Proxy

Padrão de Projeto

Prof. Igor Avila Pereira
igor.pereira@riogrande.ifrs.edu.br

Instituto Federal de Educaçãoo, Ciência e Tecnologia do Rio Grande do Sul (IFRS)
Campus Rio Grande
Divisão de Computação

Agenda

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.

Motivação

Exemplo

Considere um editor de documentos que pode embutir objetos gráficos em um documento.

- Alguns objetos gráficos, tais como grandes imagens, podem ser muito caros para serem criados.
- A abertura de documentos deveria ser rápida, assim, deveríamos evitar a criação, de uma só vez, de todos os objetos caros quando o documento é aberto.
- De qualquer forma, isso não é necessário porque nem todos esses objetos estarão visíveis no documento ao mesmo tempo.

Motivação

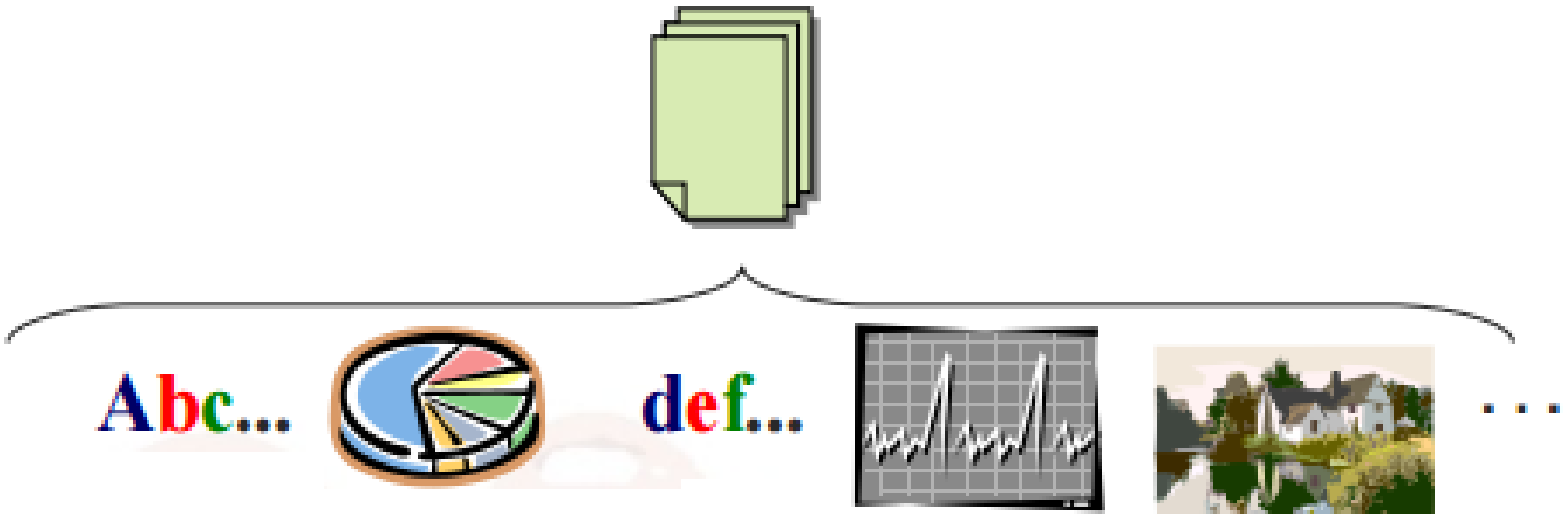


Figura: Exemplo

Motivação

Solução

A solução é usar outro objeto, um **proxy** (procurador), que funciona como um substituto temporário da imagem real. O proxy funciona exatamente como a imagem e toma conta da sua instanciação quando a mesma for necessária.

