

# Diagramas de Sequência e Comunicação

# Introdução

- Faltam informações aos modelos 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?

# Introdução

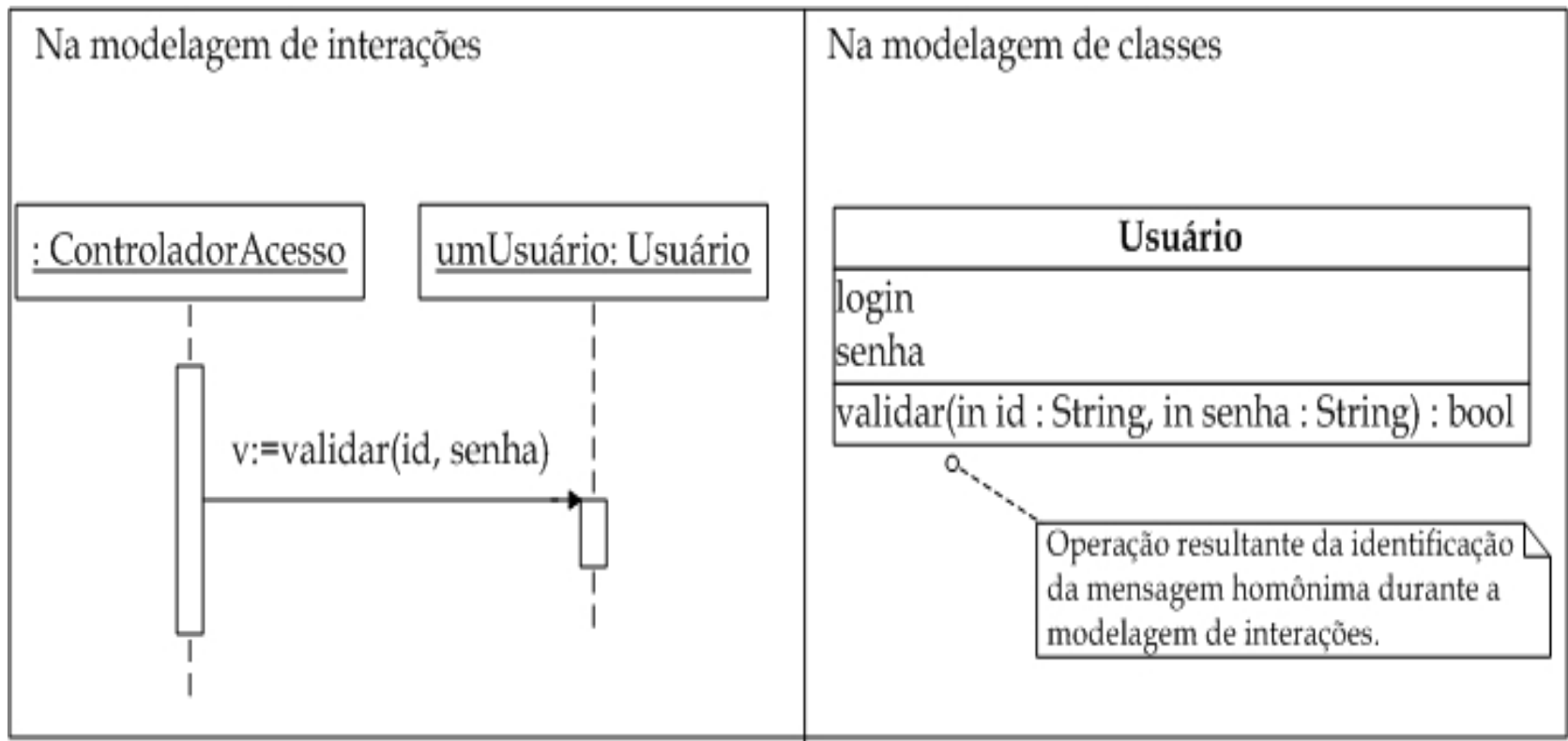
- Para responder às questões anteriores, o ***modelo de interações deve ser criado.***
- Esse modelo representa mensagens trocadas entre objetos para a execução de cenários dos casos de uso do sistema.

# Objetivos da criação de modelos de interação

- Obter informações adicionais para completar e aprimorar outros modelos (principalmente o modelo de classes)
  - Quais os métodos de uma classe?
  - Quais os objetos participantes da realização de um caso de uso?
  - Para cada método, qual a sua assinatura?
  - Uma classe precisa de mais atributos?
- Fornecer uma visão detalhada dos objetos e mensagens envolvidos na realização dos casos de uso.

# Mensagens

- Qual o objetivo da construção dos diagramas de interação?
  - Identificar **mensagens**



# O que uma mensagem indica?

- 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.

# Exemplos de mensagens

- 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)

# Tipos de diagrama de interação

- Há três tipos de diagrama de interação na UML 2.0 : **diagrama de sequência, diagrama de comunicação e *diagrama de visão geral da interação*.**
- O diagrama de sequência e o diagrama de comunicação são equivalentes.
- ***Diagrama de sequência: foco nas mensagens enviadas no decorrer do tempo.***
- ***Diagrama de comunicação: foco nas mensagens enviadas entre objetos que estão relacionados.***

