

4.4.2-Refinando por meio de padrões

Os padrões *GRASP* são utilizados para uniformizar estratégias para criação de estruturas orientadas a objetos, em nível de relacionamento entre classes e em nível de interação por mensagens.

Podemos refinar um diagrama de colaboração aplicando sucessivamente os seguintes padrões:

Padrão Especialista

FUNÇÃO : atribuir responsabilidade ao especialista da informação - a classe que tem a informação necessária para satisfazer uma responsabilidade.

Baseado na idéia de WIRFS-BROOKS - classes-responsabilidades-colaboração (CRC), auxilia na determinação de métodos das classes e na determinação de relacionamentos entre elas.

Passos de utilização do padrão:

- 1-Definir objetos de domínio do problema dentro do use case
- 2-A partir dos eventos do use case, fazer a pergunta "de quem é a responsabilidade por realizar isto?". Em primeira instância, seria da classe que tem as informações necessárias para tal.
- 3-Definir agora que COLABORAÇÕES são necessárias (informações necessárias) de outros objetos para realizar a RESPONSABILIDADE do objeto considerado.
- 4-Para cada objeto responsável pelas colaborações (que têm a responsabilidade de realizá-las), repetir o item 3.

Seja, por exemplo, uma operação denominada *CalcularTotal*. O seguinte diálogo demonstra a utilização deste padrão:

Q: De quem é a responsabilidade por calcular o total de vendas?

R: Quem tem as informações necessárias para tal é a classe VENDA.

Q: Que informação (colaboração) é necessária para determinar este total?

R: Precisamos conhecer instâncias de Item de Venda e a soma de seus subtotais. Assim, a responsabilidade por calcular os subtotais é da classe Item de Venda. Temos então uma mensagem indo de Venda para Item de Venda.

Q: Para calcular o subtotal, o que precisa Item de Venda conhecer?

R: O preço do produto. Quem tem esta informação é o objeto Produto. Logo, cria-se uma mensagem de Item de Venda para Produto (retornarPreço)

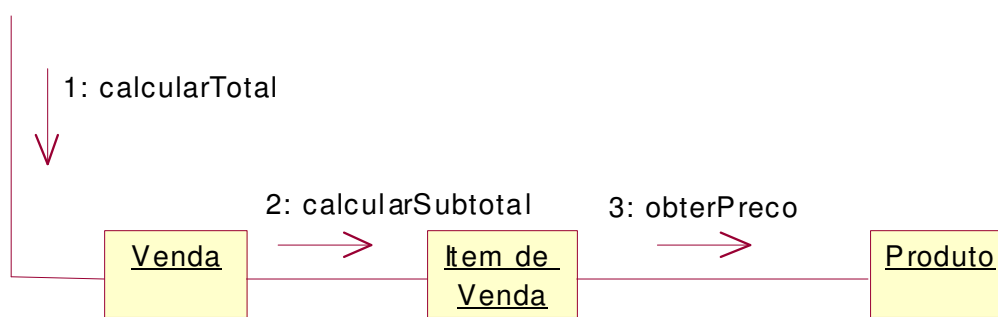


Figura 4.7-Diagrama de colaboração da operação CalcularTotal

Padrão criador

FUNÇÃO: Atribuir à classe B a responsabilidade de criar uma instância da classe A se:

- B contém objetos A (todo-parte);
- B usa objetos A;
- B possui dados de inicialização que serão passados para A quando for criado.

No exemplo anterior, a responsabilidade pela criação de Itens de Venda deve ser de classe que contenha instâncias dos mesmos. Então a responsabilidade de criação da mesma deve ser da classe Venda.

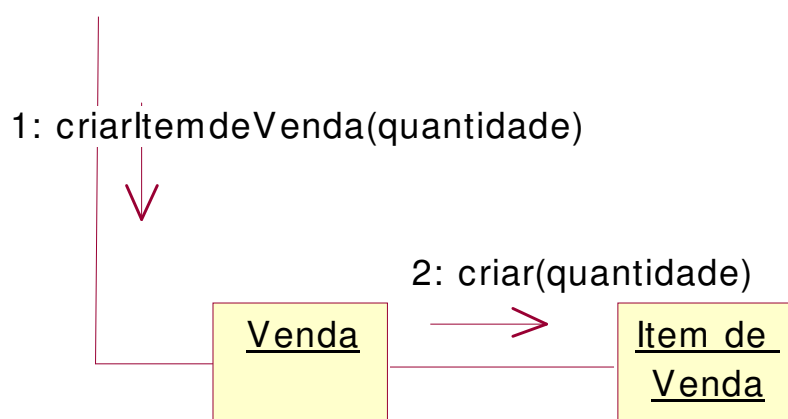


Figura 4.8-Responsabilidade de criação da instância de Item de Venda é da classe Venda

Padrão acoplamento fraco

FUNÇÃO: Atribuir uma responsabilidade de forma a conseguir acoplamento fraco entre classes.