- O proxy de imagem cria a imagem real somente o editor de documentos solicita ao mesmo exibir a si próprio invocando sua operação *Draw*.
- O proxy repassa as solicitações subsequentes diretamente para a imagem.
- Portanto, ele deve manter uma referência para a imagem após criá-la

Motivação

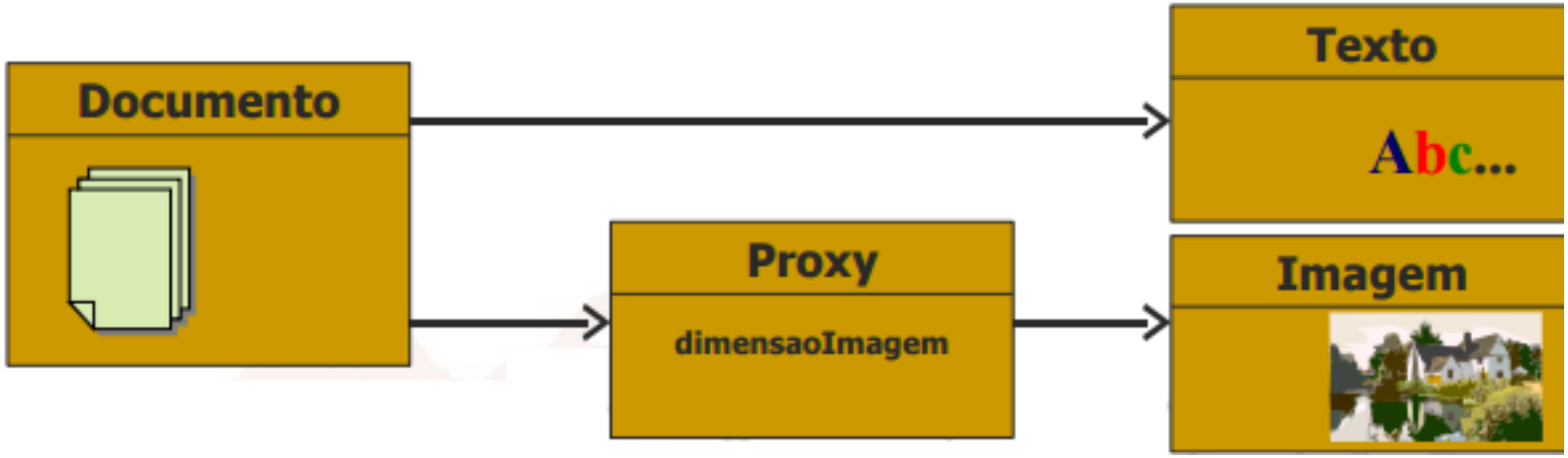


Figura: Exemplo

Motivação

Vamos assumir que as imagens são armazenadas em arquivos separados.

- Neste caso, podemos usar o nome do arquivo como referência para o objeto real.
- O proxy também armazena sua extensão, ou seja, sua largura e altura.
- A extensão permite ao proxy esconder as solicitações sobre o seu tamanho, oriundas do formatador, sem ter que efetivamente instanciar a imagem.

