



Projeto Orientado a Objetos

Definição das Operações





Responsabilidade x Operação

Responsabilidades de um objeto são as suas obrigações:

- responsabilidades de fazer: fazer um cálculo, criar um objeto, inicializar uma ação em outros objetos, controlar atividades em outros objetos...
- responsabilidades de conhecer: dados privados, dados que precisam ser calculados, objetos relacionados...

As responsabilidades são implementadas usando operações que agem sozinhas ou colaboram com outras operações e objetos.

As responsabilidades são atribuídas às classes de objetos durante o projeto de objetos.

➔ A definição das operações é uma parte crítica do desenvolvimento OO.

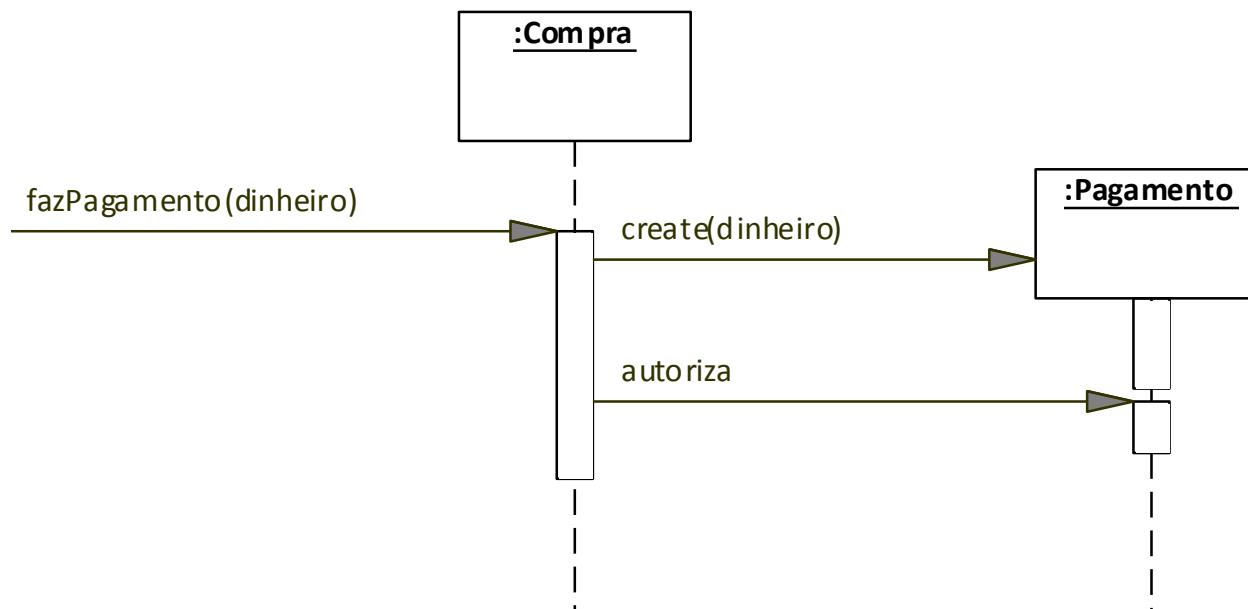


Definição das Operações

As operações são definidas durante a realização dos casos de uso.

Realização de um caso de uso: descreve como um caso de uso é realizado em termos dos objetos que colaboram.

- utilizam diagramas de interação





Definição das Operações

Durante a definição das operações são considerados:

- Padrão de Projeto Criador
- Padrão de Projeto Especialista da Informação
- Princípio de Baixo Acoplamento
- Princípio de Alta Coesão
- Padrão de Projeto Fachada
- Outros Padrões de Projeto





Criador

Durante a definição das operações são considerados:

- Padrão de Projeto Criador
- Padrão de Projeto Especialista da Informação
- Princípio de Baixo Acoplamento
- Princípio de Alta Coesão
- Padrão de Projeto Fachada
- Outros Padrões de Projeto





Criador

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

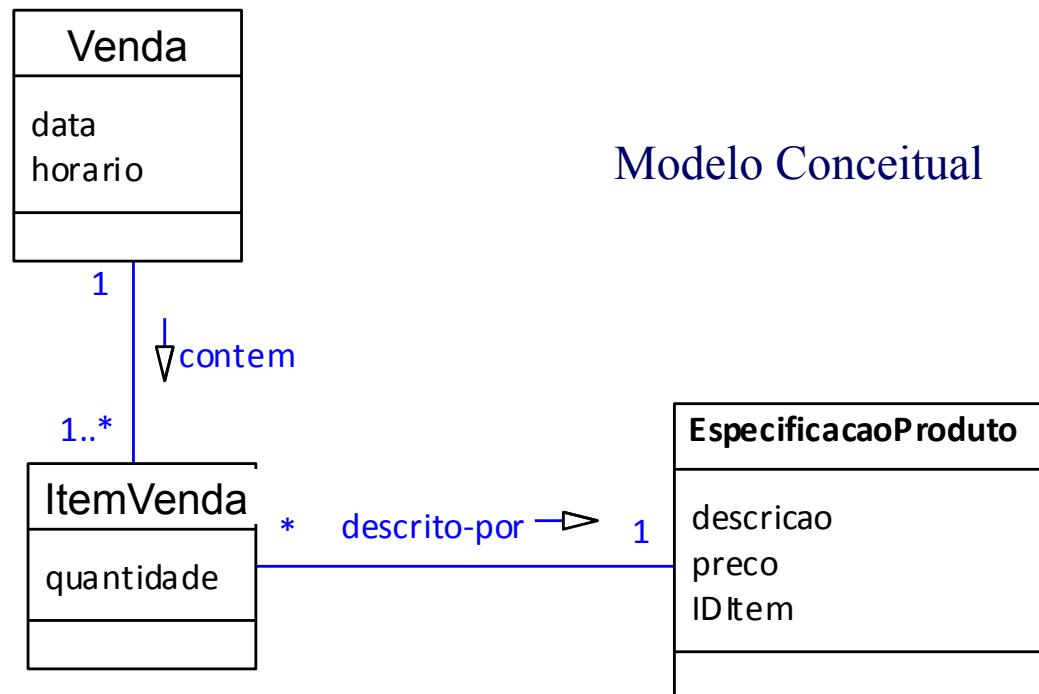
- A contém ou agrega instâncias de B; ou
- A usa instâncias de B; ou
- A tem os dados de inicialização que serão passados para B quando ele é criado.



Criador

Exemplo - Ponto de Venda:

- Quem deverá ser o responsável pela criação de uma instância de ItemVenda?



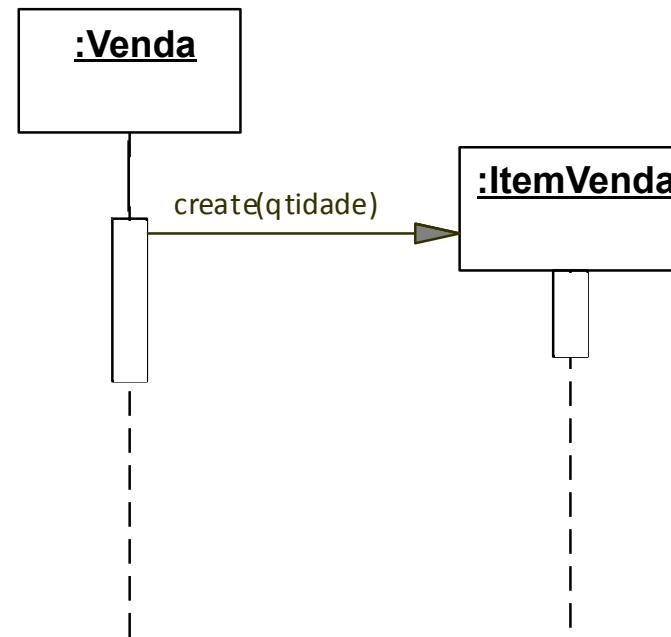


Criador

Exemplo - Ponto de Venda:

- Quem deverá ser o responsável pela criação de uma instância de ItemVenda?

Como a classe **Venda** contém vários objetos do tipo **ItemVenda**, ela pode ser a responsável.

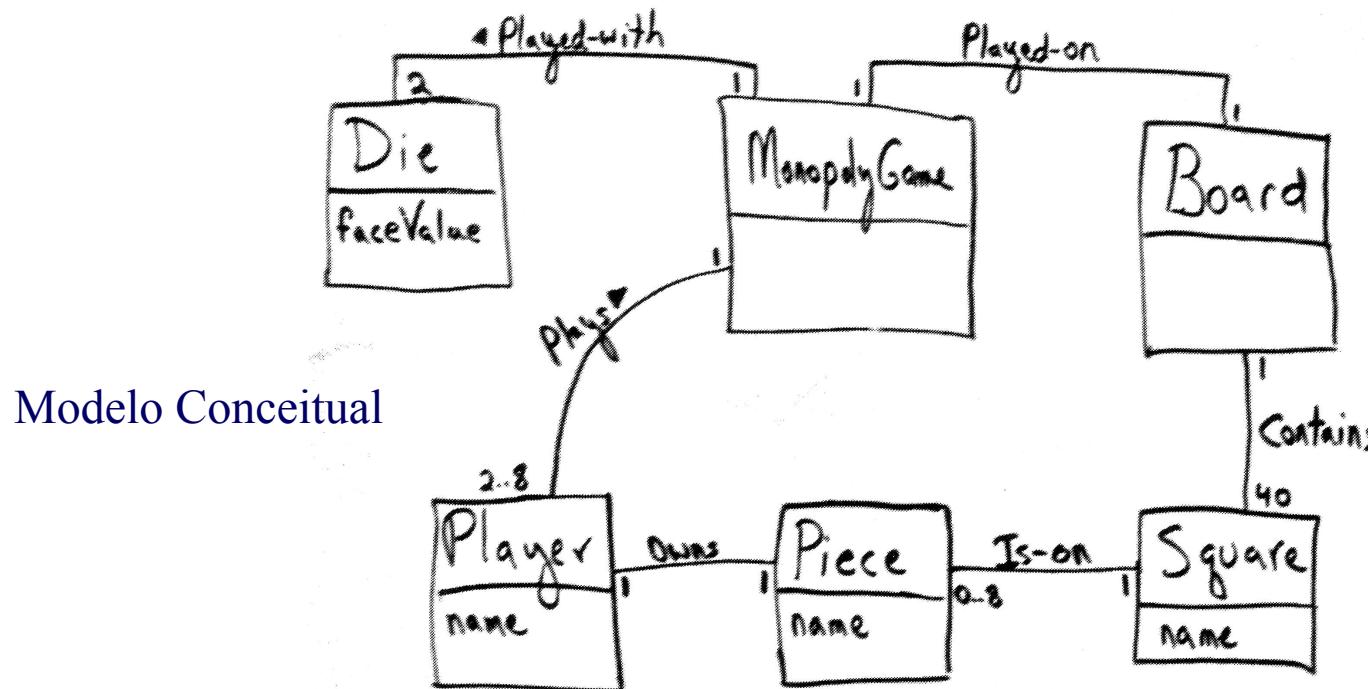




Criador

Exemplo - Banco Imobiliário:

- Quem deverá ser o responsável pela criação de uma instância de Casa (Square) do Banco Imobiliário?

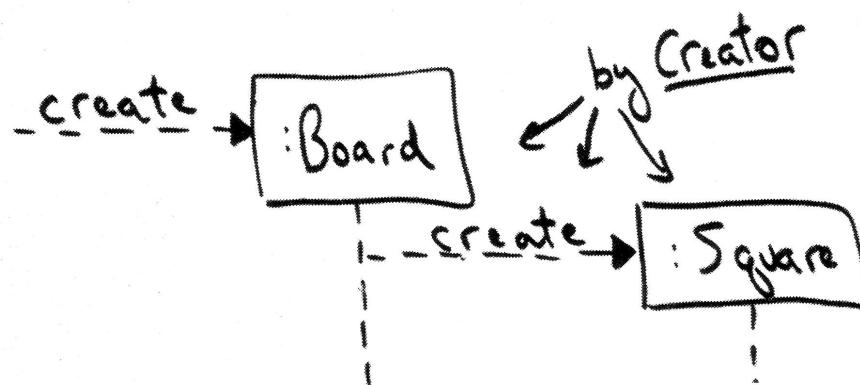


Criador

Exemplo - Banco Imobiliário:

- Quem deverá ser o responsável pela criação de uma instância de Casa (Square) do Banco Imobiliário?

