

PROGRAMAÇÃO ORIENTADA A OBJETOS

Diagrama de Sequência - UML



Visão estática

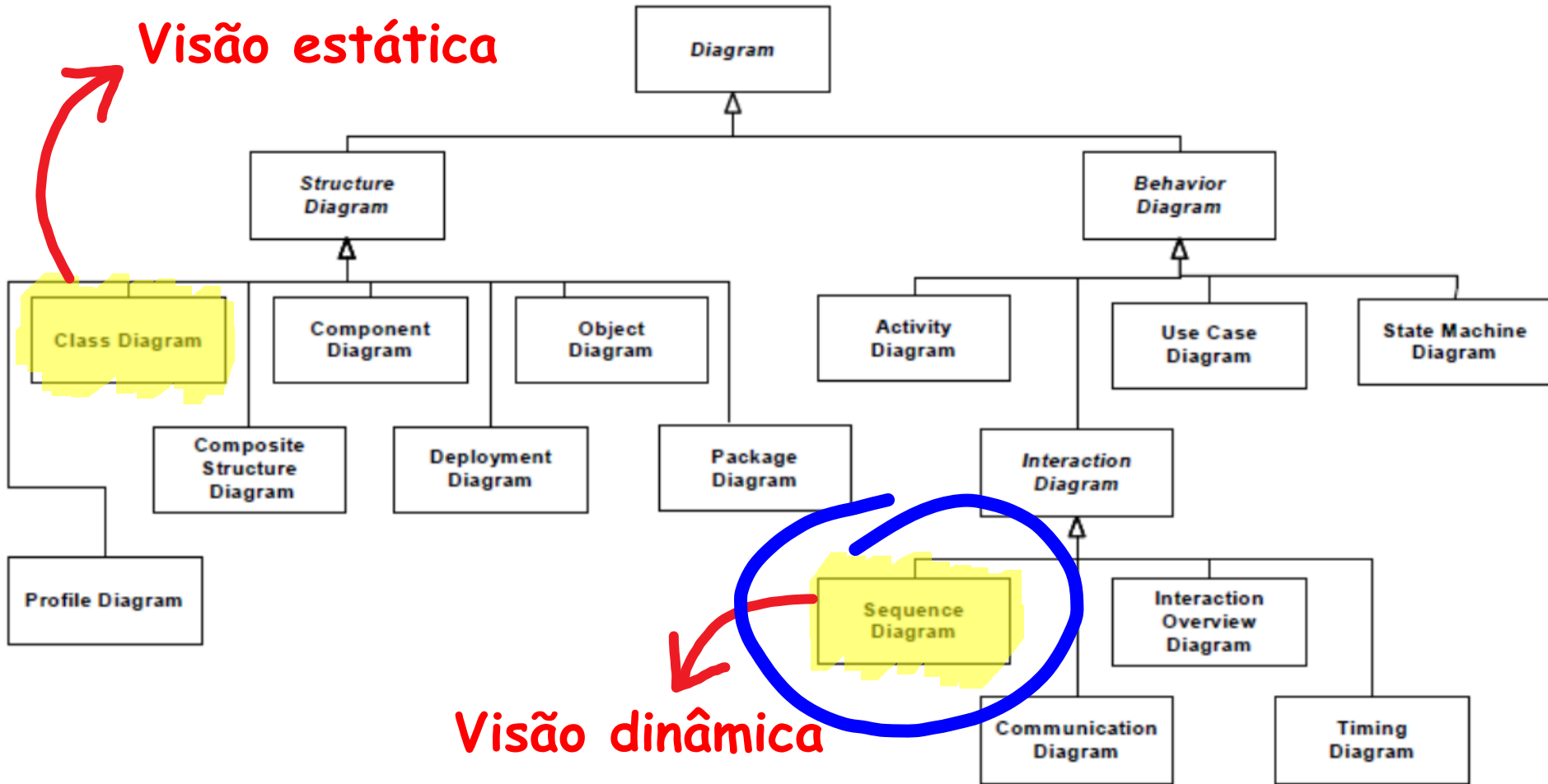


Diagrama de Sequência



3

□ Diagrama Comportamental

▣ Diagrama de Interação

- Mostra a interação de objetos organizada no tempo
- Permite visualizar a troca de mensagens entre os objetos
- Mostra a implementação de uma ou mais “operações”
 - Mostra os “métodos” destas operações

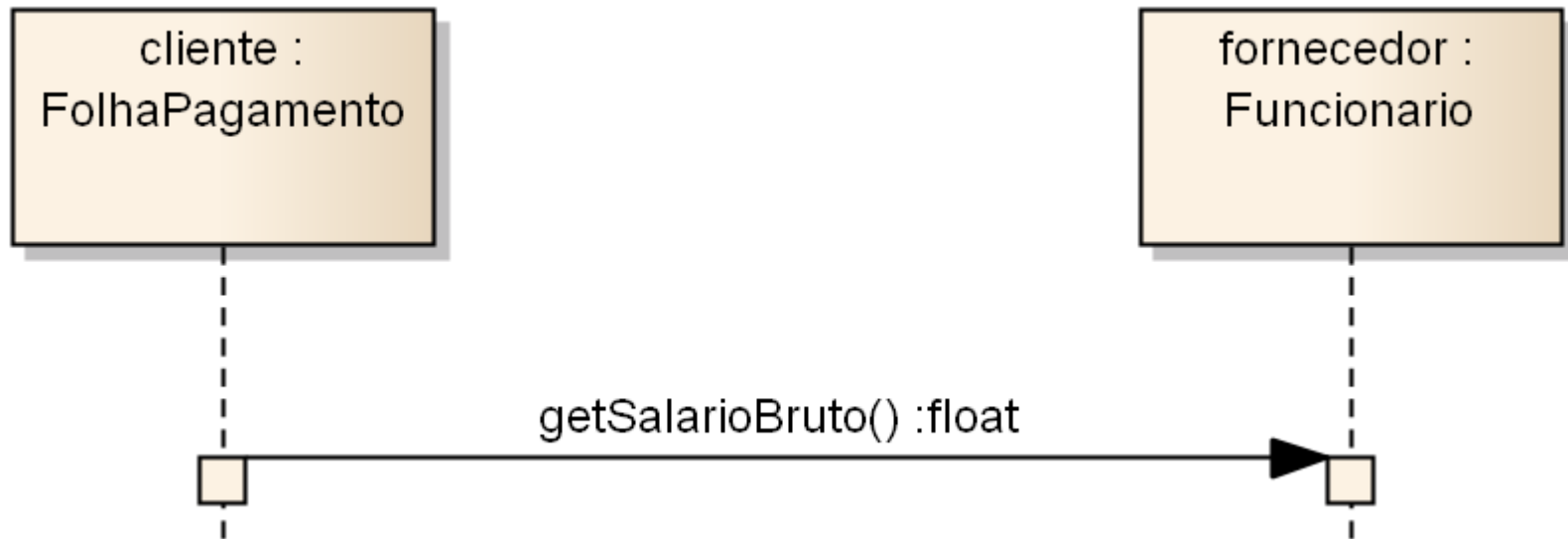
Operações e mensagens



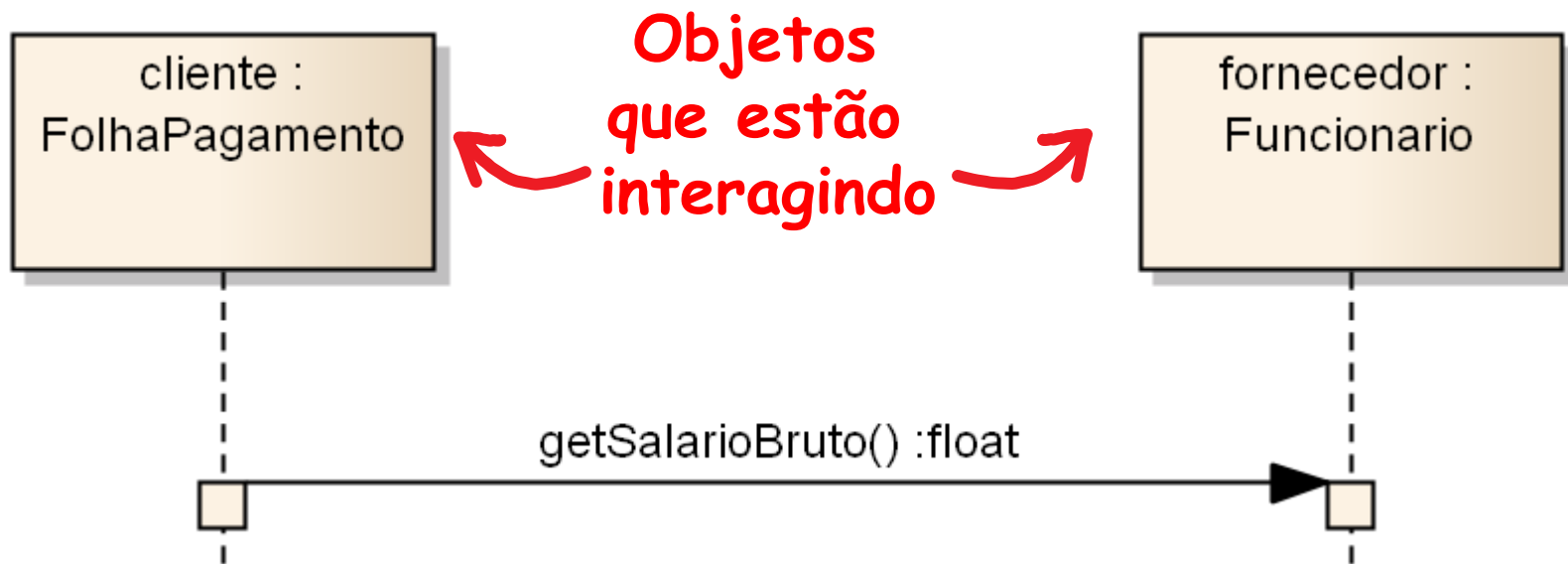
4

- Uma mensagem é uma solicitação feita por um objeto **cliente** a um objeto **fornecedor**
 - ▣ Como resultado desta solicitação, o objeto fornecedor irá modificar seu estado ou irá retornar algum valor
 - ▣ Para invocar uma operação de um objeto, deve-se enviar uma mensagem para este objeto

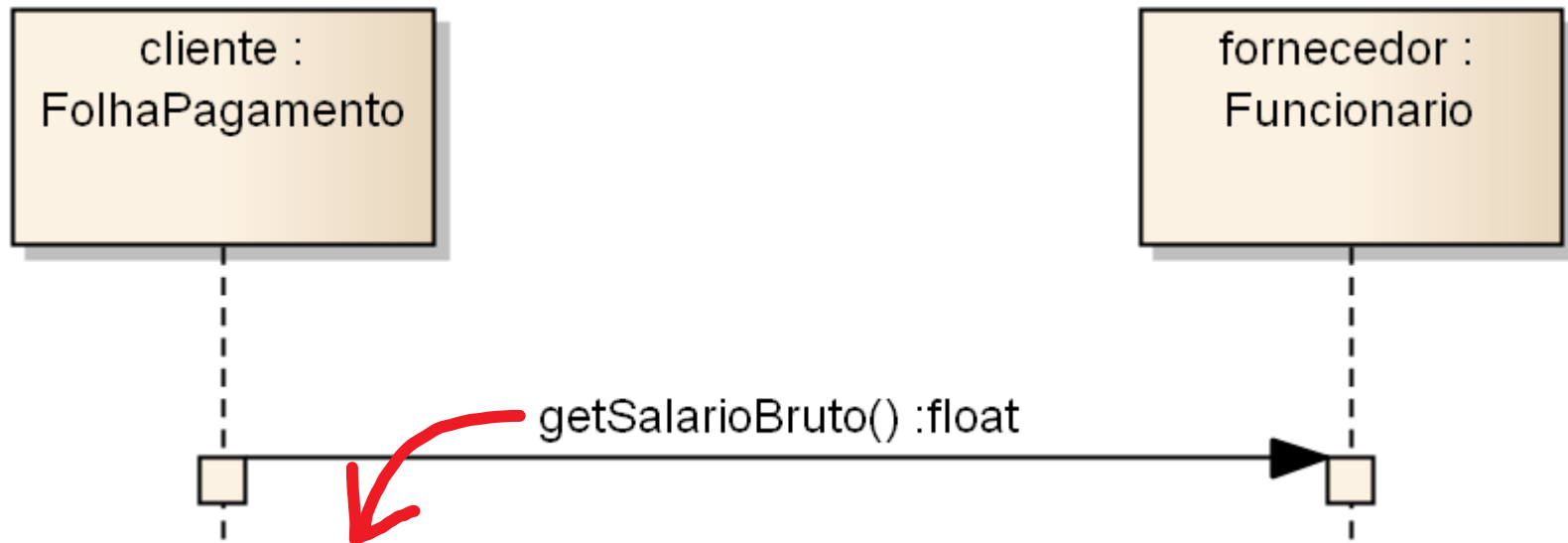
Representação de uma mensagem...



Representação de uma mensagem...

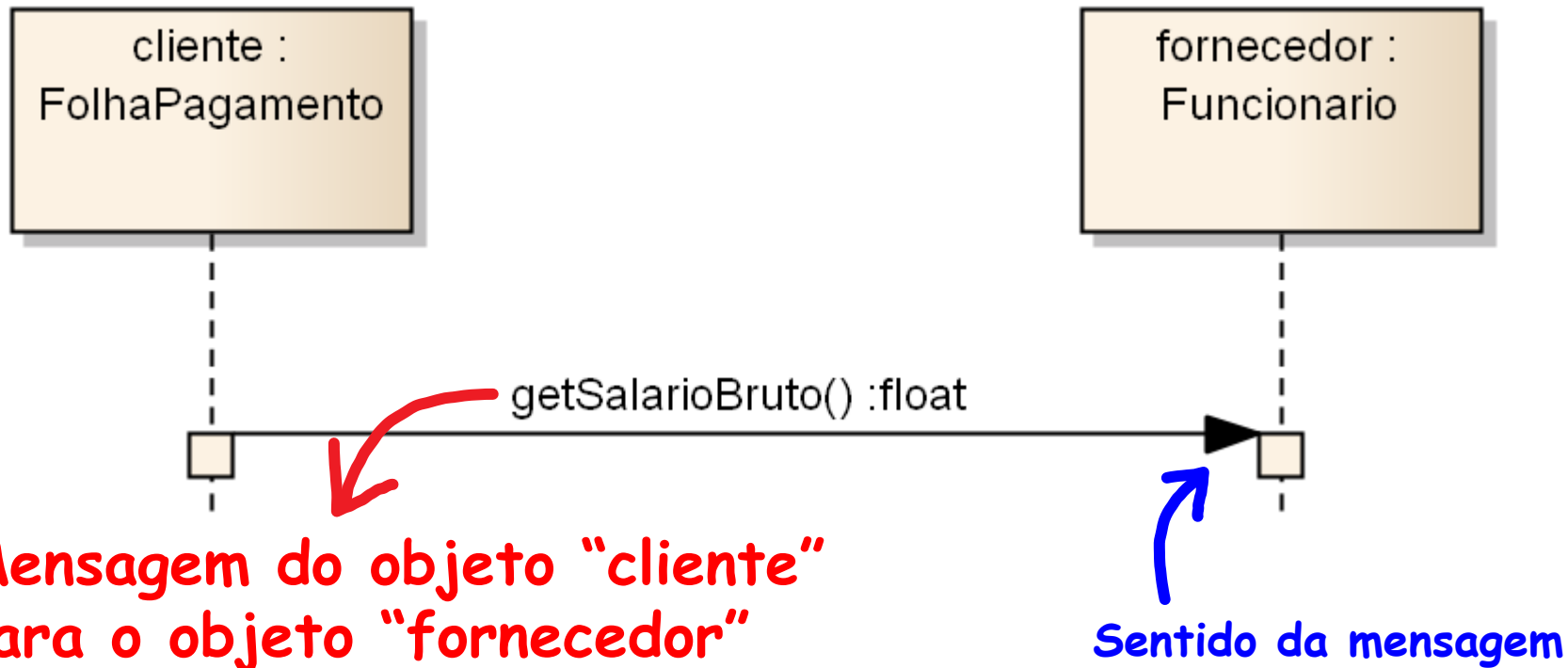


Representação de uma mensagem...

