

Análise e Projeto de Sistemas

Universidade Federal do Ceará – UFC
Campus de Quixadá

Prof. Marcos Antonio de Oliveira
(deoliveira.ma@gmail.com)

Somente após a construção de diagramas de interação para os cenários de um caso de uso, pode-se ter certeza de que todas as responsabilidades que os objetos devem cumprir foram identificadas.”
(Ivar Jacobson)

MODELAGEM DE INTERAÇÕES

Esses slides são uma adaptação das notas de aula do professor Eduardo Bezerra autor do livro Princípios de Análise e Projeto de Sistemas com UML

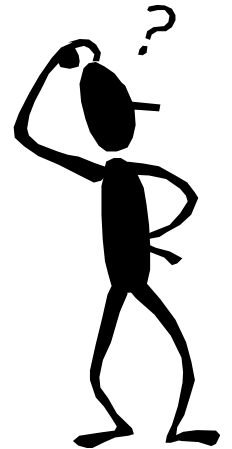
Índice

- Introdução
- Diagrama de seqüência
- Diagrama de comunicação
- Modularização de interações
- Construção do modelo de interações
- Modelo de interações em um processo iterativo

INTRODUÇÃO

Introdução

- O objetivo dos modelos vistos até agora é fornecer um entendimento do problema correspondente ao sistema a ser desenvolvido
- Entretanto, esses modelos deixam algumas perguntas sem respostas



Introdução

No Modelo de Casos de Uso

Quais são as operações que devem ser executadas internamente ao sistema?

A que classes estas operações pertencem?

Quais objetos participam da realização deste caso de uso?

No Modelo de Classes de Análise

De que forma os objetos colaboram para que um determinado caso de uso seja realizado?

Em que ordem as mensagens são enviadas durante esta realização?

Que informações precisam ser enviadas em uma mensagem de um objeto a outro?

Será que há responsabilidades ou mesmo classes que ainda não foram identificadas?

Sessões CRC pode ajudar a identificar quais são as responsabilidades de cada objeto e com que outros objetos ele precisa colaborar. Mas sessões CRC não fornecem um modo de documentar essas interações.

Introdução

- Para responder às questões anteriores, o modelo de ***interações deve ser criado***
- Esse modelo representa mensagens trocadas entre objetos para a execução de cenários dos casos de uso do sistema
- A construção dos ***diagramas de interação*** é uma consolidação do entendimento dos aspectos dinâmicos do sistema, iniciado nas sessões CRC
- A modelagem de interações é uma parte da modelagem dinâmica de um sistema OO

Introdução

Diagramas de interação representam como o sistema age internamente para que um ator atinja seu objetivo na realização de um caso de uso. A modelagem de um sistema OO normalmente contém diversos diagramas de interação. O conjunto de todos os diagramas de interação de um sistema constitui o seu ***modelo de interações***.

Introdução

- Os objetivos da construção do modelo de interação são
 1. Obter informações adicionais para completar e aprimorar outros modelos
 - Quais as operações de uma classe?
 - Quais os objetos participantes da realização de um caso de uso (ou cenário deste)?
 - Para cada operação, qual a sua assinatura?
 - Uma classe precisa de mais atributos?
 2. Fornecer aos programadores uma visão detalhada dos objetos e mensagens envolvidos na realização dos casos de uso

Mensagem

- O conceito básico da interação entre objetos é a **mensagem**
- Um sistema OO é uma rede de objetos que trocam mensagens
 - Funcionalidades são realizadas pelos objetos, que só podem interagir através de mensagens
 - Um objeto envia uma mensagem para outro objeto quando o primeiro deseja que o segundo realize alguma tarefa

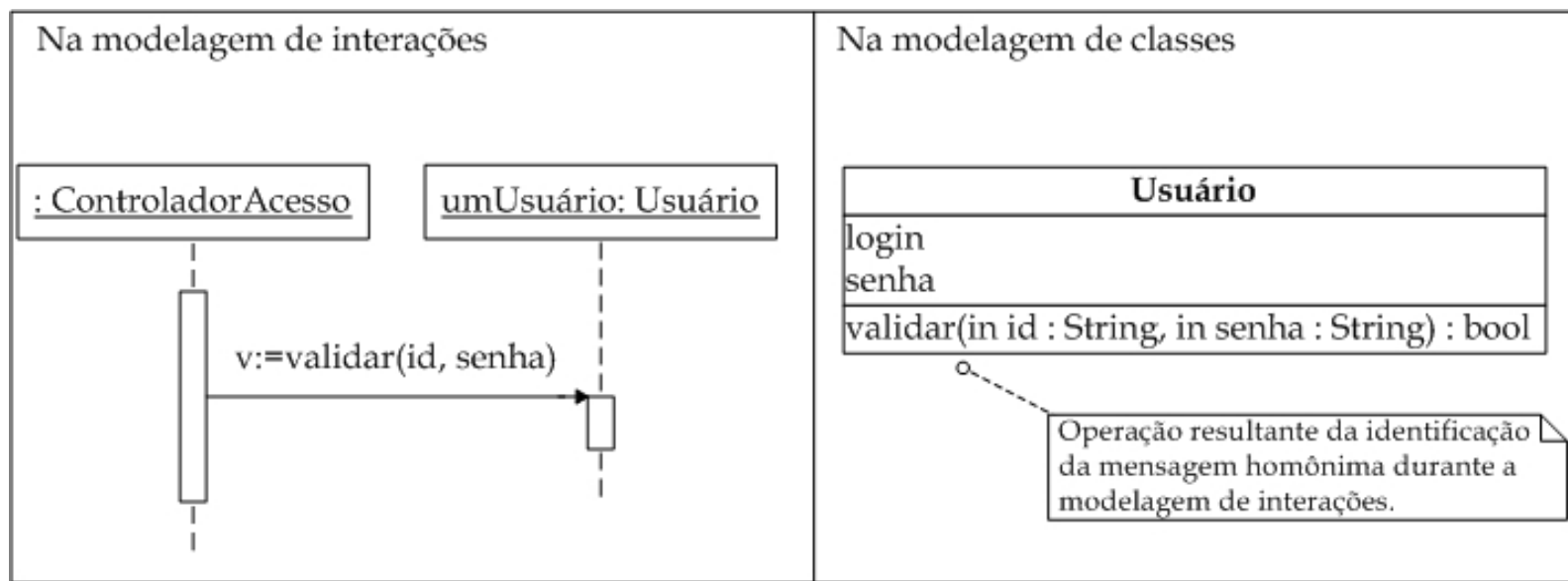
Mensagem

- O fato de um objeto “precisar de ajuda” indica a necessidade de este **enviar mensagens**
- Na construção de diagramas de interação, mensagens de um objeto a outro implicam em **operações** que classes **devem ter**

Uma mensagem representa a requisição de um objeto remetente a um objeto receptor para que este último execute alguma operação definida para sua classe. Essa mensagem deve conter informação suficiente para que a operação do objeto receptor possa ser executada.

Mensagens versus Responsabilidades

- Qual o objetivo da construção dos diagramas de interação?
 - Identificar **mensagens** e, em última análise, **responsabilidades (operações e atributos)**



Uma mensagem implica na existência de uma operação no objeto receptor. A resposta do objeto receptor ao recebimento de uma mensagem é a execução da operação correspondente.

