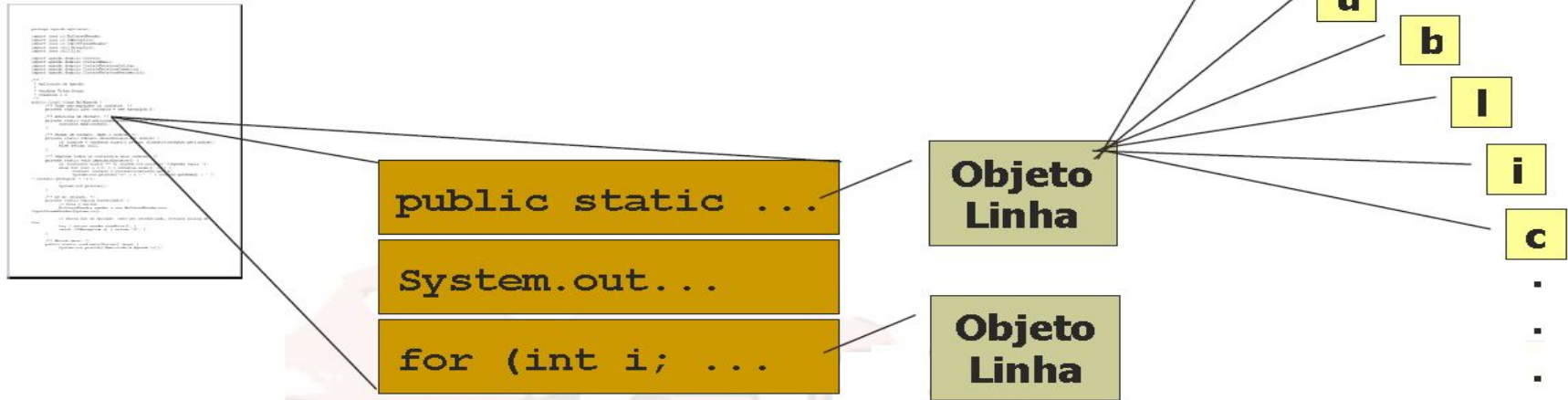
Usar quando:

- A aplicação usa um grande número de objetos;
- O custo de armazenamento é alto por causa desta quantidade;
- O estado dos objetos pode ser externalizado;
- Objetos podem ser compartilhados assim que seu estado é externalizado.

Flyweight Problema

Desenvolver um editor de texto onde cada caractere é representado por um objeto:

- Granularidade muito pequena;
- Não haverá recursos (memória) suficiente para textos grandes.

