

Arquitetura de Software

Padrões de Projetos – Padrões Estruturais

Nairon Neri Silva

Sumário

Padrões de Projeto Estruturais

1. *Adapter*
2. *Bridge*
3. *Composite*
4. *Decorator*
5. *Façade*
6. *Flyweight*
- 7. *Proxy***

Proxy

Também conhecido como ***Surrogate***

Intenção

- Fornece um substituto (*surrogate*) ou marcador da localização de outro objeto para controlar o acesso ao mesmo

Motivação

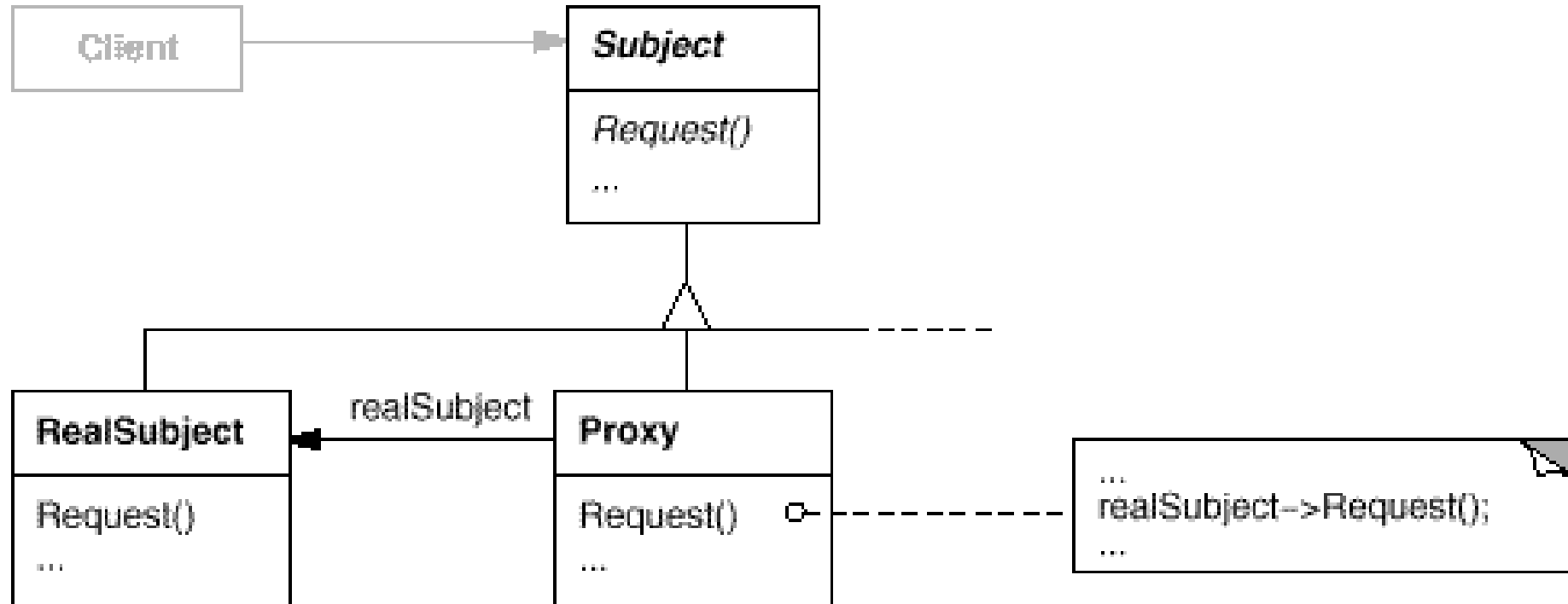
- Uma razão para controlar o acesso a um objeto é adiar o custo integral de sua criação e inicialização até o momento em que realmente necessitamos usá-lo
- Exemplo: considere um editor de documentos que pode embutir objetos gráficos num documento
 - Tais como grandes imagens – podem demorar muito para serem criados
 - A abertura de um documento deveria ser rápida – deveríamos evitar a criação, de uma só vez, de todos os objetos necessários (principalmente os mais complexos)

