

Diretrizes para projeto de objetos: padrões GRASP

Análise e Projeto Arquitetural de
Software

Prof. Thiago

Objetivos

- Após esta aula, você deverá ser capaz de:
 - Compreender o conceito de padrões na modelagem orientada a objeto;
 - Compreender o conceito de projeto guiado por responsabilidades;
 - Aplicar os primeiros padrões GRASP ao projeto de objetos.

Motivação

Algumas questões podem surgir ao nos iniciarmos na modelagem orientada a objetos:

Criar classes e depois “preenchê-las” com métodos?

Identificar operações, criar as classes e depois associar as operações como métodos?



Motivação

Se for possível registrar e transmitir o conhecimento sobre o que é um “bom” projeto de objetos, a tarefa pode ser mais fácil!



Motivação

- Vamos apresentar duas formalizações do conceito de qualidade no projeto de objetos:
 - **Projeto guiado por responsabilidades (PGR)**
 - **Padrões GRASP para projeto de objetos**

Projeto Guiado por Responsabilidades (PGR)

- Direciona o projeto de objetos para a definição e atribuição de responsabilidades e colaborações
- **Responsabilidade:** obrigação de um objeto, em termos do seu papel
- **Colaboração:** objetos que colaboram com o atendimento de uma responsabilidade

Projeto Guiado por Responsabilidades (PGR)

- Responsabilidades podem ser de dois tipos:

- **Conhecer**

Ter conhecimento sobre os dados internos do objeto, sobre objetos relacionados ou sobre coisas que o objeto pode gerar ou calcular

- **Fazer**

Executar cálculos, criar objetos, iniciar ações em outros objetos, controlar e coordenar atividades em outros objetos

Projeto Guiado por Responsabilidades (PGR)

As responsabilidades do tipo *conhecer* estão normalmente ligadas aos atributos, e as do tipo *fazer*, aos métodos

Padrões GRASP

- A atribuição de responsabilidades é apoiada pelos padrões GRASP
 - *General Responsibility Assignment Software Patterns*
- GRASP define diretrizes para situações típicas de atribuição de responsabilidades
- É apenas uma das possibilidades de aplicação de padrões ao projeto de software...

Conceito de padrão

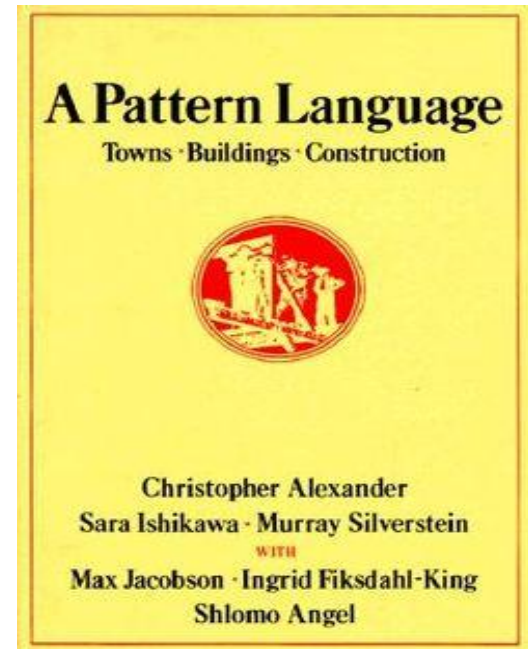
Padrão



Descrição de um problema que ocorre muitas vezes, e de uma solução genérica que pode ser aplicada toda vez que o problema ocorre

Histórico

- A idéia de padrões foi inicialmente desenvolvida na área de arquitetura
- Christopher Alexander escreveu em 1977 o livro *A Pattern Language: Towns, Buildings, and Construction*, que serviu de inspiração para os desenvolvedores de software



Exemplos de padrões na arquitetura

- **Problema:** Em lugares nos quais há ação e movimento de pessoas, os lugares mais convidativos são aqueles altos o suficiente para dar um ponto de vantagem, e baixos o suficiente para colocar as pessoas em ação
- **Solução:** escadas unindo níveis, diretamente acessíveis para as pessoas