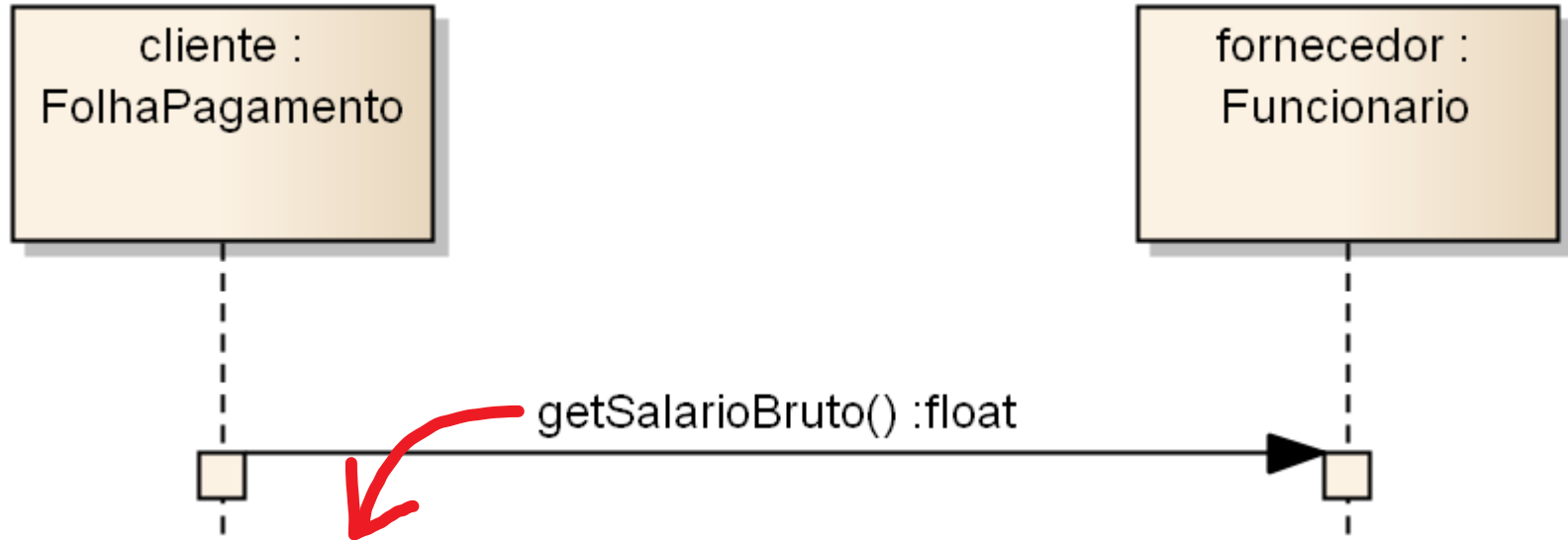


**Mensagem do objeto "cliente"
para o objeto "fornecedor"**

Representação de uma mensagem...



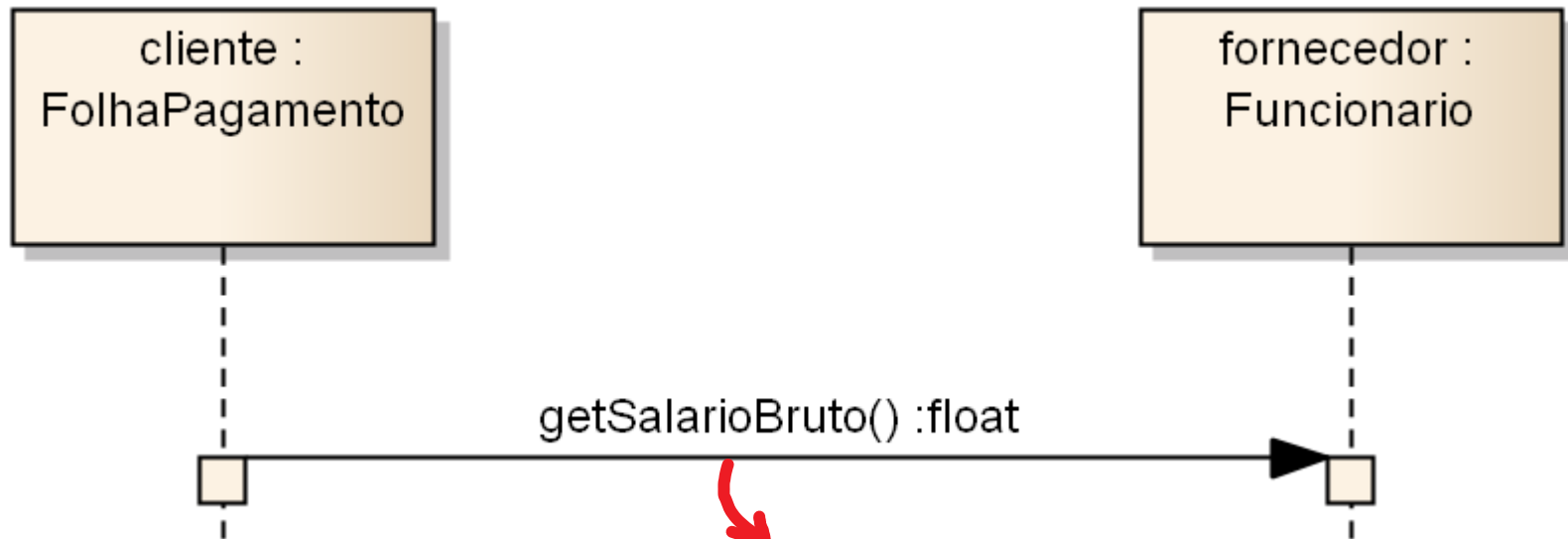
Representação de uma mensagem...



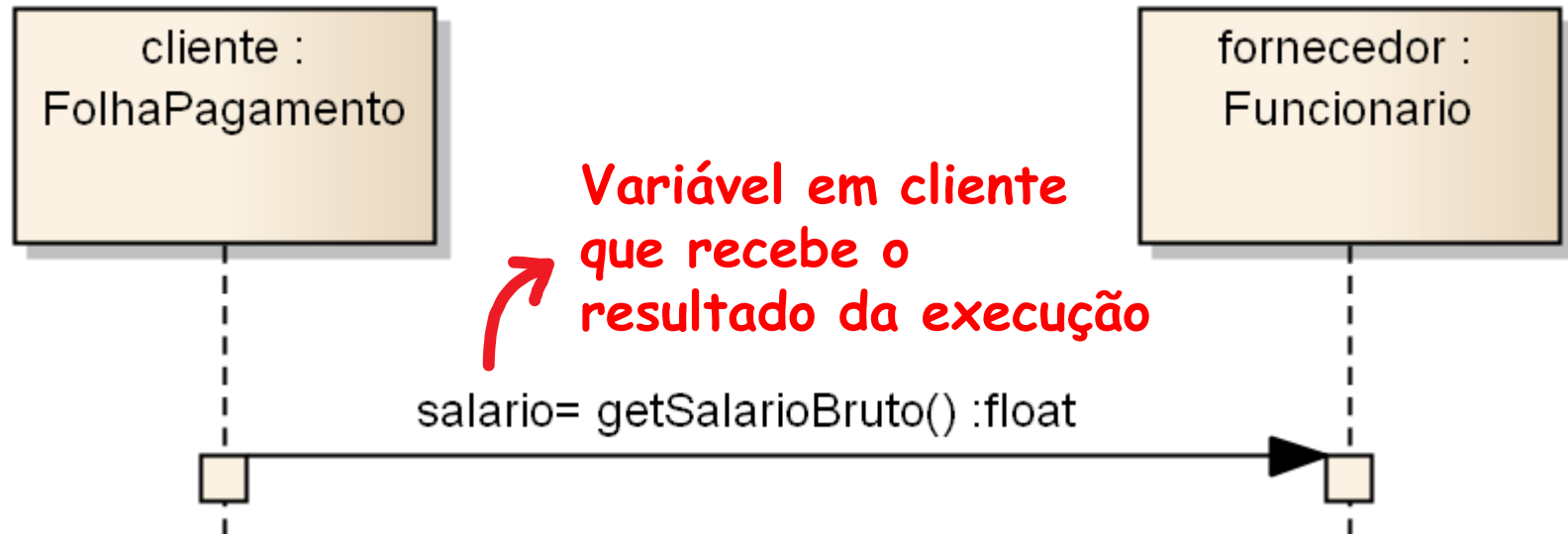
fornecedor.getSalarioBruto()

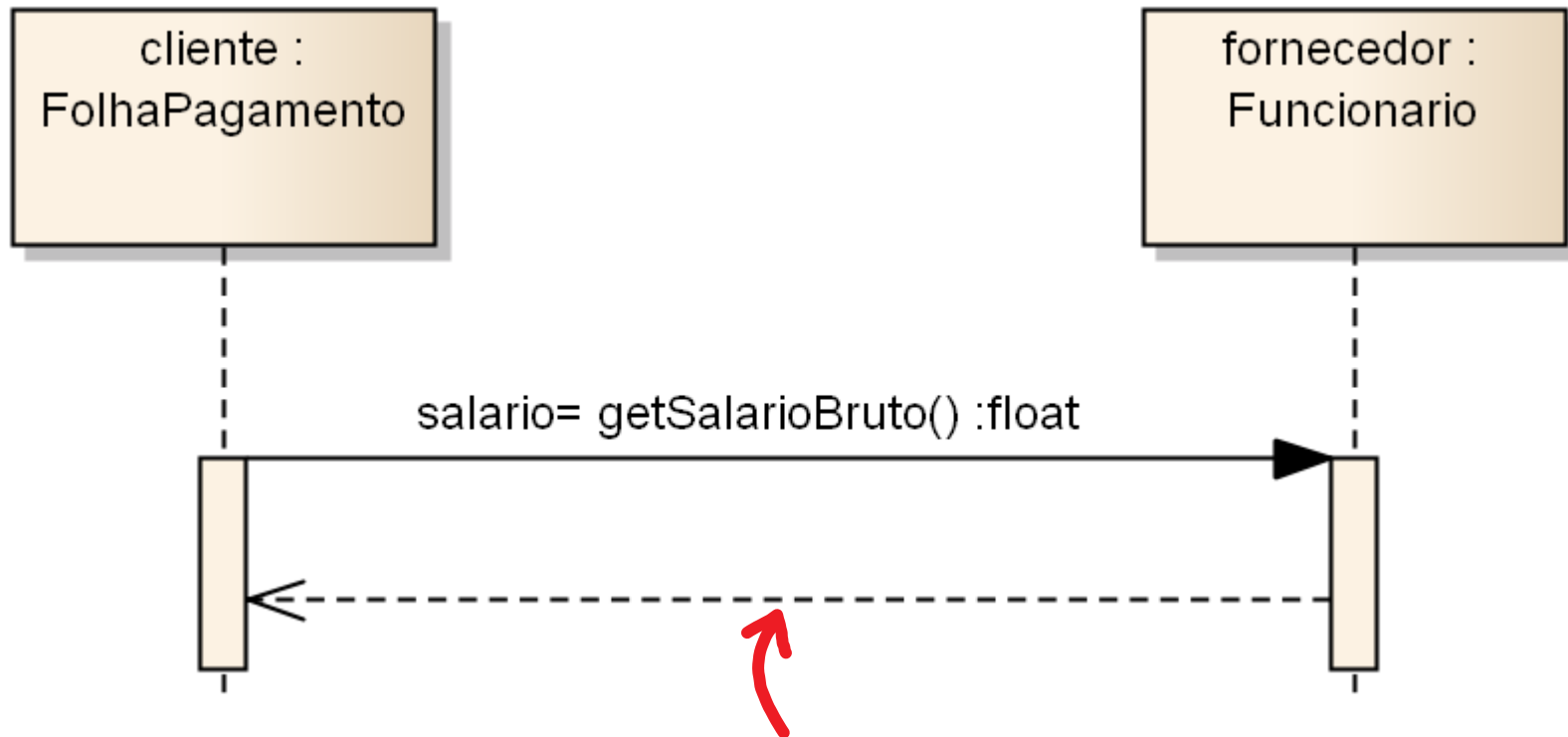
getSalarioBruto() é uma operação da classe "Funcionario"

Representação de uma mensagem...

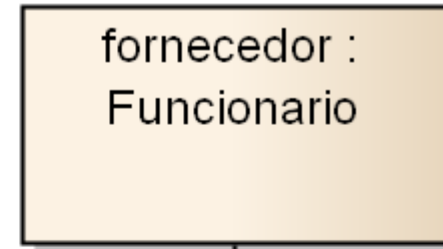
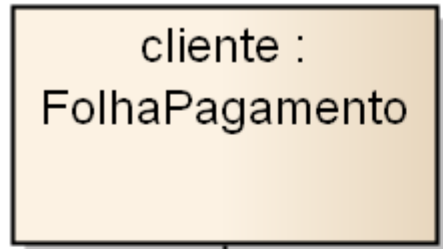


cliente está invocando o método da operação `getSalarioBruto()` que será executado pelo objeto fornecedor






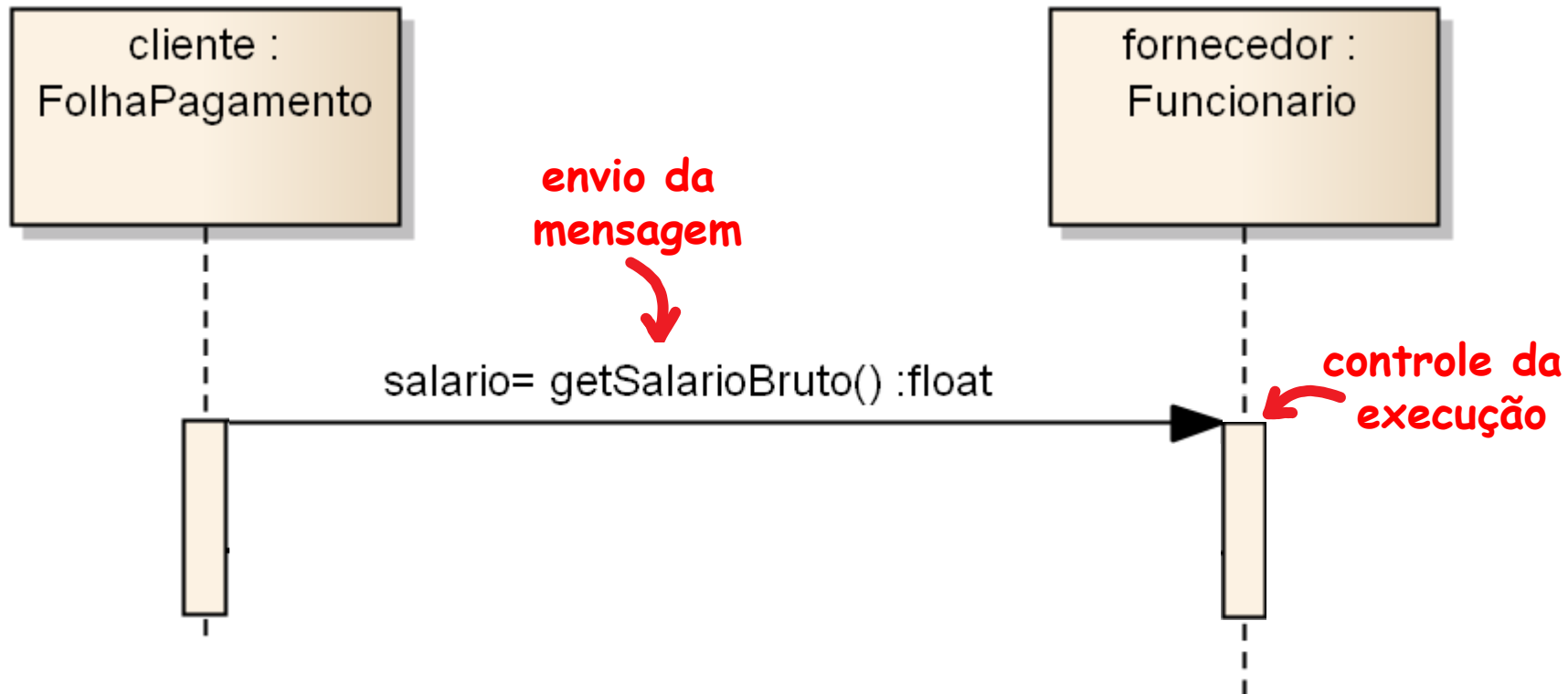
**Notação para representar o
retorno do controle da execução
(retorno da chamada)**