Como a classe **Tabuleiro** contém vários objetos do tipo Casa, ela pode ser a responsável.





Especialista da Informação

Durante a definição das operações são considerados:

- Padrão de Projeto Criador
- Padrão de Projeto Especialista da Informação
- Princípio de Baixo Acoplamento
- Princípio de Alta Coesão
- Padrão de Projeto Fachada
- Outros Padrões de Projeto





Especialista da Informação

Especialista da Informação: Atribuir uma responsabilidade ao especialista da informação, ou seja, à classe que tem a informação necessária para cumpri-la.

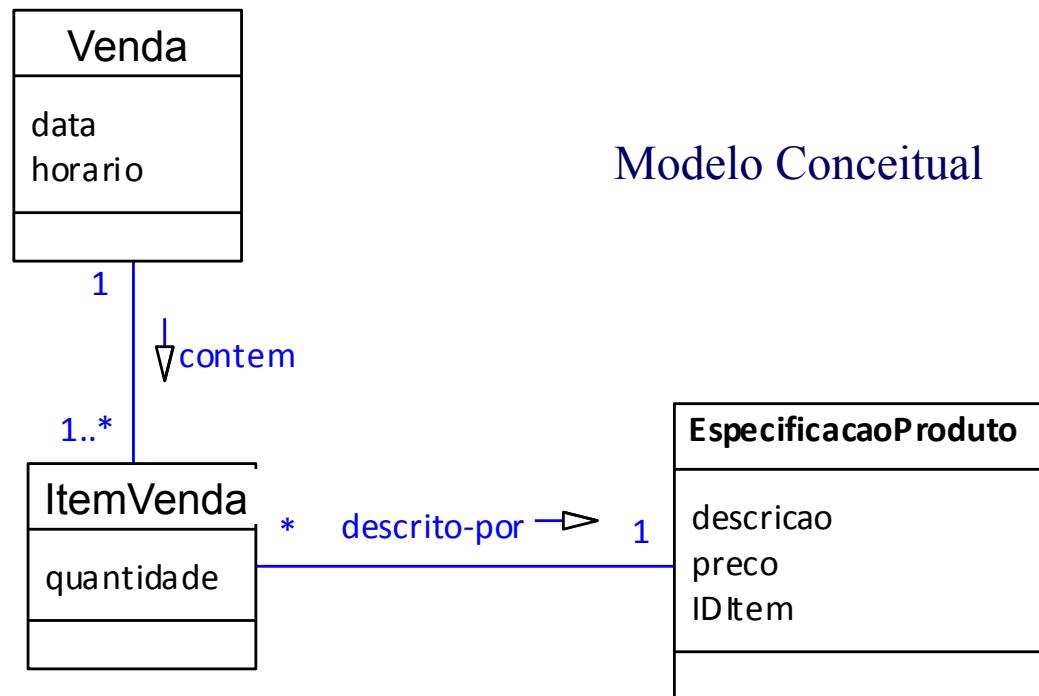




Especialista da Informação

Exemplo - Ponto de Venda:

- Quem deverá ser o responsável por saber o total de uma venda?





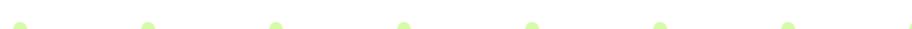
Especialista da Informação

Exemplo - Ponto de Venda:

- Quem deverá ser o responsável por saber o total de uma venda?

Para determinar o total da venda é necessário conhecer todas as instâncias de **ItemVenda** de uma venda e a soma de seus sub-totais.

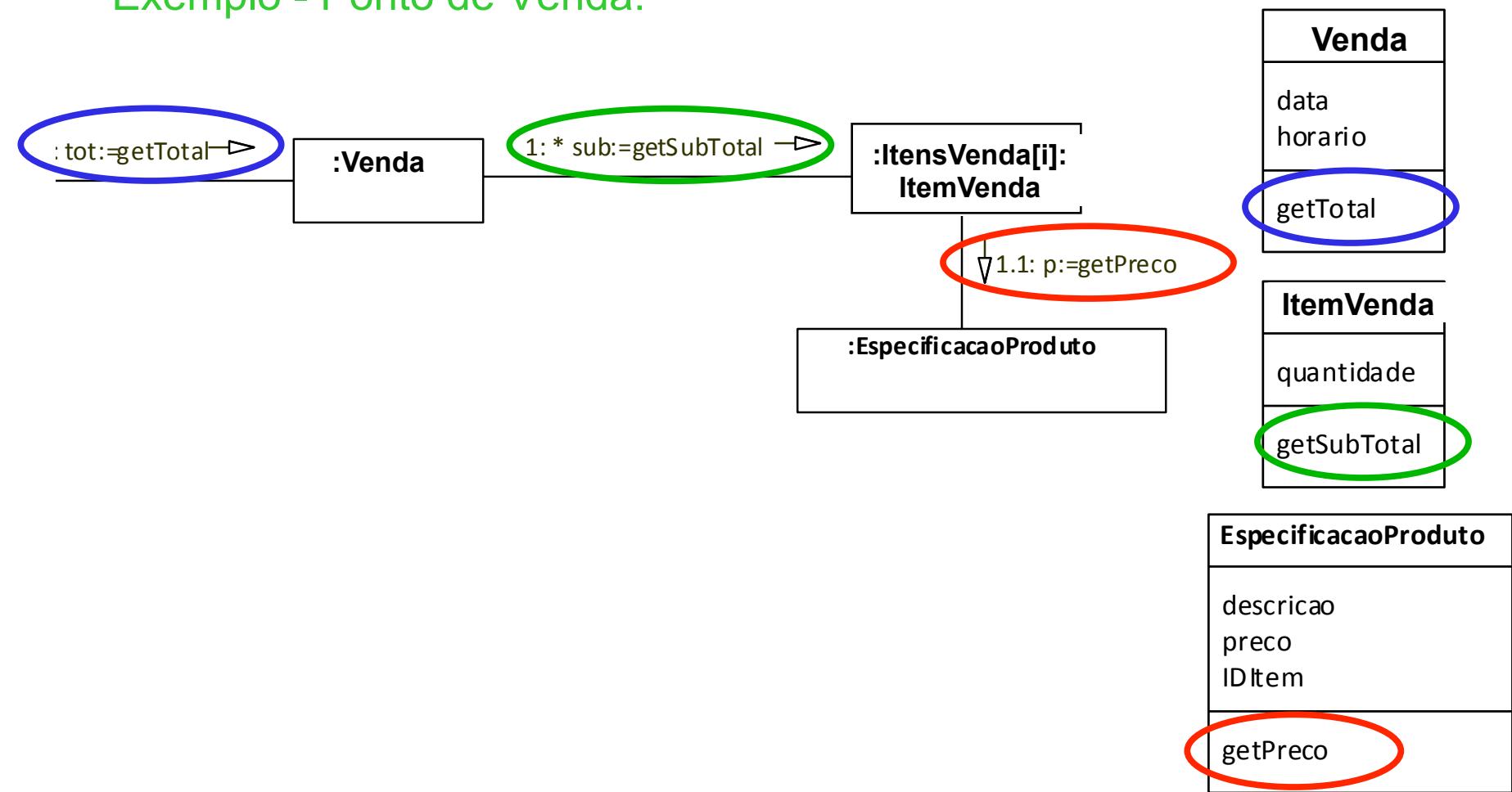
- Como a classe **Venda** conhece todas as instâncias de **ItemVenda**, ela pode ser responsável por conhecer o total da venda.
- Como a classe **ItemVenda** conhece a quantidade dos itens e está associada com a especificação do produto, que possui o preço de cada item, ela pode ser responsável por determinar o sub-total de um item de venda.





Especialista da Informação

Exemplo - Ponto de Venda:

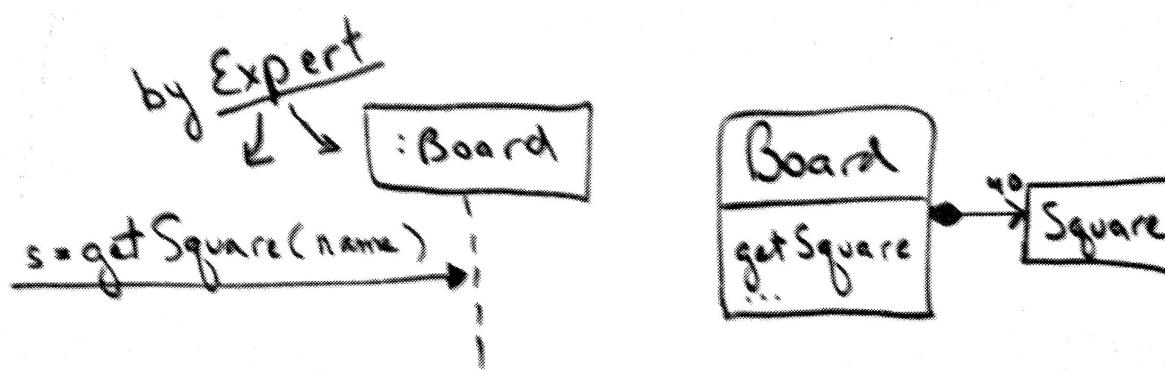




Especialista da Informação

Exemplo - Banco Imobiliário:

- Quem deverá ser o responsável por retornar uma casa (square), a partir de uma chave?
- Como o Tabuleiro agregará todos os objetos casas (squares), ele pode ser a classe responsável por retornar uma casa.





Especialista da Informação

Vantagens:

- O encapsulamento das informações é mantido, pois os objetos usam suas próprias informações para cumprir as tarefas.
- O comportamento é distribuído através das classes que têm as informações necessárias.





Baixo Acoplamento

Durante a definição das operações são considerados:

- Padrão de Projeto Criador
- Padrão de Projeto Especialista da Informação
- **Princípio de Baixo Acoplamento**
- Princípio de Alta Coesão
- Padrão de Projeto Fachada
- Outros Padrões de Projeto





Baixo Acoplamento

Acoplamento é uma medida da intensidade na qual um elemento (classe, subsistema, sistema, etc) está conectado a, depende de ou tem conhecimento a respeito de outros elementos.

Formas comuns de acoplamento em linguagens OO:

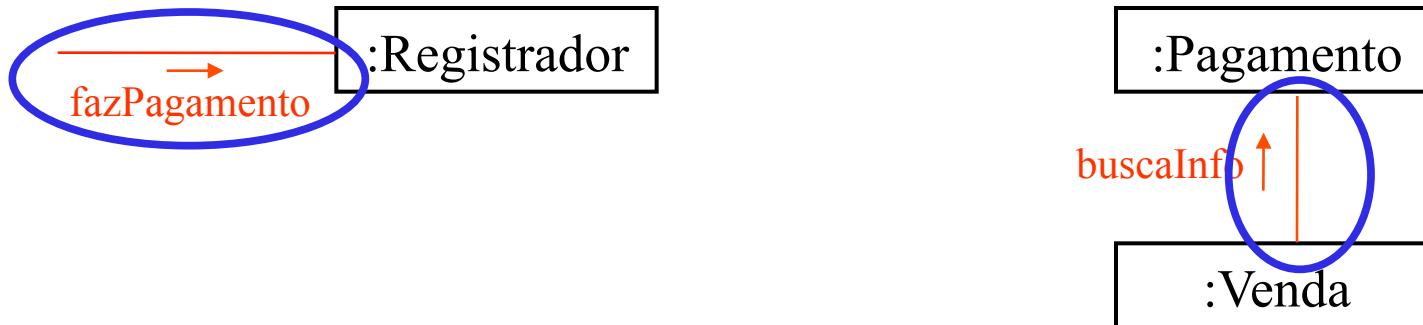
- Classe X tem um atributo que referencia uma instância da classe Y;
- Classe X tem um método que referencia uma instância da classe Y;
- Classe X é uma subclasse direta ou indireta da classe Y;
- Y é uma interface e classe X implementa esta interface.

→ Um elemento com baixo acoplamento não é dependente de muitos outros elementos.