Sintaxe UML para Mensagens

- O rótulo de uma mensagem deve seguir a seguinte sintaxe

[[expressão-seqüência] controle:] [v :=] nome [(argumentos)]

- Onde o termo controle pode ser uma **condição** ou um **iteração**

*[cláusula-iteração]

[cláusula-condição]

- O único termo **obrigatório** corresponde ao **nome** da mensagem

Exemplos (Mensagens UML)

Mensagem simples, sem cláusula alguma

1: adicionarItem(item)

Mensagem com cláusula de condição

3 [a > b]: trocar(a, b)

Mensagem com cláusula de iteração e com limites indefinidos

2 *: desenhar()

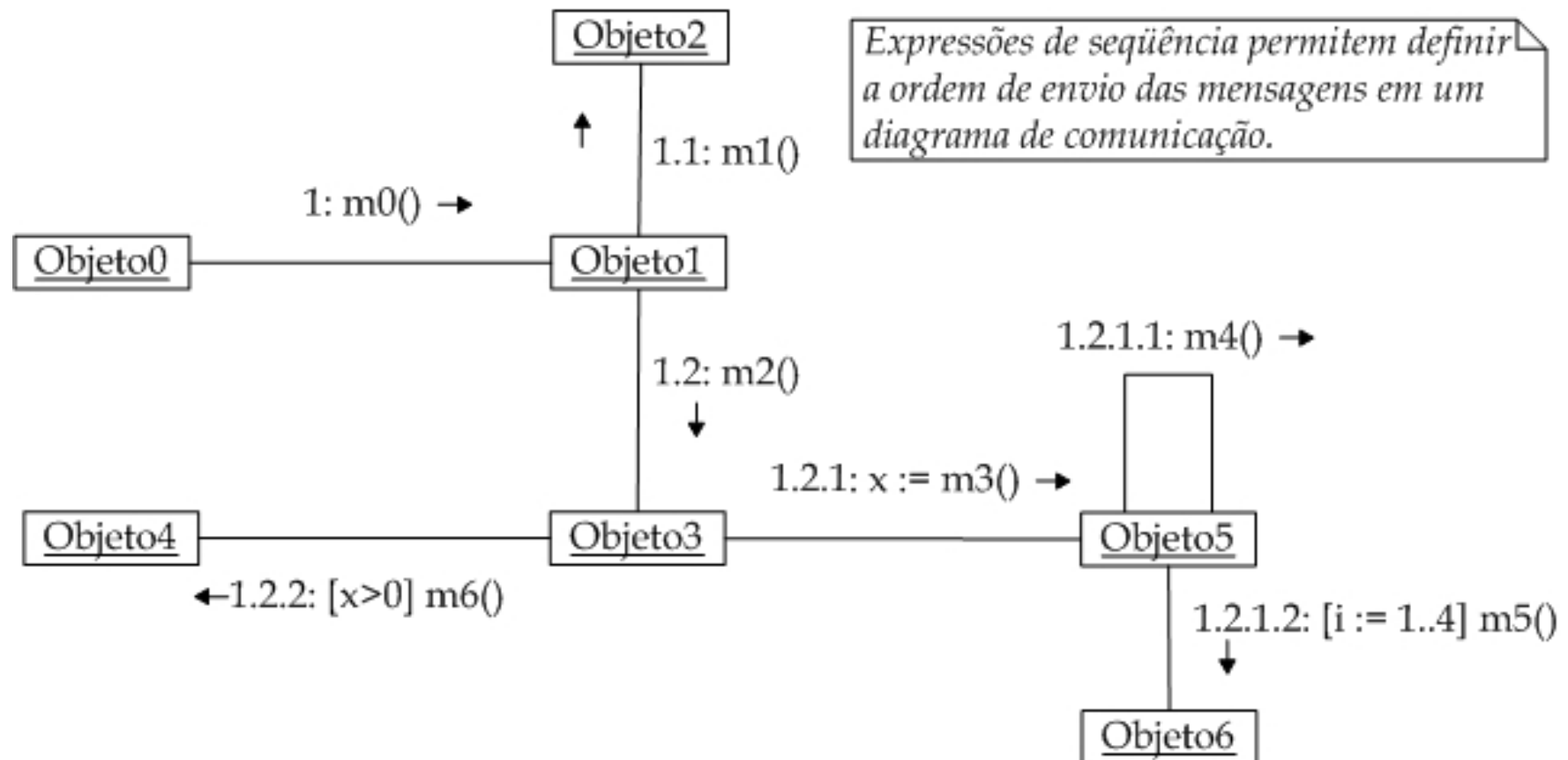
Mensagem com cláusula de iteração e com limites definidos

2 *[i := 1..10]: figuras[i].desenhar()

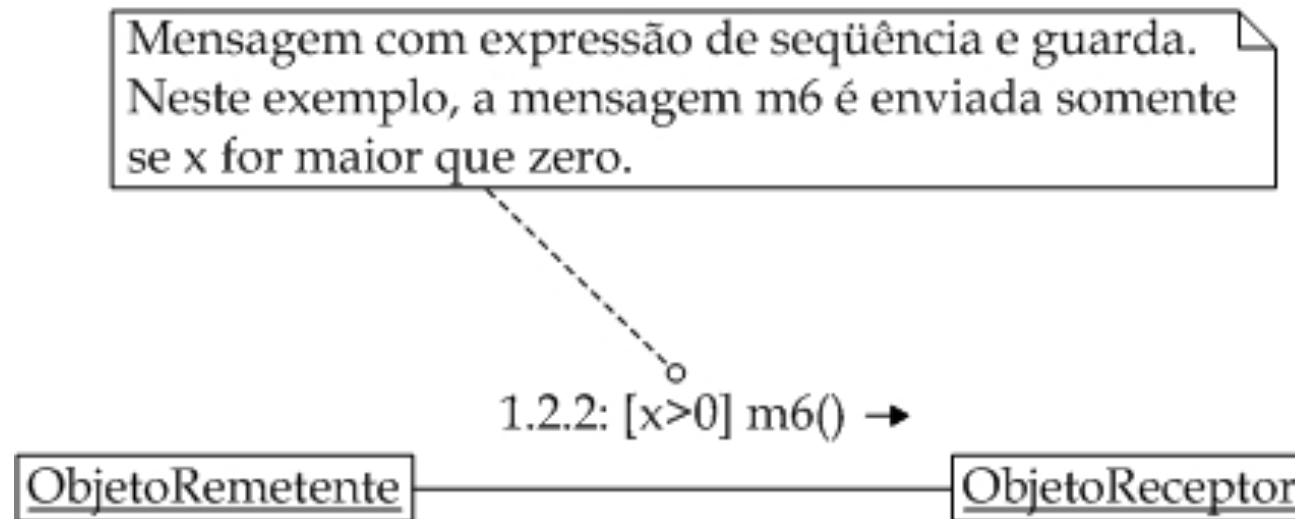
Mensagem aninhada com retorno armazenado na variável x

1.2.1: x := selecionar(e)

Exemplos (Mensagens UML)



Exemplos (Mensagens UML)



Notação para Objetos

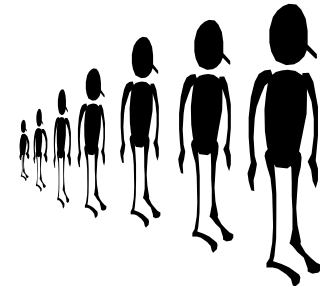
- Objetos são representados em um diagrama de interação utilizando-se a mesma notação do **diagrama de objetos**
- Pode-se representar objetos **anônimos** ou objetos **nomeados**, dependendo da situação
- Elementos de uma coleção também podem ser representados

Notação para Objetos

- Classes também podem ser representadas
 - Para o caso de mensagens enviadas para a classe
 - Uma mensagem para uma classe dispara a execução de uma operação estática
 - A representação de uma classe em um diagrama de seqüência é a mesma utilizada para objetos, porém o nome da classe não é sublinhado