Motivação

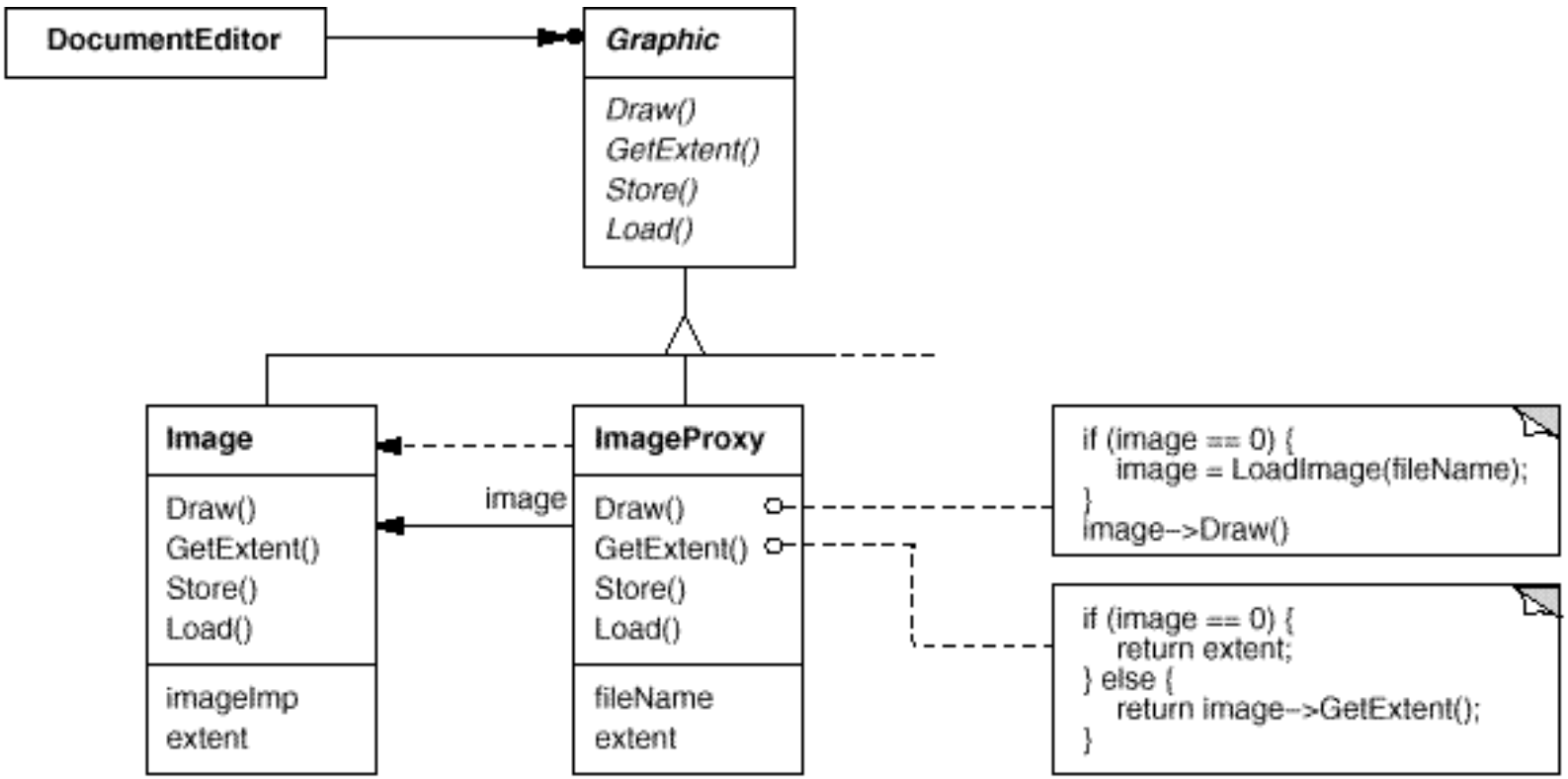


Figura: Estrutura

Motivação

- O editor de documentos acessa as imagens embutidas através da interface definidas pela classe abstrata *Graphic*.
- ImageProxy é uma classe para imagens que são criadas sob demanda. ImageProxy mantém o nome do arquivo como uma referência para a imagem no disco.
- O nome do arquivo é passado como um argumento para construtor de ImageProxy.

Motivação

- Um ImageProxy também armazena os limites da imagem e uma referência para a instância real de Image (*filename*). Essa referência não será válida até que o Proxy instancie a imagem real.
- A operação Draw garante que a imagem é instanciada antes de repassar a ela a solicitação.
- *GetExtent* repassa a solicitação para a imagem somente se ela estiver instanciada; caso contrário, ImageProxy retorna a extensão armazenada.

Aplicabilidade

O padrão Proxy é aplicável sempre que há necessidade de uma referência mais versátil, ou sofisticada, do que um simples apontador para um objeto.

Aplicabilidade

Aqui apresentamos diversas situações comuns nas quais o padrão Proxy é aplicável:

- Um **remote proxy** fornece um representante local para um objeto num espaço de endereçamento diferente.
- Um **virtual proxy** cria objetos caros sob demanda. O ImageProxy descrito na seção de Motivação é um exemplo de um proxy deste tipo.
- Um **protection proxy** controla acesso ao objeto original. Os proxies de proteção são úteis quando os objetos devem ter diferentes direitos de acesso.

