

Métodos Avançados de Programação

PADRÕES GOF

Dr^a. ALANA MORAIS

Aula Passada

- Composite
- Decorator



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.

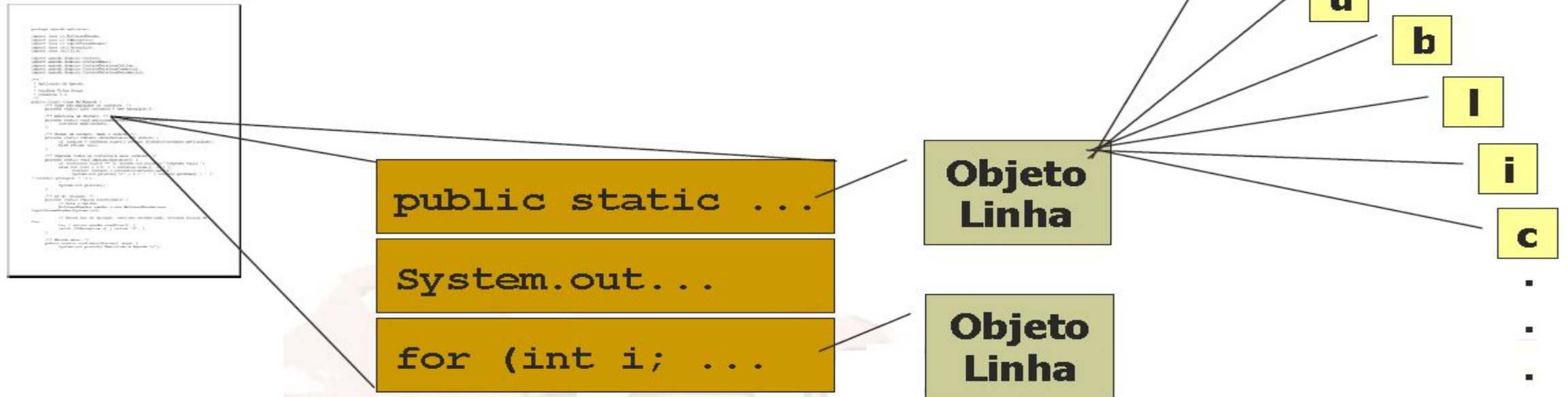
Usar quando:

- 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:

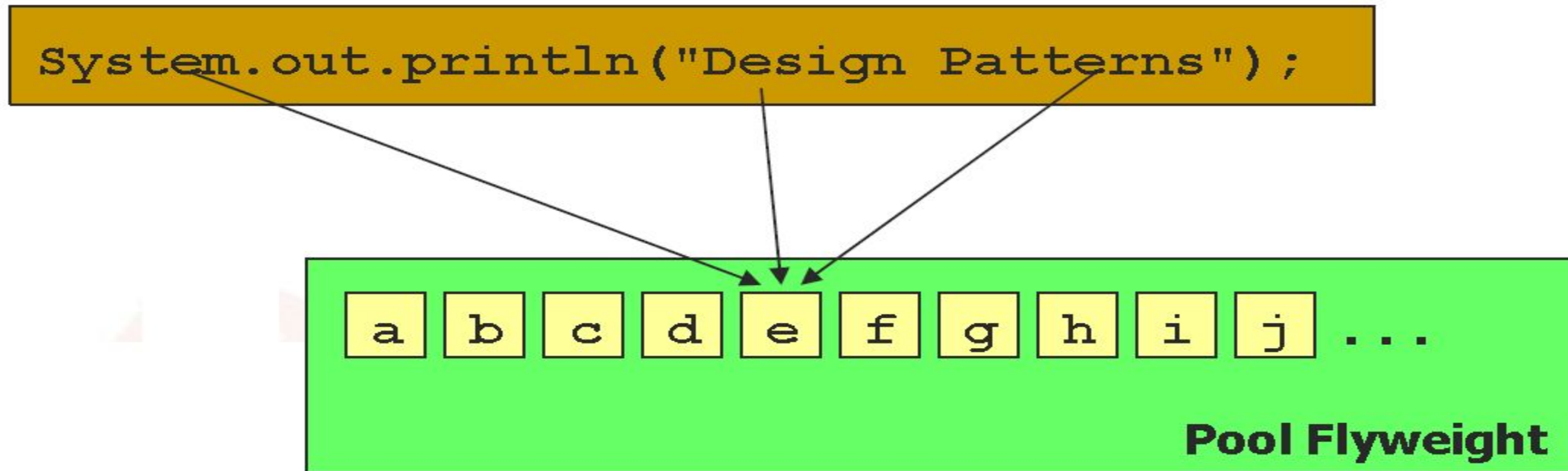
- Granularidade muito pequena;
- Não haverá recursos (memória) suficiente para textos grandes.