Baixo Acoplamento

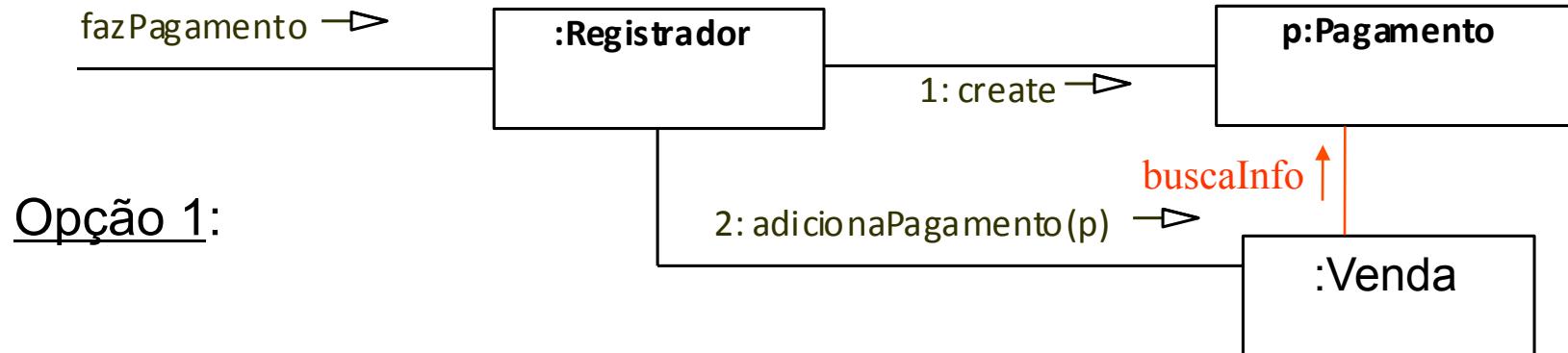
Exemplo - Ponto de Venda:



- Suponha que a venda precise conhecer o pagamento para buscar algumas informações dele, ou seja, ela está associada à instância de Pagamento.
- Suponha que o registrador é o responsável em fazer um pagamento e adicioná-lo à venda.
- Suponha que é necessário criar uma instância de Pagamento.
- Qual classe deverá ser a responsável pela criação de um pagamento?



Baixo Acoplamento



- Adiciona acoplamento de Registrador ao Pagamento e de Registrador à Venda.





Acoplamento

Problemas das classes com alto acoplamento:

- Mudanças nas classes relacionadas podem gerar mudanças locais;
- São mais difíceis de serem entendidas isoladamente;
- São mais difíceis de serem reusadas pois o uso requer a presença adicional das classes das quais ela depende.





Baixo Acoplamento

→ Use o Princípio de Baixo Acoplamento para avaliar alternativas de projeto. Se todo o resto está igual, prefira a solução com baixo acoplamento.





Alta Coesão

Durante a definição das operações são considerados:

- Padrão de Projeto Criador
- Padrão de Projeto Especialista da Informação
- Princípio de Baixo Acoplamento
- **Princípio de Alta Coesão**
- Padrão de Projeto Fachada
- Outros Padrões de Projeto





Alta Coesão

Coesão é uma medida da intensidade na qual as responsabilidades de um elemento (classe, subsistema, sistema, etc) estão relacionadas.

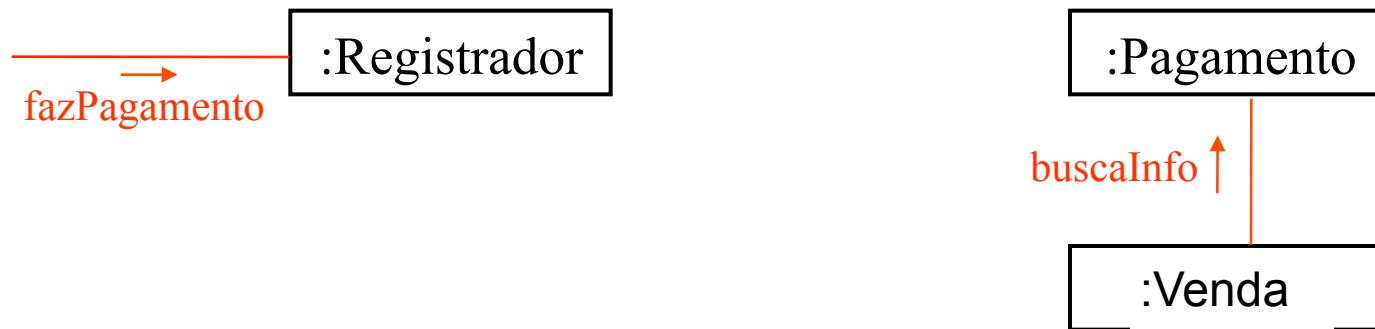
Um elemento com responsabilidades altamente relacionadas e que não tem uma quantidade extremamente grande de trabalho, tem alta coesão.





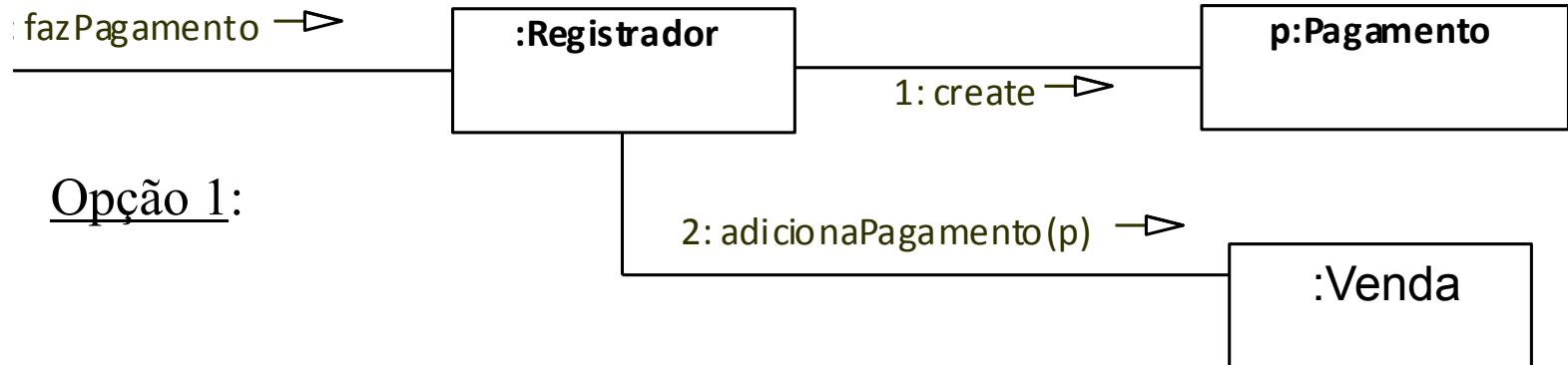
Alta Coesão

Exemplo - Ponto de Venda:

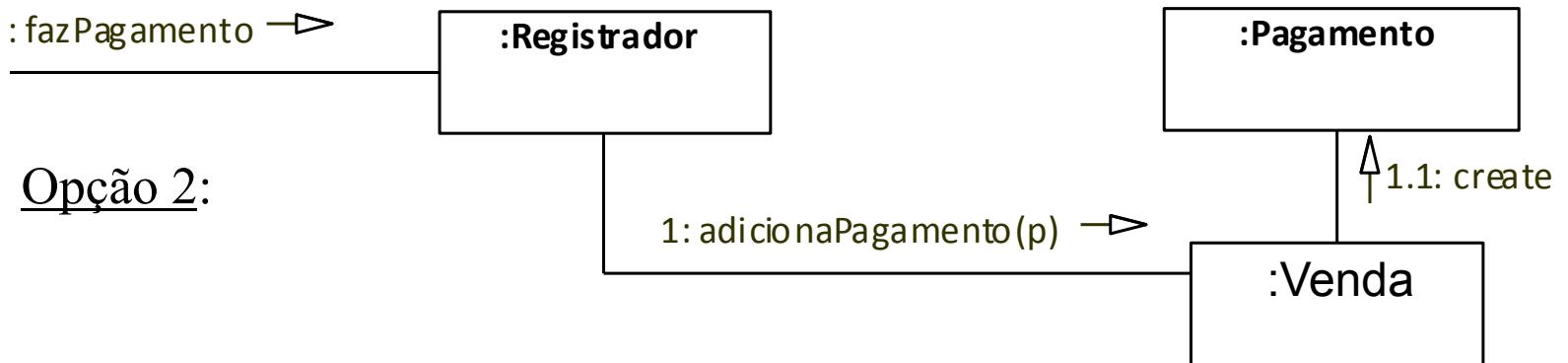


- Suponha que o Registrador será responsável por várias operações do sistema e que é necessário criar uma instância de Pagamento e associá-la a uma instância de Venda.
- Qual projeto suporta alta coesão?

Alta Coesão



- Supondo que o Registrador será responsável por várias outras operações do sistema, ele ficará sobrecarregado de tarefas.

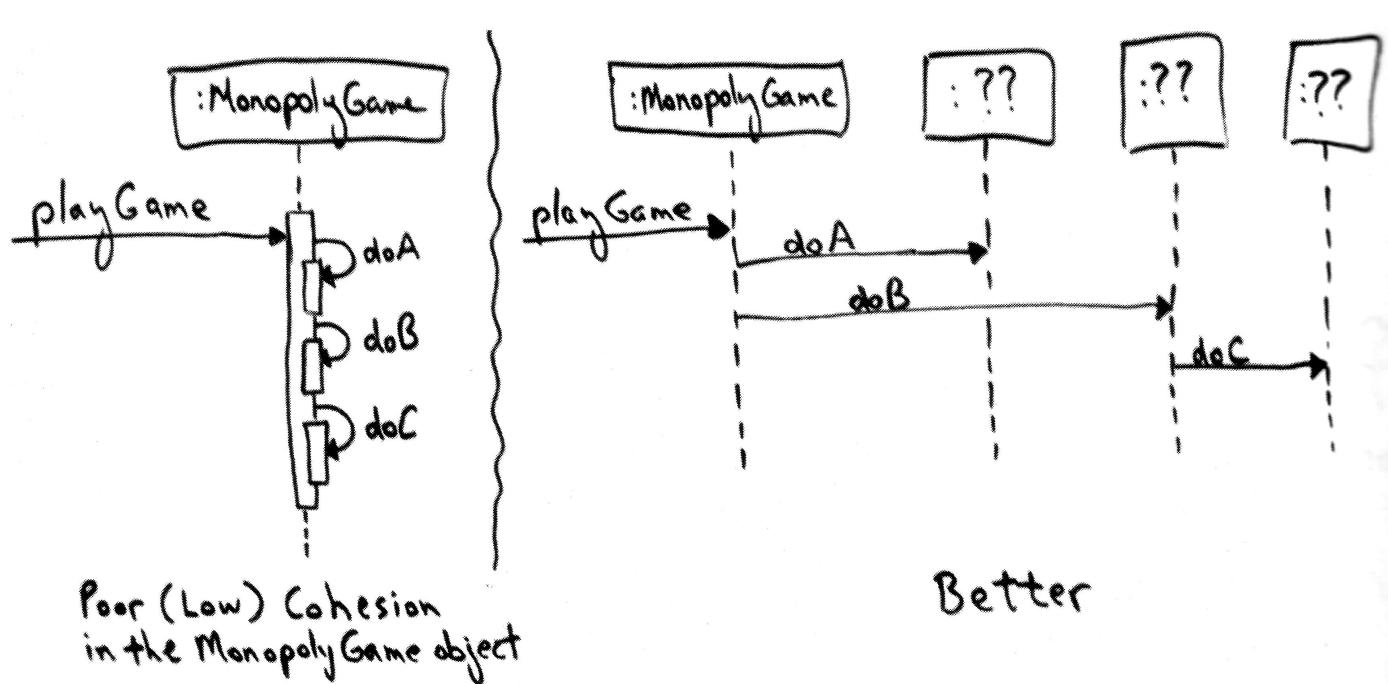


- Delega a responsabilidade da criação do Pagamento para a Venda, dando suporte a uma coesão mais alta no Registrador.



Alta Coesão

Exemplo - Banco Imobiliário:



- Esquerda: PIOR - o próprio objeto Bancomobiliario (`MonopolyGame`) faz todo o trabalho
- Direita: MELHOR - o objeto Bancomobiliario (`MonopolyGame`) delega e distribui o trabalho entre outros objetos.