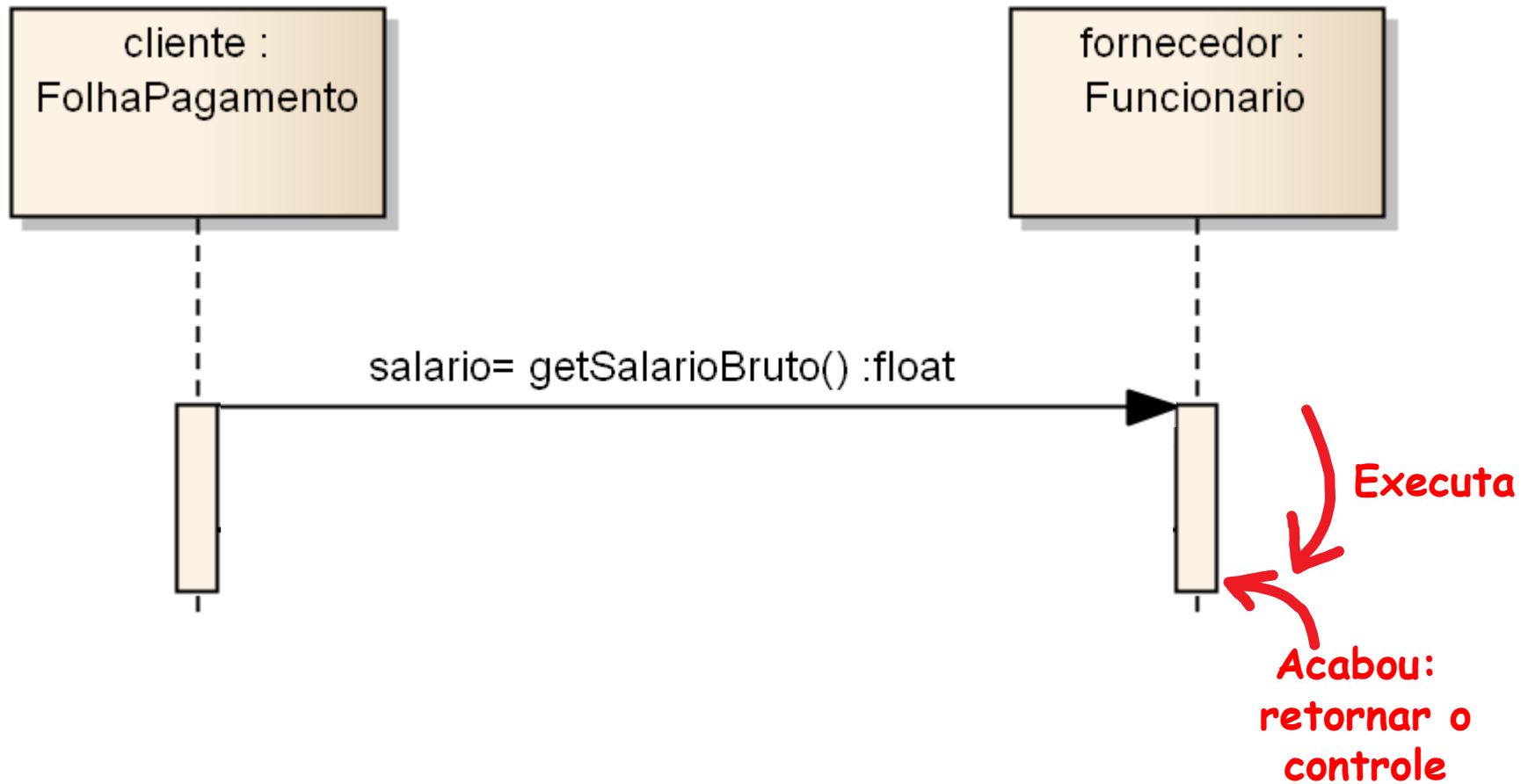


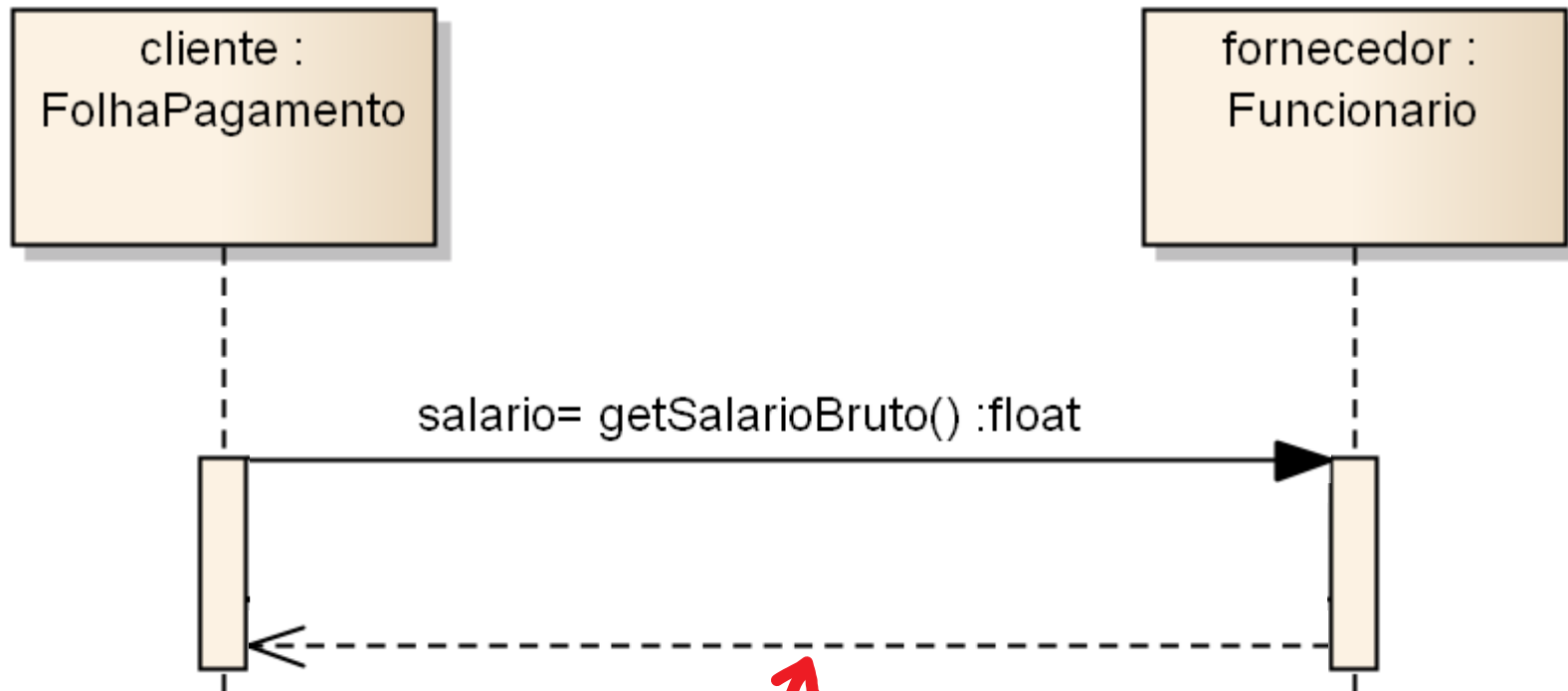
controle da
execução



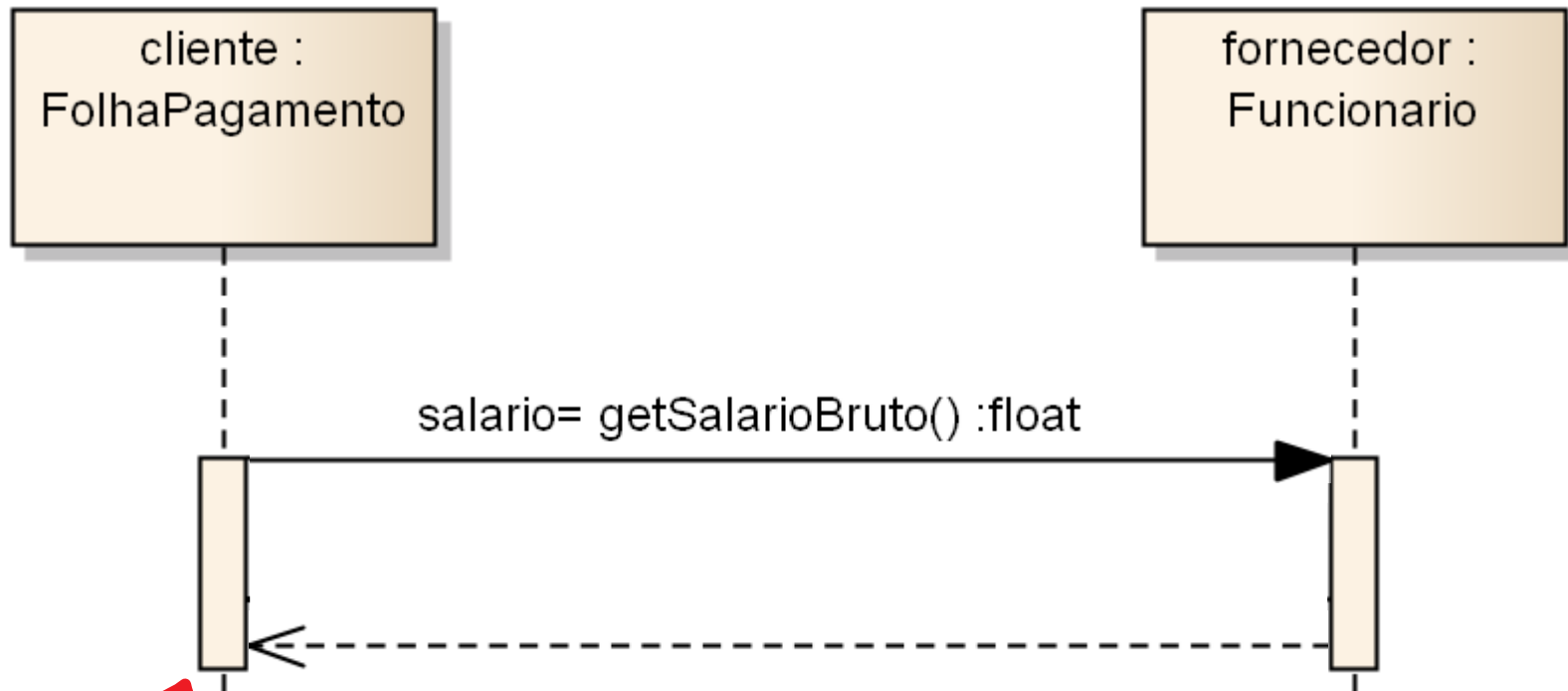
Como funciona...



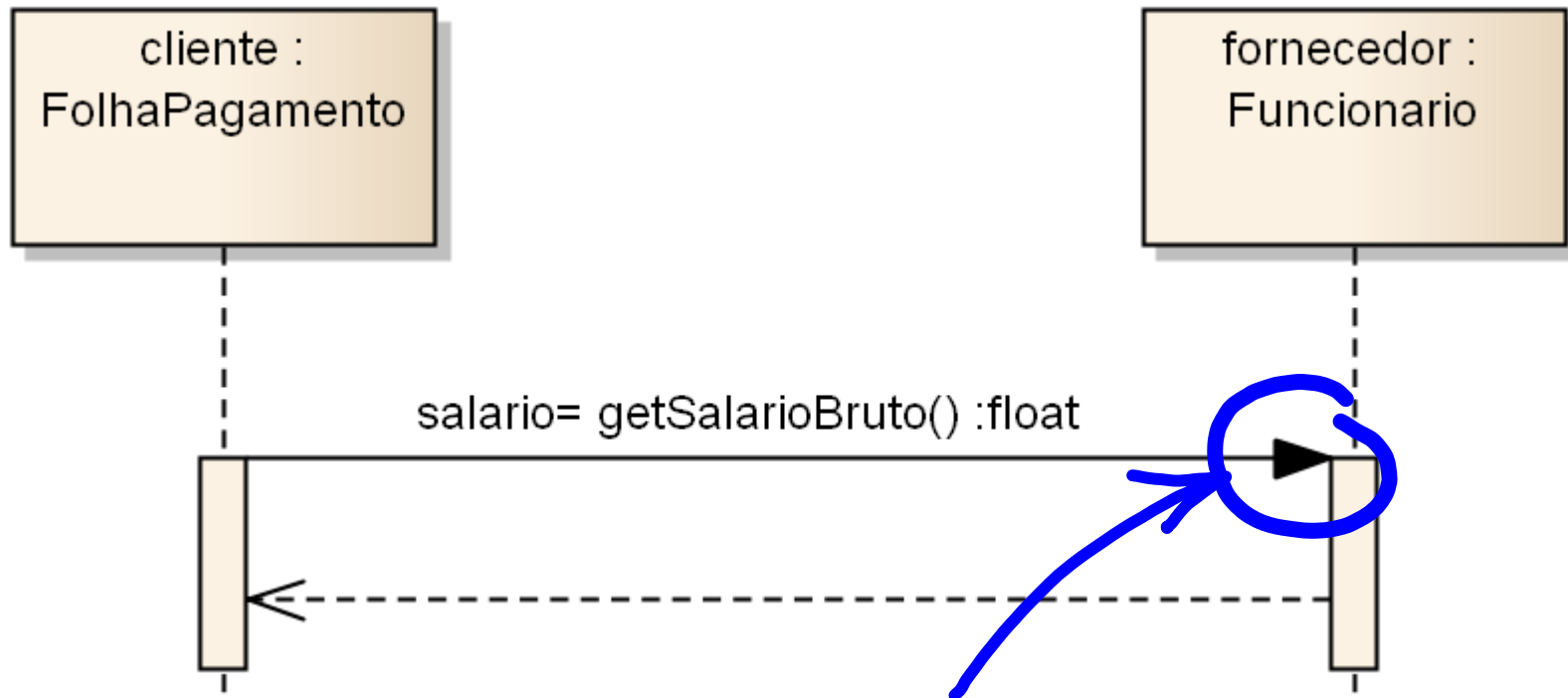




Retorno do controle da execução



O controle da execução retornou para o objeto que mandou a mensagem



**Chamada de procedimento
(mensagem síncrona)**

Mensagens síncronas



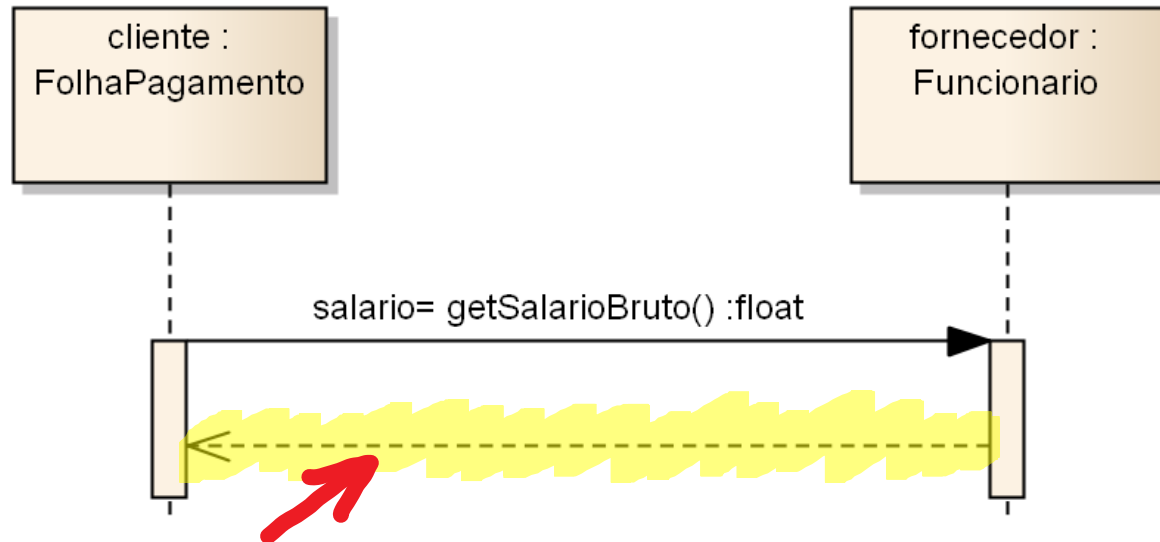
- Representa um fluxo de controle aninhado
 - ▣ Chamada a uma operação
 - ▣ Mensagem bloqueante
 - O objeto “cliente” fica esperando (bloqueado) pela execução da operação no objeto “fornecedor”
 - O objeto “cliente” só poderá fazer uma nova ação quando receber o controle de volta