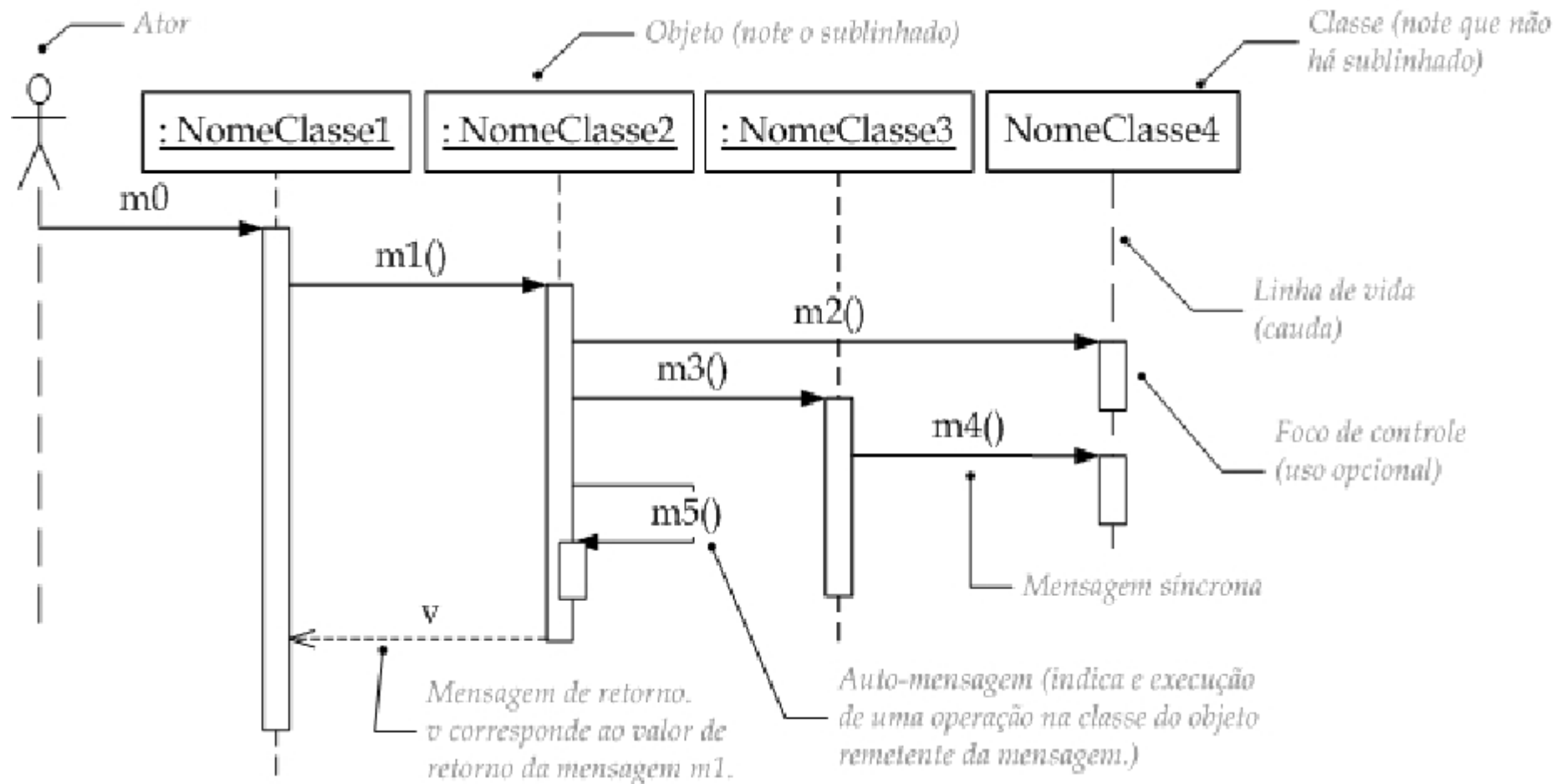
# Diagrama de Sequência

- 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.
- O foco de controle indica que 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.

# Elementos gráficos de um diagrama de sequência

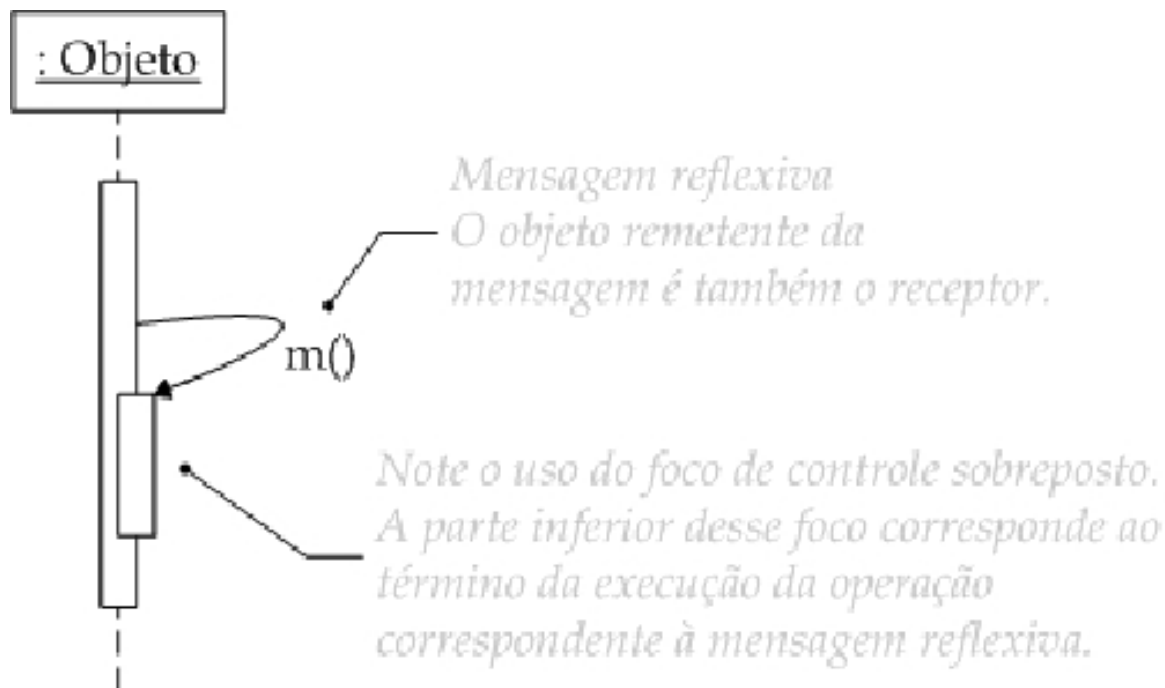
- Elementos básicos em um diagrama de sequência:
  - Atores
  - Objetos e classes
  - Mensagens
  - Linhas de vida e focos de controle
  - Criação e destruição de objetos
  - Iterações