Coesão

Problemas das classes com baixa coesão (classes que fazem muitas coisas não relacionadas ou fazem trabalhos demais):

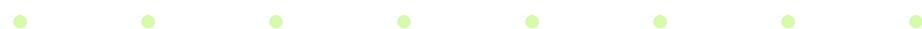
- difícil de compreender;
- difícil de usar;
- difícil de manter;
- constantemente afetada por mudanças.





Alta Coesão

→ Use o Princípio de Alta Coesão para avaliar alternativas de projeto.
Se todo o resto está igual, prefira a solução com alta coesão.





Controlador

Durante a definição das operações são considerados:

- Padrão de Projeto Criador
- Padrão de Projeto Especialista da Informação
- Princípio de Baixo Acoplamento
- Princípio de Alta Coesão
- **Padrão de Projeto Fachada**
- Outros Padrões de Projeto





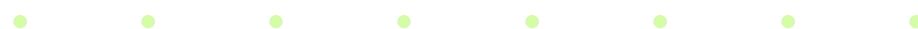
Fachada

Fachada: Classe responsável pelo recebimento e manipulação de eventos gerados por atores externos. Ou seja, classe atrás da camada de interface com o usuário que deverá receber a mensagem da camada de interface com o usuário.

Não é uma classe de interface com o usuário.

→ objetos de interface com o usuário não devem conter lógica da aplicação ou negócio.

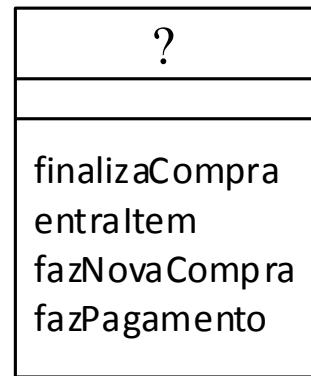
- Classe que representa o sistema como um todo ou um subsistema (**Fachada**).
- Classe que representa um manipulador de todos os eventos do sistema dentro de um caso de uso (**Controlador**).





Fachada

Exemplo - Ponto de Venda:

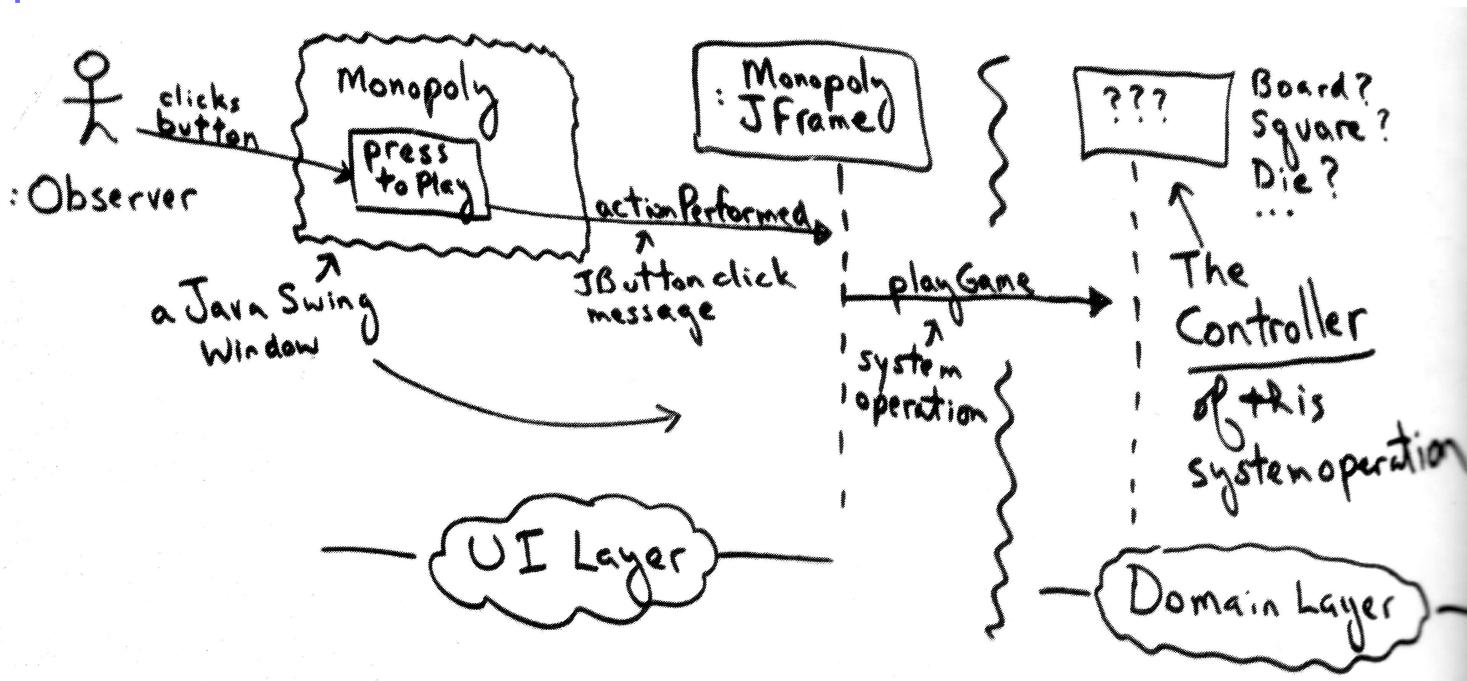


- Quem deverá ser a fachada dos eventos do sistema, como entraItem e finalizaVenda?
 - classe que representa o sistema como um todo ou um subsistema (fachada): **Registrador**



Fachada

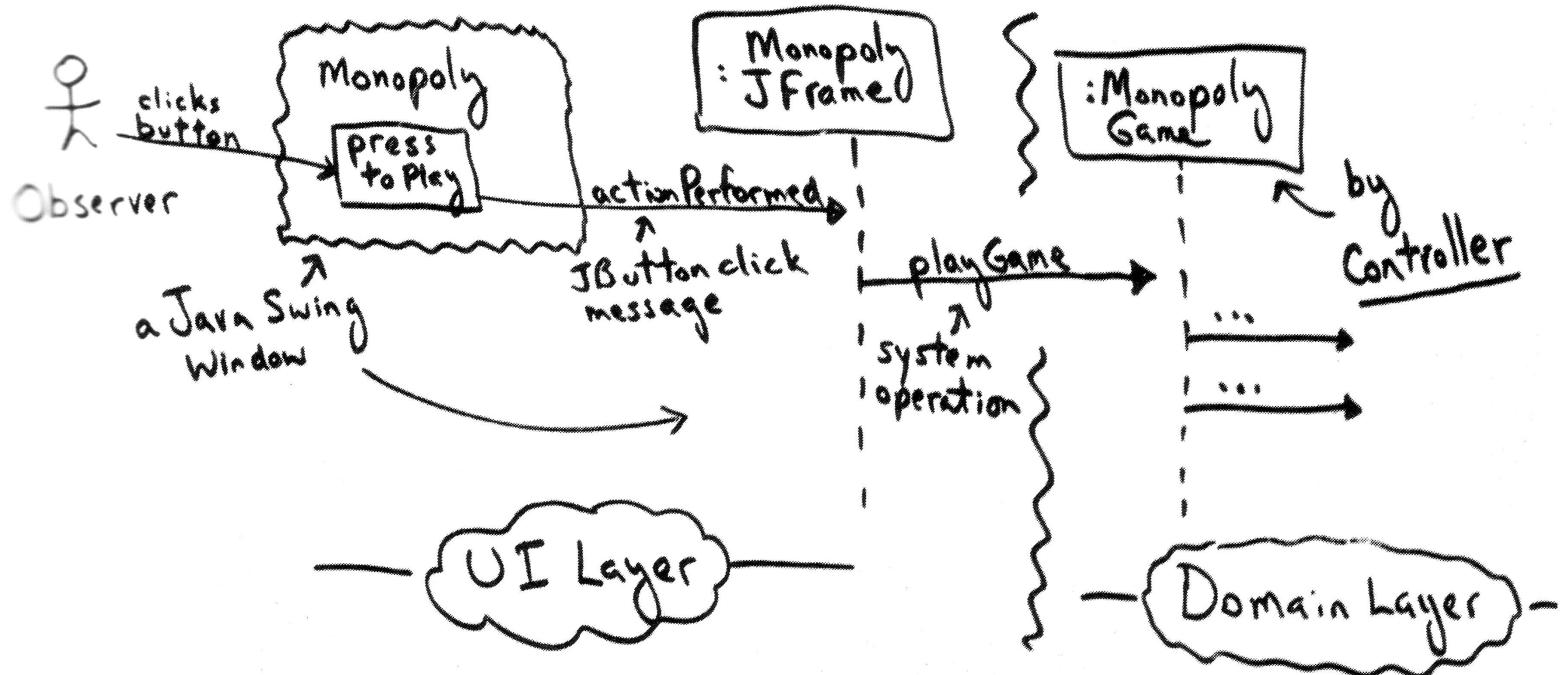
Exemplo - Banco Imobiliário:



- Como conectar a camada de interface com o usuário com a camada de domínio?
- Qual objeto deverá receber a mensagem **playGame** da camada de interface com o usuário?

Fachada

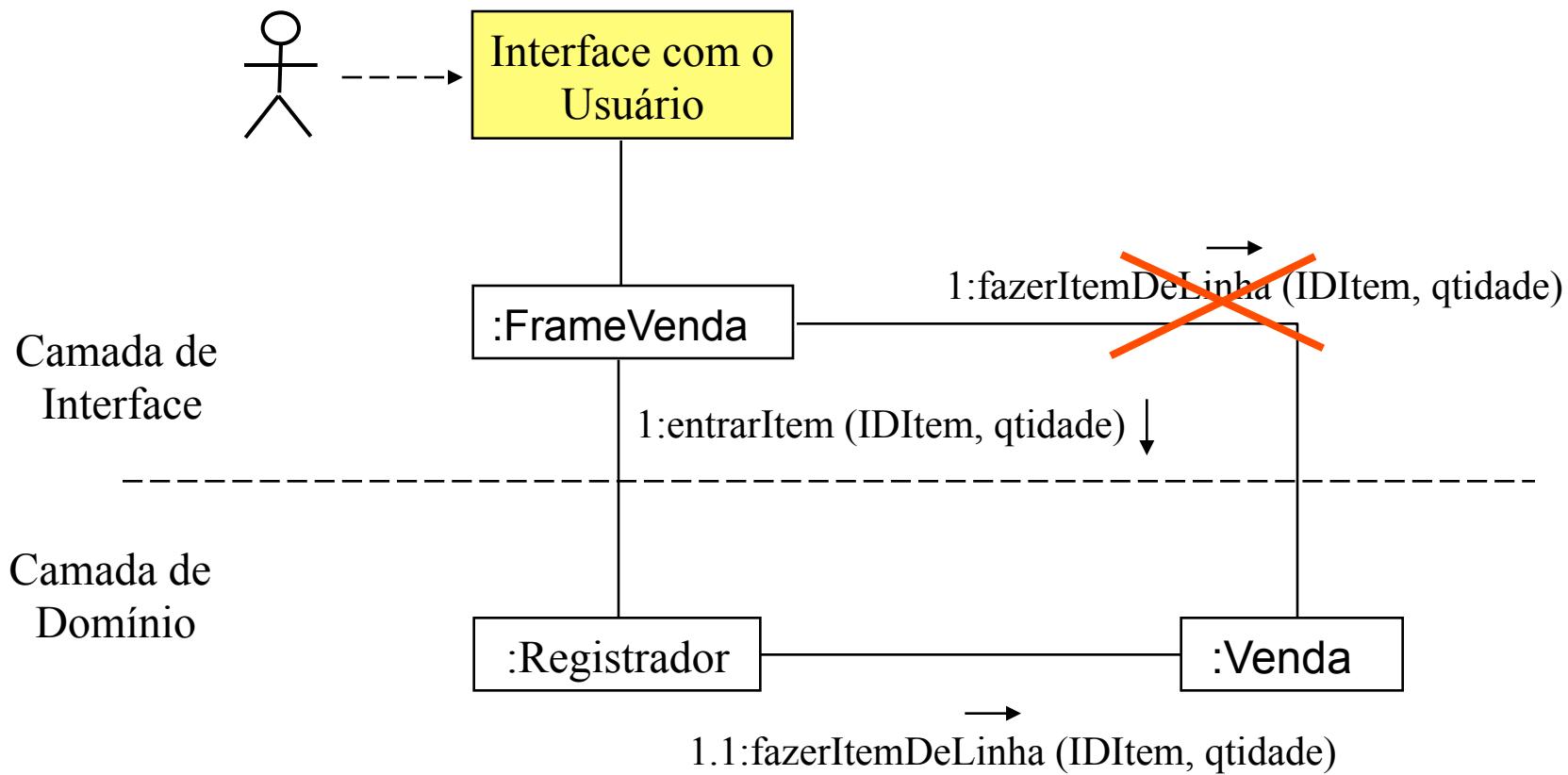
Exemplo - Banco Imobiliário:



- Como existem poucas operações do sistema, podemos definir uma classe que representa todo o sistema, BancoImobiliario (MonopolyGame).