Mensagens síncronas

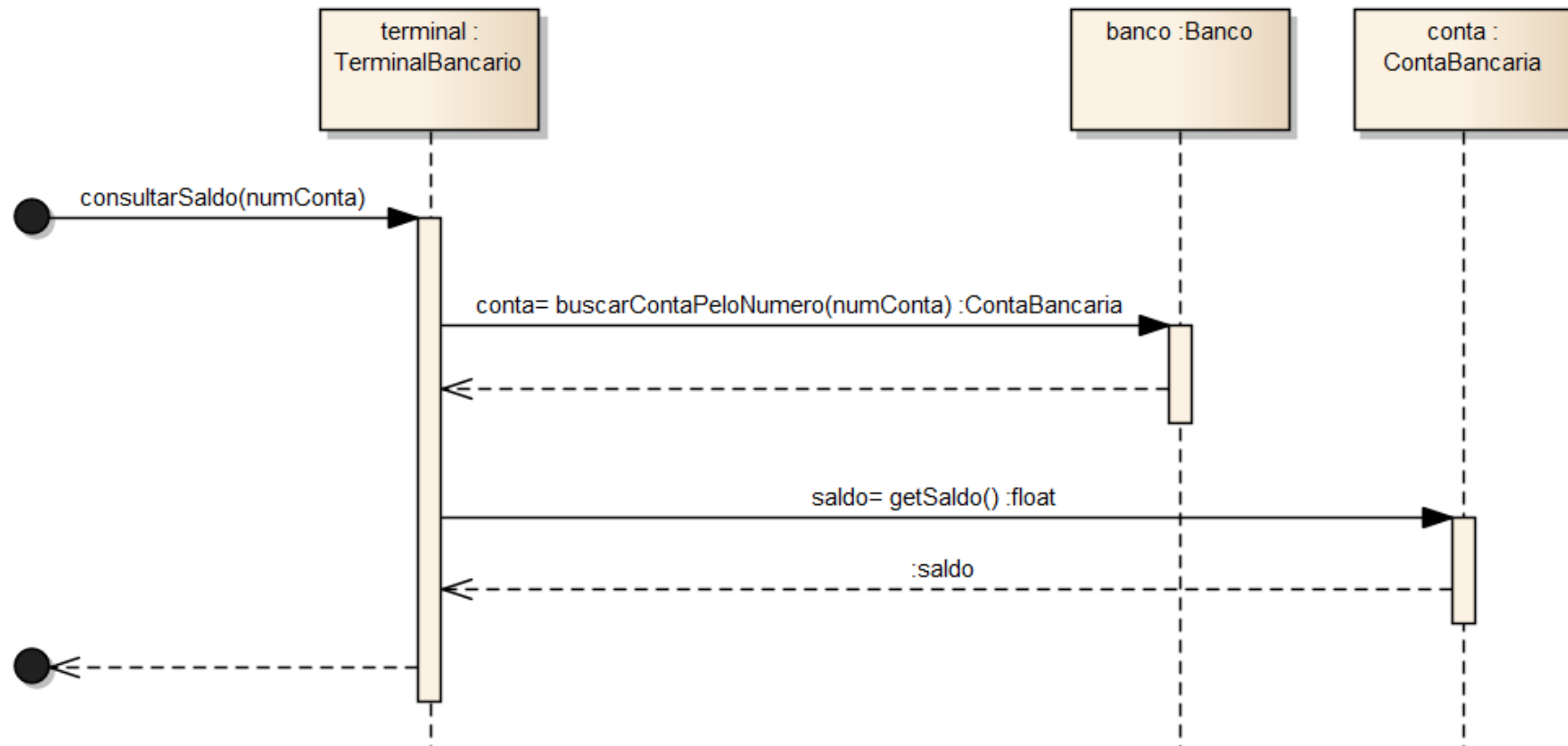


20

- Por ser uma mensagem bloqueante, utilizamos uma notação para indicar o **retorno do controle** ao objeto que originou a mensagem

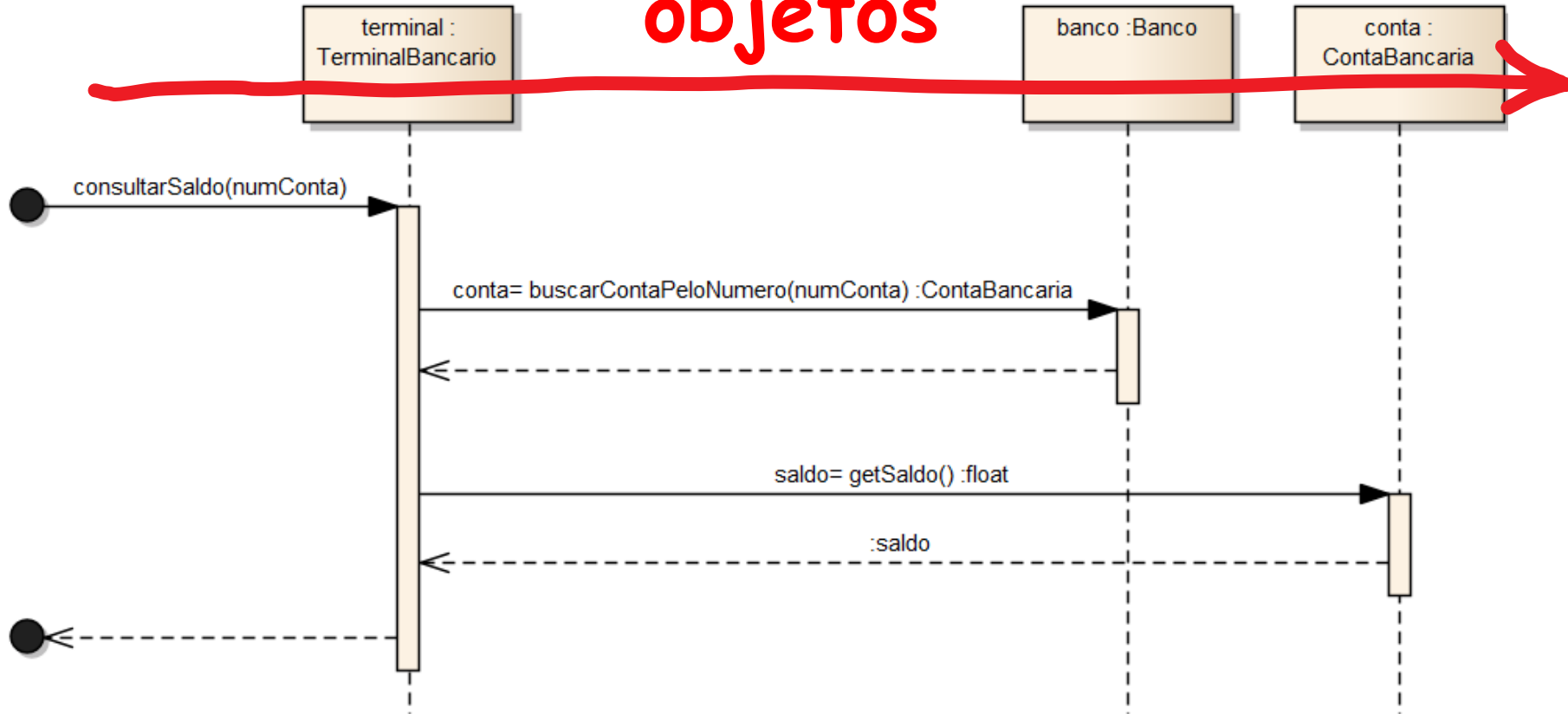


Um exemplo...

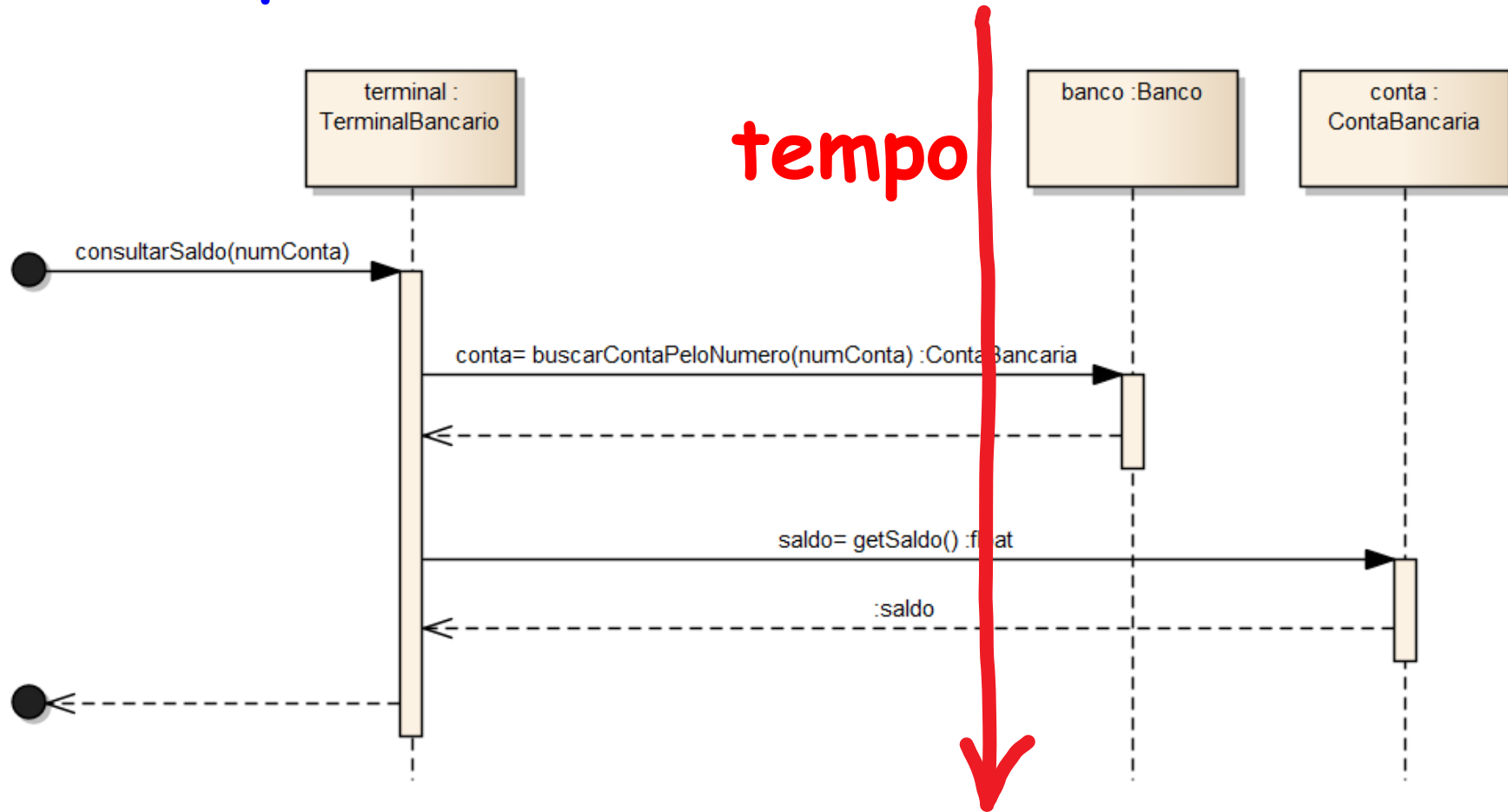


Um exemplo...

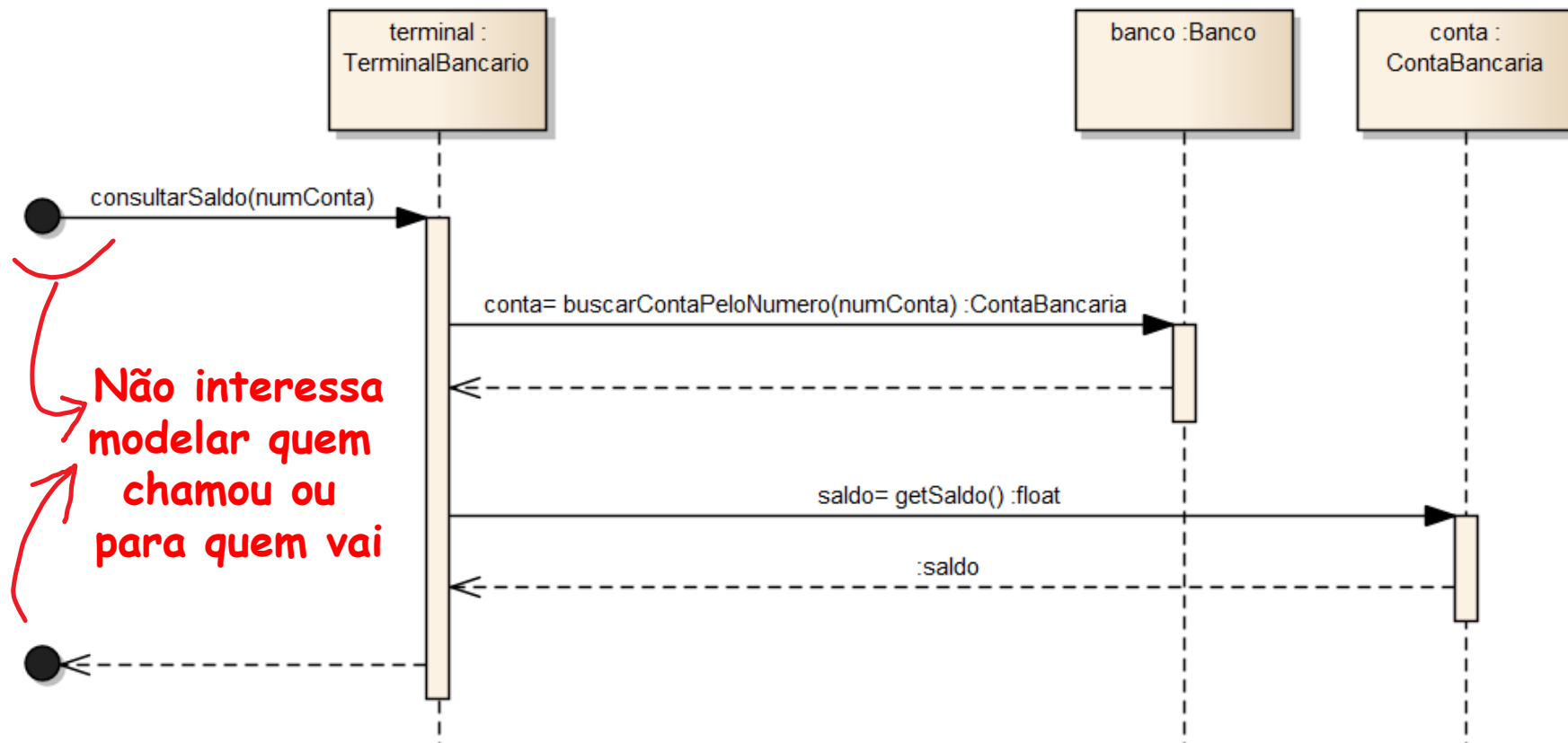
objetos



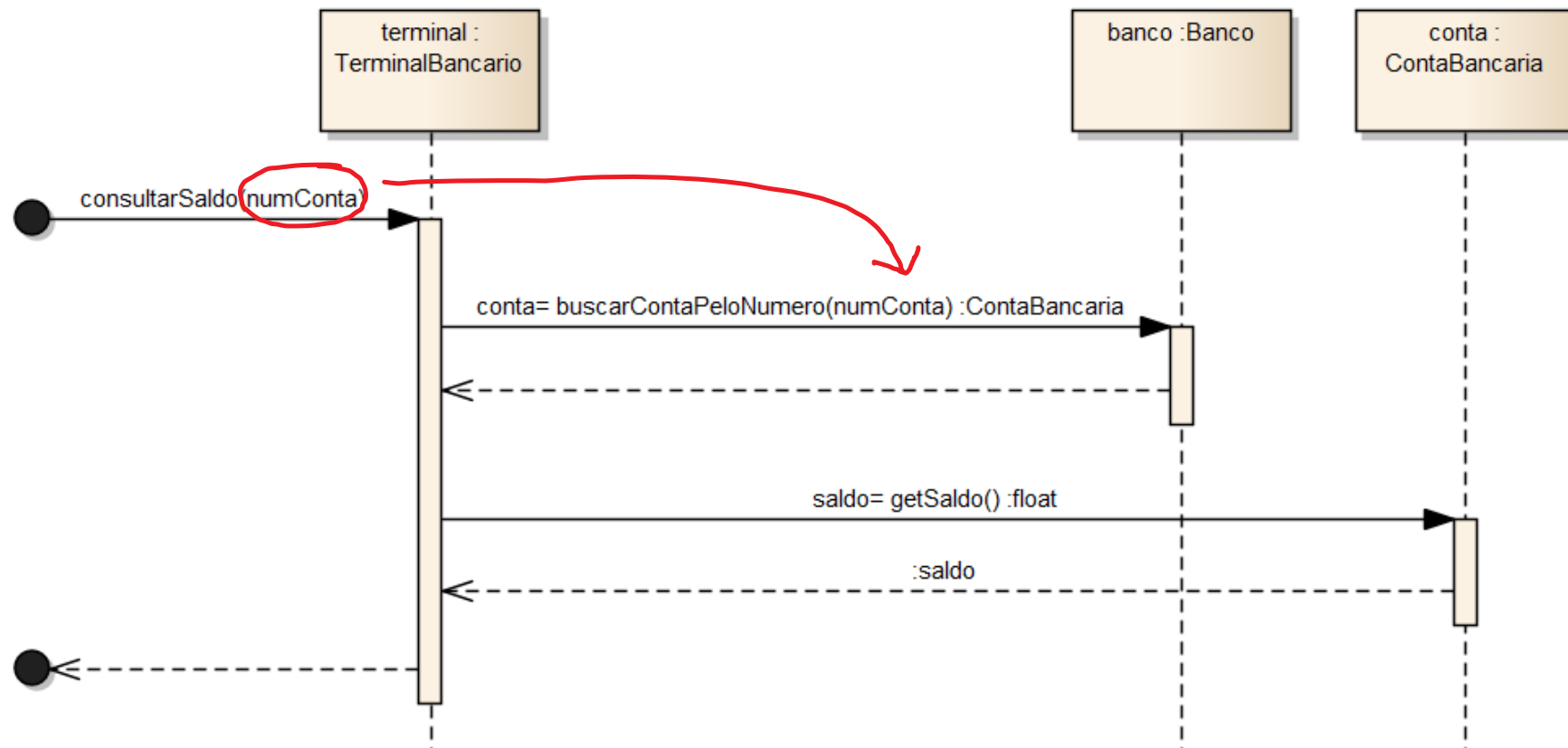
Um exemplo...