Aplicabilidade

- Um **smart reference** é um substituto para um simples *pointer* que executa ações adicionais quando um objeto é acessado.
  - contar o número de referências para o objeto real, de modo que o mesmo possa ser liberado automaticamente quando não houver mais referências
  - carregar um objeto persistente para a memória persistente quando ele for referenciado pela primeira vez
  - verificar se o objeto real está bloqueado antes de ser acessado, para assegurar que nenhum outro objeto possa mudá-lo

Aplicabilidade

O consagrado framework Hibernate também utiliza o pattern Proxy, por exemplo, ao fazer o "lazy-loading", técnica utilizado para acessar o banco de dados apenas quando for necessário.

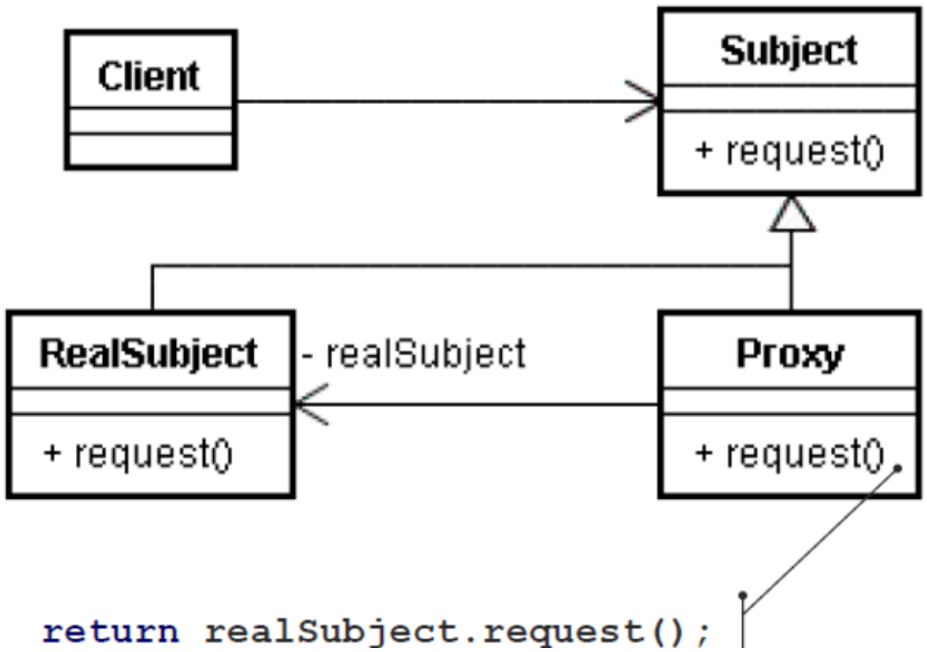
Muitas vezes quando trabalhamos com o Hibernate, e uma busca é realizada, por exemplo usando o método "session.load(id)", um Proxy para o objeto real é retornado.

Aplicabilidade

Neste caso o objeto ainda não está completamente preenchido, pois nenhum SQL foi realizado até este momento.

Apenas quando uma propriedade deste objeto (métodos getX) ou um relacionamento, como por exemplo "empresa.getFuncionarios()"forem chamados, a consulta no banco será realizada. Tudo isto de forma transparente para o cliente.

Estrutura



É possível substituir a classe **Subject** por uma classe abstrata ou uma interface

Participantes

- **Proxy** (ImageProxy)
  - mantém uma referência que permite ao proxy acessar o objeto real (*real subject*). O proxy pode referenciar um Subject se as interfaces de RealSubject e Subject forem as mesmas;
  - fornece uma interface idêntica a de Subject, de modo que o proxy possa substituir o objeto real (*real subject*)
  - controla o acesso ao objeto real e pode ser responsável pela sua criação e exclusão
  - outras responsabilidades dependem do tipo de proxy

Participantes

- **Subject**
  - define uma interface comum para RealSubject e Proxy, de maneira que um Proxy possa ser usado em qualquer lugar em que um RealSubject é esperado
- **RealSubject**
  - Define o objeto real que o proxy representa.