# Exemplo de diagrama de sequência

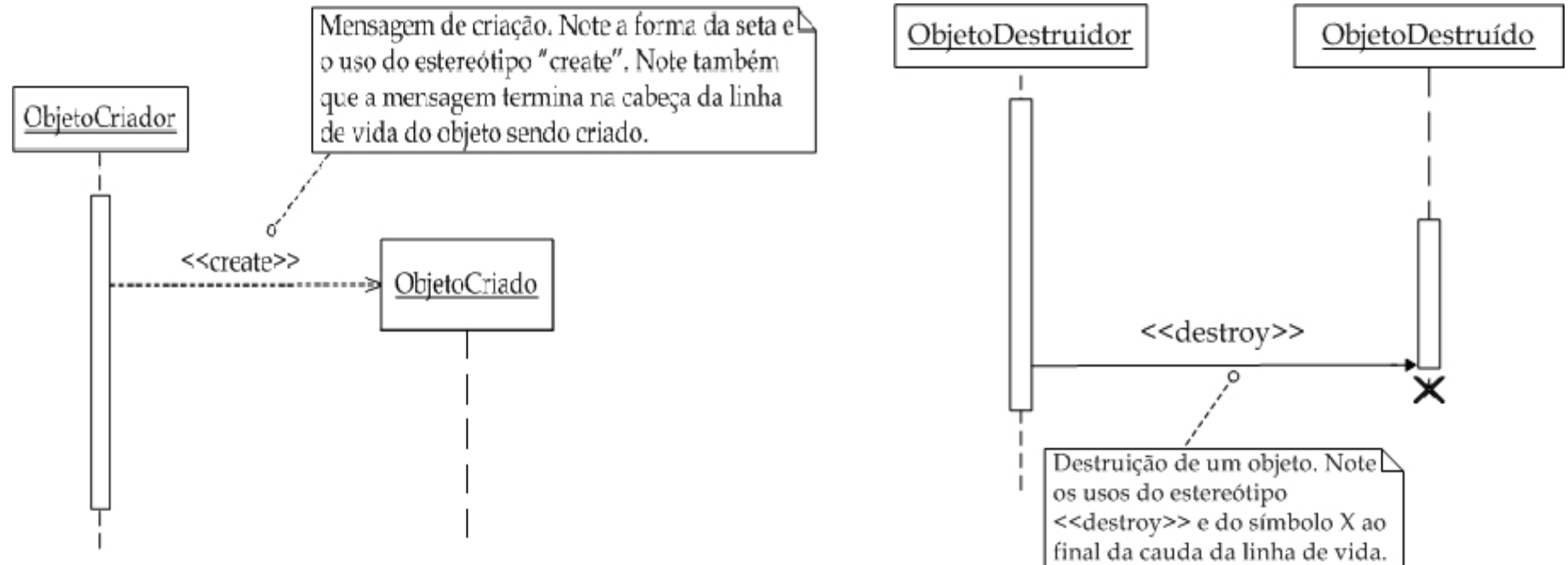


# Mensagens reflexivas

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



# Criação/destruição de objetos



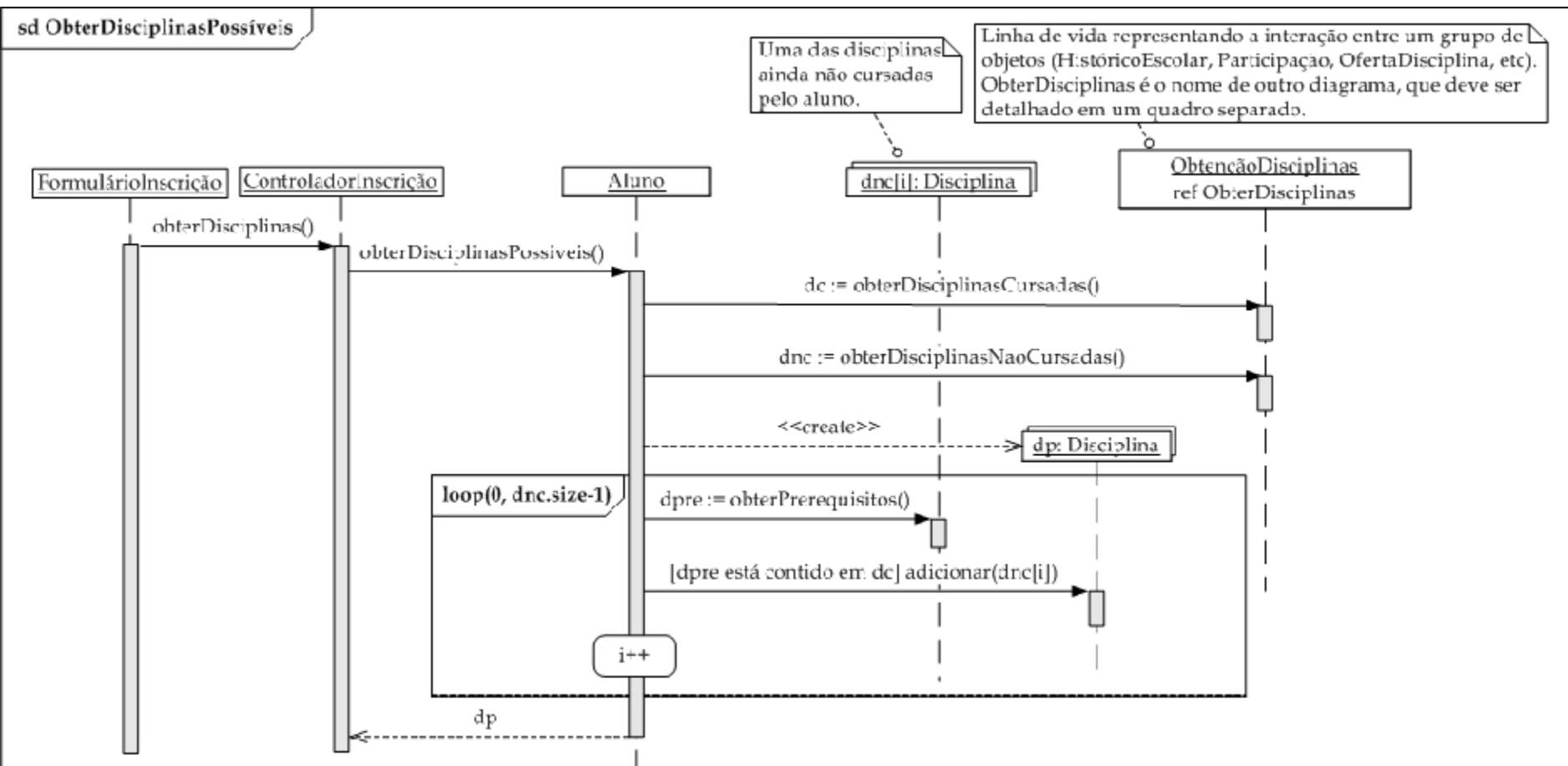
# Quadros de interação

- Elemento gráfico, que serve para modularizar a construção de diagramas de sequência (ou de comunicação).



Um diagrama (ou um nome de um diagrama) é posicionado no interior do quadro.

# Exemplo



Uma das disciplinas  
ainda não cursadas  
pelo aluno.

Linha de vida representando a interação entre um grupo de objetos (HistóricoEscolar, Participação, OfertaDisciplina, etc). **ObterDisciplinas** é o nome de outro diagrama, que deve ser detalhado em um quadro separado.