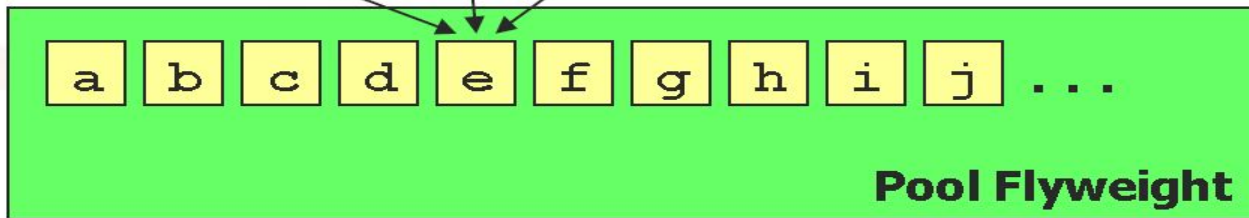


Flyweight Solução

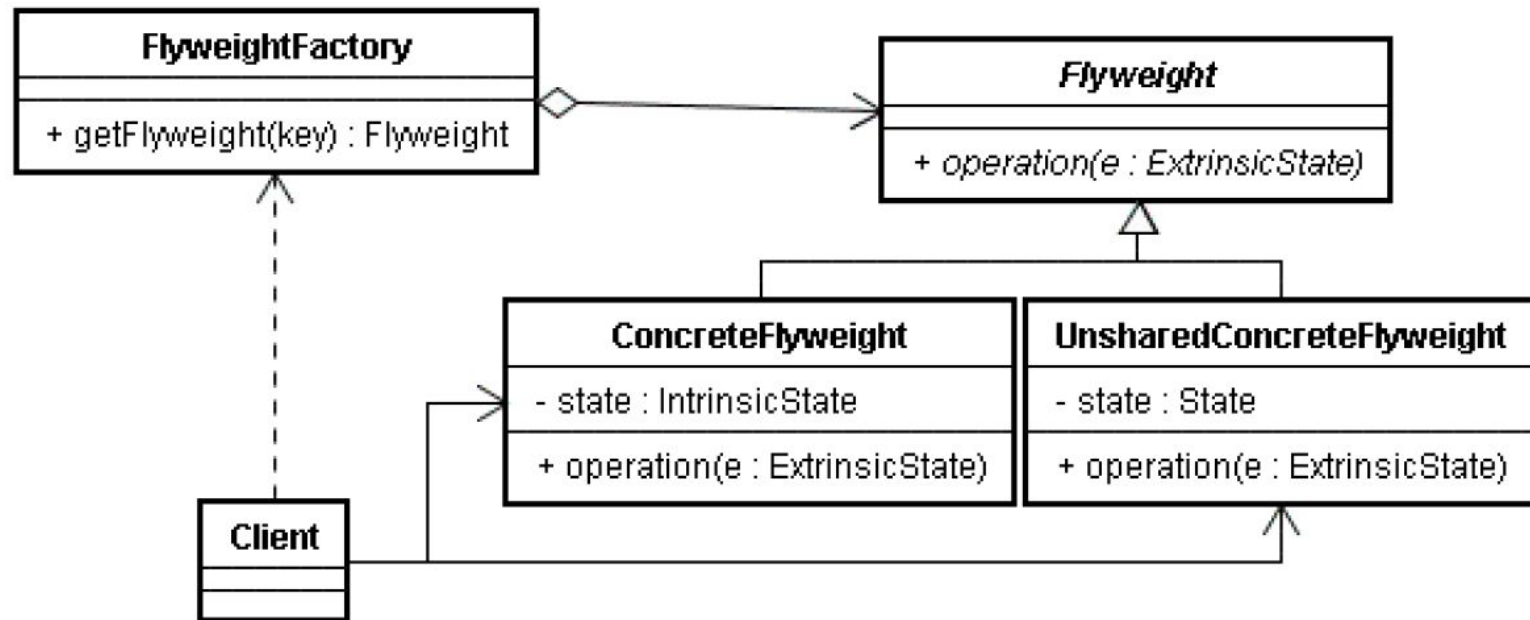
Monta-se um pool de objetos compartilhados;

- Cada caractere tem um objeto.
- Com 100 objetos (tabela ASCII) poderíamos montar textos de qualquer tamanho.

```
System.out.println("Design Patterns");
```



Flyweight Estrutura



Flyweight Exemplo

- No desenvolvimento de jogos são utilizadas várias imagens.
 - Elas representam as entidades que compõem o jogo, por exemplo, cenários, jogadores, inimigos, entre outros.
- Ao criar classes que representam estas entidades, é necessário vincular a elas um conjunto de imagens, que representam as animações.
 - Técnica denominada Sprites



Flyweight

Prós e Contras

Flyweight é uma solução para construção de aplicações usando objetos imutáveis

- Ideal para objetos que oferecem serviços (guardados em caches e em pools)
- Ideal para objetos que podem ser usados para construir outros objetos

Problemas

- Possível impacto na performance (se houver muitas representações diferentes, elas não podem ser alteradas, e é preciso criar muitos objetos)

Proxy

PROVER UM SUBSTITUTO OU PONTO POR ONDE UM OBJETO POSSA CONTROLAR O ACESSO A OUTRO.

Proxy

Intenção:

- Prover um representante ou ponto de acesso que controle ao acesso de um objeto.
- Prover um acesso mais versátil a um objeto do que um ponteiro:
 - Remote proxy (acesso remoto);
 - Virtual proxy (exemplo da imagem);
 - Protection proxy (controla acesso)
 - Smart Proxy

Também conhecido como:

- Surrogate.