http://vasarhelyi.eu/books/A_pattern_language_book

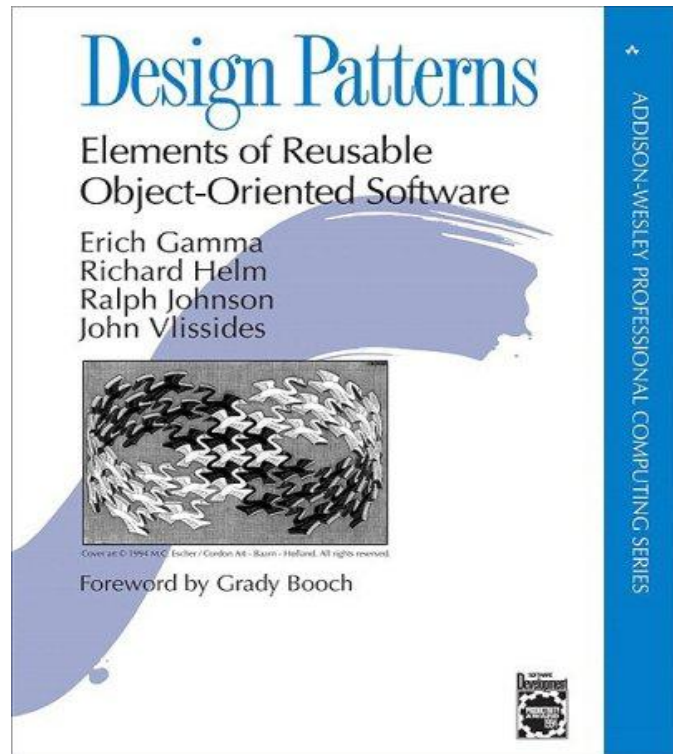


Estrutura de um bom padrão

- Um bom padrão deve ter:
 - **Nome**, para ser facilmente referenciado;
 - **Descrição do problema genérico** ao qual ele se aplica;
 - **Descrição da solução** que ele traz ao problema

Padrões em computação

- O livro *Design Patterns*, de Gamma, Johnson, Vlissides e Helm, publicado em 1995, trouxe o conceito de padrões para a arquitetura de software
- O impacto foi enorme na comunidade de desenvolvimento, e o livro vendeu milhares de cópias
- Os padrões definidos ficaram conhecidos como **padrões GoF** (de Gang of Four)



Padrões GRASP

- Definidos por Larman, são nove padrões para atribuição de responsabilidades:
 - **Especialista na informação**
 - **Criador**
 - **Acoplamento baixo**
 - **Coesão Alta**
 - **Polimorfismo**
 - **Controlador**
 - **Invenção Pura**
 - **Indireção**
 - **Variações Protegidas**

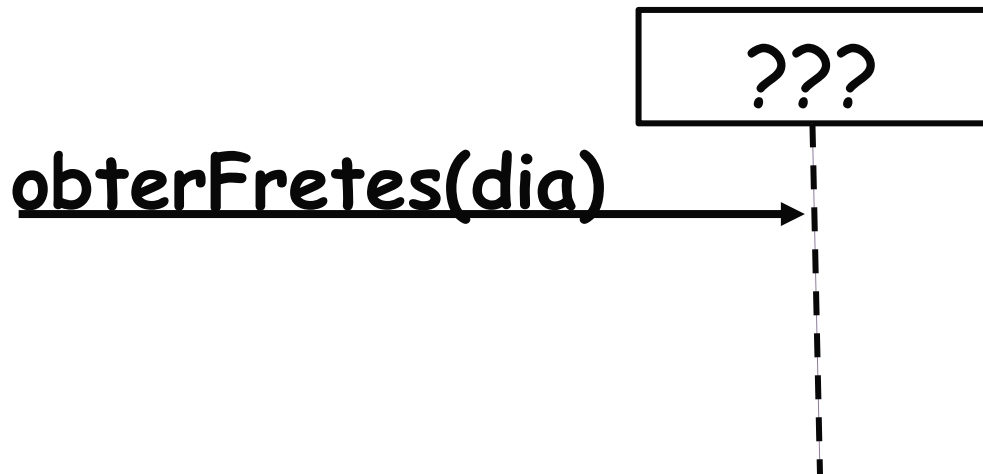
Especialista na informação

Problema: Qual um princípio geral de atribuição de responsabilidades?

Solução: Atribua a responsabilidade à classe que é *especialista na informação* necessária para satisfazer a responsabilidade

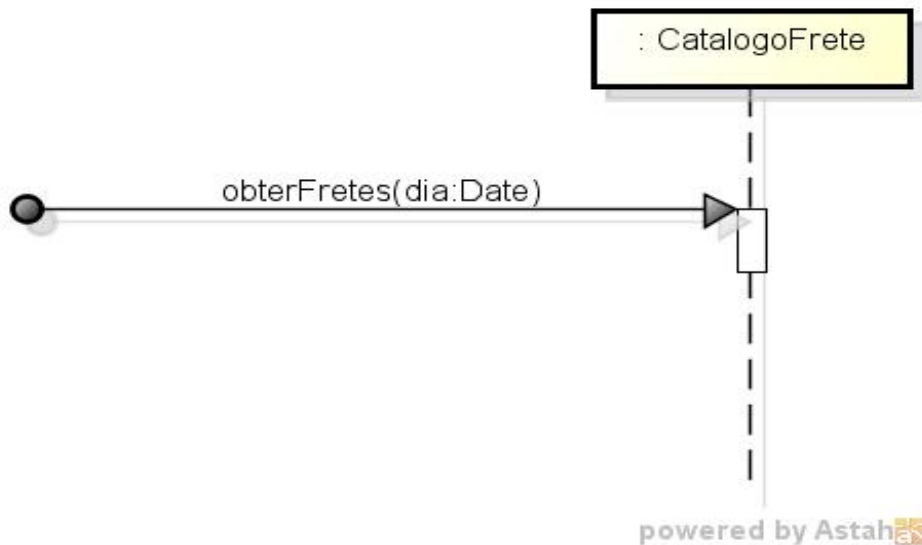
Especialista na informação - exemplo

Quem deveria saber os fretes a serem feitos em um determinado dia?