Fachada

Importante: Os objetos de interface não devem ter a responsabilidade de manipular os eventos do sistema.





Exemplos de Definição das Operações do Ponto de Venda





Exemplos de Definição de Operações

Exemplo - Ponto de Venda: caso de uso Processar Venda

Fluxo Básico:

1. O cliente chega no caixa com os produtos e/ou serviços para comprar.
2. O caixa inicia uma nova venda.
3. O caixa entra com o identificador do item.
4. O sistema registra o item e apresenta a sua descrição, preço e o subtotal. O preço calculado é oriundo de um conjunto de regras de preço.

O caixa repete os passos 3 e 4 até indicar que terminou.

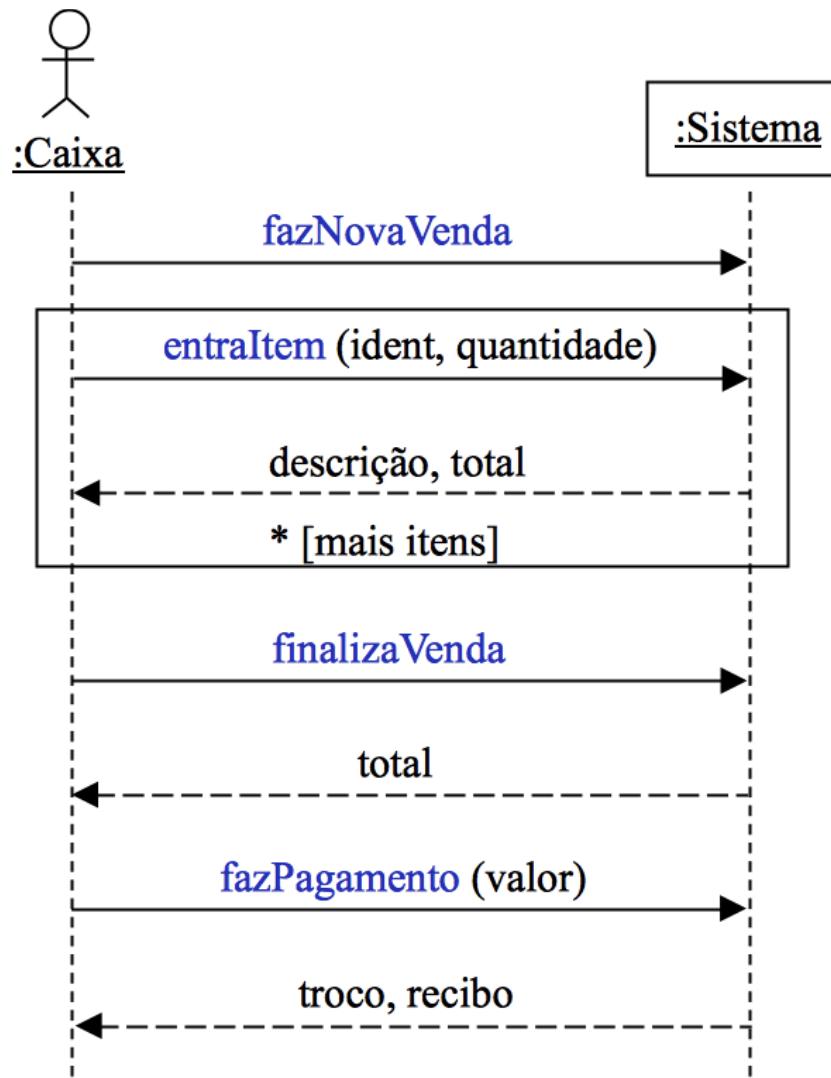
5. O sistema apresenta o total com as taxas calculadas.
6. O caixa diz ao cliente o total e pergunta ao cliente a forma de pagamento.
7. O cliente paga e o sistema processa o pagamento.
8. O sistema registra que a venda foi completada e manda a informação da compra e do pagamento para o sistema externo de contabilidade (para a contabilidade e comissão) e para o sistema de estoque.

...