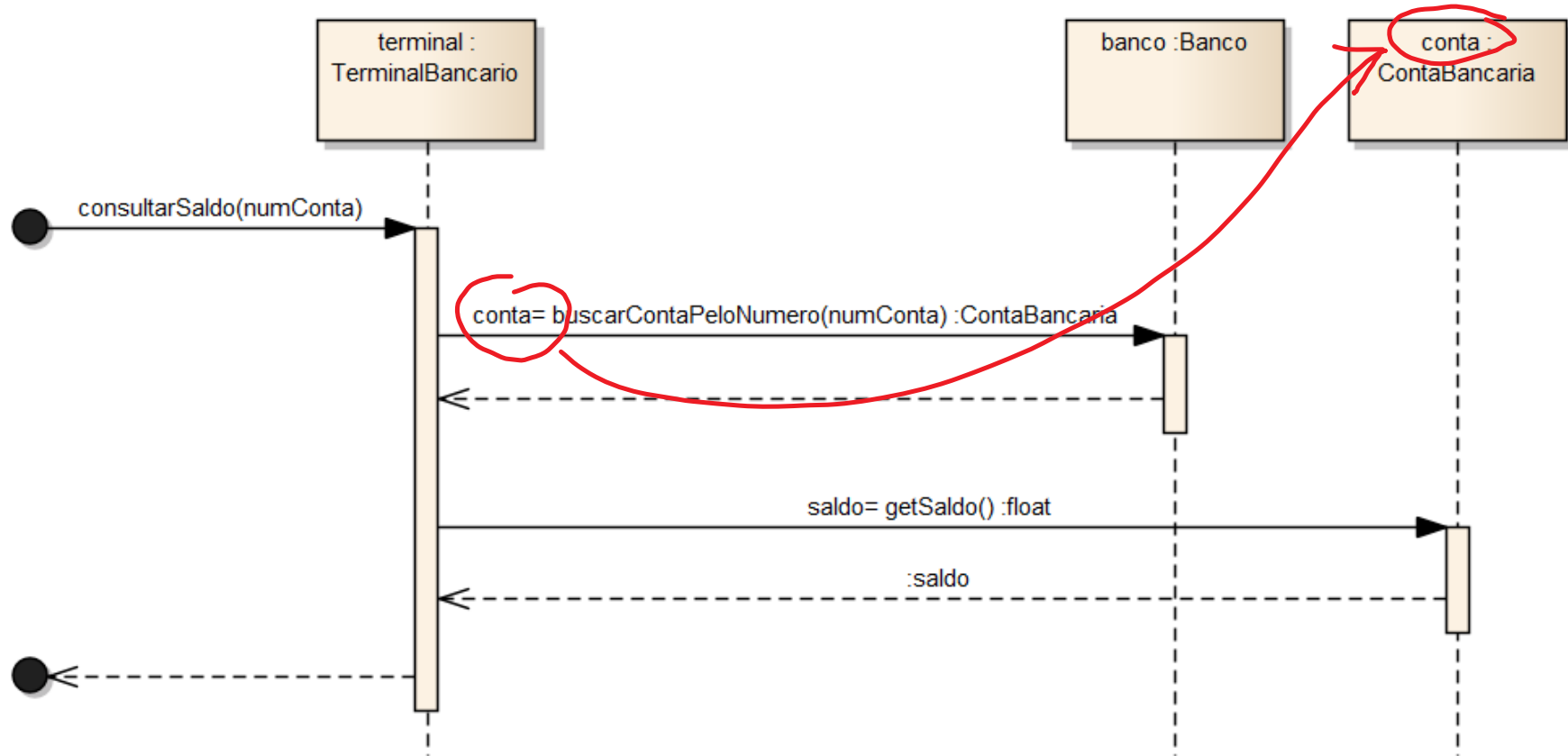
Um exemplo...



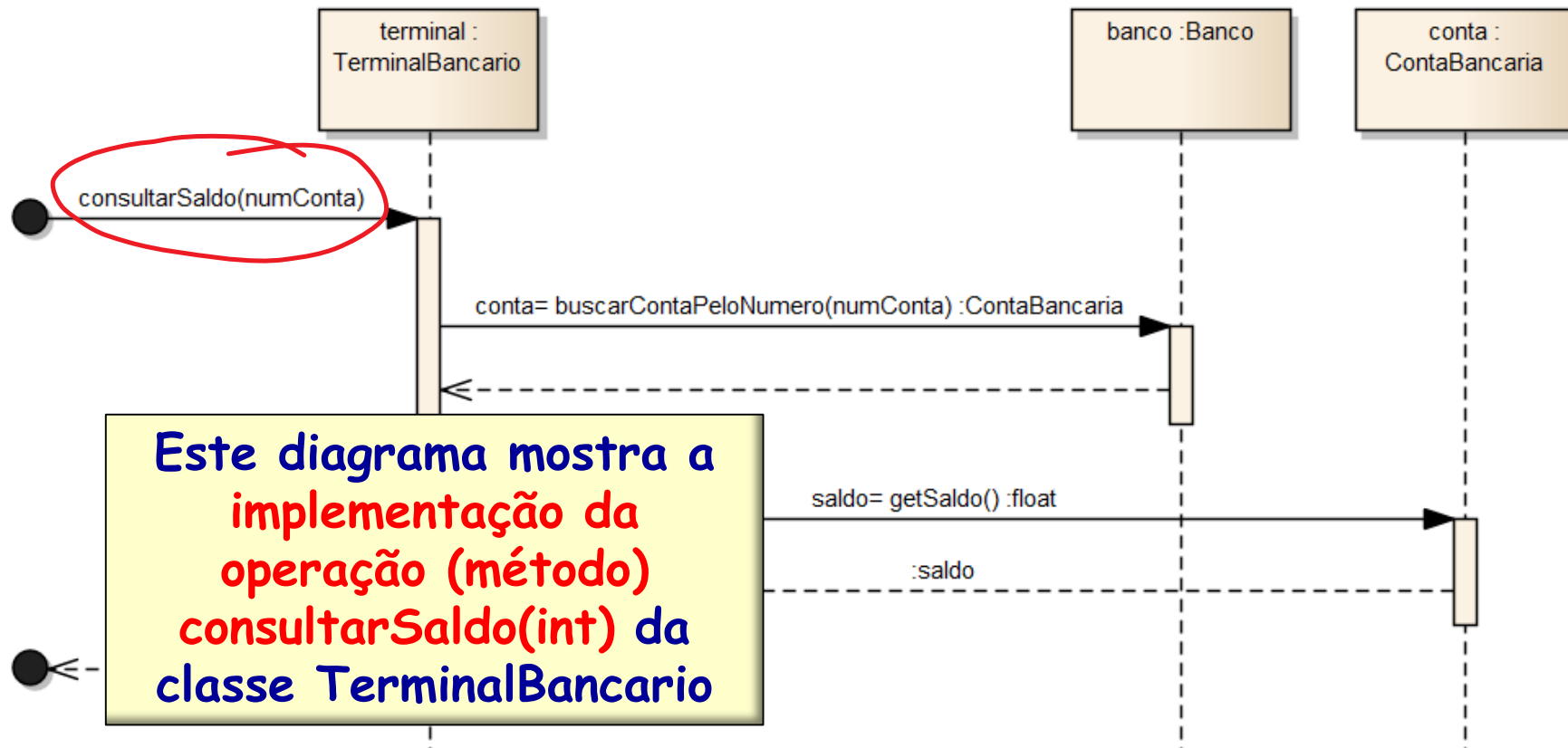
Um exemplo...

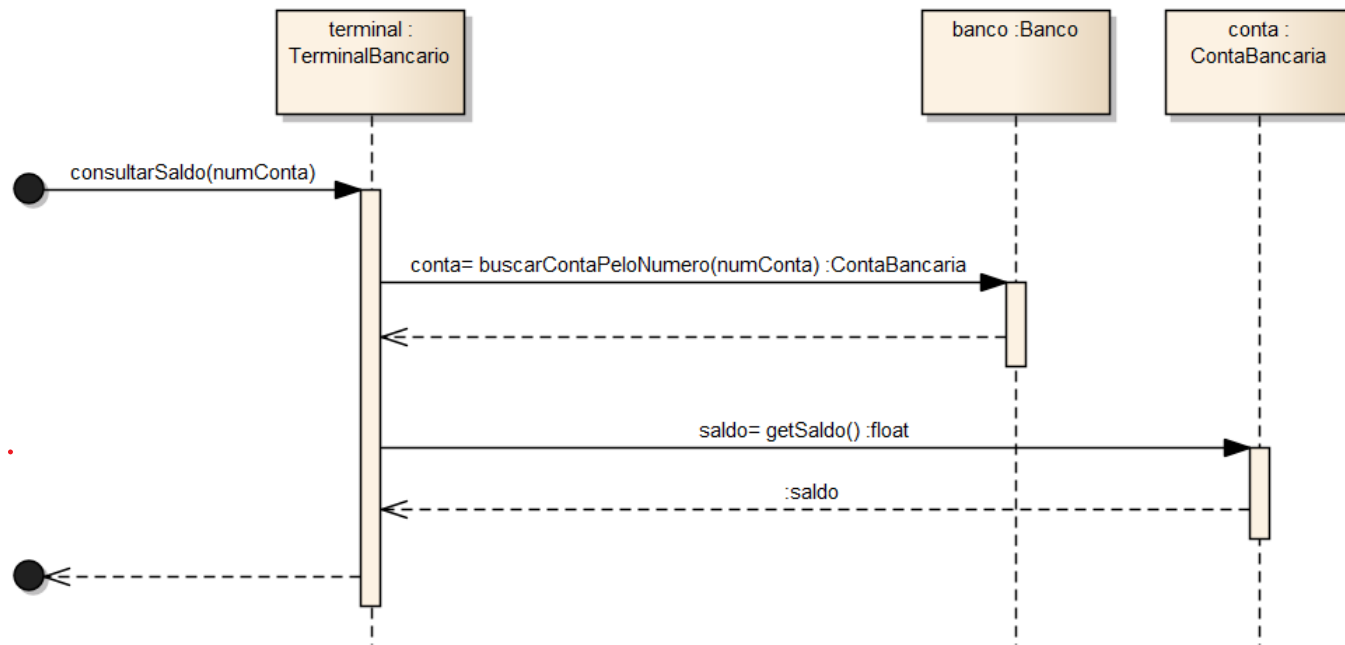


Um exemplo...

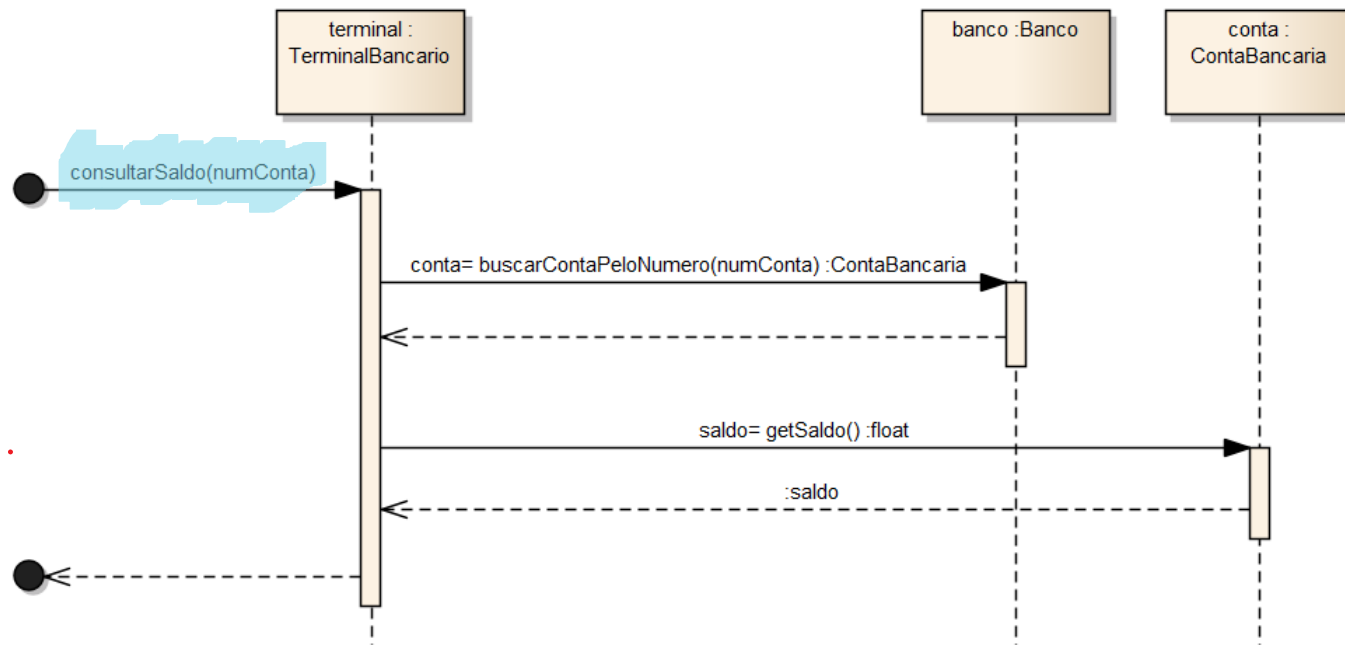


Um exemplo...

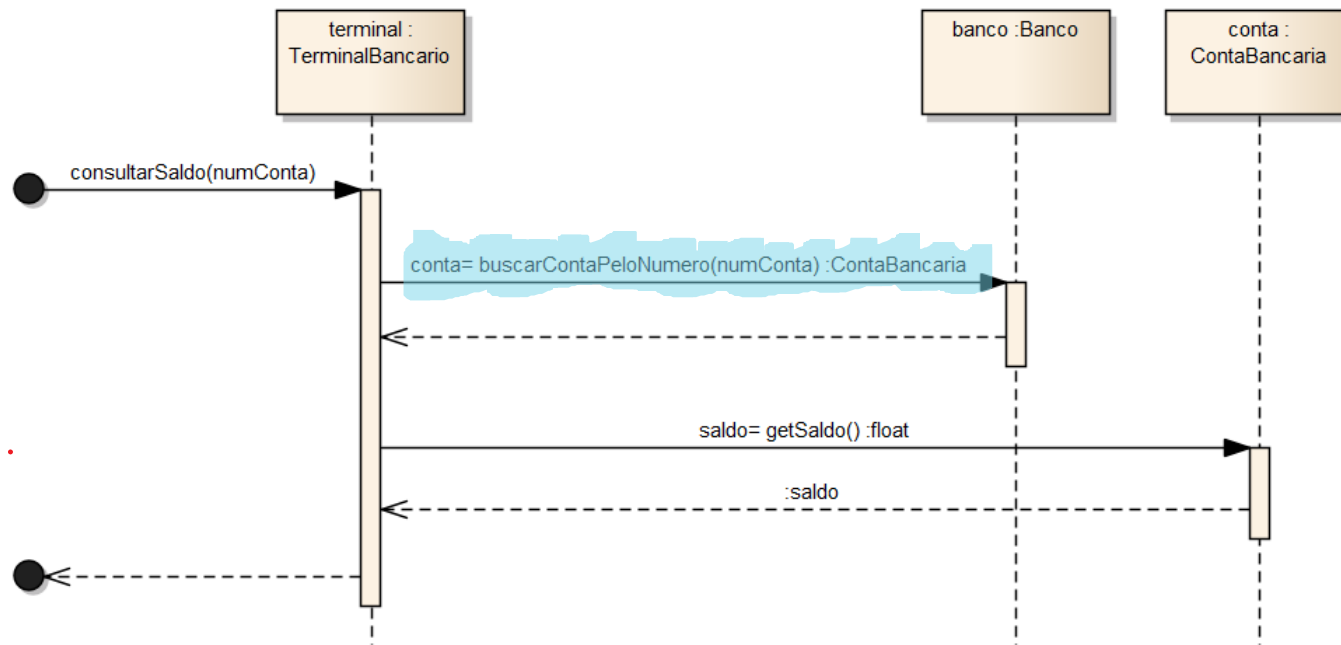




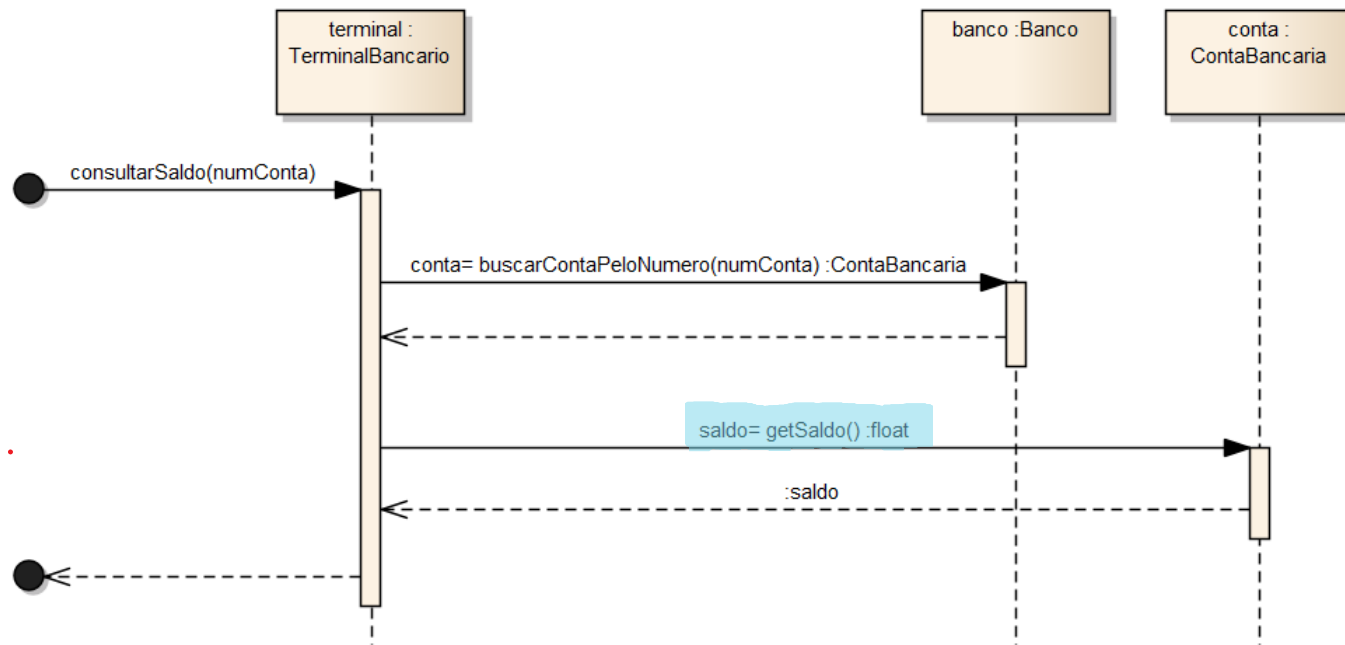
```
public void consultarSaldo(int numConta) {  
    ContaBancaria conta = banco.buscarContaPeloNumero(numConta);  
    float saldo = conta.getSaldo();  
    System.out.println(saldo);  
}
```



```
public void consultarSaldo(int numConta) {  
    ContaBancaria conta = banco.buscarContaPeloNumero(numConta);  
    float saldo = conta.getSaldo();  
    System.out.println(saldo);  
}
```