Especialista na informação - exemplo

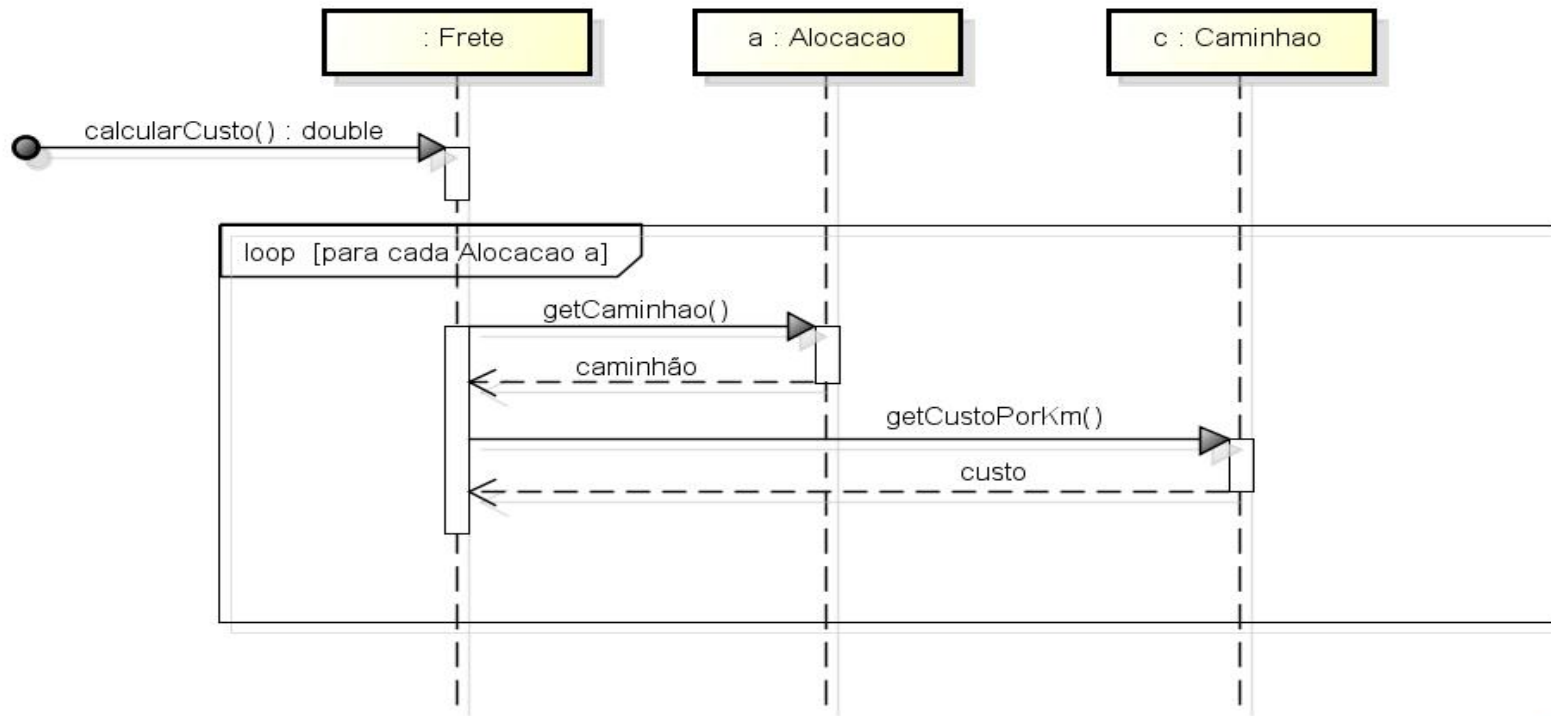
O CatalogoFrete, que tem informação sobre todos os fretes já criados no sistema



Especialista na informação - exemplo

O Frete tem a responsabilidade de calcular o custo, pois ele detém informação sobre Alocação e, por consequência, sobre o custo por quilômetro rodado pelo Caminhão