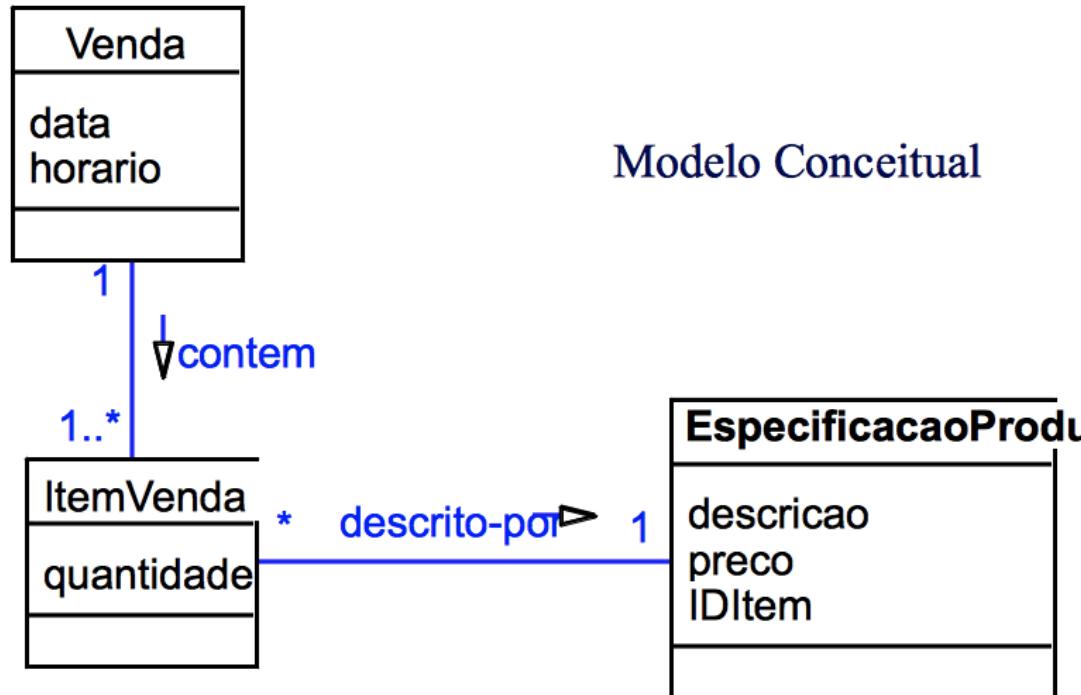


Exemplos de Definição de Operações



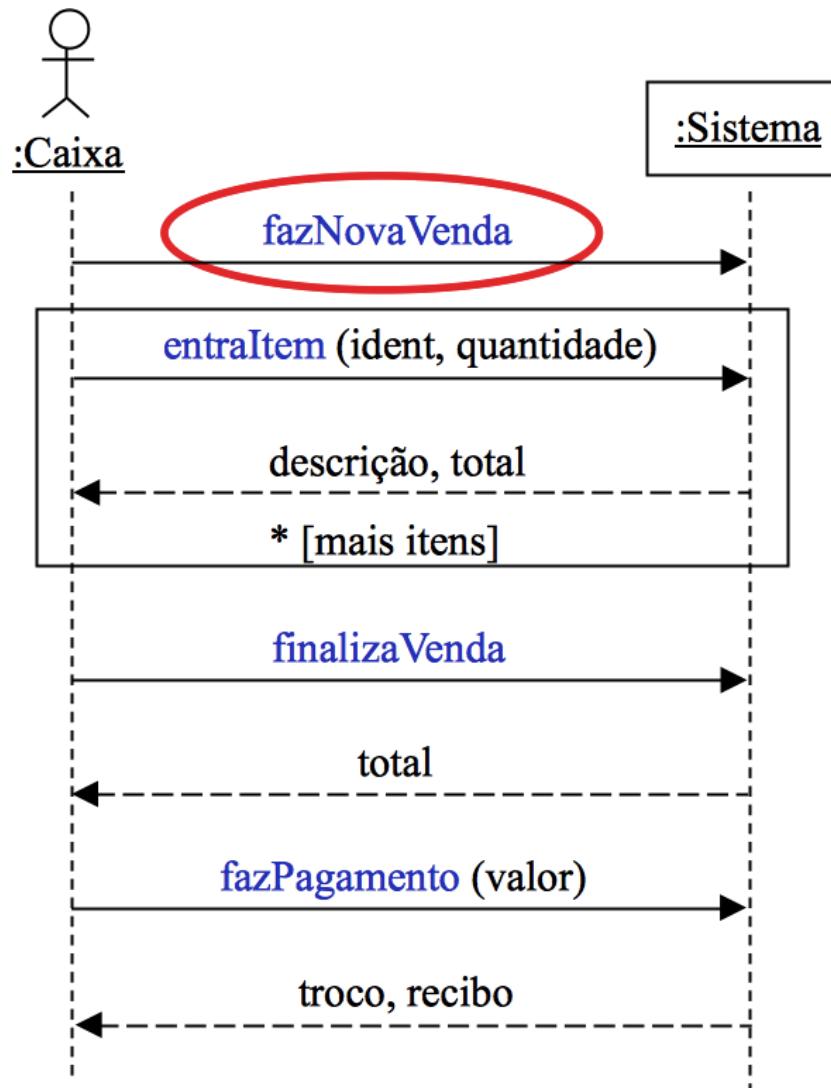


Definição da Operação fazNovaVenda





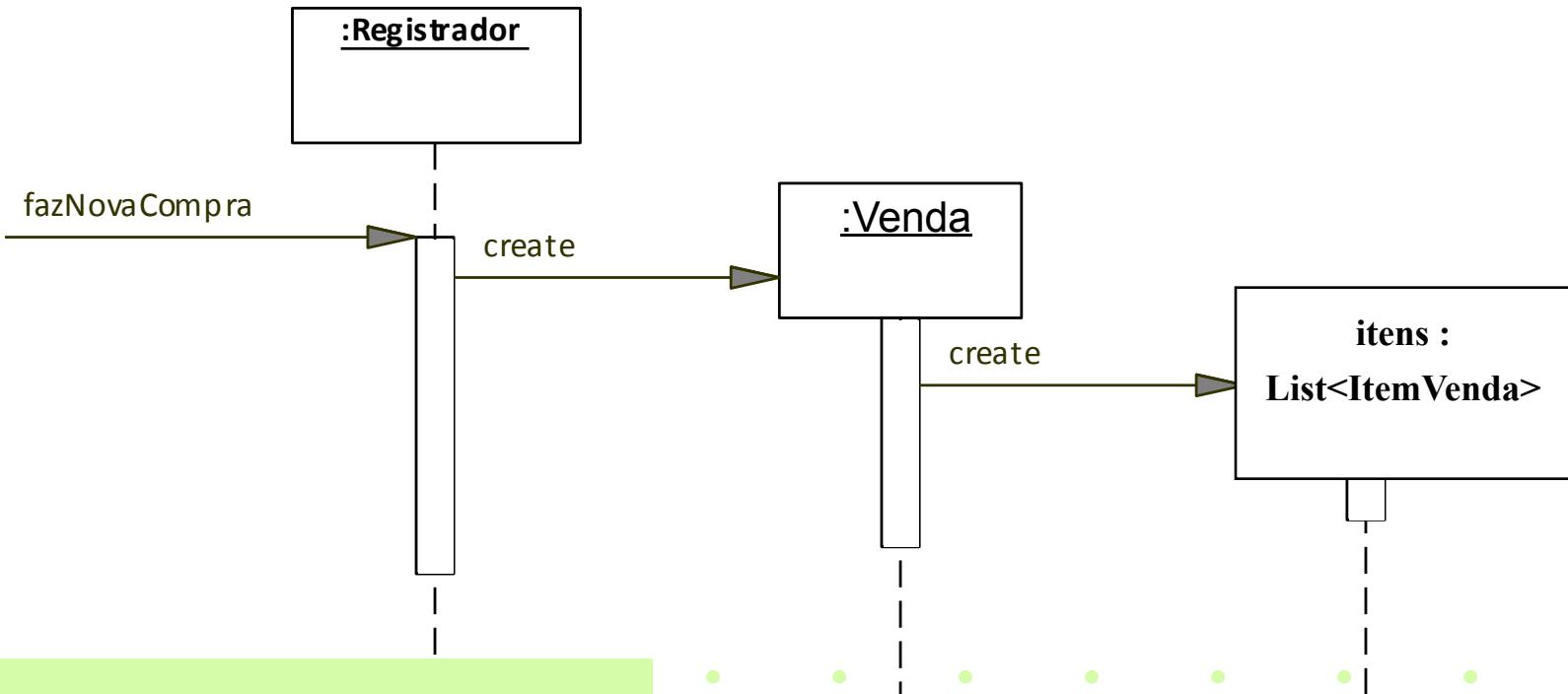
Definição da Operação fazNovaVenda



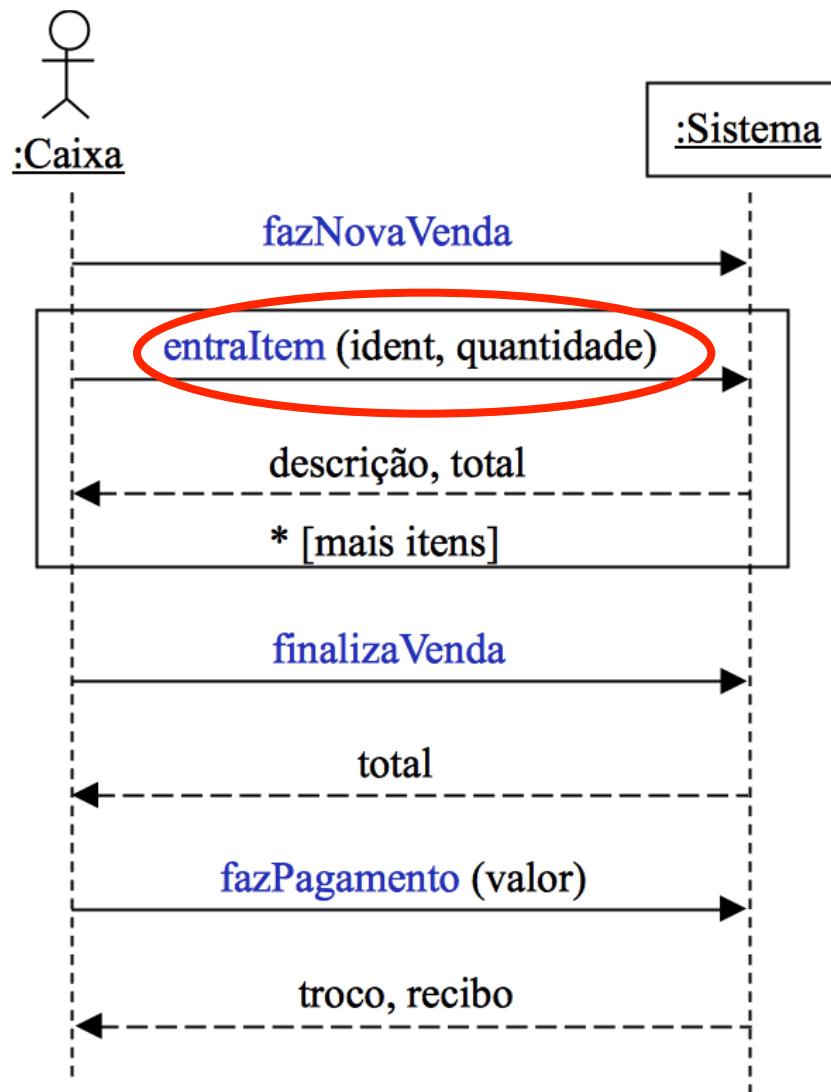


Definição da Operação fazNovaVenda

- Registrador escolhido como fachada porque existem poucas operações do sistema.
- Registrador pode ser visto como aquele que cria a Venda.
- Quando uma venda é criada, ela deve criar uma coleção vazia para registrar todas as instâncias de ItemVenda que serão adicionadas.



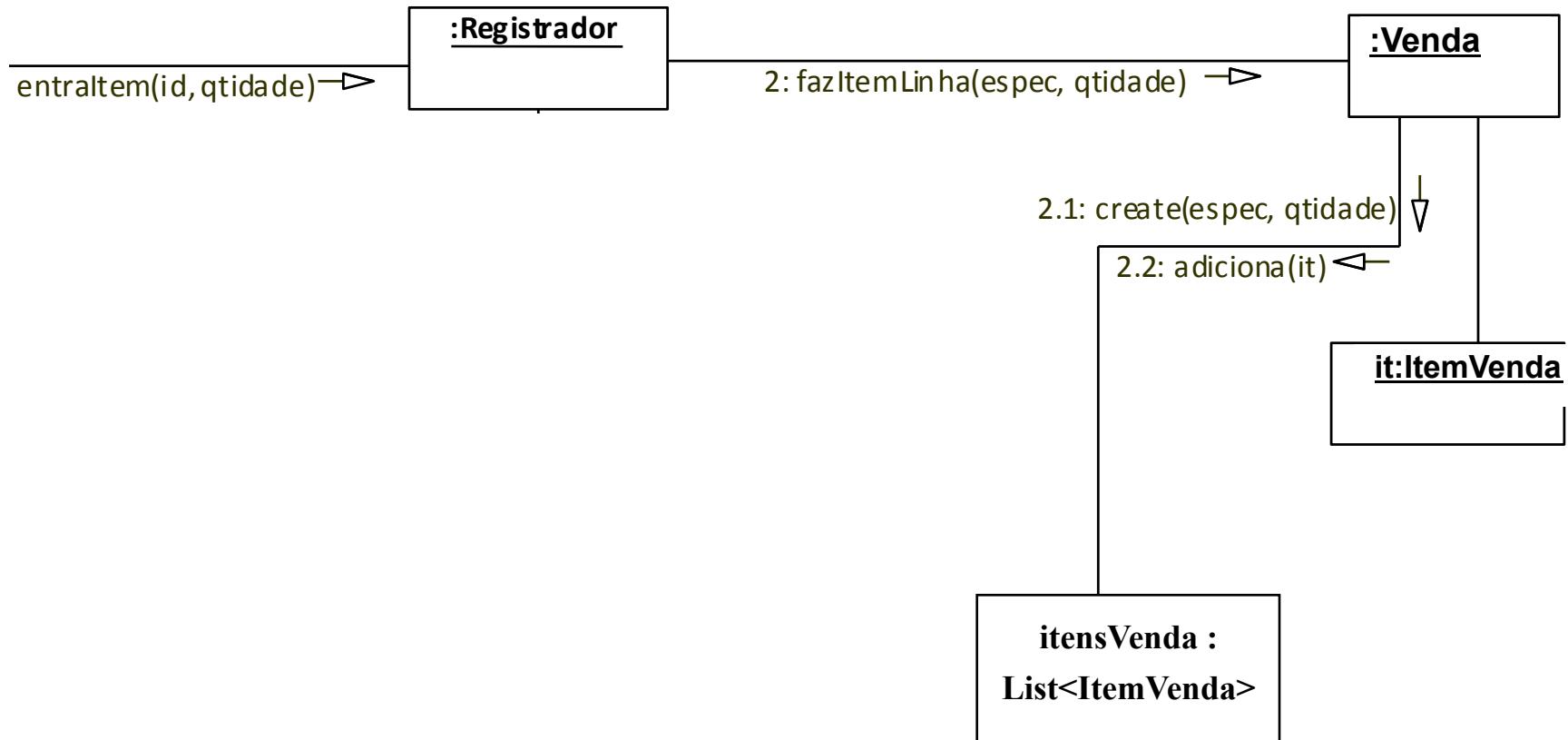
Definição da Operação `entraItem`





Definição da Operação `entraltem`

- Registrador continua como controlador.
- Venda cria um item de venda porque ela contém itens de venda. O item de venda criado pode ser armazenado na coleção de itens



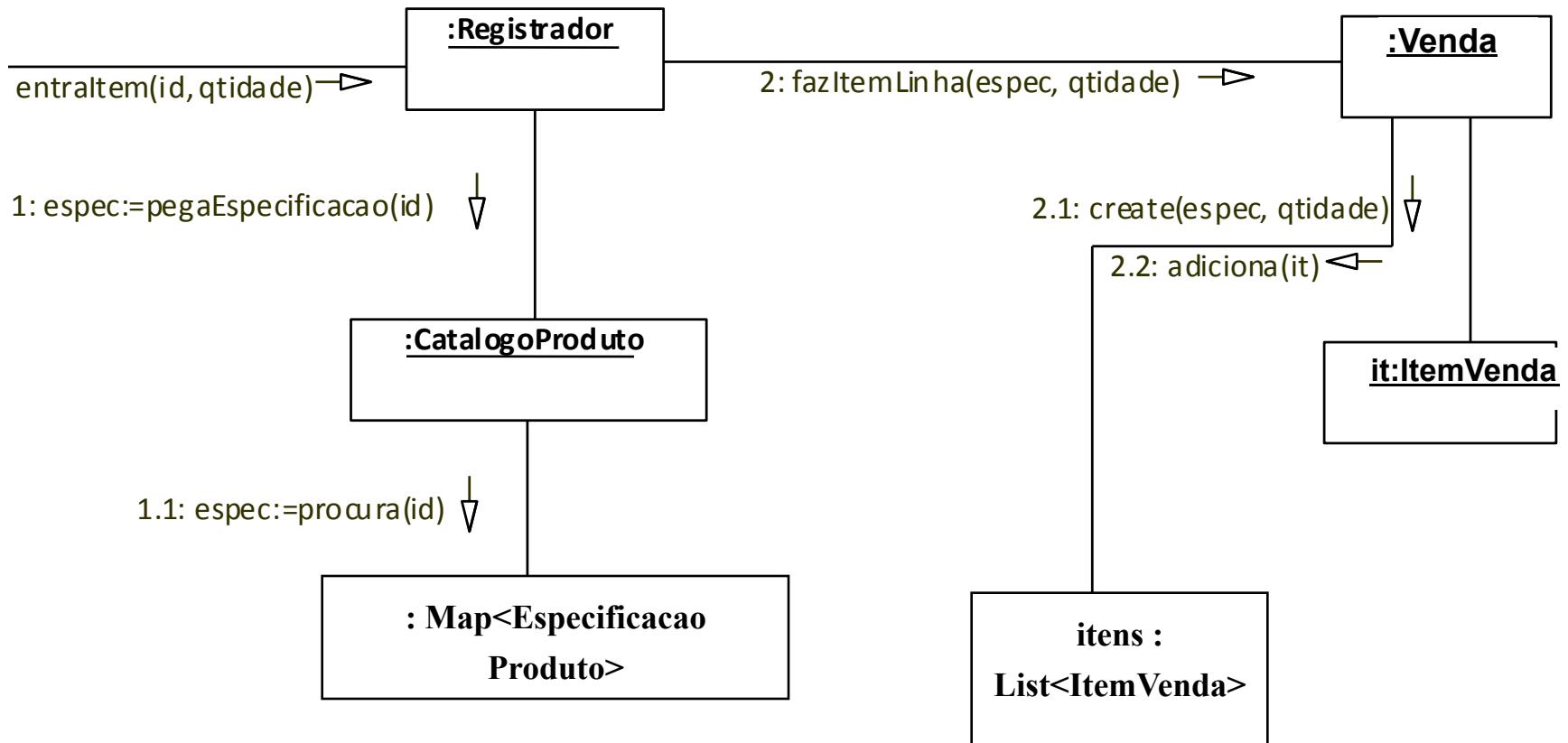


Definição da Operação entraltem

- Registrador continua como fachada.
- Venda cria um item de linha porque ela contém itens de venda. O item de venda criado pode ser armazenado na coleção de itens de venda.
- A criação de um item de venda precisa da quantidade (parâmetro fornecido na operação) e da especificação do produto. Registrador envia mensagem para o CatalogoProduto para buscar a especificação do produto, pois ele conhece as especificações dos produtos.