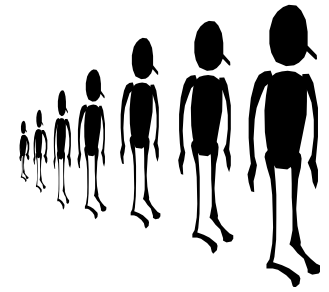


Multiobjetos



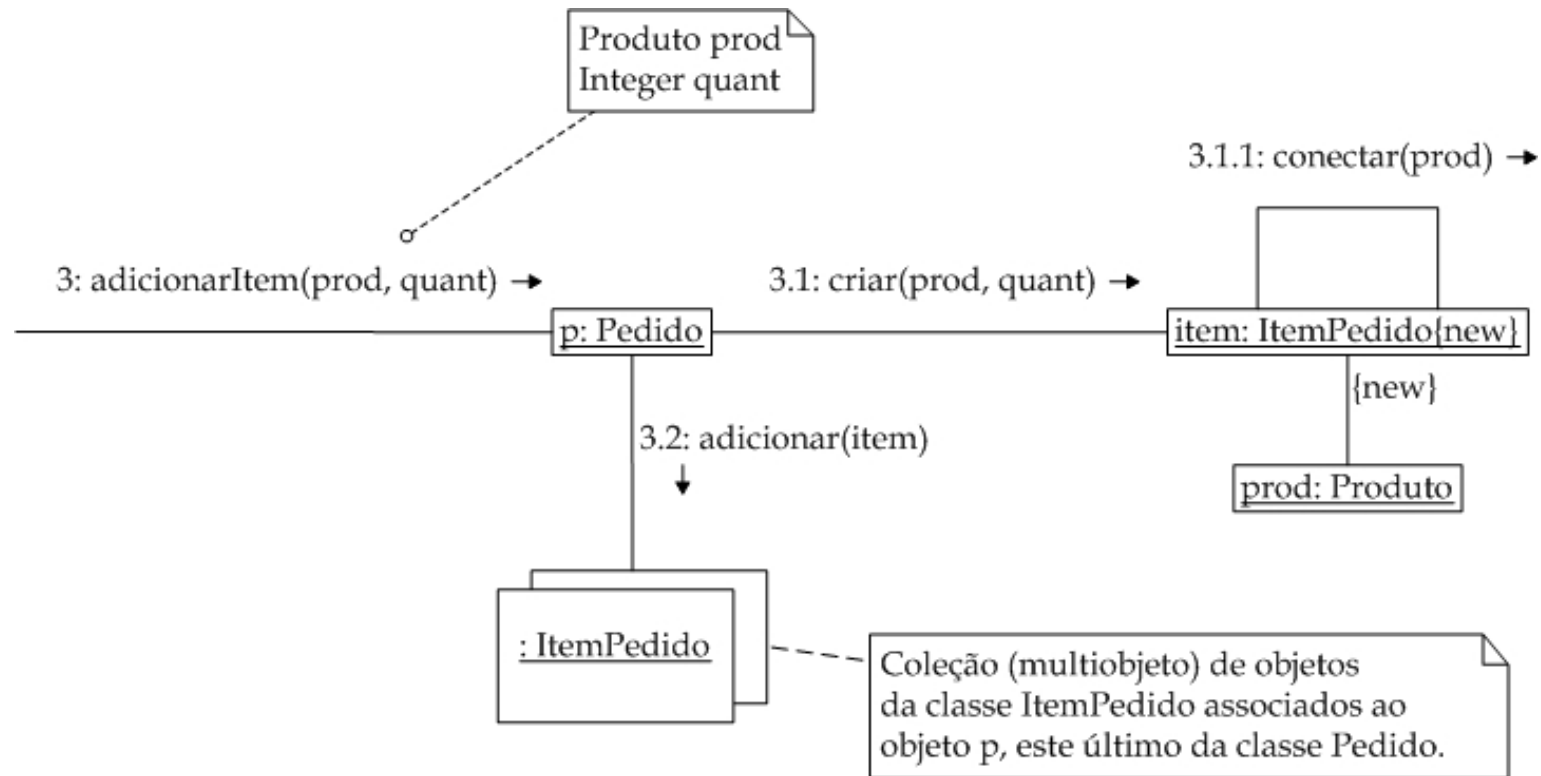
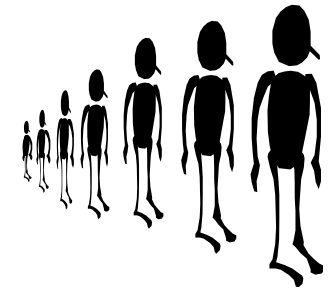
- Um **multiobjeto** é o nome que a UML dá para uma *coleção* de objetos de uma mesma classe. Pode ser utilizado para
 - Representar o lado muitos de uma associação de conectividade um para muitos
 - Representar uma lista (temporária ou não) de objetos sendo formada em uma colaboração

Multiobjetos



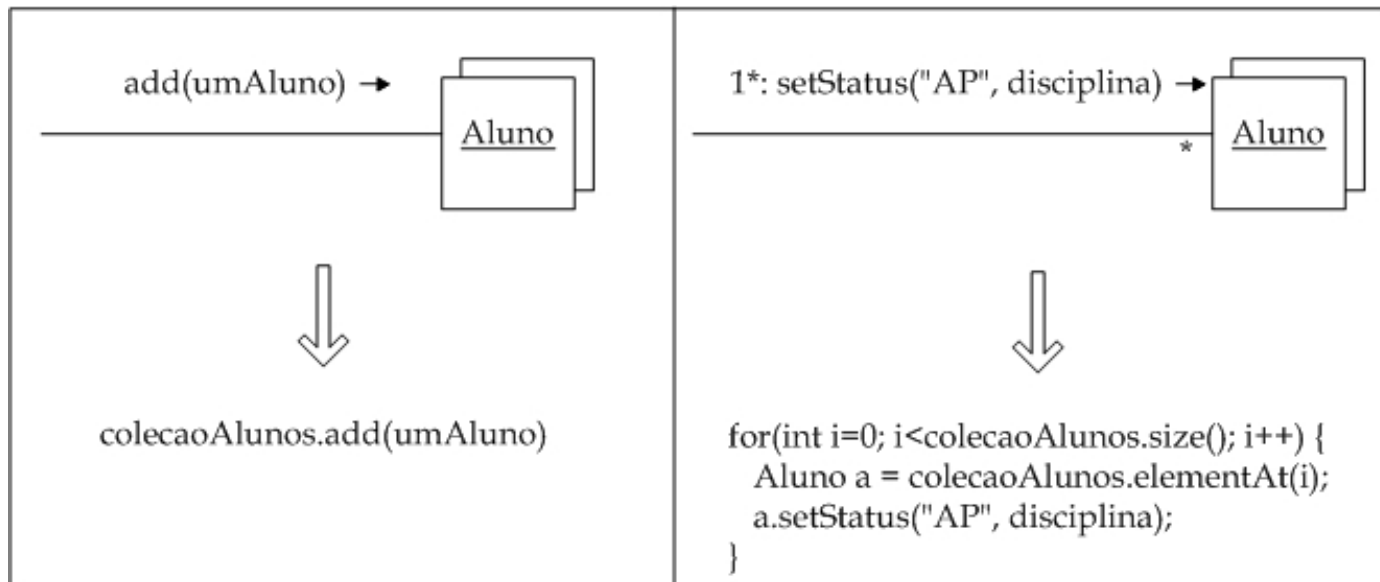
- Um multiobjeto é representado na UML através de dois retângulos superpostos
 - A superposição dos retângulos evita a confusão com a notação usada para objetos
 - O nome do multiobjeto é apresentado no retângulo que fica por cima e segue a mesma nomenclatura utilizada para objetos
 - **Convenção:** usar o nome da classe de seus elementos para nomear o multiobjeto

Exemplo (Multiobjetos)



Mensagens para Objetos/Coleção

- Uma mensagem pode ser enviada para um multiobjeto, ou pode ser enviada para um único objeto (elemento) do multiobjeto
- Quando o símbolo de iteração não é usado, convencionou-se que a mensagem está sendo enviada para o próprio multiobjeto



Implementação de Multiobjetos

Multiobjetos são normalmente implementados através de alguma estrutura de dados que manipule uma **coleções**.