Especialista na informação - exemplo



Criador

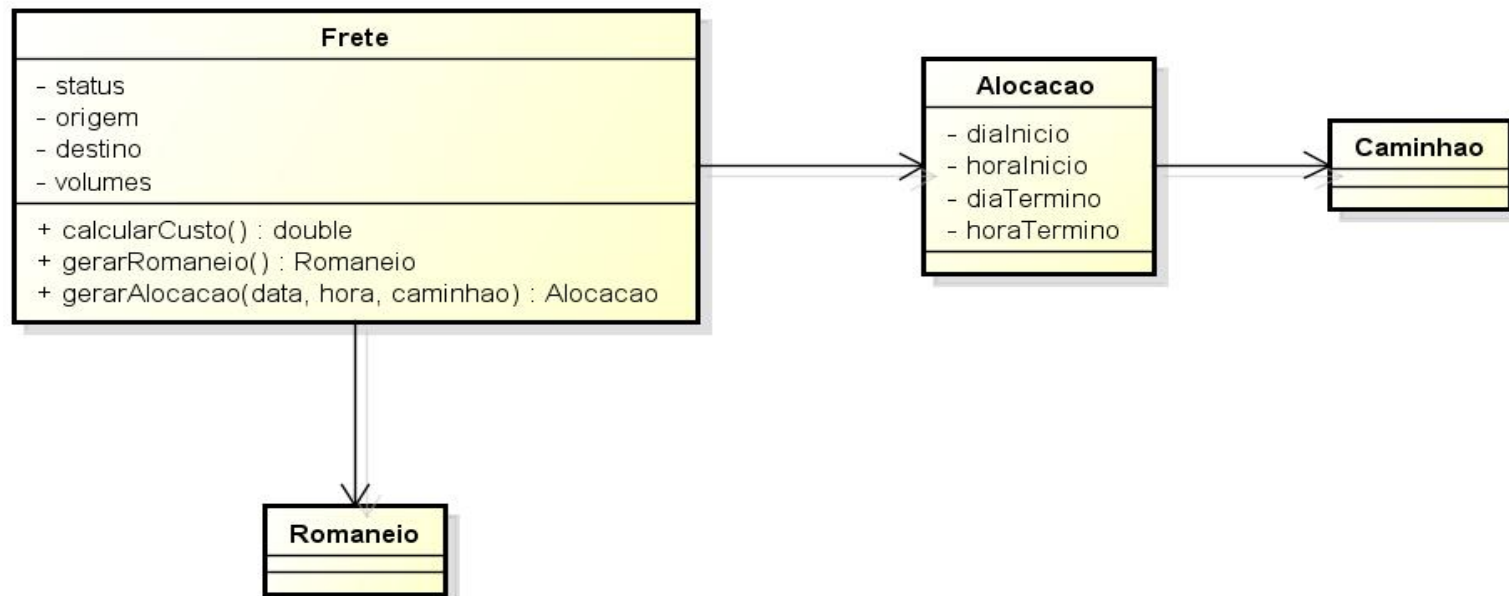
Problema: Quem deve ser responsável por criar instâncias de uma classe?

Solução: A classe B deve criar instâncias da classe A se:

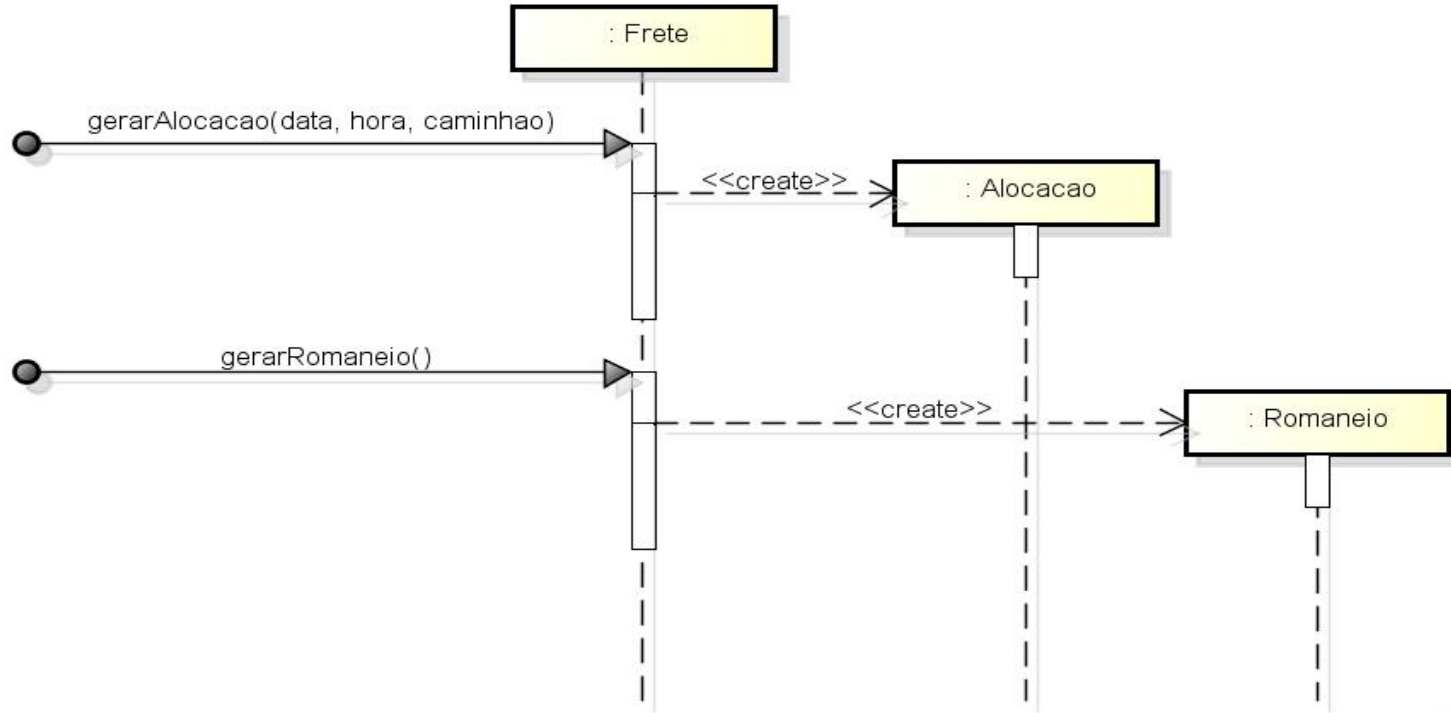
- B “contém” ou agrega A de modo composto;
- B registra A
- B usa A de uma maneira muito próxima
- B tem os dados iniciais para criar uma instância de A