Consequências

O padrão Proxy introduz um nível de referência indireta no acesso a um objeto. A referência indireta adicional tem muitos usos, dependendo do tipo de proxy

1. Um proxy remoto pode ocultar o fato de que um objeto reside em um espaço de endereçamento diferente
2. Um proxy virtual pode executar otimizações, tais como a criação de um objeto sob demanda
3. Tanto proxies de proteção como smart references permitem tarefas adicionais de organização quando um objeto é acessado

Padrões Relacionados

- **Decorator**: embora decoradores possam ter implementações semelhantes às de proxies, os decoradores têm uma finalidade diferente. Um decorador acrescenta uma ou mais responsabilidades a um objeto, enquanto que um proxy controla o acesso a um objeto.

Também com o Padrão Adapter

Discussão

- Como Decorator, o padrão Decorator compõe o objeto e fornece uma interface idêntica para os clientes.
- Diferentemente do Decorator, o padrão Proxy não está preocupado em incluir e excluir propriedades dinamicamente e não está projetado para composição recursiva.
- Sua intenção é fornecer um substituto para um objeto quando for inconveniente ou indesejável acessá-lo diretamente porque, por exemplo, está numa máquina remota, tem acesso restrito ou é persistente.

Discussão

- No padrão Proxy, o objeto fornece as funcionalidades-chave e o proxy fornece (ou recusa) acesso ao objeto. No Decorator, o componente fornece somente parte da situação em que a funcionalidade e um ou mais decoradores fornecem o restante.
- O Decorator trata a situação em que a funcionalidade total de um objeto não pode ser determinada em tempo de compilação, pelo menos não de maneira conveniente.