Mensagens Típicas que Podemos Esperar que um Multiobjeto

Posicionar o cursor da coleção no primeiro elemento

Retornar o i-ésimo objeto da coleção

Retornar o próximo objeto da coleção

Encontrar um objeto de acordo com um identificador único

Adicionar um objeto na coleção

Remover um objeto da coleção

Obter a quantidade de objetos na coleção

Retornar um valor lógico que indica se há mais objetos a serem considerados

Implementação de Multiobjetos

A interface List da linguagem Java apresenta operações típicas de um multiobjeto.

```
public interface List<E> extends Collection<E> {  
    E get(int index);  
    E set(int index, E element);  
    boolean add(E element);  
    void add(int index, E element);  
    E remove(int index);  
    abstract boolean addAll(int index, Collection<? extends E> c);  
    int indexOf(Object o);  
    int lastIndexOf(Object o);  
    ListIterator<E> listIterator();  
    ListIterator<E> listIterator(int index);  
    List<E> subList(int from, int to);  
}
```


Tipos de Diagrama de Interação

- Há três tipos de diagrama de interação na UML 2.0: diagrama de seqüência, diagrama de comunicação e diagrama de visão geral da interação. O diagrama de seqüência e o diagrama de comunicação são equivalentes

Diagrama de seqüência: foco nas mensagens enviadas no decorrer do tempo.

Diagrama de comunicação: foco nas mensagens enviadas entre objetos que estão relacionados.

Diagrama de visão geral de interação. Pode ser utilizado para apresentar uma visão geral de diversas interações entre objetos, cada uma delas representada por um diagrama de interação. Diagrama é útil para **modularizar** a construção do diagramas de seqüência (ou de comunicação).

DIAGRAMA DE SEQÜÊNCIA

Diagrama de Seqüência

Descrição Geral

Os objetos participantes da interação são organizados na horizontal.

Abaixo de cada objeto existe uma linha (linha de vida).

Cada linha de vida possui o seu foco de controle. Quando o objeto está fazendo algo.

As mensagens entre objetos são representadas com linhas horizontais rotuladas partindo da linha de vida do objeto remetente e chegando a linha de vida do objeto receptor.

A posição vertical das mensagens permite deduzir a ordem na qual elas são enviadas.

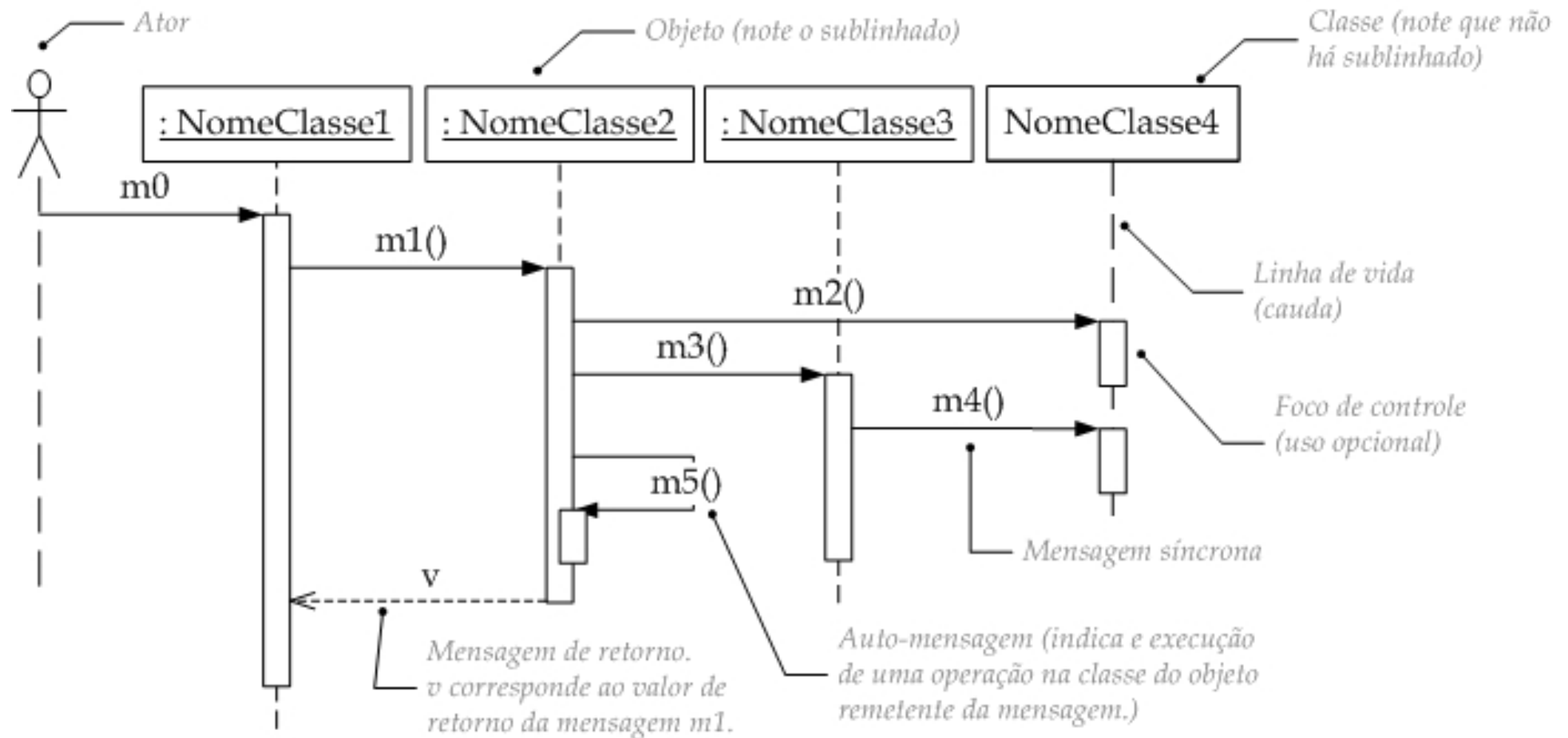
Ordem de envio de mensagens em um diagrama de seqüência pode ser deduzida a partir das expressões de seqüência.

Criação e destruição de objetos podem ser representadas.

Diagrama de Seqüência

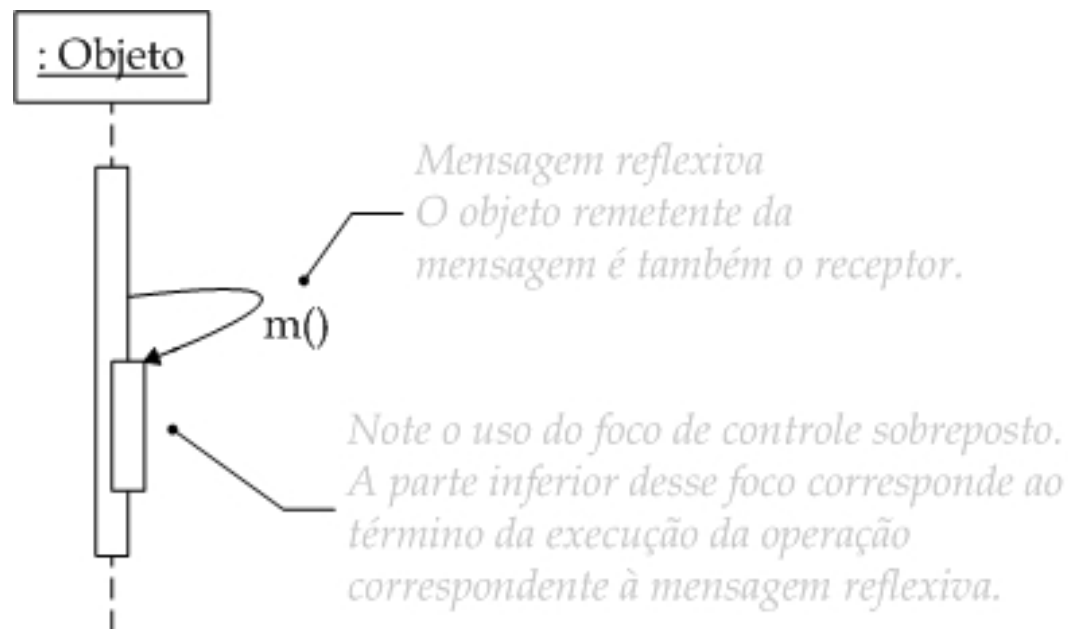
- Elementos básicos em um diagrama de seqüência
 - Atores
 - Objetos, multiobjetos e classes
 - Mensagens
 - Linhas de vida e focos de controle
 - Criação e destruição de objetos
 - Iterações