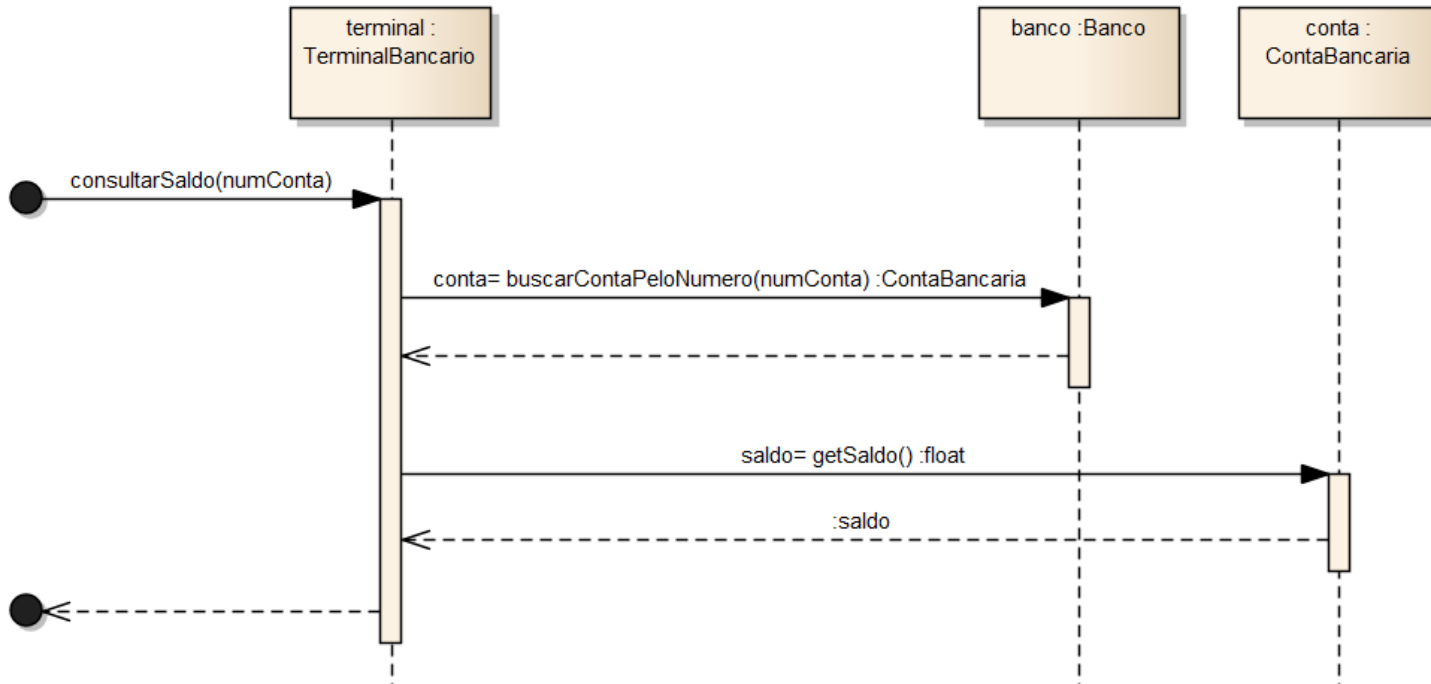
```
public void consultarSaldo(int numConta) {  
    ContaBancaria conta = banco.buscarContaPeloNumero(numConta);  
    float saldo = conta.getSaldo();  
    System.out.println(saldo);  
}
```



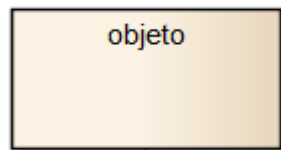
```
public void consultarSaldo(int numConta) {  
    ContaBancaria conta = banco.buscarContaPeloNumero(numConta);  
    float saldo = conta.getSaldo();  
    System.out.println(saldo);  
}
```

Por convenção, objetos são colocados da esquerda para a direita, conforme a participação deles na interação

↪ Mas os objetos podem ser colocados em qualquer ordem

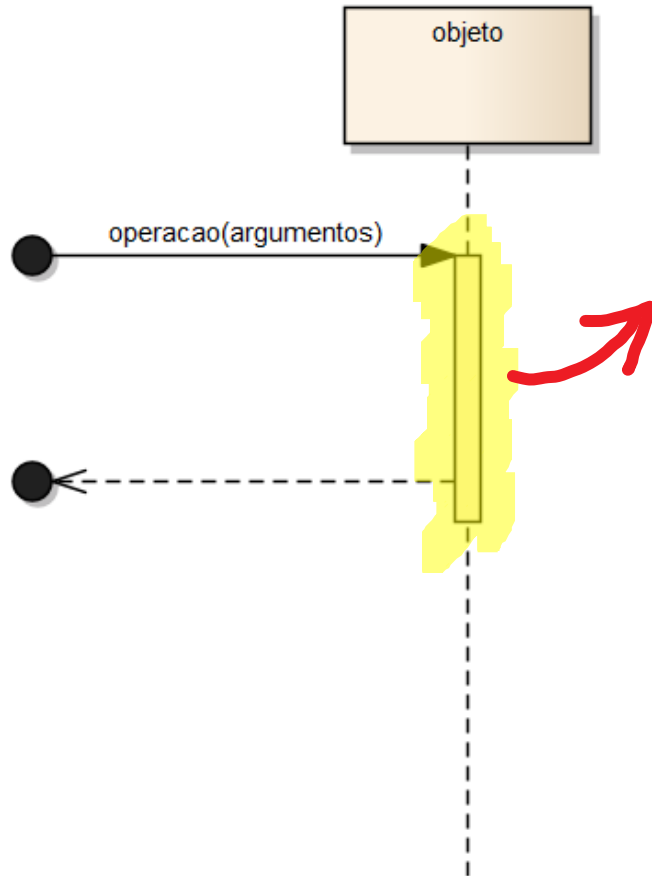


Linha de vida (Lifeline)



- Representa um objeto que participa da interação
 - Não faz sentido definir multiplicidade
 - A linha de vida (linha tracejada) representa o tempo de vida do objeto
 - Somente objetos ainda vivos podem enviar e receber mensagens
- - Notação para indicar que o objeto não existe mais
- O objeto foi destruído

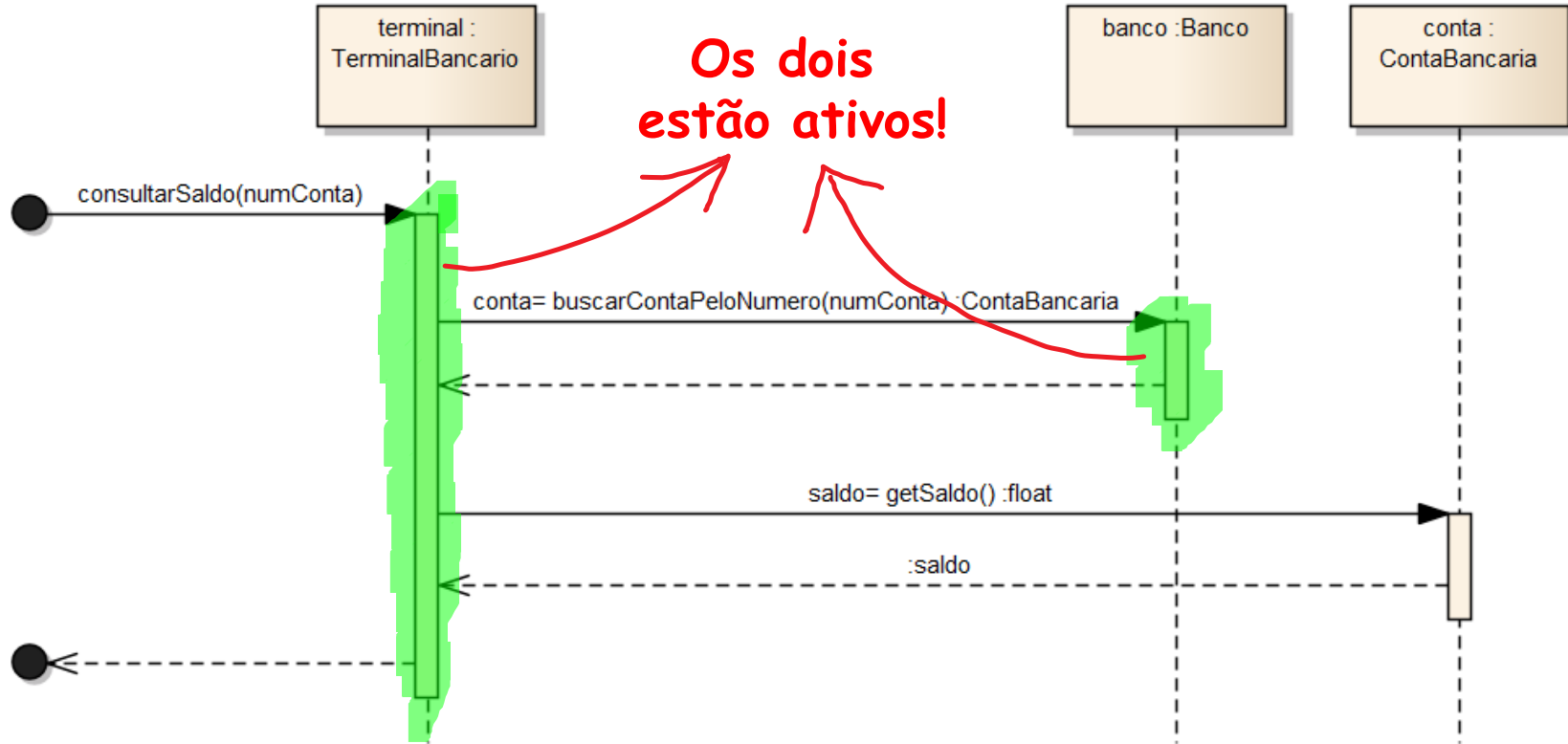
Foco de controle / Ativação



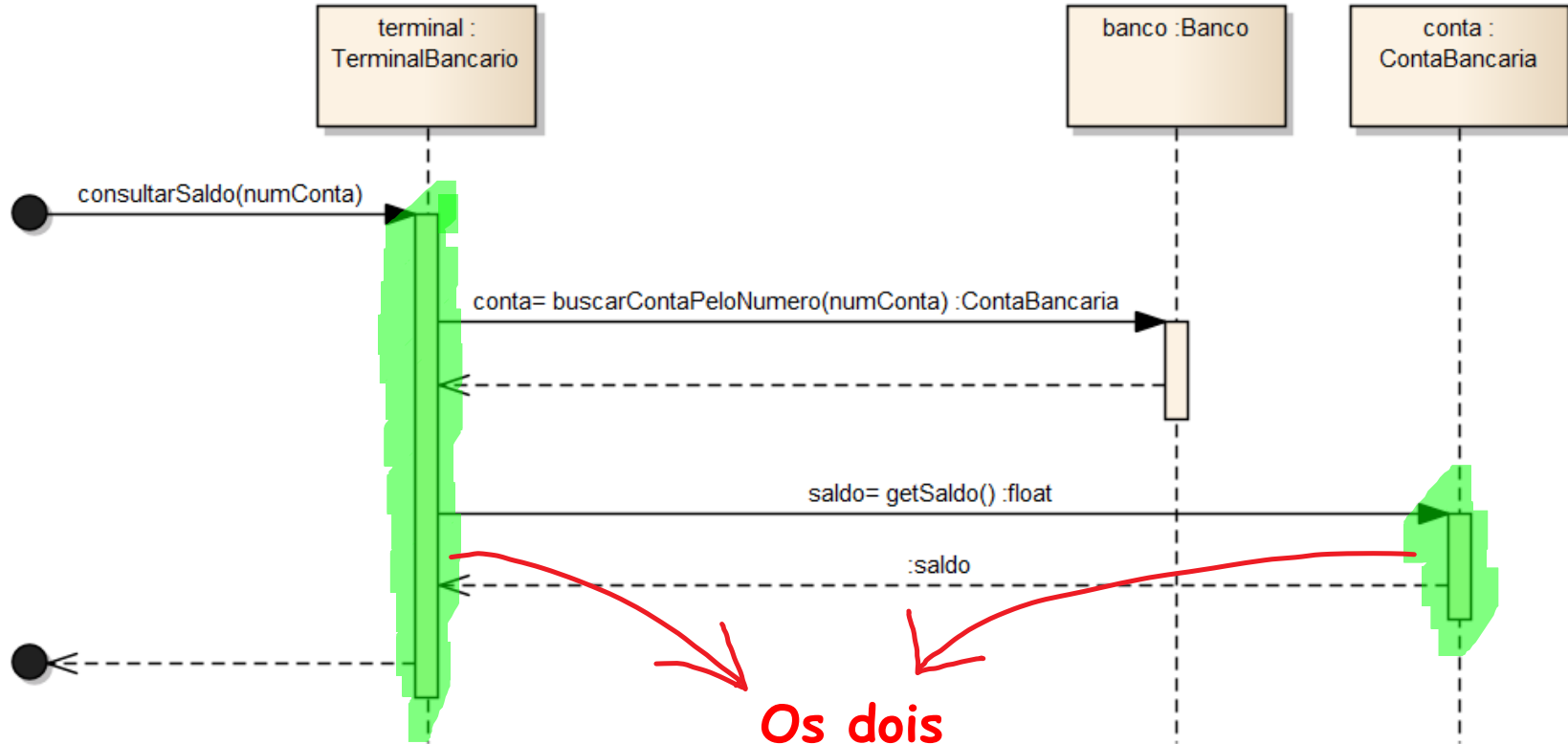
- ▣ Indica os períodos em que o objeto participa ativamente da interação
- Quando o objeto está executando um método

É possível ter dois objetos ativos ao mesmo tempo?