Criador - exemplo

- O Frete tem informação para criar o Romaneio
- O Frete utiliza diretamente a Alocação (além de ter a informação necessária para criá-la)



Criador - exemplo



powered by Astah

Contra-indicações

- Às vezes, a criação de objetos envolve passos muito complexos (ex: reutilizar instâncias anteriores devido a questões de desempenho)
- Nesses casos, é mais adequado delegar a criação para uma classe auxiliar → Fábrica Concreta ou Fábrica Abstrata (padrões GoF)

Acoplamento baixo

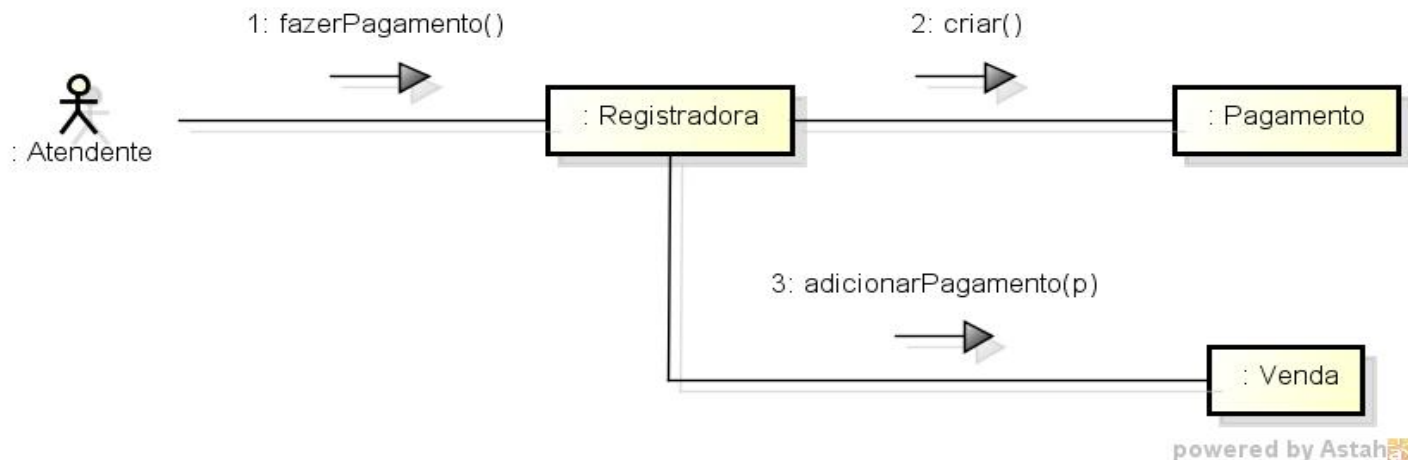
Problema: como apoiar dependência baixa, baixo impacto de modificação e aumento do reúso?

Acoplamento é uma medida de quão fortemente um elemento está conectado a, tem conhecimento de, ou depende de outros elementos

Solução: atribua responsabilidades de forma a reduzir o acoplamento

Acoplamento baixo - exemplo

Sistema de frente de caixa: Registradora é uma classe de controle, as classes Venda e Pagamento são classes de domínio.