Obtenção Disciplinas  
ref Obter Disciplinas

## Formulário Inscrição

Controlador	Inscrição
-------------	-----------

Aluno

dnc[i]: Disciplina

obterDisciplinas()

```
obterDisciplinasPossiveis()
```

```
dc := obterDisciplinasCursadas()
```

```
dnc := obterDisciplinasNaoCursadas()
```

&lt;&lt;create&gt;&gt;

dp: Disciplina

```
loop(0, dnc.size-1)
```

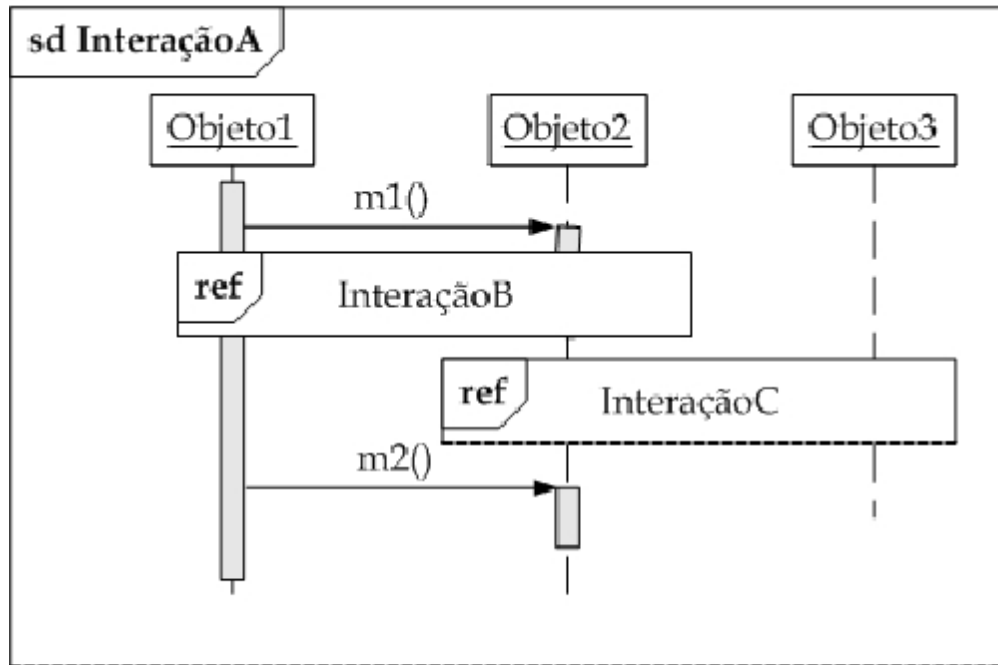
```
dpre := obterPrerequisitos()
```

```
[dpre está contido em: dc] adicionar(dnc[i])
```