Elementos Gráficos de um DS



Mensagem Reflexiva em um DS

- Em uma mensagem reflexiva (ou auto-mensagem) o remetente é também o receptor
 - Corresponde a uma mensagem para *this* (*self*)



Criação/Destruição de Objetos em DS

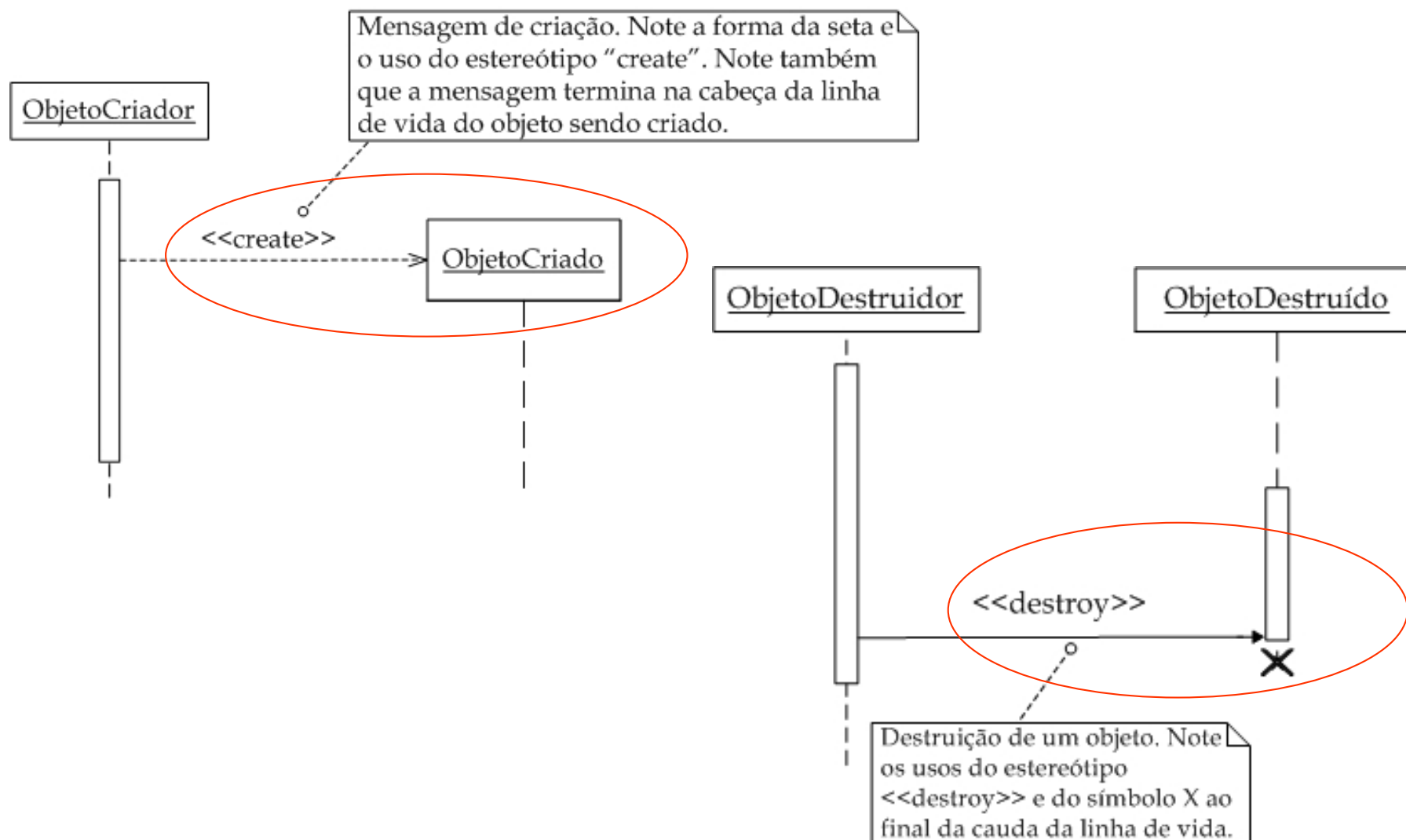


DIAGRAMA DE COMUNICAÇÃO

Diagrama de Comunicação

- Chamado de **diagrama de colaboração** na UML 1.X
- Estruturalmente, é bastante semelhante a um diagrama de objetos
 - A diferença é que são adicionados setas e rótulos de mensagens nas ligações entre esses objetos
- As ligações (linhas) entre objetos correspondem a relacionamentos existentes entre os objetos
 - Deve haver consistência com o diagrama de classes

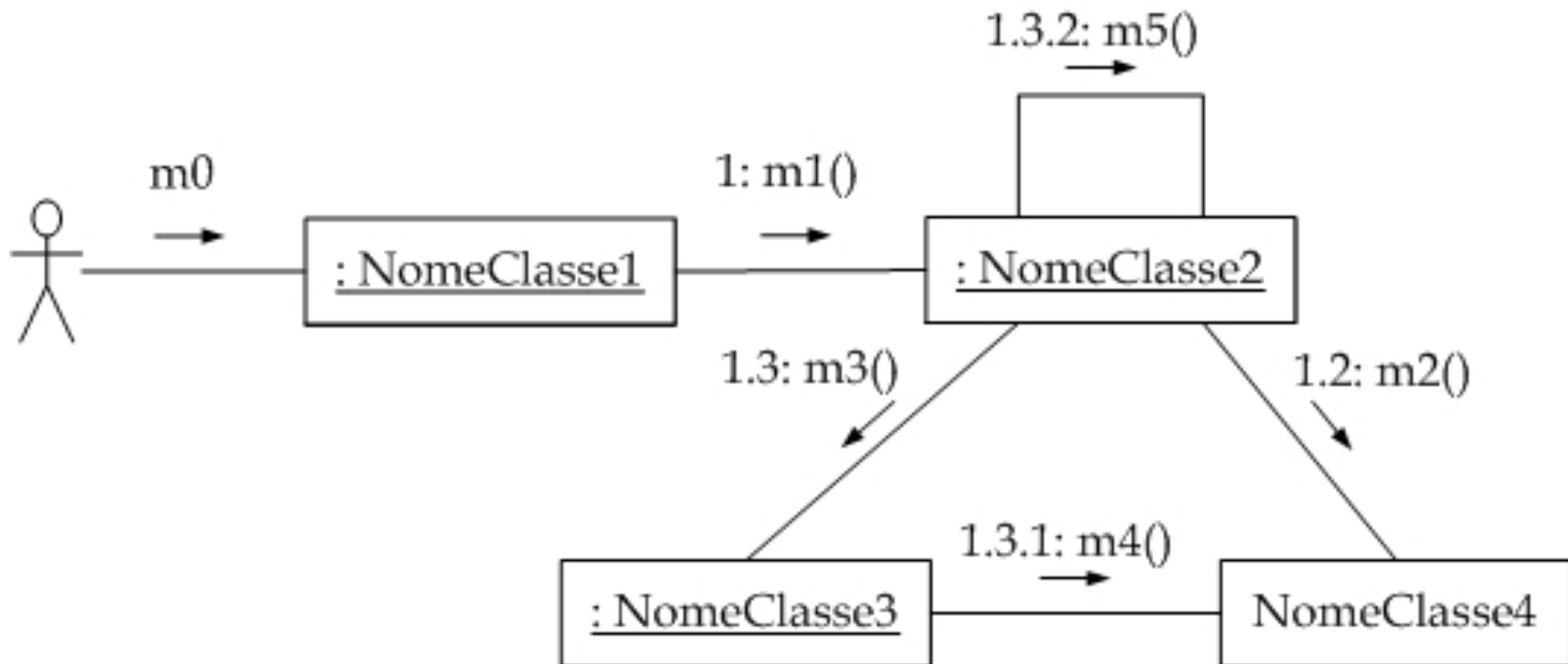
Diagrama de Comunicação

- Os objetos estão distribuídos em duas dimensões
 - Vantagem
 - Normalmente permite construir desenhos mais legíveis comparativamente aos diagramas de seqüência
 - Desvantagem
 - Não há como saber a ordem de envio das mensagens a não ser pelas expressões de seqüência
- Direção de envio de mensagem é indicada por uma seta próxima ao rótulo da mensagem.