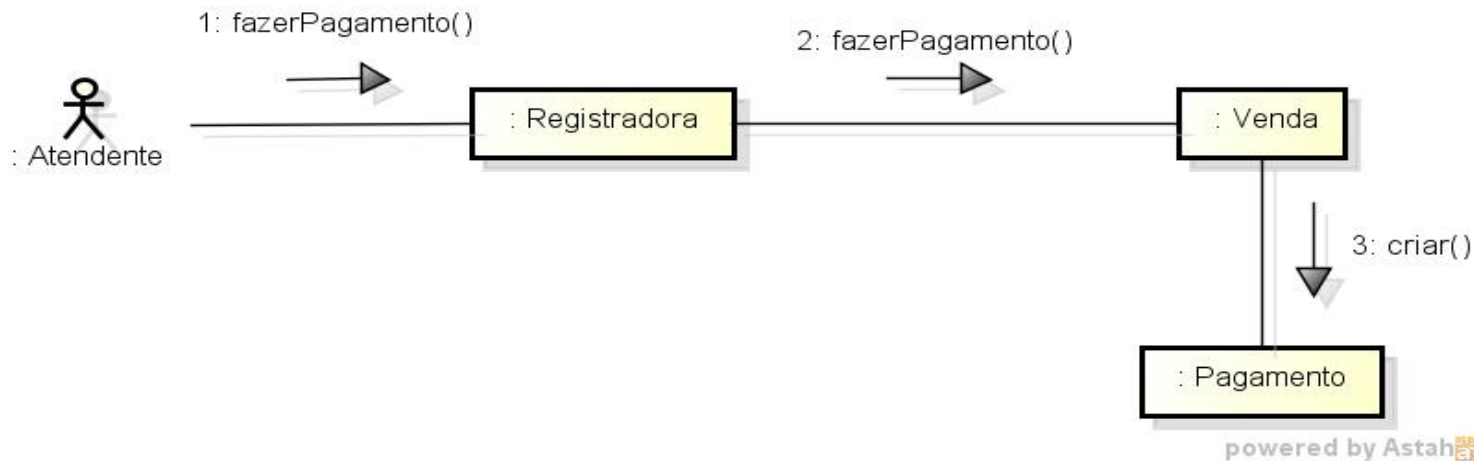


Se Registradora cria o Pagamento e o associa a Venda, ela tem acoplamento com ambas as classes. Mas...

Acoplamento baixo - exemplo

Se Venda é responsável por criar seus Pagamentos, Registradora não precisa mais “saber” da existência de Pagamento.

Uma eventual mudança no sistema de pagamento não afetaria a classe Registradora.



Coesão

Problema: como manter os objetos bem focados, inteligíveis e gerenciáveis e como efeito colateral apoiar Acoplamento Baixo?

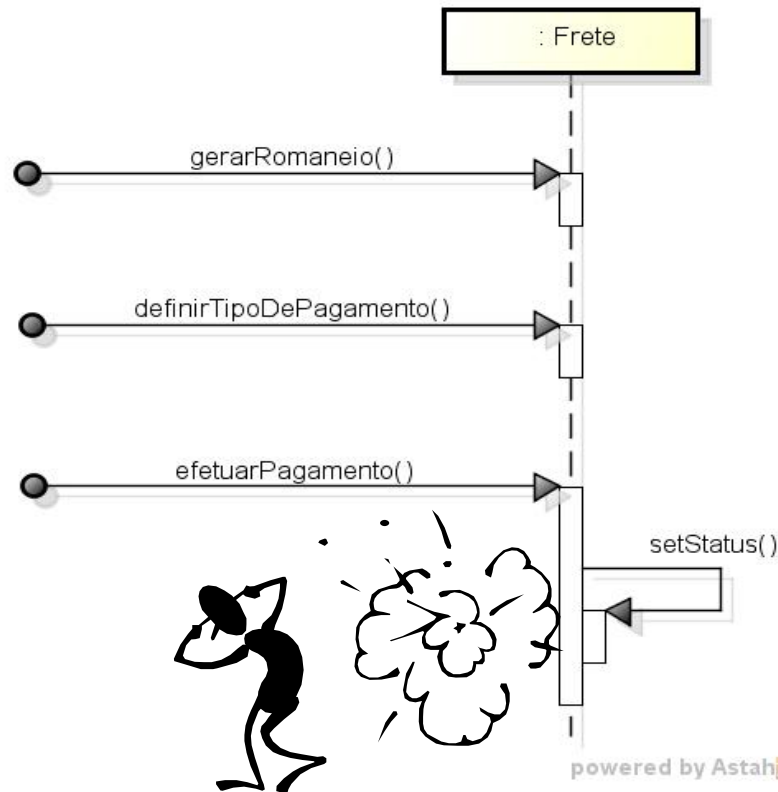
Coesão é uma medida de quanto as responsabilidades de um elemento estão fortemente relacionadas e focalizadas.

Solução: atribuir responsabilidades de forma que a coesão permaneça alta.