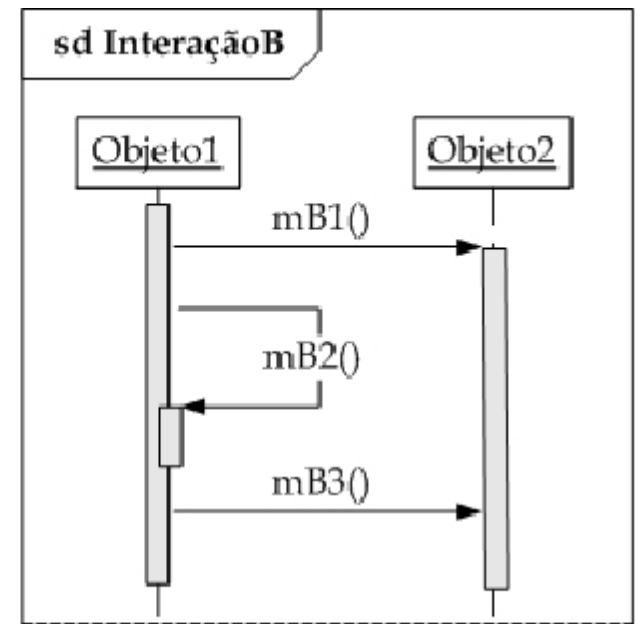
i++

dp

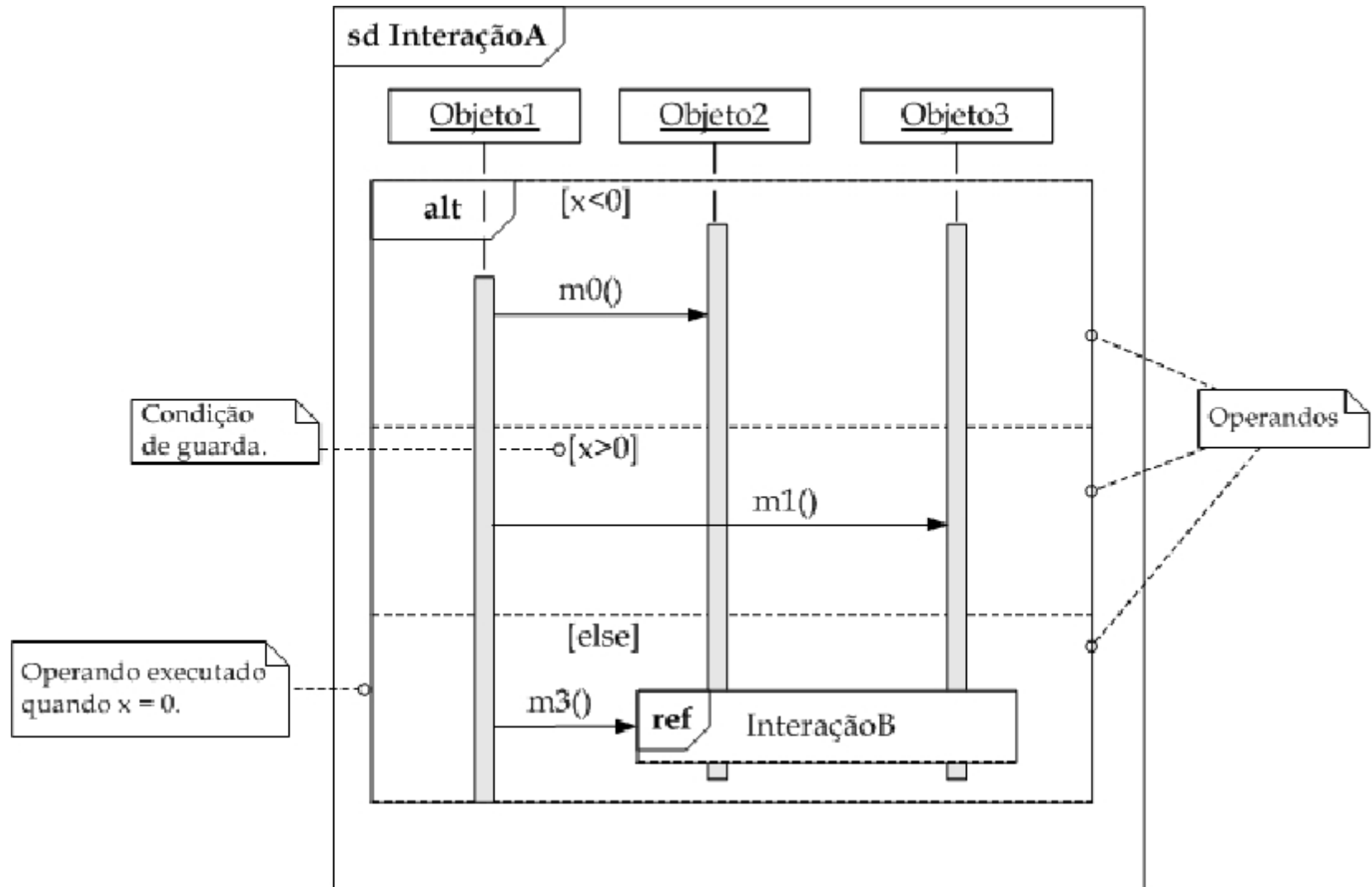
# Diagramas referenciados



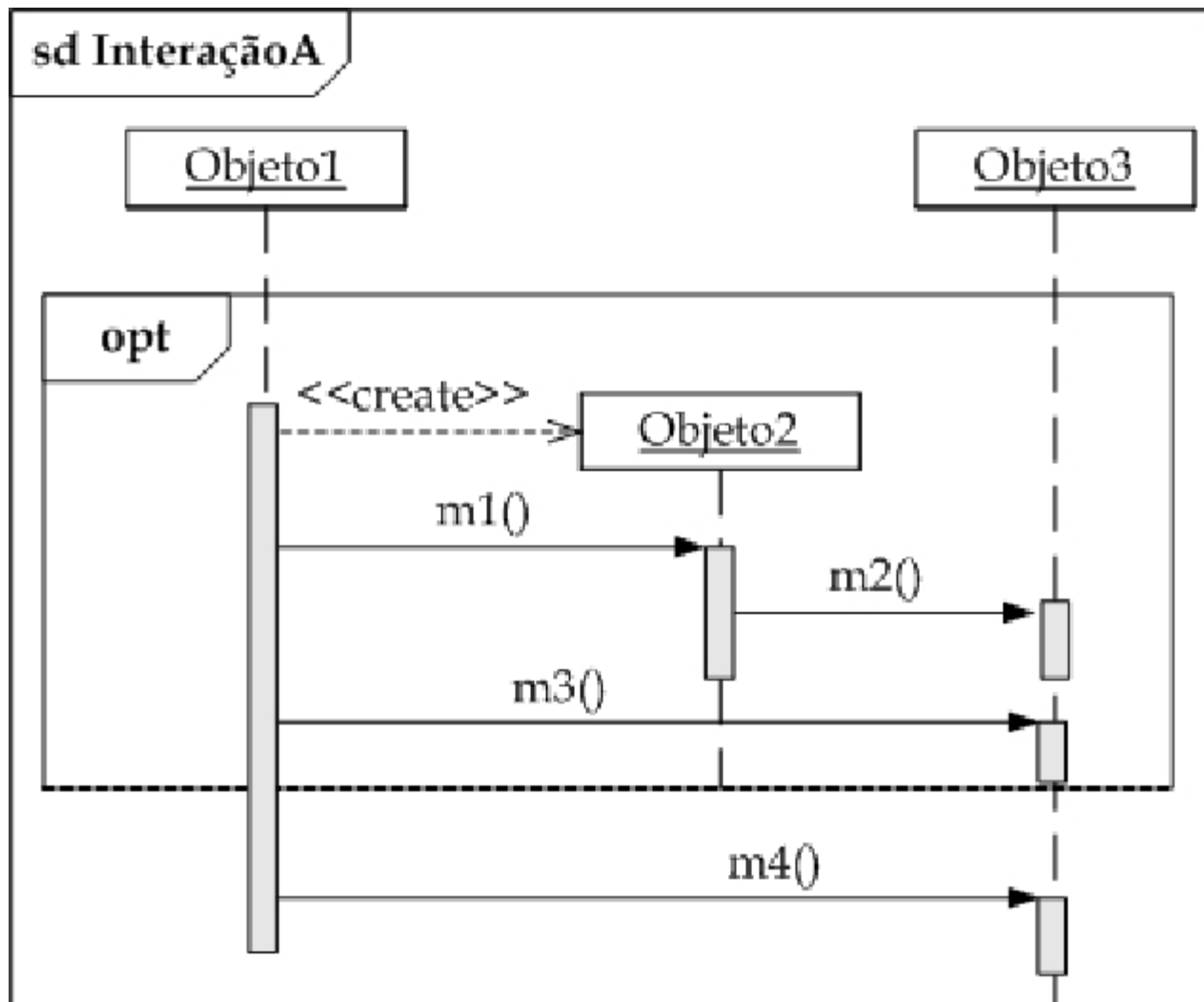
InteraçãoB e InteraçãoC são nomes de diagramas que apresentam mensagens trocadas entre os objetos Objeto1 e Objeto2. Note que os quadros correspondentes são rotulados com "ref" e posicionados sobre as linhas de vida dos objetos.