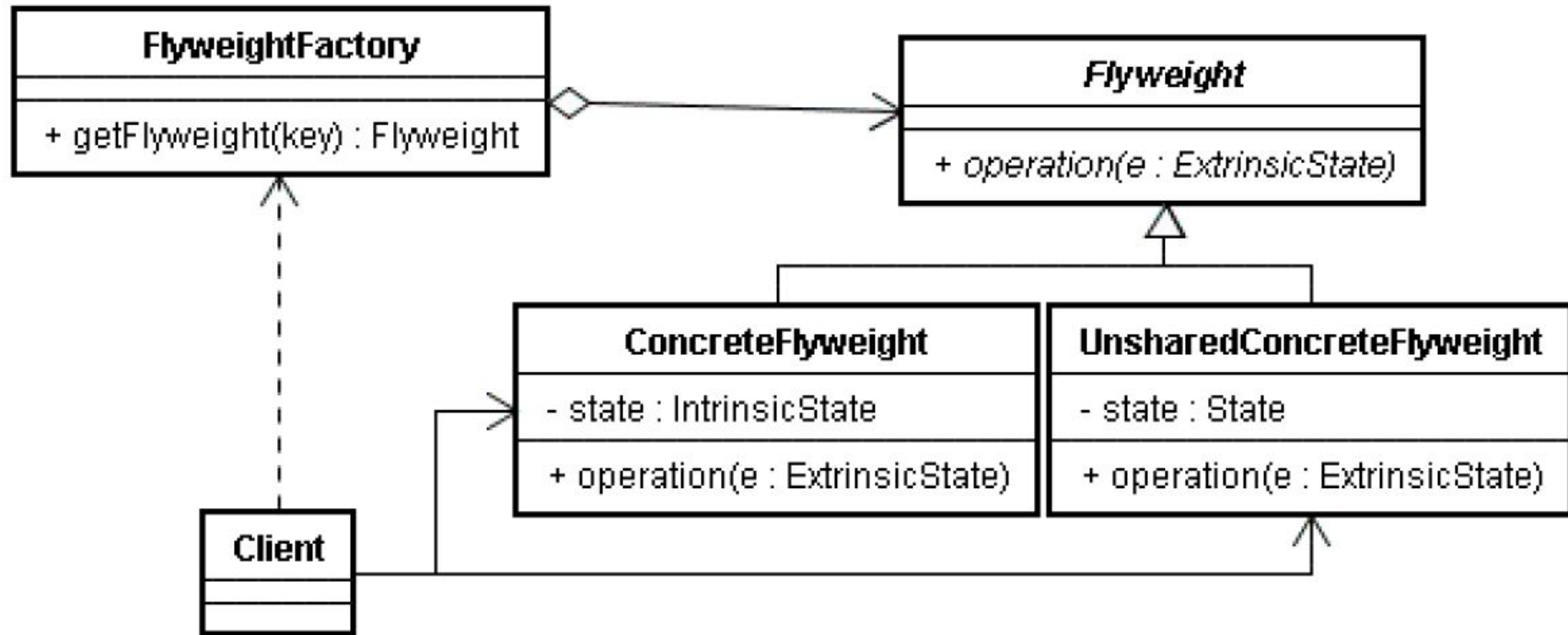
Flyweight Solução

Monta-se um pool de objetos compartilhados;

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



Flyweight Estrutura



Flyweight Exemplo

No desenvolvimento de jogos são utilizadas várias imagens.