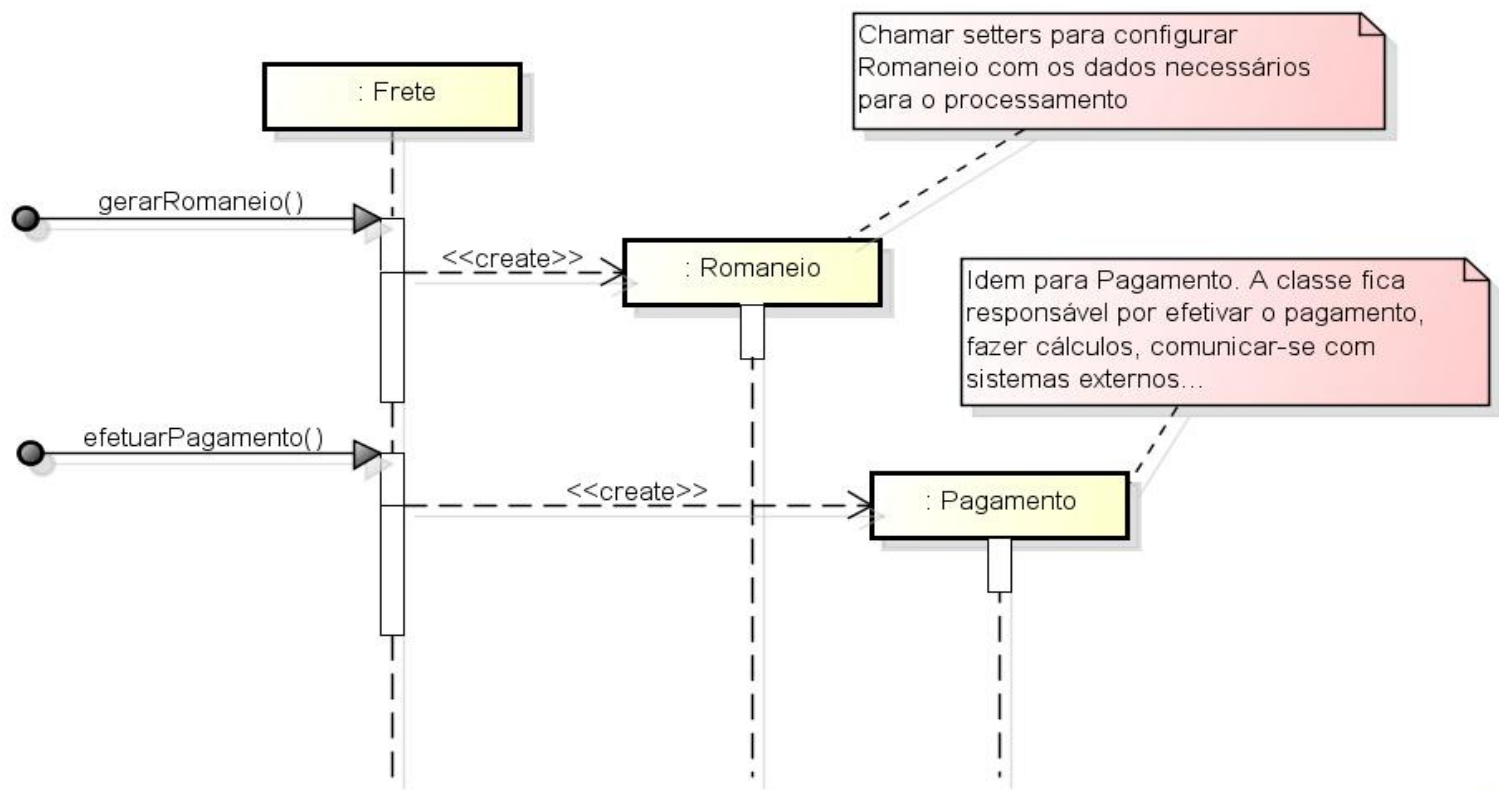
Coesão - exemplo

Na transportadora, o Frete poderia cuidar do romaneio e do pagamento.

Mas o que aconteceria se as formas de pagamento começassem a variar?