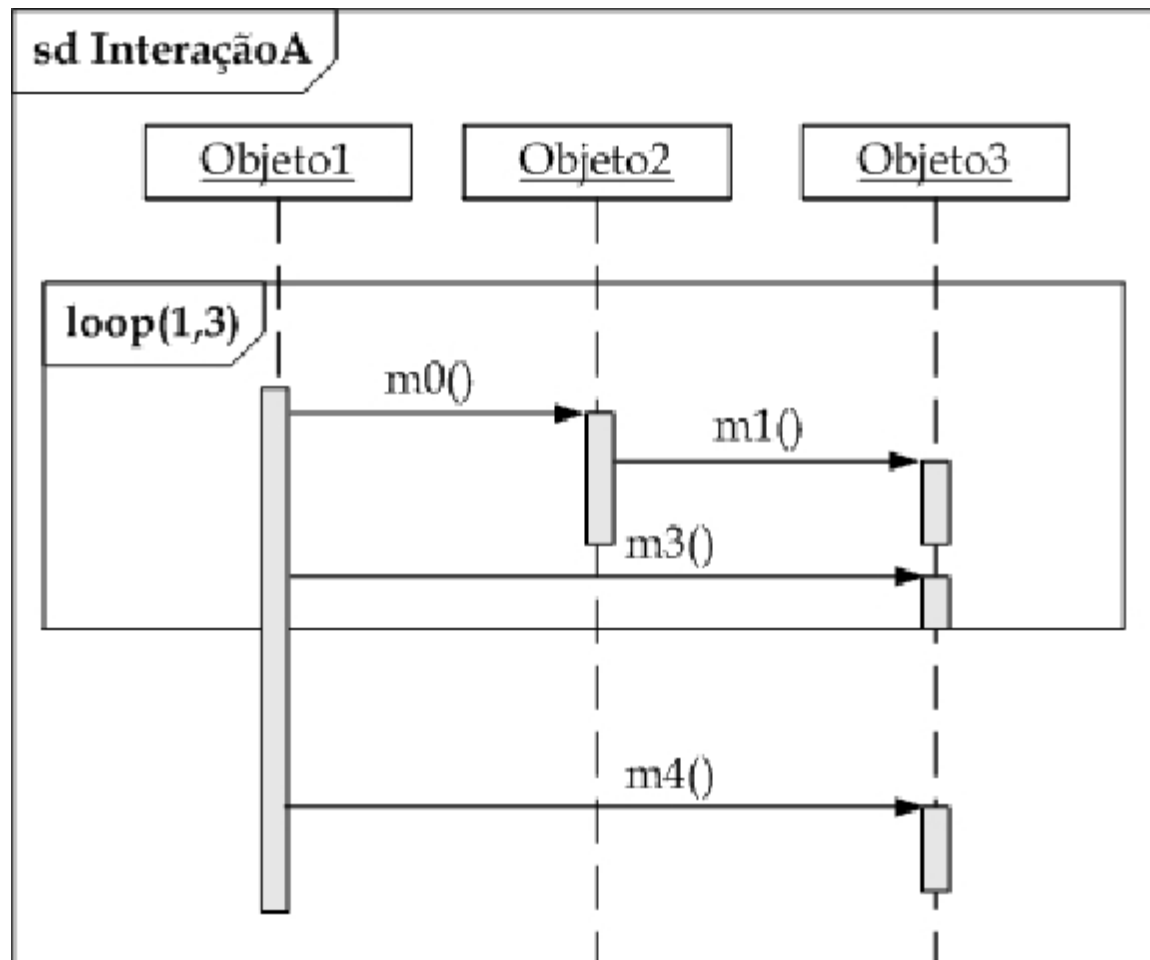
# Alternativas



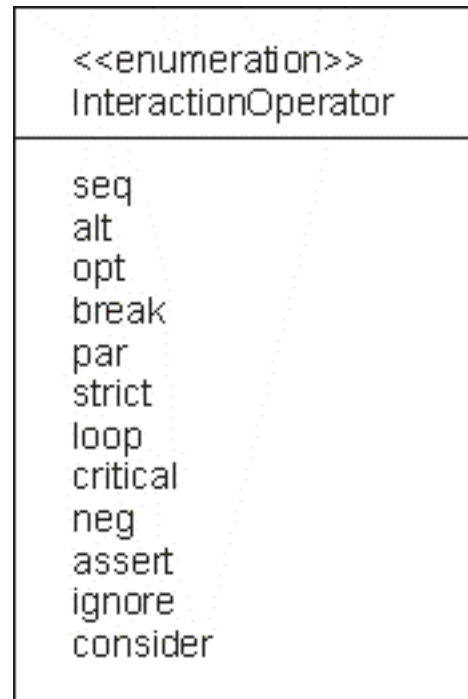
# Opções



# Iterações



# Fragmentos de um diagrama de sequência



# Heurísticas para modelagem de diagramas de sequência

- Na primeira coluna deve ser colocado o ator que iniciou o caso de uso.
- Na segunda coluna deve ser colocado um objeto de fronteira (usado pelo ator para iniciar o caso de uso).
- Na terceira coluna deve ser colocado o objeto de controle responsável pelo caso de uso.
- Objetos de entidade são acessíveis tanto por objetos de fronteira (raro) como por objetos de controle.
- Objetos de entidade nunca acessam objetos de fronteira ou de controle.
- Objetos de entidade normalmente são compartilhados por outros diagramas de sequência