Elementos Gráfico do DC

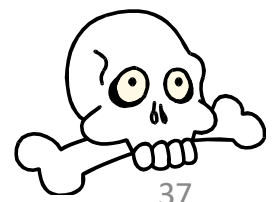
- Elementos básicos em um diagrama de comunicação
 - Atores
 - Objetos, multiobjetos e classes
 - Mensagens
 - Ligações entre objetos
 - Criação e destruição de objetos
 - Iterações

Elementos Gráfico do DC



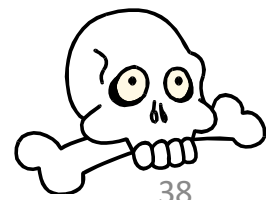
Criação de Objetos em um DC

- Durante a execução de um cenário de caso de uso, objetos podem ser criados e outros objetos podem ser destruídos
- Alguns objetos podem sobreviver à execução do caso de uso. Já outros podem nascer e morrer durante essa execução

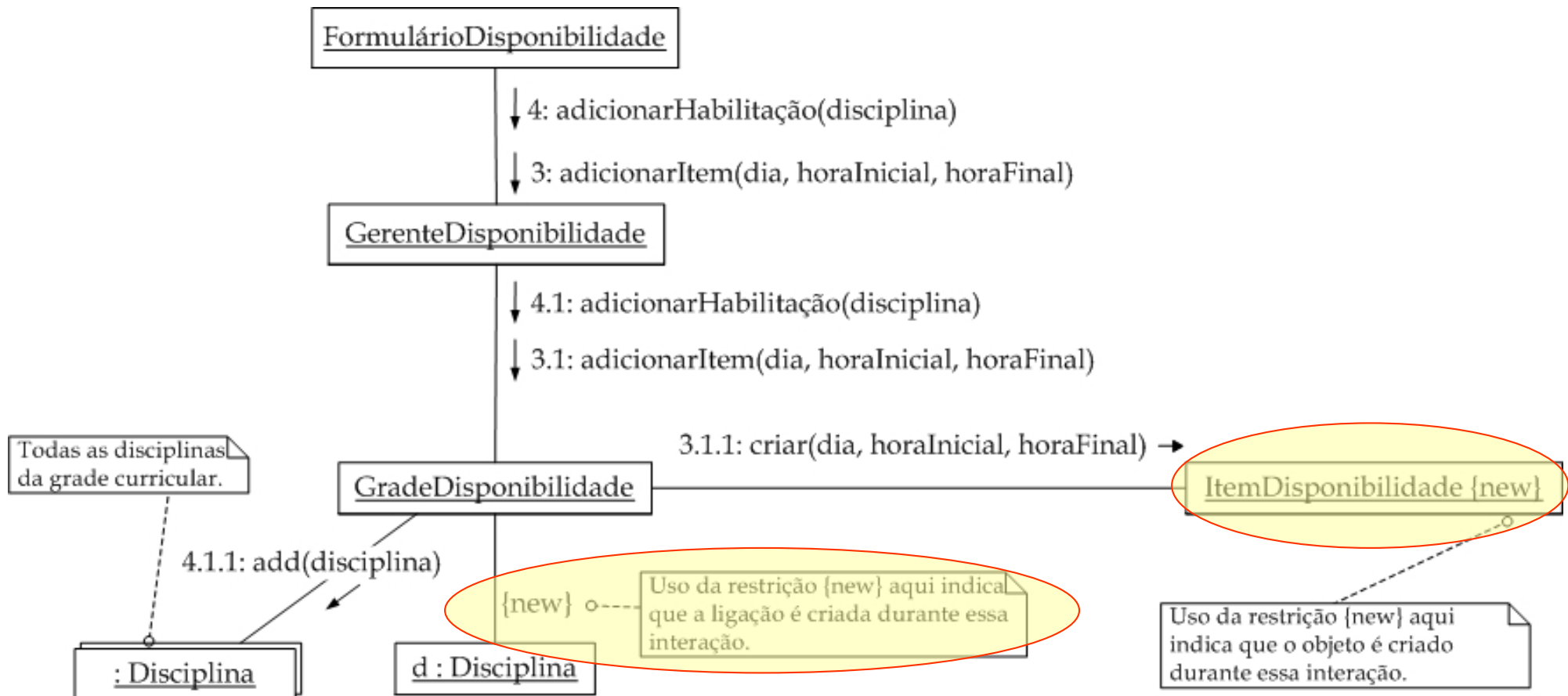


Criação de Objetos em um DC

- A UML define etiquetas (tags) para criação e destruição de objetos (ou de ligações entre objetos) no diagrama de comunicação.
 - **{new}**: objetos ou ligações criados durante a interação
 - **{destroyed}**: objetos ou ligações destruídos durante a interação
 - **{transient}**: objetos ou ligações destruídos e criados durante a interação



Criação de Objetos em um DC



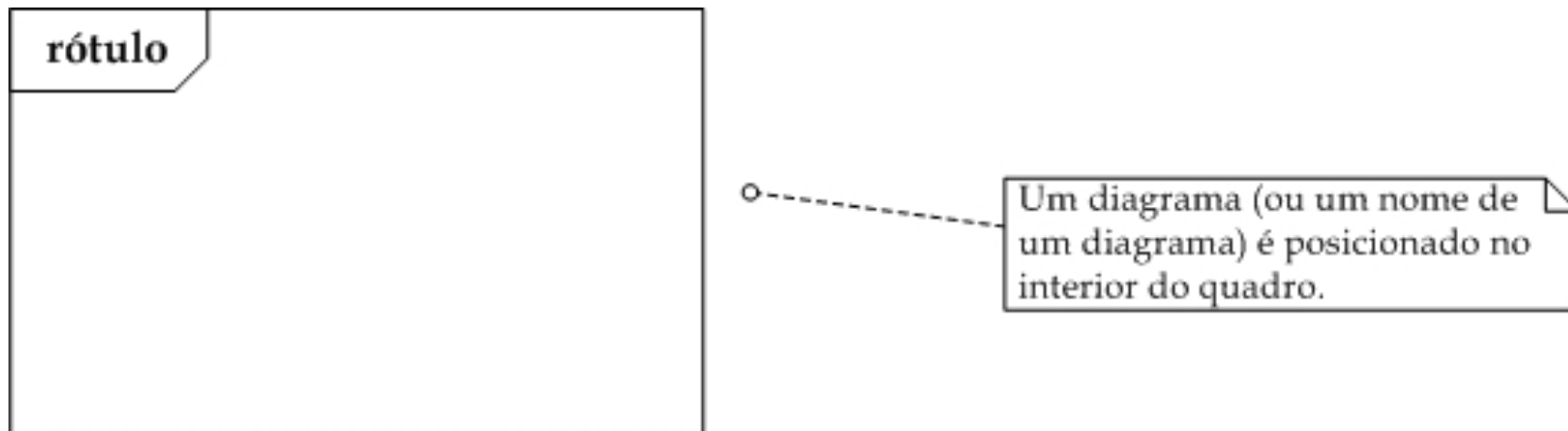
MODULARIZAÇÃO DE INTERAÇÕES

Quadro de Interação

- Elemento gráfico, que serve para modularizar a construção de diagramas de seqüência (ou de comunicação)
- Objetivos específicos
 - Dar um nome ao diagrama que aparece dentro do quadro
 - Fazer referência a um diagrama definido separadamente
 - Definir o fluxo de controle da interação