Tipos de Proxy

Protection Proxy

- Ele controla o acesso aos objetos, por exemplo, verificando se quem chama possui a devida permissão.

Virtual Proxy

- O objeto real é criado apenas quando um cliente requerer ou acessar o objeto.

Tipos de Proxy

Remote Proxy

- Fornece um representante local para um objeto em outro espaço de endereçamento. Por exemplo, precisamos codificar todas as solicitações enviadas ao banco, utilizaríamos um Remote Proxy que codificaria a solicitação e só então faria o envio.

Smart Reference

- Este proxy é apenas um substituto simples para executar ações adicionais quando o objeto é acessado, por exemplo para implementar mecanismos de sincronização de acesso ao objeto original.

Proxy Problema

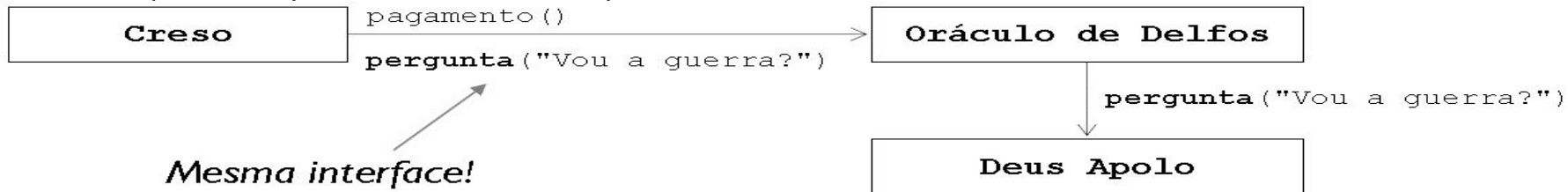
Sistema quer utilizar objeto real



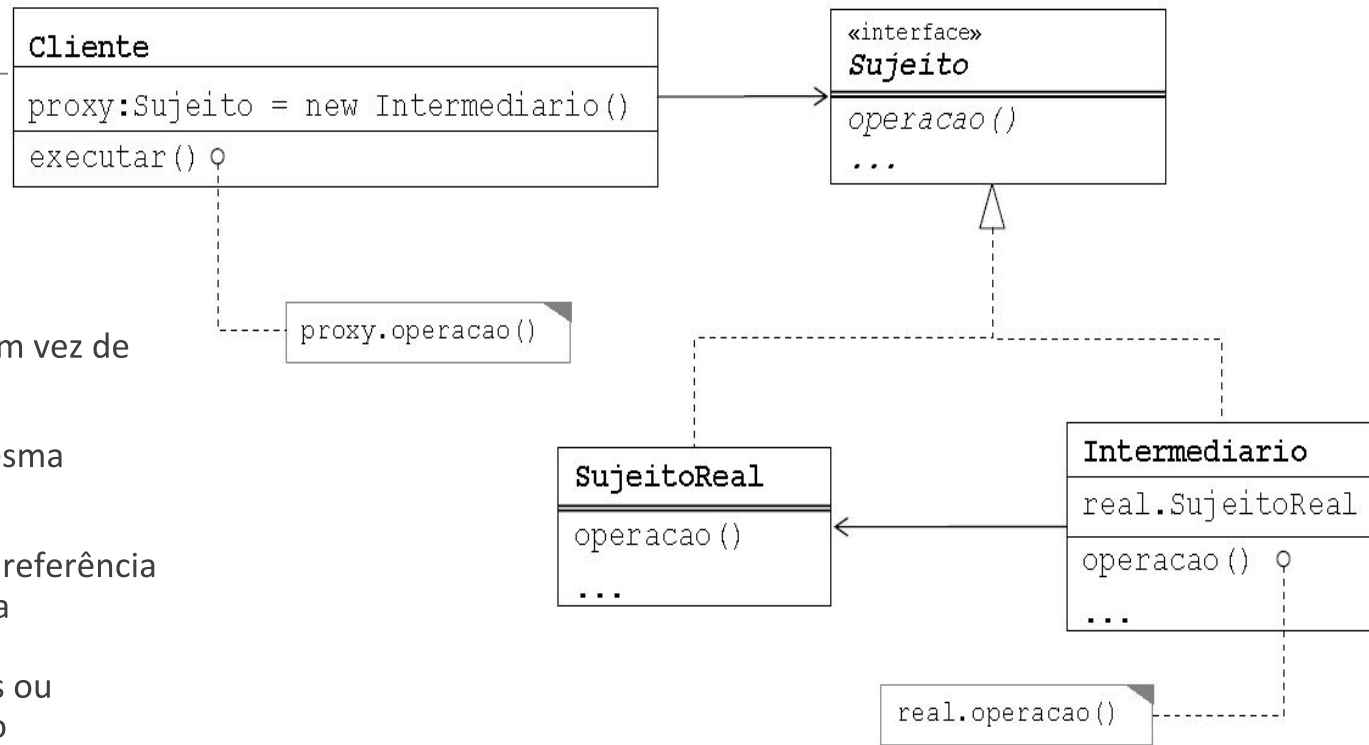
Mas ele não está disponível (remoto, inacessível, ...)