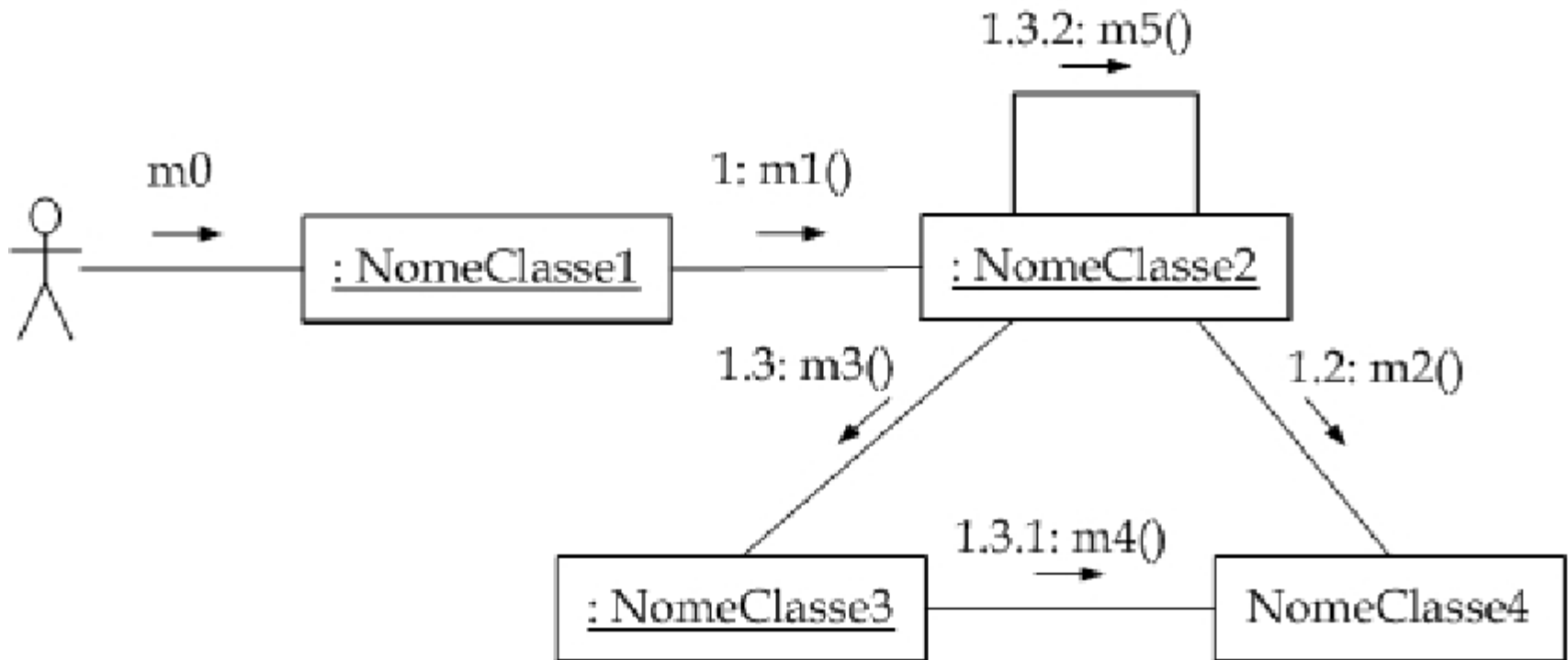
# Diagrama de comunicação

- 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.

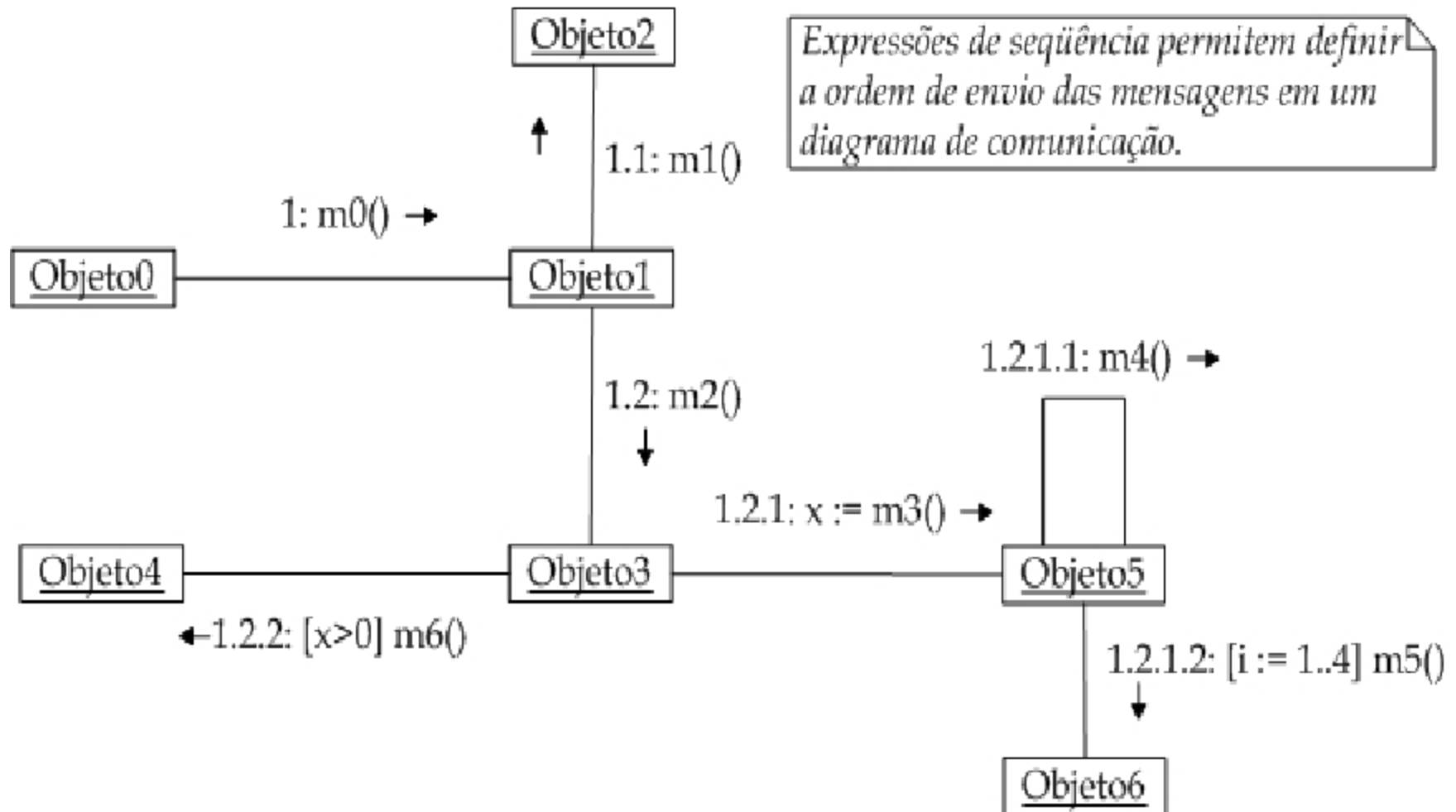
# Diagrama de Comunicação

- Os objetos estão distribuídos em duas dimensões
  - Vantagem: normalmente permite construir modelos 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.

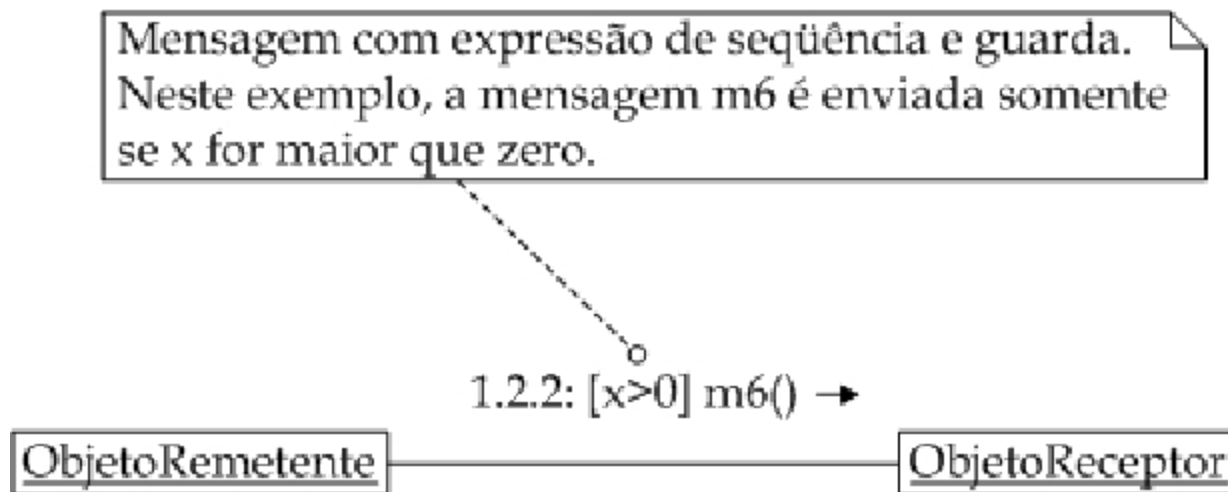
# Elementos gráficos de um diagrama de comunicação