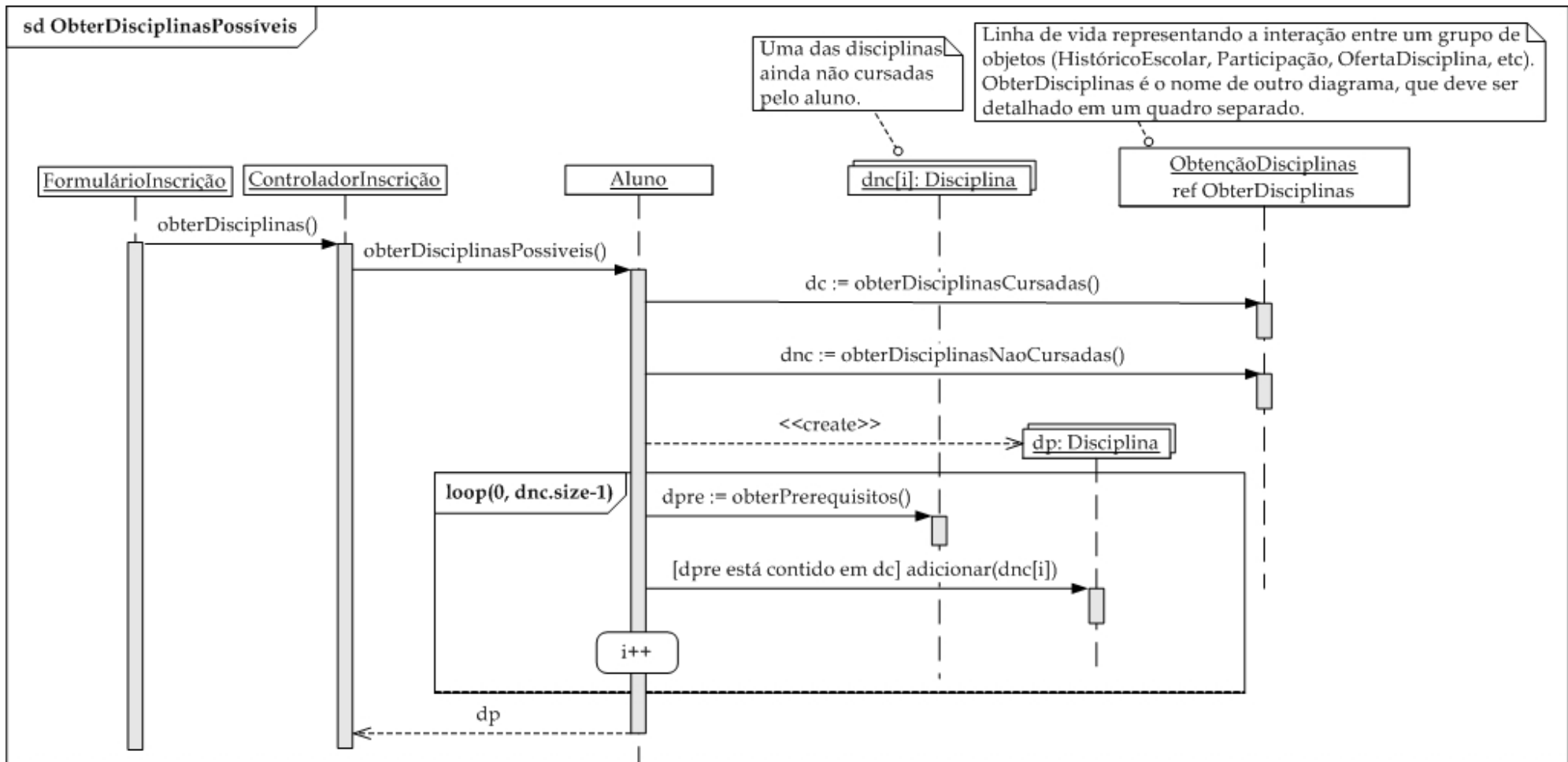
Quadro de Interação

- Notação



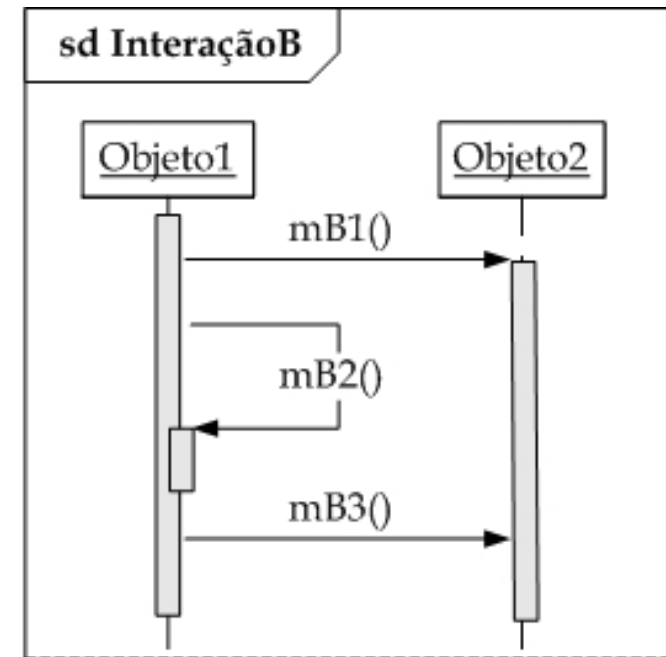
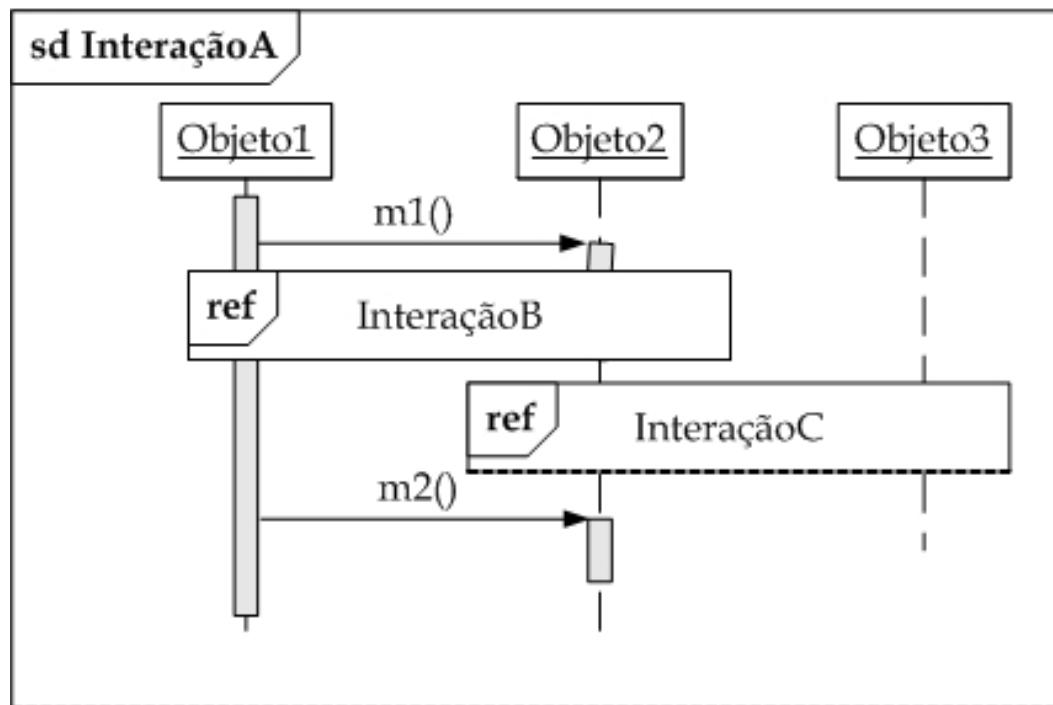
Diagramas Nomeados