Elas representam as entidades que compõe o jogo, por exemplo, cenários, jogadores, inimigos, entre outros.

Utilizando técnica de Sprites

Ao criar classes que representam estas entidades, é necessário vincular a elas um conjunto de imagens, que representam as animações.



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)

Também conhecido como:

- Surrogate.

Proxy Problema

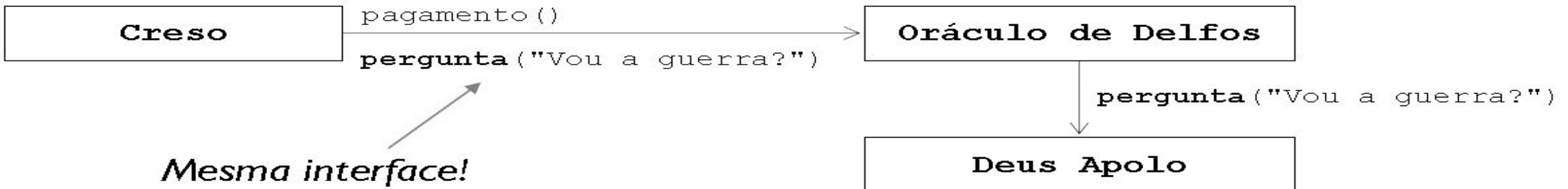
Sistema quer utilizar objeto real



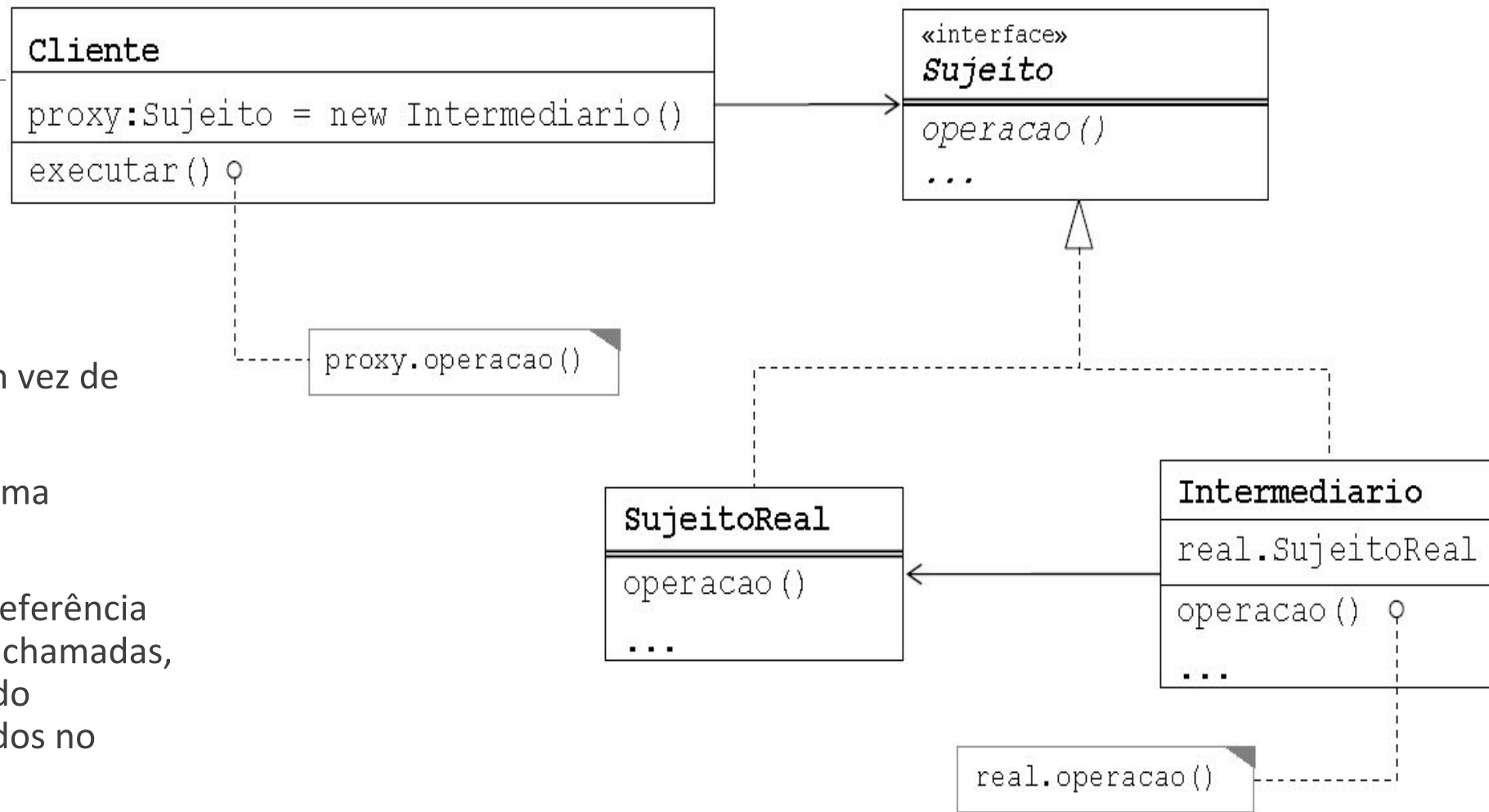
Mas ele não está disponível (remoto, inacessível, ...)