# Exemplo de diagrama de comunicação



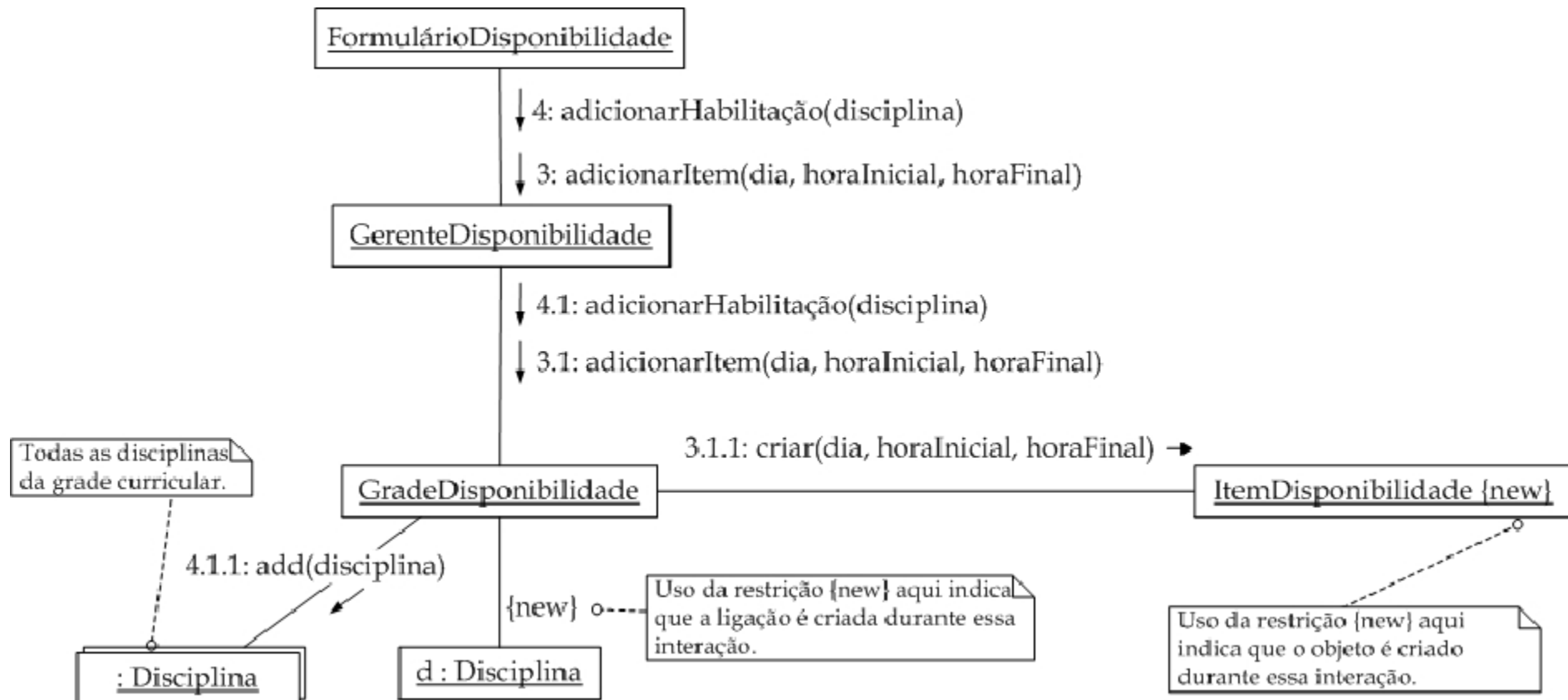
# Exemplo de diagrama de comunicação



# Criação de objetos em um diagrama de comunicação

- Durante a execução de um cenário de caso de uso, objetos podem ser criados e outros objetos podem ser destruídos.
- A UML define etiquetas (tags) para criação e destruição de 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 diagrama de comunicação



# Heurísticas para construção do MI

- ***Verifique a consistência dos diagramas de interação (modelos de interação) em relação aos casos de uso e ao modelo de classes.***
  - Cada cenário relevante para cada caso de uso foi considerado na modelagem de interações.
  - Durante a construção do diagrama de interação, pode-se identificar novas classes.
  - Atributos, associações e operações também surgem como subproduto da construção dos diagramas de interação.

# Heurísticas para construção do MI

- ***O objeto de controle realiza apenas a coordenação da realização do caso de uso.***
  - Todas as ações do ator resultam em alguma atividade realizada por esse objeto de controle.
  - Isso pode levar ao alto acoplamento; no pior caso, o controlador tem conhecimento de todas as classes participantes do caso de uso.
  - Sempre que for adequado, segundo os princípios de coesão e de acoplamento, as classes de domínio devem enviar mensagens entre si, aliviando o objeto de controle.

# MI em um processo iterativo e incremental

- São construídos para os casos de uso.
- Há controvérsias sobre o momento de início da utilização desse modelo (análise vs. projeto).
- Inicialmente (+análise), pode exibir apenas os objetos participantes e mensagens exibindo somente o nome da operação.
- Posteriormente (+projeto), pode ser refinado.
  - Criação e destruição de objetos, tipo e assinatura completa de cada mensagem.

# MI em um processo iterativo e incremental

- A construção do Modelo de Interação fornece informações úteis para transformar o modelo de classes de análise no modelo de classes de projeto.
- Em particular, Modelo de Interação fornece os seguintes itens para refinar o modelo de classes de análise:
  - Detalhamento de métodos
  - Detalhamento de associações
  - Novos métodos
  - Novos atributos
  - Novas classes

# MI em um processo iterativo e incremental