Definição da Operação `entraltem`





Definição da Operação inicializar

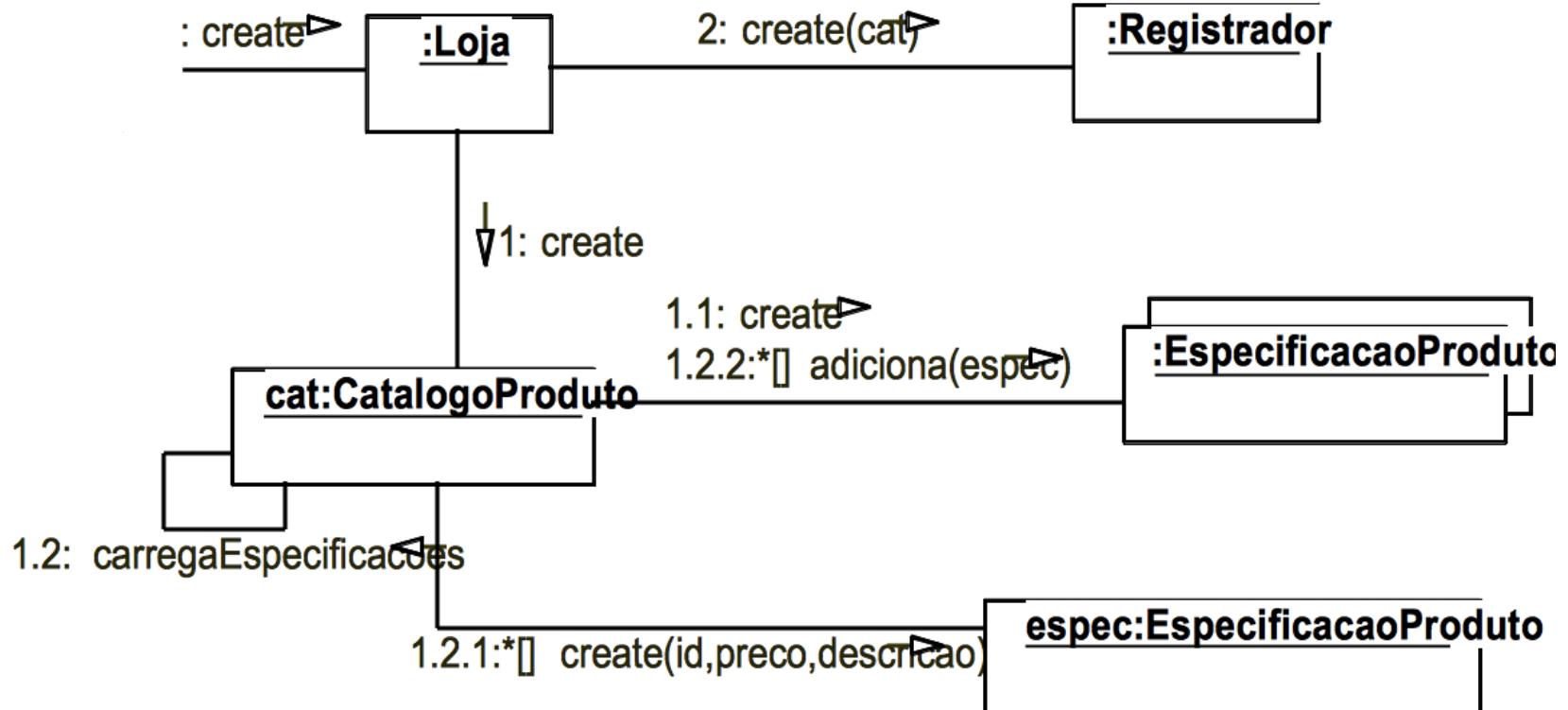
Exemplo: inicialização do sistema ([Start Up](#))

O diagrama de interação da operação de inicialização da aplicação é definido após todas as operações do sistema terem sido projetadas.

- São criados os objetos “iniciais” da aplicação.
- O objeto inicial é uma classe na raiz da hierarquia de agregação dos objetos do domínio (ou perto dela): uma fachada ou um objeto que contém a maioria dos outros objetos.
- Se a aplicação tem interface gráfica, geralmente, o objeto inicial não controla o processo.



Definição da Operação inicializar





Exemplo de Definição de Operações

Conexão da Camada de Interface com o Usuário e da Camada de Domínio

- A camada de interface com o usuário não deverá ter qualquer responsabilidade lógica do domínio;
- A camada de interface com o usuário deverá ser responsável somente pelas tarefas de interface com o usuário, como a atualização dos widgets;
- A camada de interface com o usuário deverá reenviar todas as requisições sobre as tarefas do domínio para a camada de domínio.



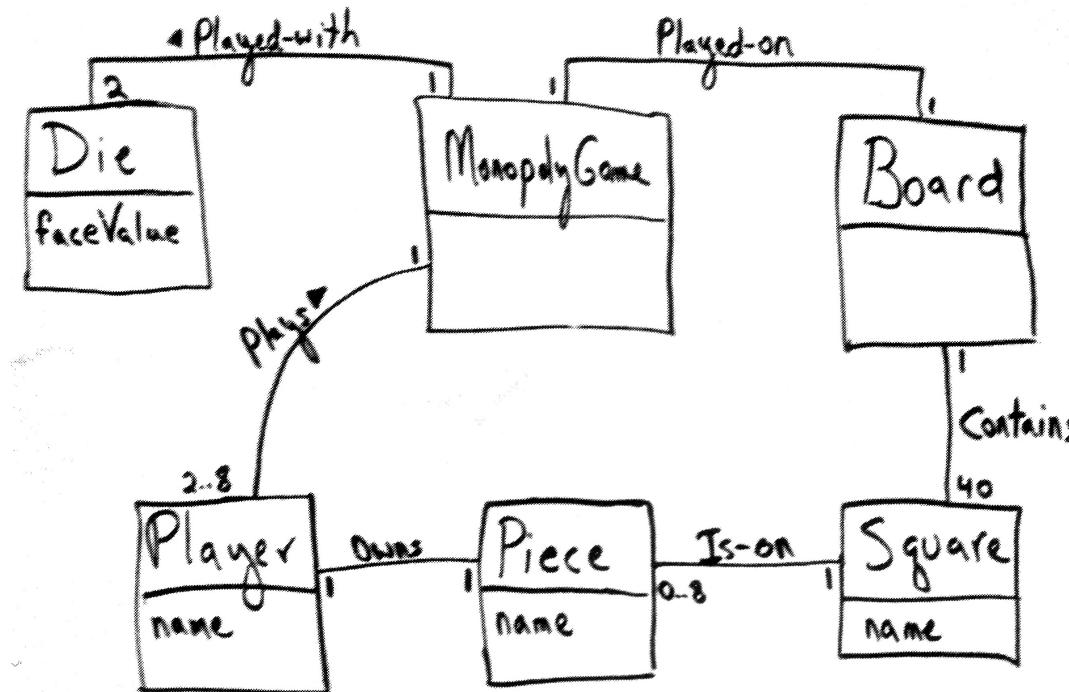


Exemplos de Definição das Operações do Banco Imobiliário



Exemplo do Banco Imobiliário

Resultado do Modelo Conceitual



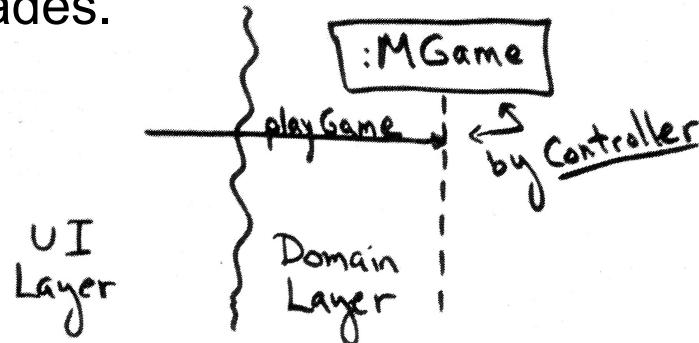
→ será usado como inspiração para o projeto da camada de domínio do Modelo de Projeto



Definição da Operação Jogar

Exemplo 1: Como projetar a operação de sistema jogar (playGame)?

1. Escolher a fachada: JogoBancolmobilario satisfaz porque existem poucas operações do sistema e a fachada não tem muitas responsabilidades.



2. Escolher o responsável pelo algoritmo Game-Loop.

para N rodadas

para cada Jogador j

j faz uma jogada

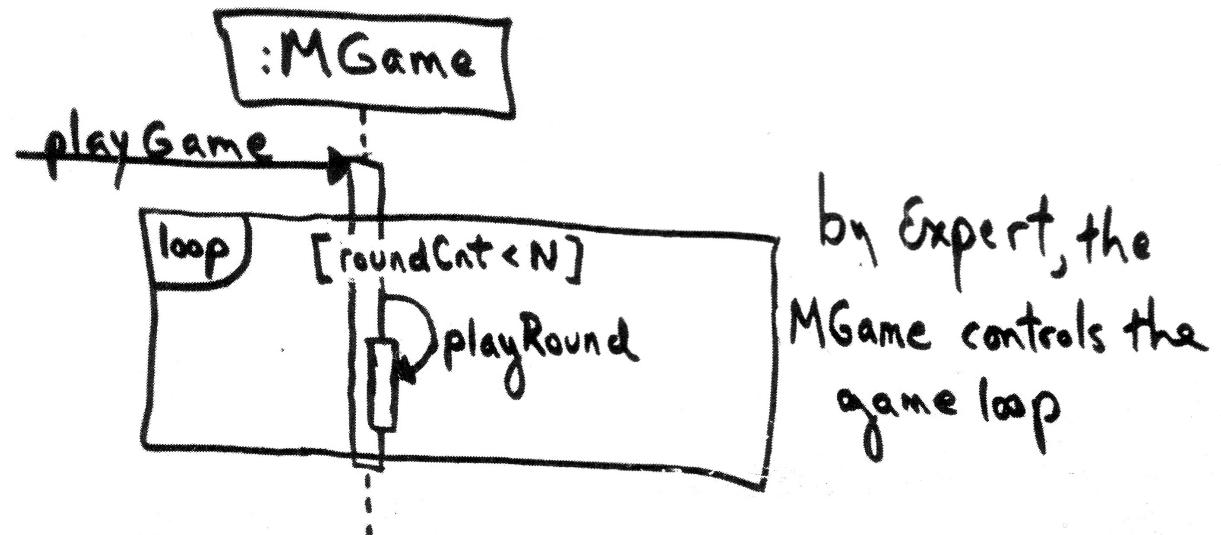
- jogada (turn): um jogador rola os dados e move a peça
- rodada (round): todos os jogadores fazem uma jogada



Definição do Algoritmo Game-Loop

2A. Quem será o responsável pelo algoritmo Game-Loop?

- É necessário conhecer o contador da rodada atual: nenhum objeto tem esta informação, mas, para ficar mais próximo da realidade, o BancoImobiliario poderia ter
- É necessário conhecer todos os jogadores: baseando-se no modelo de domínio, o BancoImobiliario é um bom candidato
- BancoImobiliario irá controlar o loop do jogo e coordenar as jogadas de cada rodada.





Definição do Algoritmo Game-Loop

2B. Quem será o responsável pela jogada (turn).

- É necessário conhecer a localização atual do jogador
 - baseando-se no modelo de domínio, uma Peça conhece a sua Casa e um Jogador conhece a sua Peça. Portanto, um objeto Jogador poderia conhecer sua localização.
- É necessário conhecer os dois objetos Dados
 - baseando-se no modelo de domínio, o Bancolmobilario é um bom candidato porque os dados podem ser vistos como parte do jogo.
- É necessário conhecer todas as casas (para mover para a nova casa)
 - para ficar mais próximo da realidade, o Tabuleiro é um bom candidato





Definição do Algoritmo Game-Loop

Qual classe escolher, Jogador, Bancolmobilario ou Tabuleiro?