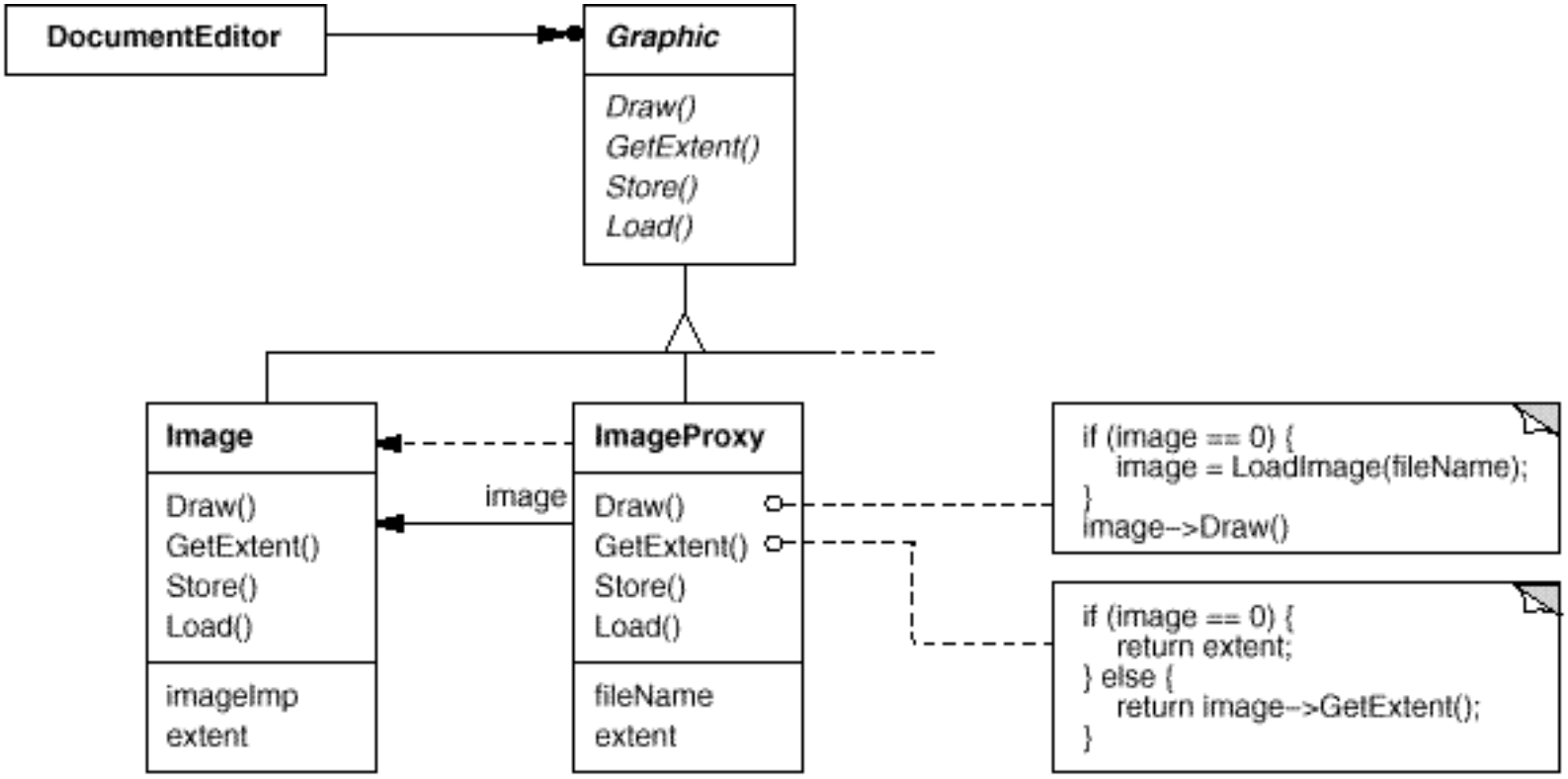
Discussão

- Essa abertura torna a composição recursiva uma parte essencial do Decorator.
- Esse não é o caso do Proxy porque o Proxy focaliza um relacionamento - entre o proxy e seu objeto - e esse relacionamento pode ser expresso estaticamente.

Essas diferenças são significativas porque capturam soluções para problemas recorrentes específicos no projeto orientado a objetos. Mas isso não significa que esses padrões não possam ser combinados.

Implementação

Vamos implementar o exemplo do editor de documento?



Trabalho

Em uma garagem coletiva existe o seguinte ambiente:

- Os empregados devem manobrar qualquer tipo de veículo que entra e saí da garagem;
- Assim, saber se o empregado possui mais de 18 anos e que tem carteira de habilitação torna-se uma exigência para que o mesmo possa manobrar os veículos presentes na garagem;
- Todavia, neste momento, existe na garagem um outro grupo de empregados adolescentes que estão em caráter de estágio e que não são maiores de 18 anos;
- Há também um grupo de empregados que trabalha na parte administrativa e que são maiores de 18 anos mas não possuem nenhuma carteira de habilitação;

Trabalho

Além disso, no sistema de gerenciamento da garagem coletiva há classes e código legado:

- **Empregado**
  - *nome*, *data de nascimento* (é possível usar nossa classe Data) e *número da carteira de motorista* (senão tiver, campo é vazio)
- **Carro**
  - *ano*, *modelo*, *marca*, *placa*, *chassi*
  - E o método *manobrar (Empregado empregado)*

Trabalho

Problema

Há necessidade de restringir o acesso aos carros somente aos empregados da garagem coletiva que são maiores de 18 anos e que possuem carteira de habilitação, ou seja, **somente empregados maiores de 18 e que possuem carteira de habilitação podem manobrar carros da garagem**.

Descrição

Implemente o padrão Proxy para adicionar esta regra de negócio ao sistema de gerenciamento da garagem coletiva.

- Essas novas regras devem ficar dentro do proxy

Proxy

Padrão de Projeto

Prof. Igor Avila Pereira
igor.pereira@riogrande.ifrs.edu.br

Instituto Federal de Educaçãoo, Ciência e Tecnologia do Rio Grande do Sul (IFRS)
Campus Rio Grande
Divisão de Computação