Dar um nome ao diagrama que aparece dentro do quadro



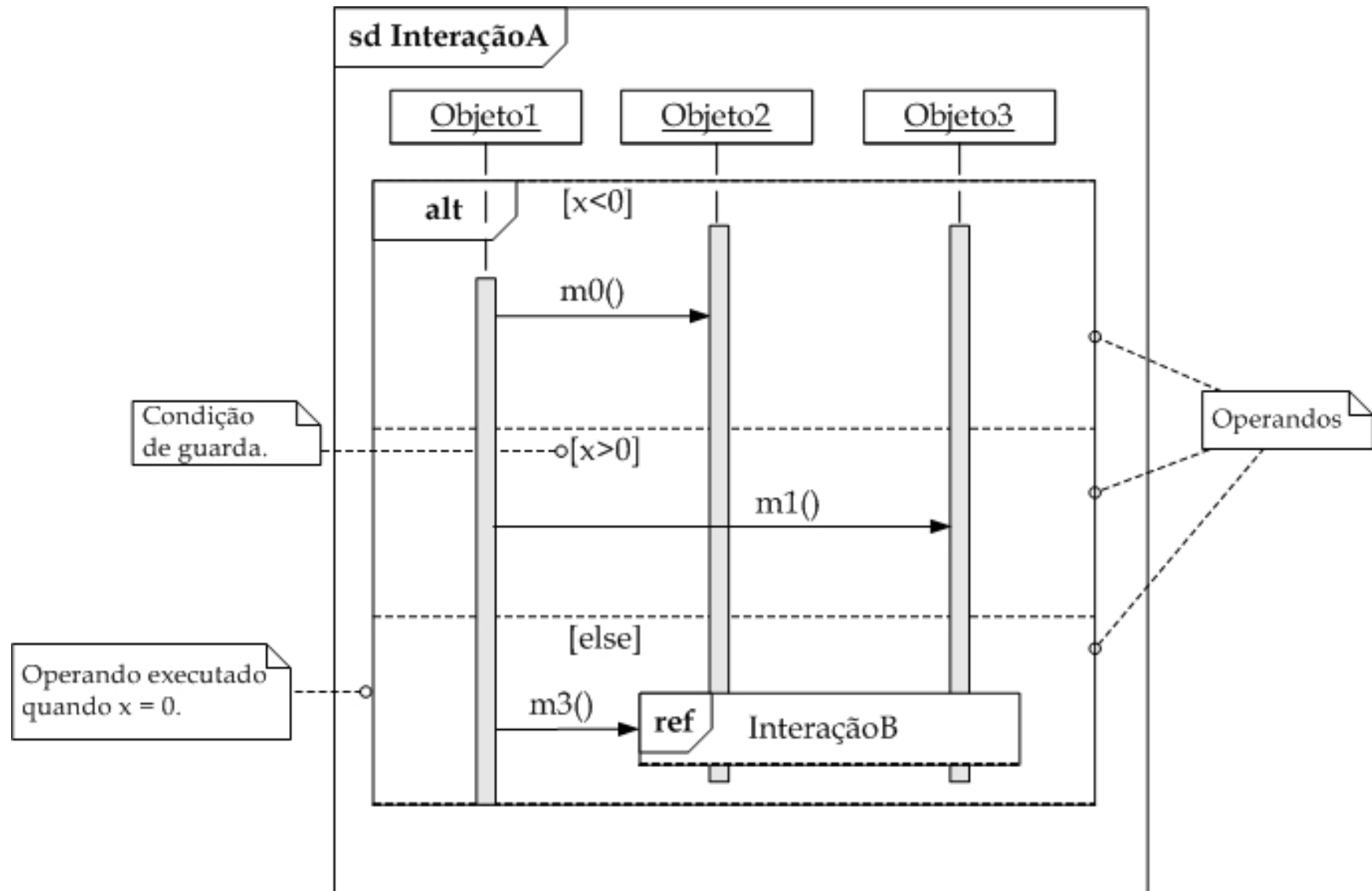
Diagramas Referenciados

Fazer referência a um diagrama definido separadamente

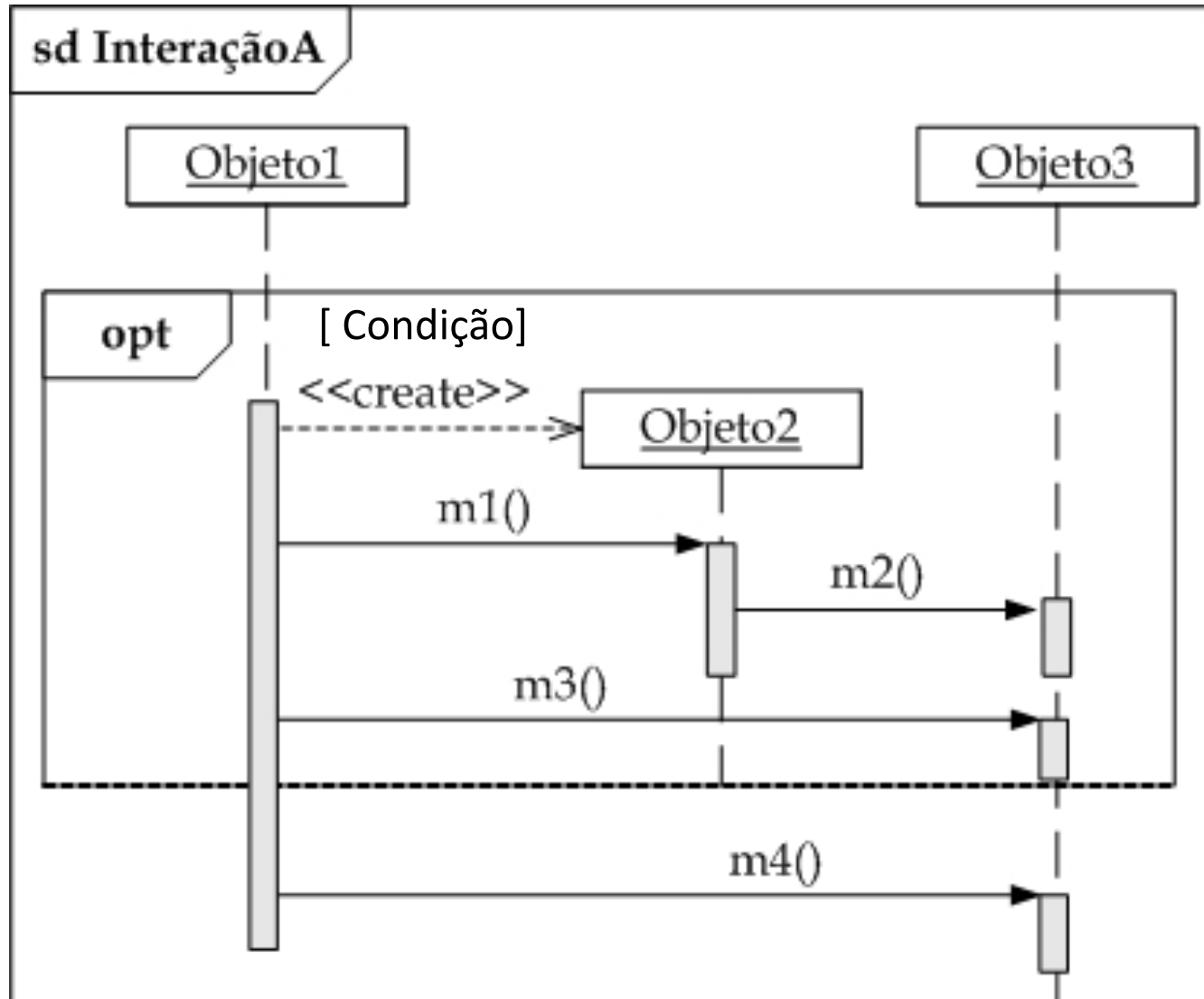


InteraçãoB e InteraçãoC são nomes de diagramas que apresentam mensagens trocadas entre os objetos Objeto1 e Objeto2. Note que os quadros correspondentes são rotulados com "ref" e posicionados sobre as linhas de vida dos objetos.

Fluxo de Controle: Alternativas



Fluxo de Controle: Opções



Fluxo de Controle: Iterações