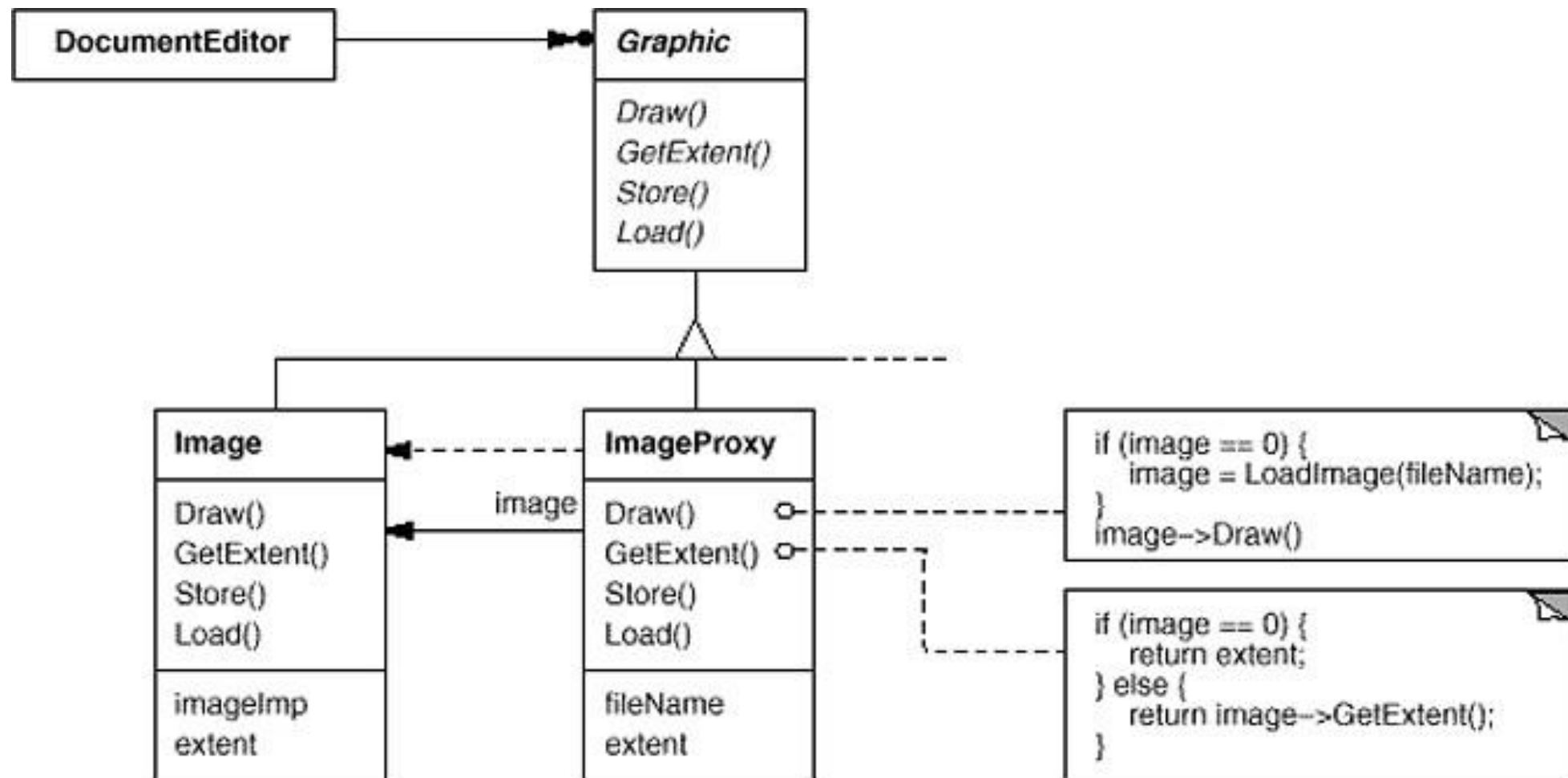
Motivação

- Continuando...
 - De qualquer forma, isto não é necessário porque nem todos estes objetos estarão visíveis o documento ao mesmo tempo
 - Estas restrições sugerem a criação de tais objetos complexos **sob demanda** – quando estes estarão visíveis
 - Mas, o que colocamos no documento no lugar da imagem?
 - Como podemos ocultar o fato de que o objeto é criado sob demanda de maneira a não complicar a implementação do editor?

Aplicabilidade

- Você quer uma ligação entre seu sistema e um sistema remoto (**proxy remoto**).
- Você tem um objeto caro para ser criado e não quer permitir acesso direto a esse objeto (**proxy virtual**).
- Você quer restringir acesso a partes da sua aplicação (**proxy de proteção**).
- Você quer fazer cache de chamadas já realizadas (**proxy inteligente** ou *smart reference*).
- Você quer interceptar quaisquer chamadas de métodos ao objeto real por qualquer motivo.