- Guideline: Quando existem vários especialistas da informação, colocar a responsabilidade no objeto que tem a maioria das informações.
→ nenhuma classe tem a maioria das informações
- Guideline: Quando existem alternativas de projeto, considerar o acoplamento e coesão.
→ Bancolmobilario já possui responsabilidades, o que pode diminuir sua coesão, principalmente em relação a Jogador e Tabuleiro, que não possuem nenhuma responsabilidade





Definição do Algoritmo Game-Loop

Qual classe escolher, Jogador, Bancolmobiliario ou Tabuleiro?

- Guideline: Considerar uma evolução futura dos objetos de software.
→ Mais tarde, executar uma jogada pode envolver a compra de uma propriedade, se o jogador tem dinheiro suficiente, ou se a cor da propriedade encaixa com a estratégia do Jogador.

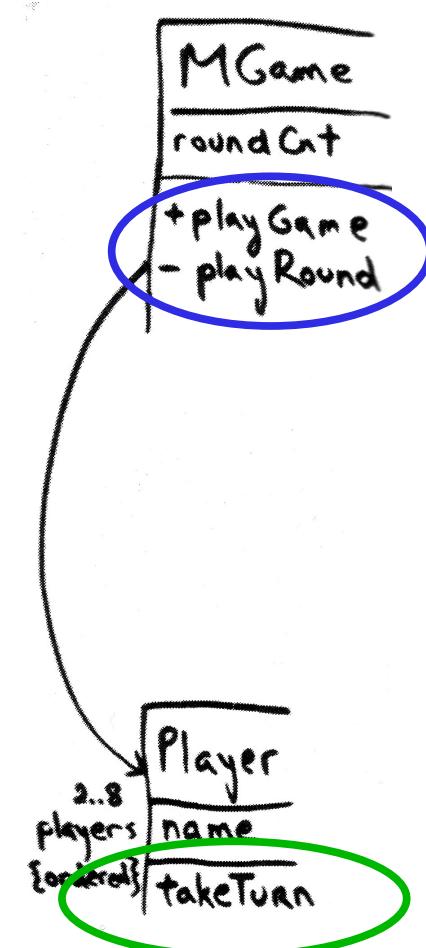
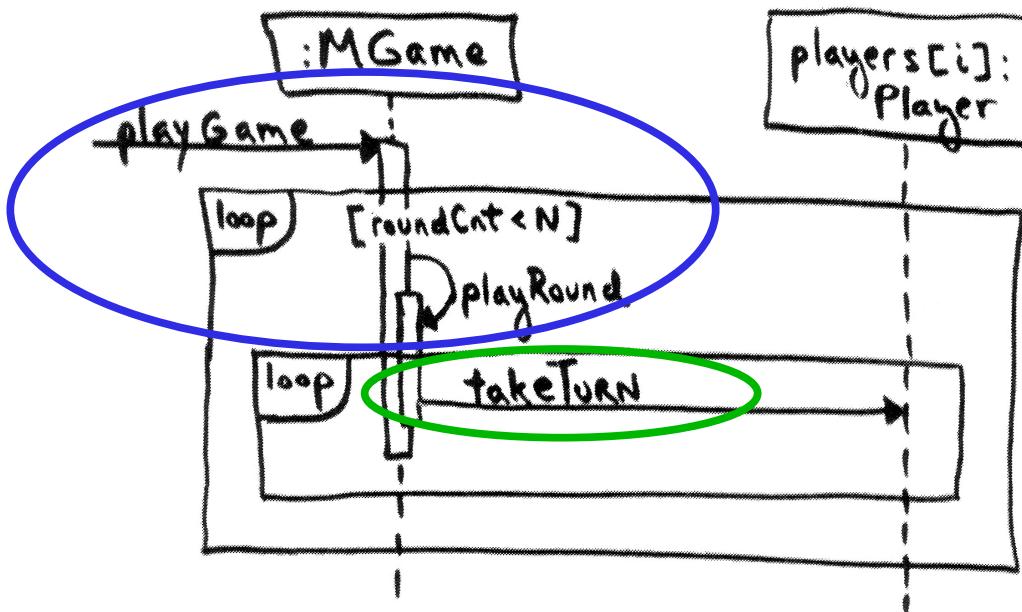
Qual o objeto indicado para saber do dinheiro do Jogador? O Jogador.

Qual o objeto indicado para saber da cor do Jogador? O Jogador.



Definição do Algoritmo Game-Loop

→ Jogador é um bom candidato, considerando as regras completas do jogo.





Definição do Algoritmo Game-Loop

2C. Execução de uma jogada (take a turn)

Executar uma jogada significa:

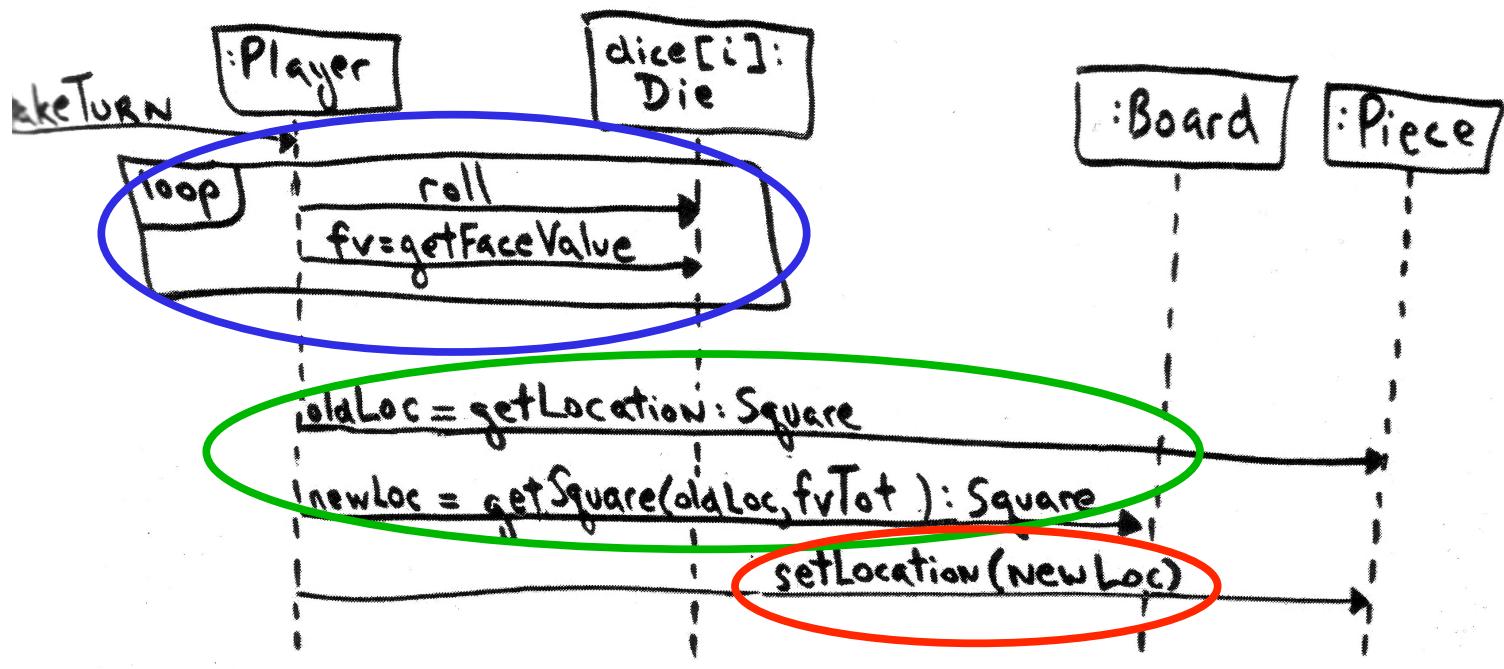
1. Calcular um número random entre 2 e 12 (soma dos 2 dados)
2. Calcular a localização da nova casa
3. Mover a peça do jogador da casa antiga para a nova casa

Estes passos serão coordenados pelo Jogador, já que ele é o responsável por executar uma jogada.

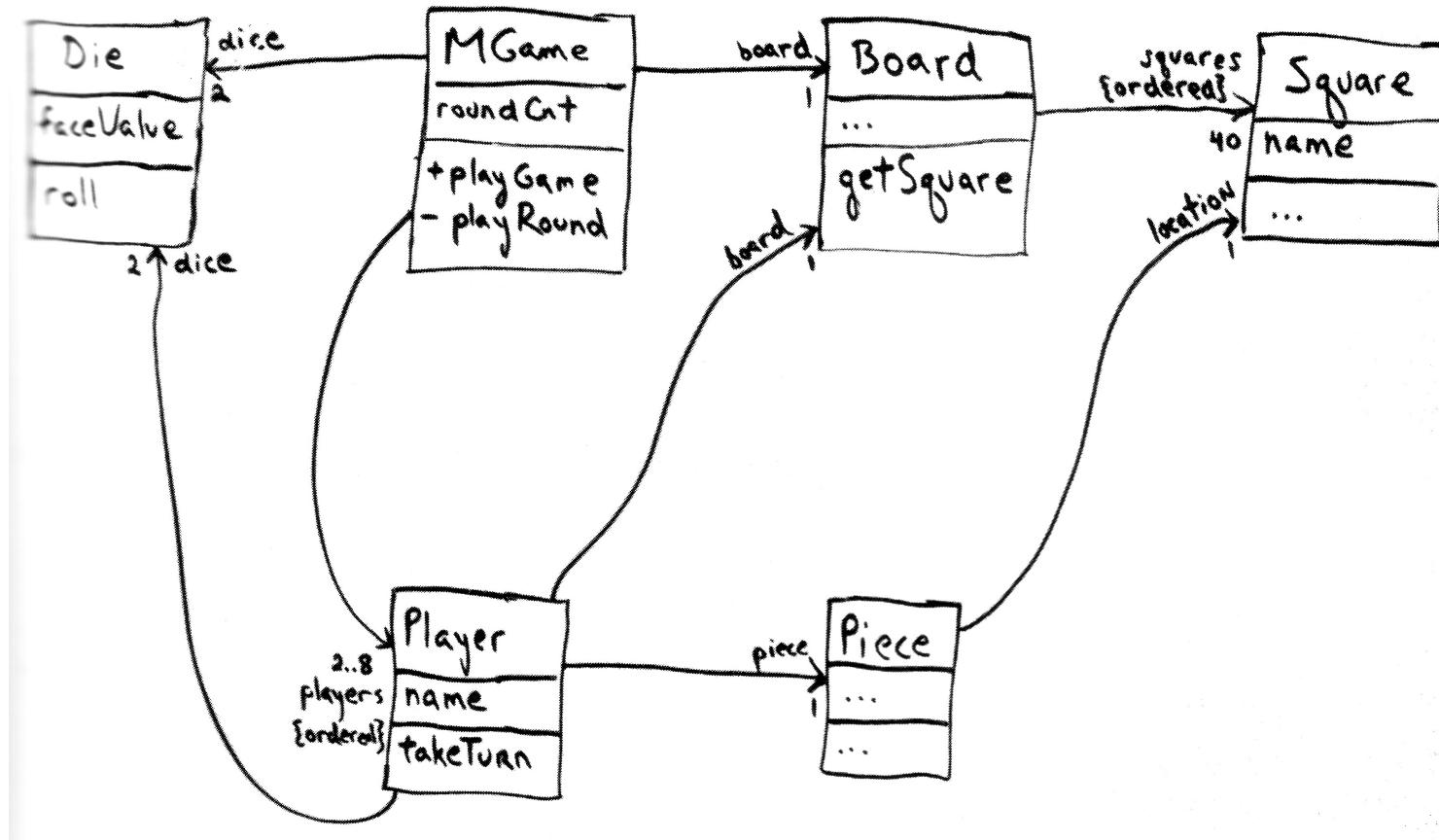


Definição do Algoritmo Game-Loop

1. Calcular um número random entre 2 e 12 (soma dos 2 dados)
2. Calcular a localização da nova casa
3. Mover a peça do jogador da casa antiga para a nova casa



Exemplo do Banco Imobiliário



Definição da Operação Inicializar

Definição da Operação do Sistema inicializar do caso de uso StartUp

- É necessário escolher o objeto raiz que criará alguns objetos
- BancoMobilario é um bom candidato