Uma classe com acoplamento forte depende de muitas outras classes.

Na Figura 4.9 abaixo a classe Controle de Venda está acoplada às classes Pagamento e Venda através da passagem das mensagens Criar e RegistrarPagamento respectivamente. Podemos diminuir o acoplamento de Controle de Vendas passando a responsabilidade pela criação da classe Pagamento para a classe Venda (Figura 4.10)

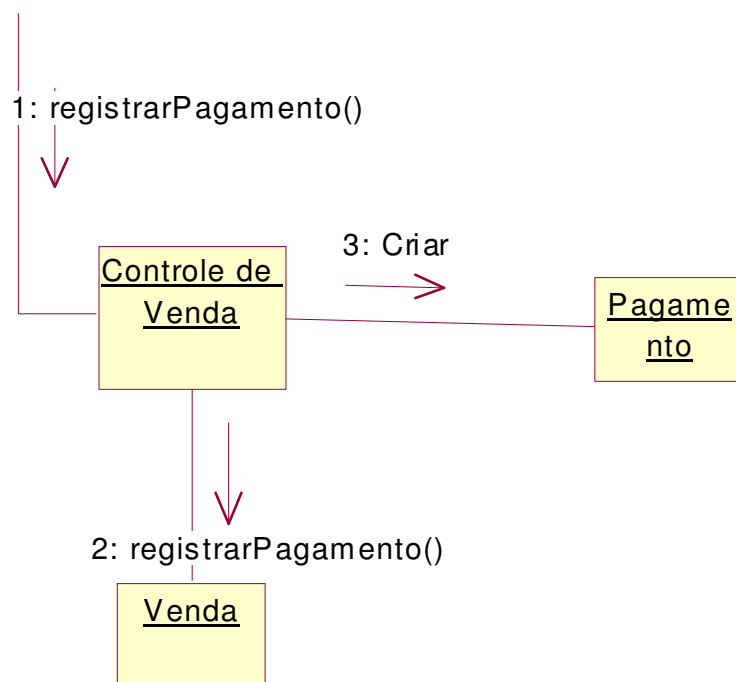
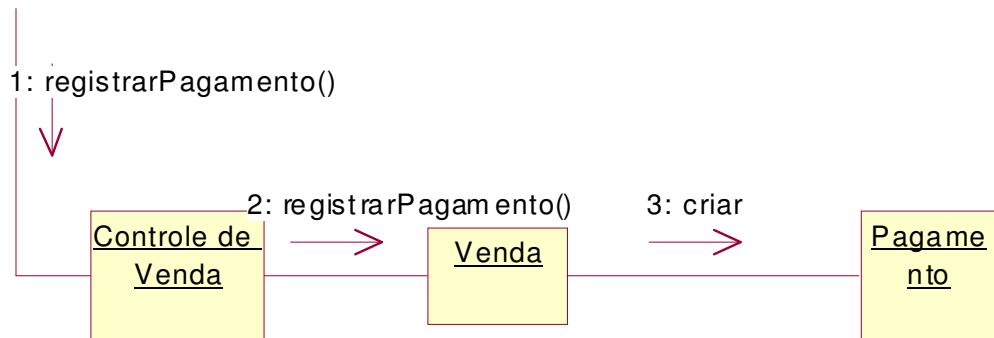


Figura 4.9-Acoplamento de Controle Venda



*Figura 4.10-Acoplamento de Controle Venda diminuído
Padrão Alta Coesão*

FUNÇÃO: Atribuir uma responsabilidade de forma a manter uma alta coesão. A coesão mede quão fortemente relacionadas e focalizadas são as responsabilidades de uma classe.

Vejam o mesmo exemplo anterior. A classe Controle de Venda segundo os padrões criador e especialista, deve criar uma instância de pagamento e registrar o pagamento via objeto venda. Isto é mostrado no primeiro diagrama. Suponha que haja cinquenta operações no sistema, todas atribuídas a Controle de Vendas. Isto faria deste último um objeto com baixa coesão. Assim sendo, ao delegarmos responsabilidades a outros objetos, como no segundo diagrama, estamos aumentando a coesão do objeto Controle de Venda.