Solução: arranjar um intermediário que saiba se comunicar com ele eficientemente



Proxy Solução



Cliente usa intermediário em vez de
sujeito real

Intermediário suporta a mesma
interface que sujeito real

Intermediário contém uma referência
para o sujeito real e repassa
chamadas, possivelmente,
acrescentando informações ou
filtrando dados no processo

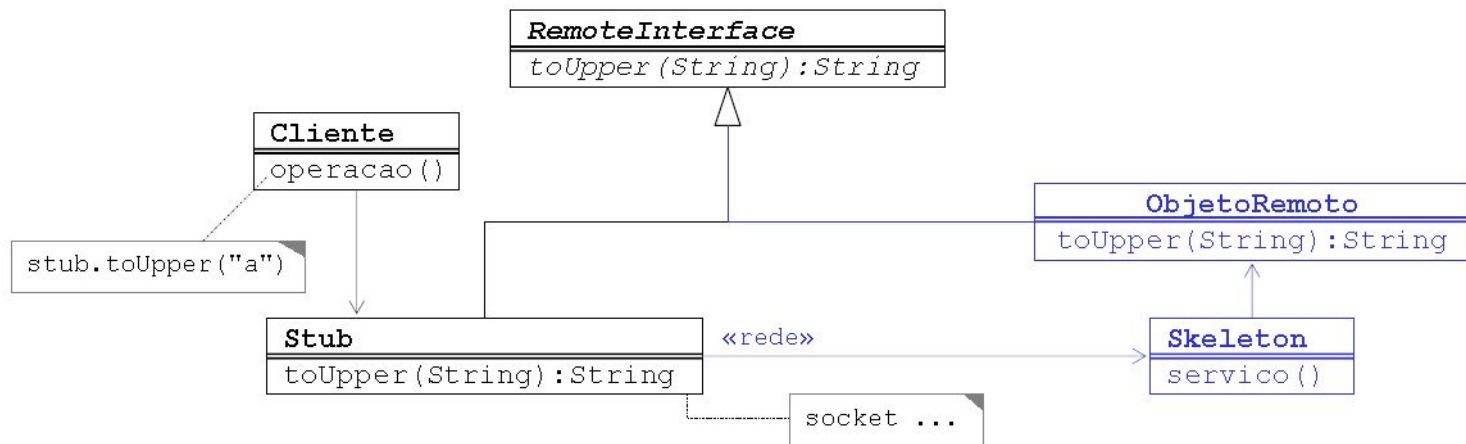
Proxy

Exemplos

A aplicação é mais comum em objetos distribuídos

Exemplo: RMI (e EJB)

- O Stub é proxy do cliente para o objeto remoto
- O Skeleton é parte do proxy: cliente remoto chamado pelo Stub



Vantagens e Desvantagens

- Otimização de algumas operações do seu projeto(cache de objetos)
- Proteção de acesso ao objeto
- Fatores externos podem causar impacto
- Conhecer bem seu comportamento para decidir quando utilizá-lo ou não

Dúvidas?

ALANAMM.PROF@GMAIL.COM