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

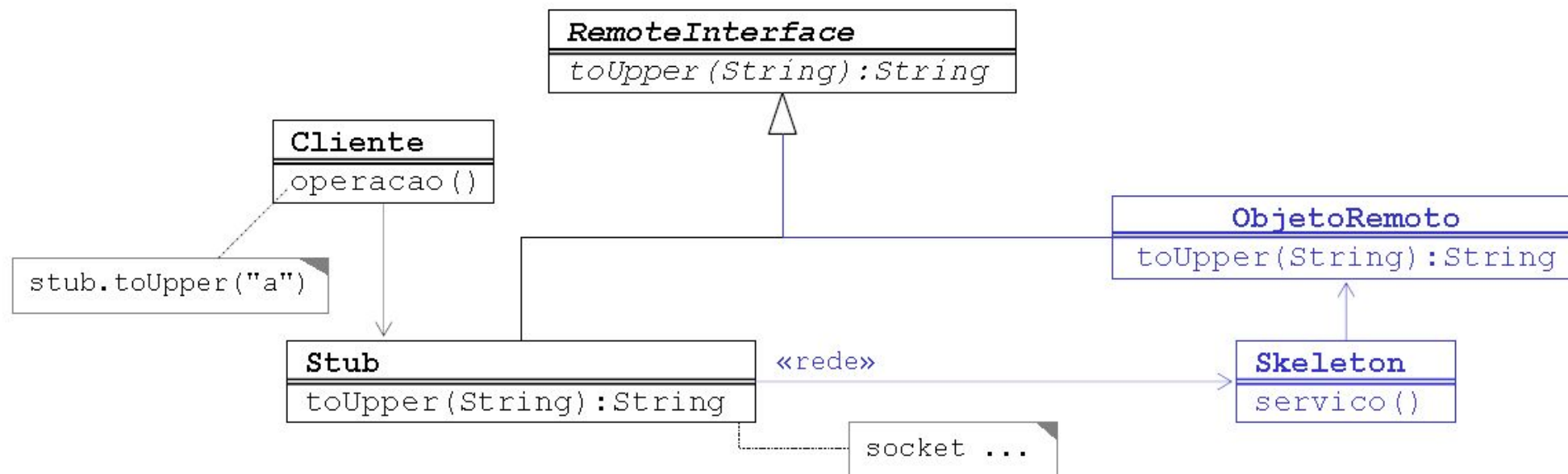
Intermediário contém uma referência para o sujeito real e repassa chamadas, possivelmente, acrescentando informações ou filtrando dados no processo

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



Dúvidas?

ALANAMM.PROF@GMAIL.COM