Classes para Romaneio e Pagamento aumentam a coesão



powered by Astah

Coesão e acoplamento

Coesão e acoplamento são dois lados da mesma moeda



Um acoplamento baixo demais não é possível, pois as classes não teriam nenhuma colaboração

Uma coesão muito alta pode envolver funcionalidades muito díspares, reduzindo o reúso

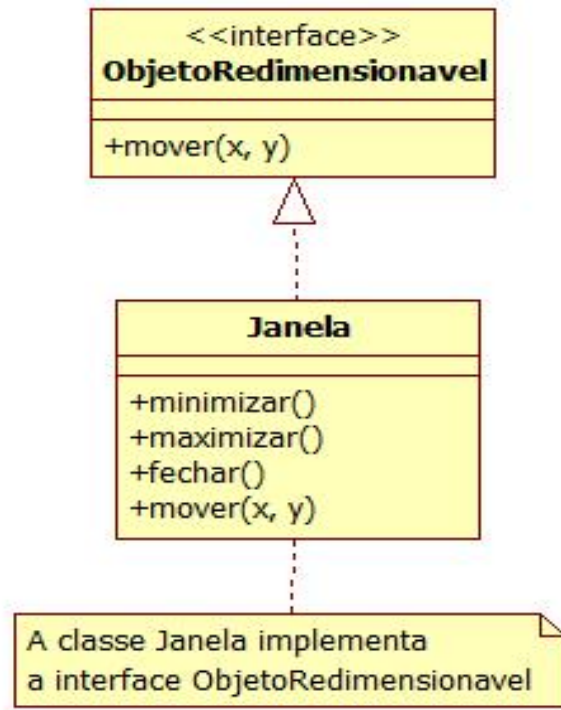
Polimorfismo

Problema: como tratar alternativas de comportamento com base no tipo da classe? Como criar componentes de software interconectáveis?

Solução: Atribua a responsabilidade pelo comportamento aos tipos para os quais o comportamento varia, utilizando operações polimórficas

Uma palavra sobre interfaces

- Uma **interface** é a definição da assinatura de um conjunto de métodos
- Uma classe pode implementar uma ou mais interfaces
- A notação mais usual é o estereótipo `<<interface>>` no Diagrama de Classes



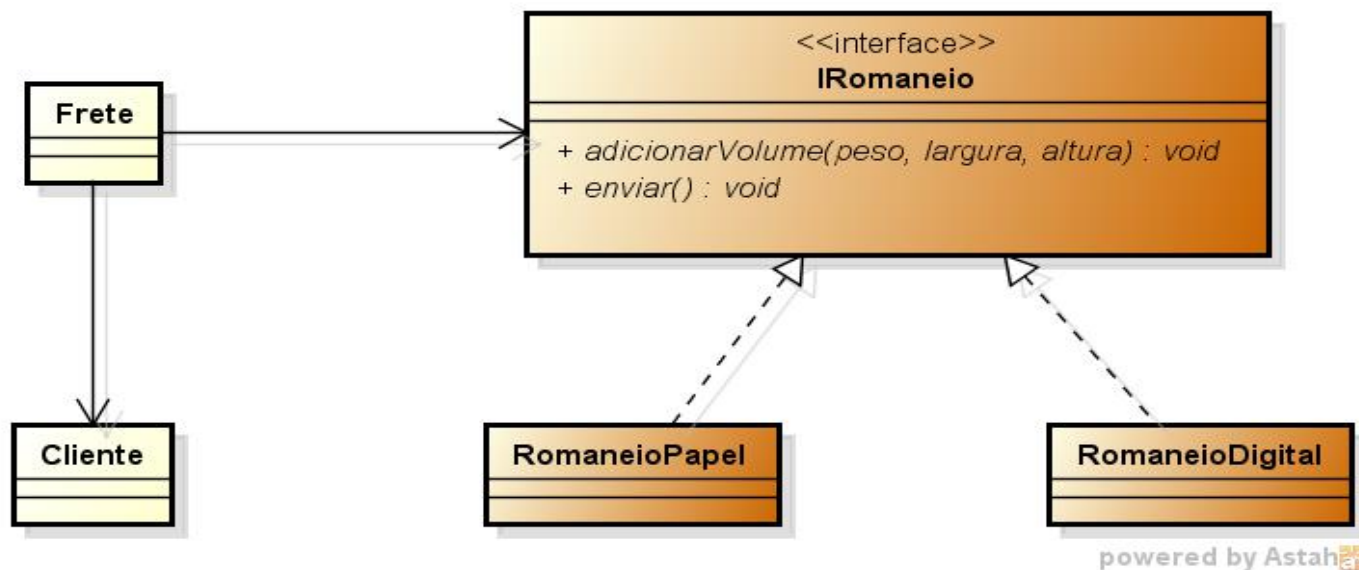
Uma palavra sobre interfaces

- Interfaces são usadas para
 - Permitir que várias classes implementem um comportamento em comum sem depender de herança
 - Permitir que a implementação de uma operação mude de forma desacoplada de quem usa a operação
- Interfaces não devem ser usadas para
 - Especificar um comportamento que hoje apenas uma classe implementa, porque “*talvez, algum dia*” outra classe o implemente

Polimorfismo - exemplo

- Se o Romaneio puder ser um documento em papel ou digital, o comportamento comum a ambas as situações pode ser modelado por uma interface
- As classes que implementam a funcionalidade modificam o comportamento do envio do Romaneio de acordo com a situação

Polimorfismo - exemplo





Obrigado!

tsbarcelos@ifsp.edu.br