Foco de controle / Ativação

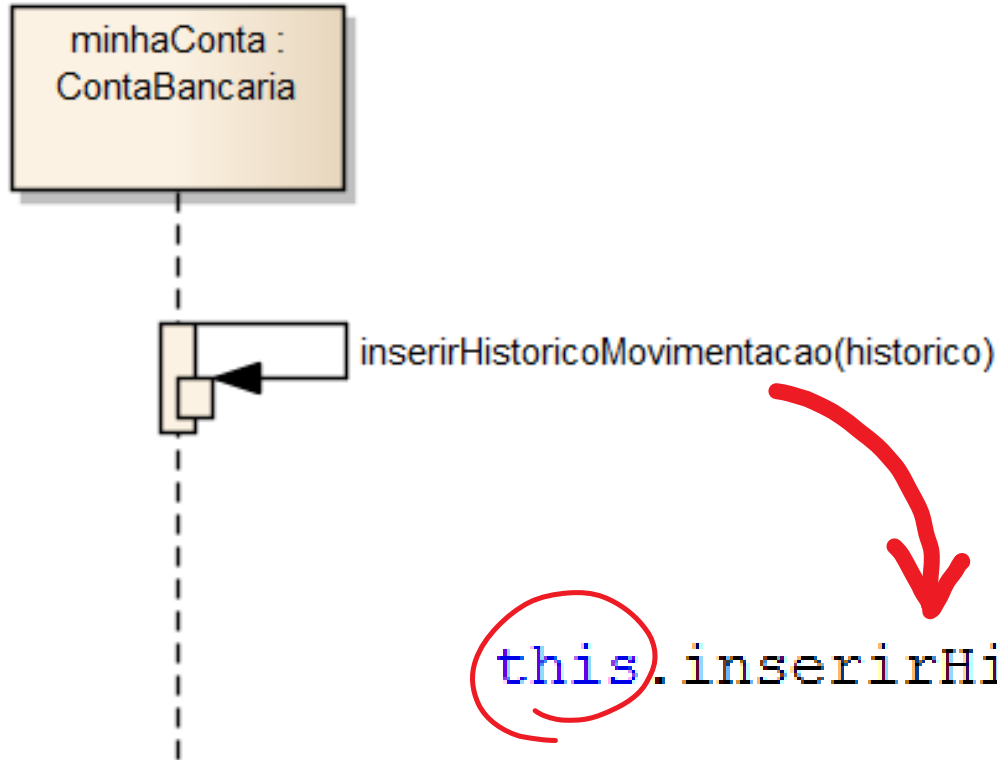


Foco de controle / Ativação



Os dois
estão ativos!

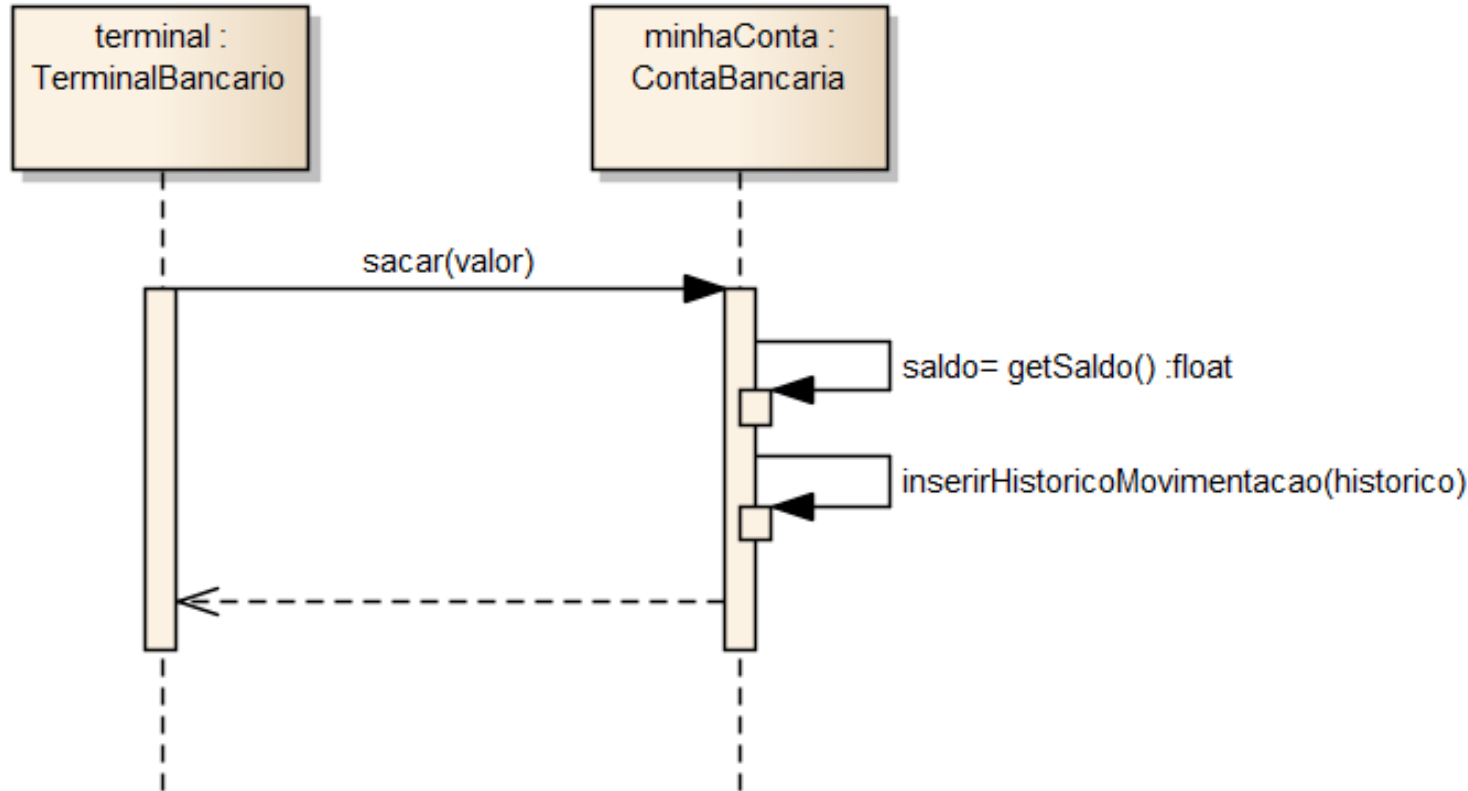
Auto-mensagem (self-message)



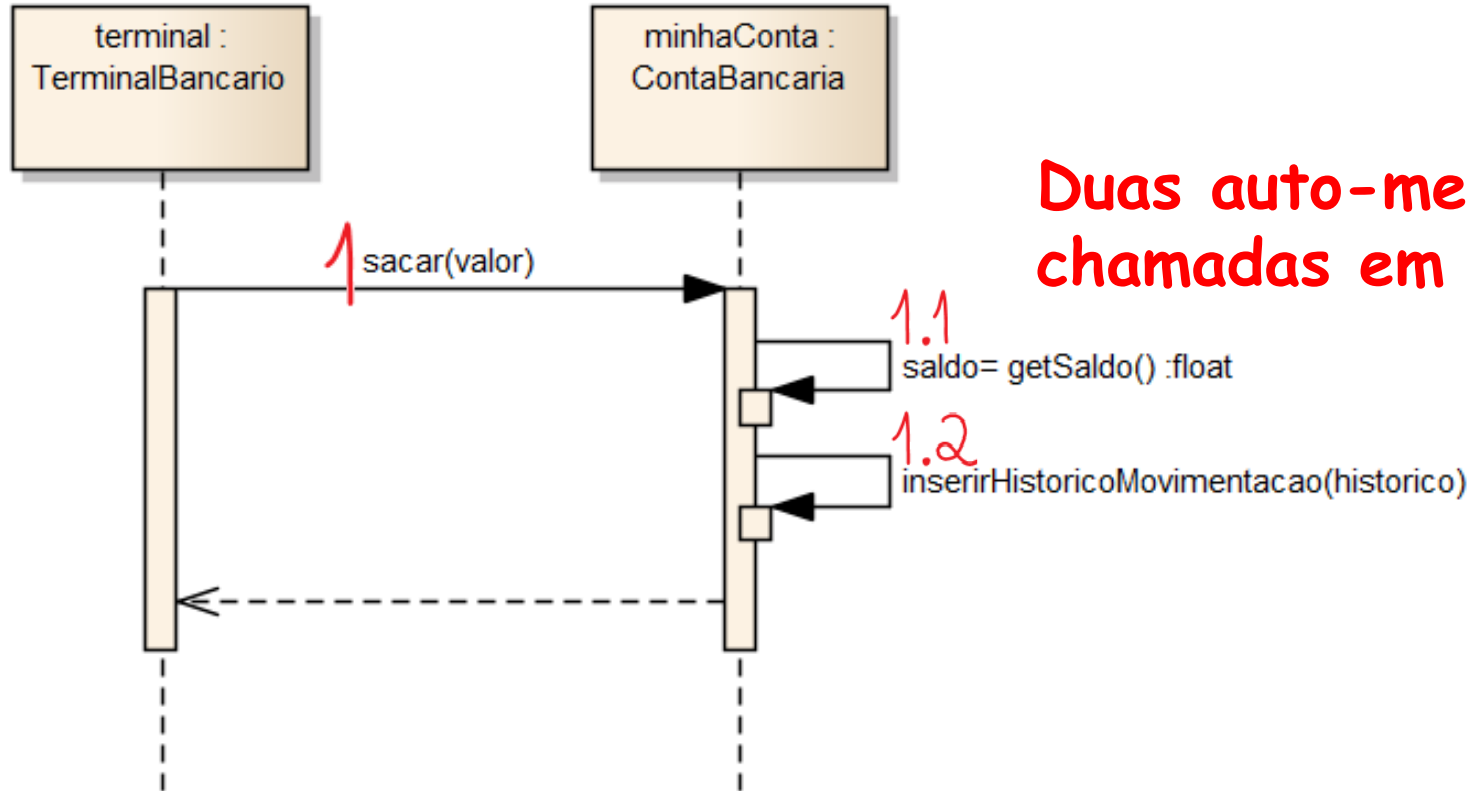
A mensagem é enviada para o próprio objeto

```
this.inserirHistorico(historico);
```

Auto-mensagem (self-message)

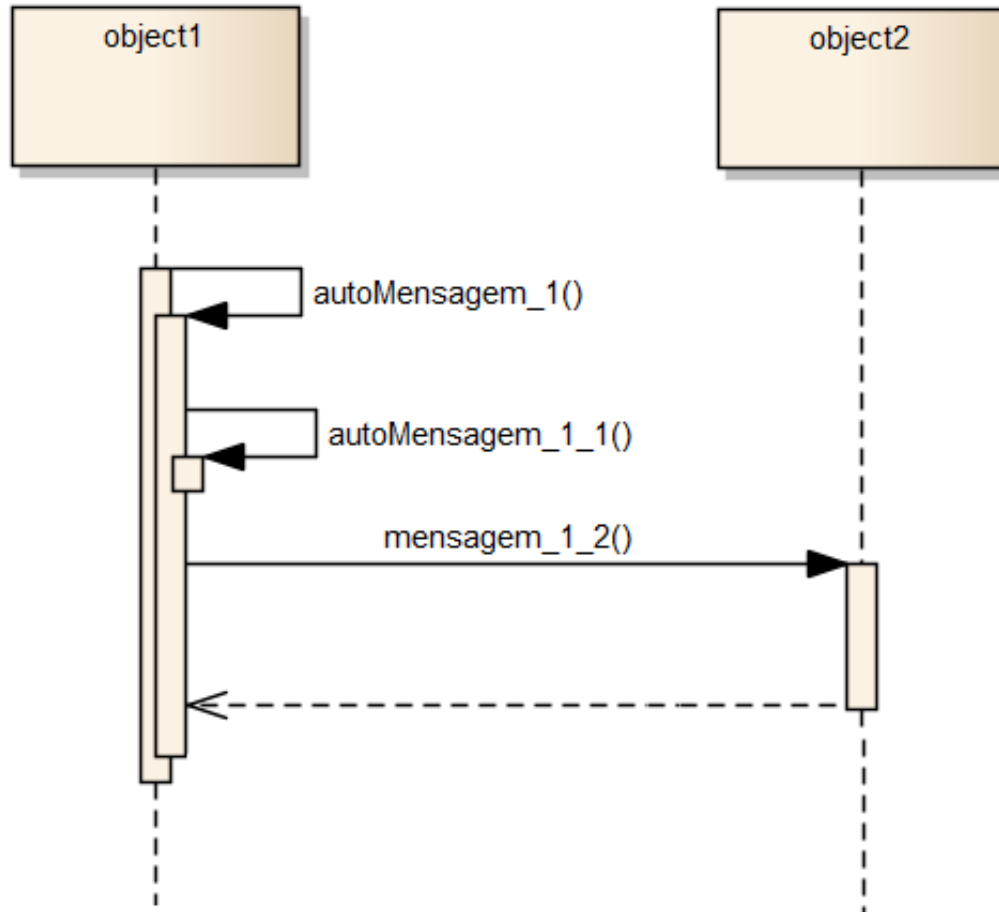


Auto-mensagem (self-message)



Duas auto-mensagens chamadas em sequência

Auto-mensagem (self-message)

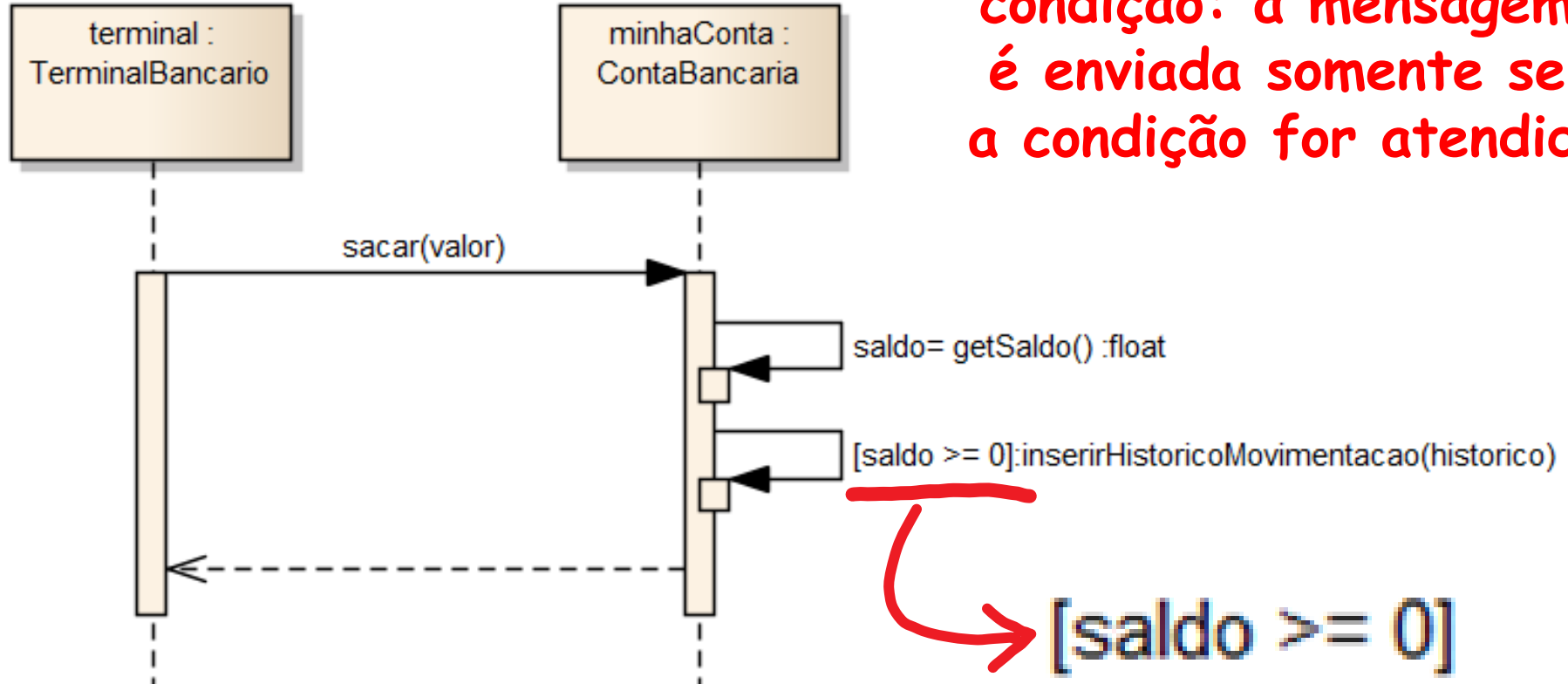


**Mostrando a
implementação da
auto-mensagem 1**

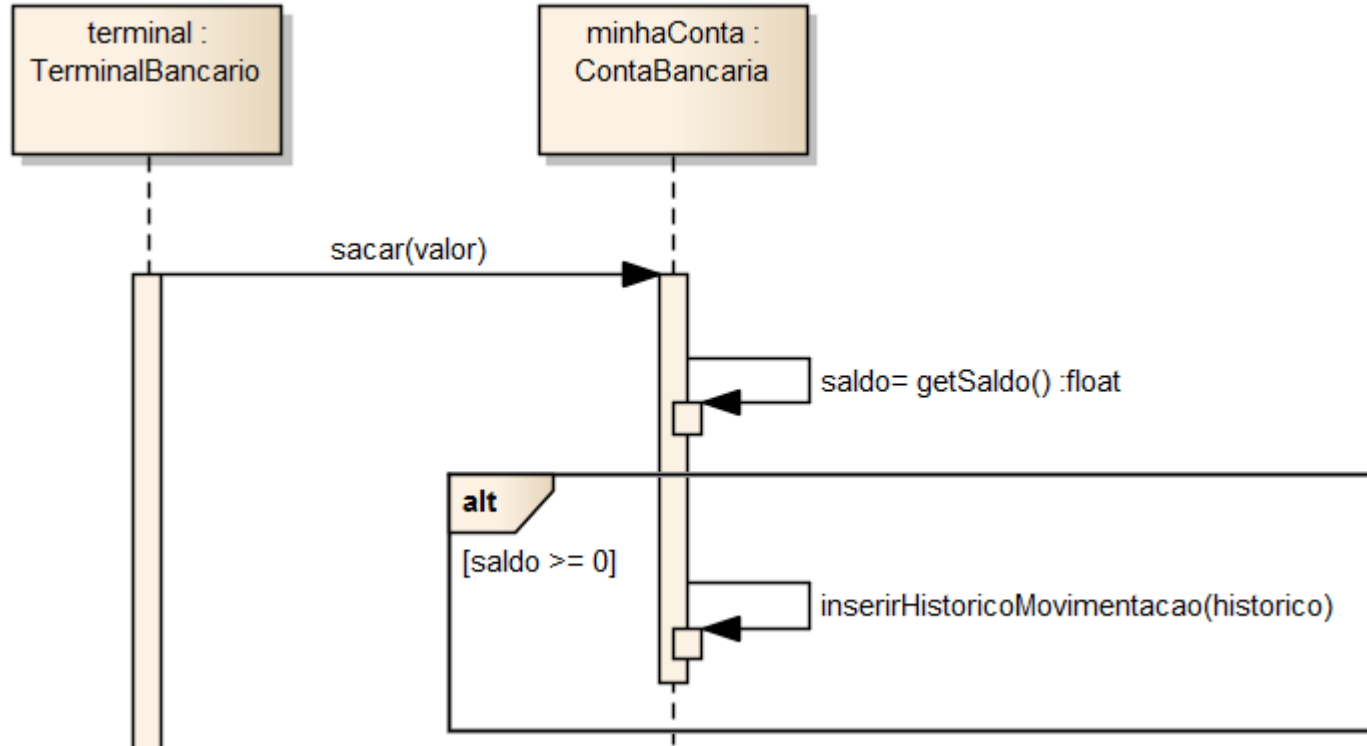
**Múltiplas camadas
de ativação**

Guarda (Guard)