Exemplo/Participantes

1. *Proxy (ImageProxy)*

- Mantém uma referência que permite ao **Proxy** acessar o objeto real
- Fornece uma interface idêntica a de **Subject**, de modo que o **Proxy** possa substituir o objeto real
- Controla o acesso ao objeto real – pode criar e destruir
- Outras responsabilidades dependem do tipo de Proxy:
 - **Remote Proxy** → montar uma solicitação com os devidos argumentos, e enviar a mesma através da rede destinada ao objeto real
 - **Virtual Proxy** → armazenar informações adicionais sobre o objeto real, de maneira que possam postergar o acesso ao mesmo
 - **Protection Proxy** → verificar se quem chama tem as permissões de acesso requeridas para executar uma consulta

Exemplo/Participantes

2. *Subject (Graphic)*

- Define uma interface comum para *RealSubject* e *Proxy*, de maneira que um *Proxy* possa ser usado como *RealSubject*

3. *RealSubject (Image)*

- Define o objeto real que o *Proxy* representa

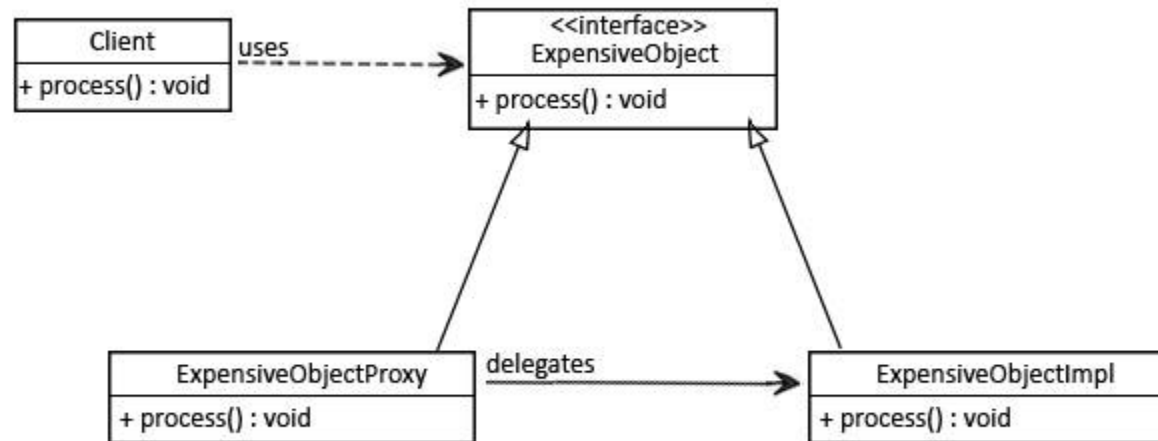
Consequências

1. Introduce um nível de referência indireta para acesso a um objeto
 - a) Um *proxy remoto* oculta o fato de que um objeto reside num espaço de endereçamento diferente
 - b) Um *proxy virtual* pode executar otimizações
 - c) Tanto *proxies de proteção* como *smart references* permitem tarefas adicionais de organização quando um objeto é acessado
2. O código cliente não precisa saber se está ou não usando um proxy (o proxy se passa pelo objeto real)
3. É possível adicionar novos proxies sem mudar o código do objeto real
4. O proxy funciona mesmo se o objeto real não estiver acessível ou pronto para uso

Exemplo prático:

Exemplo presente no site: <https://www.baeldung.com/java-proxy-pattern>

1. Considere uma situação em que temos um objeto Java muito pesado que requer alguma configuração inicial
2. Queremos que esse objeto seja inicializado sob demanda e, uma vez que seja inicializado, queremos reutilizá-lo para todas as chamadas



Saiba mais...

- <https://www.youtube.com/watch?v=EsxPyICeBPs>
- <https://www.baeldung.com/java-proxy-pattern>
- <https://refactoring.guru/pt-br/design-patterns/proxy>

Acesse os endereços e veja mais detalhes sobre o padrão Proxy

Atividade

- 1) Dentre os 7 padrões estruturais, escolha 4 e explique de forma resumida o objetivo e a aplicabilidade de cada um deles.
- 2) Cite e explique pelo menos uma situação real de aplicação (diferentes dos citados até o momento) de cada um dos padrões que você escolheu no exercício 1.