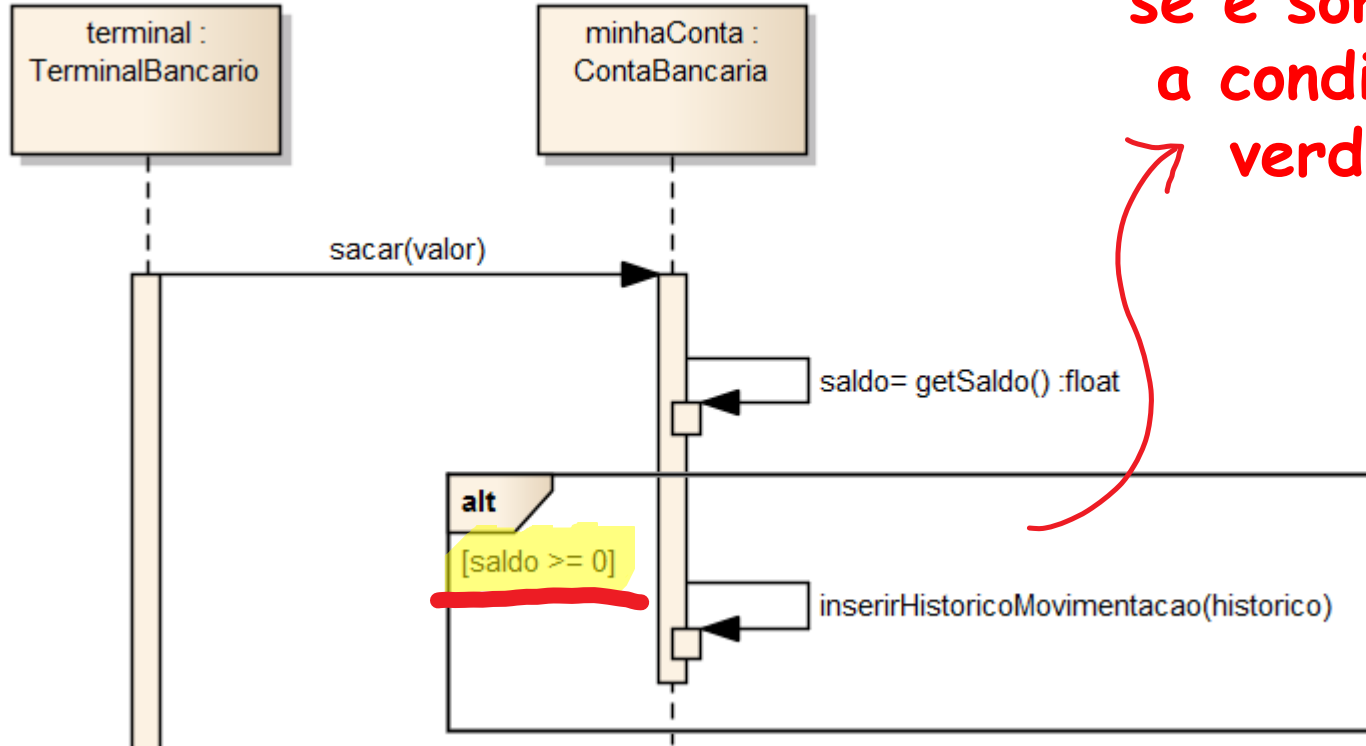
Representa uma condição: a mensagem é enviada somente se a condição for atendida



Fragmento "alt" (alternativas)

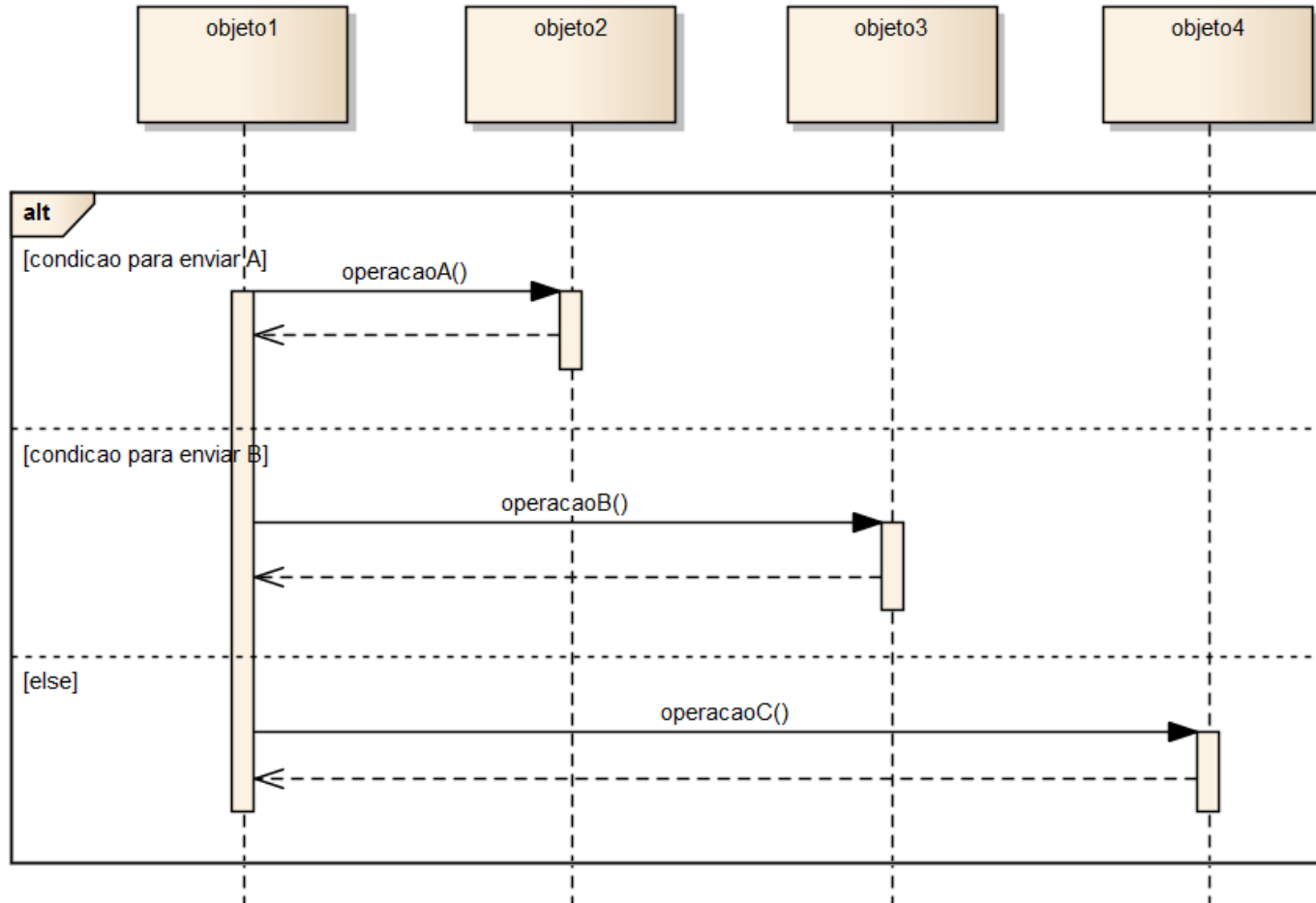


Fragmento "alt" (alternativas)

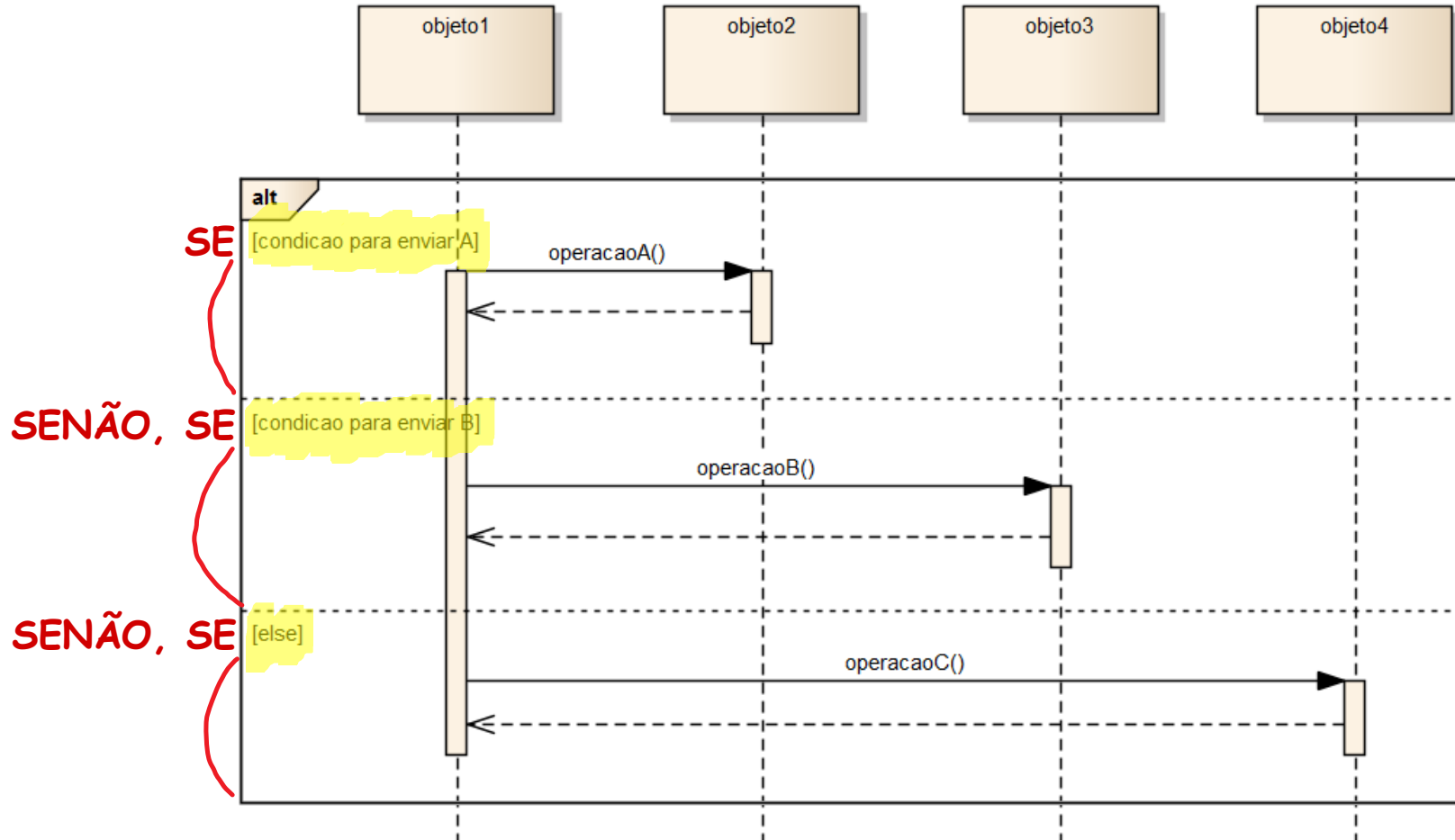


Área executada
se e somente se
a condição for
verdadeira

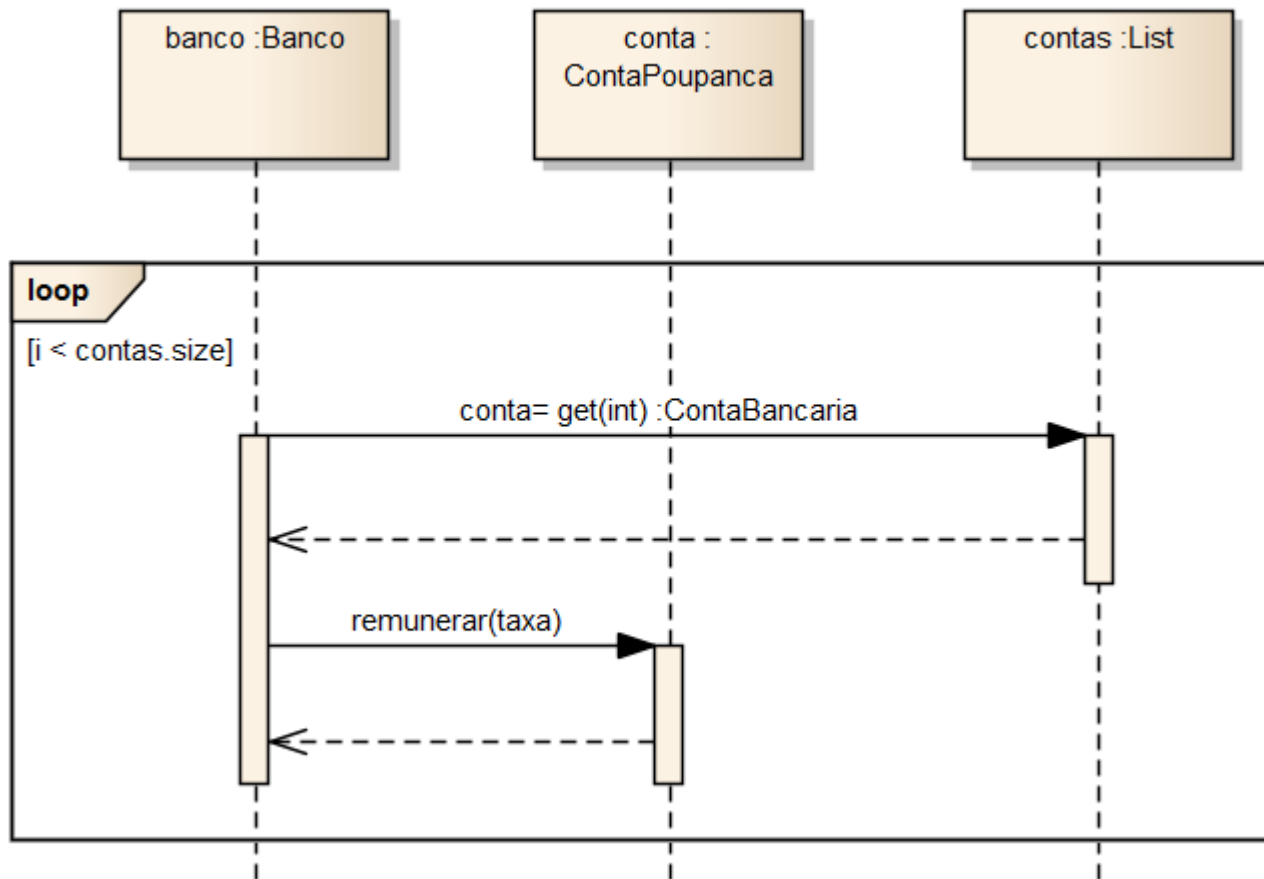
Fragmento "alt" (alternativas)



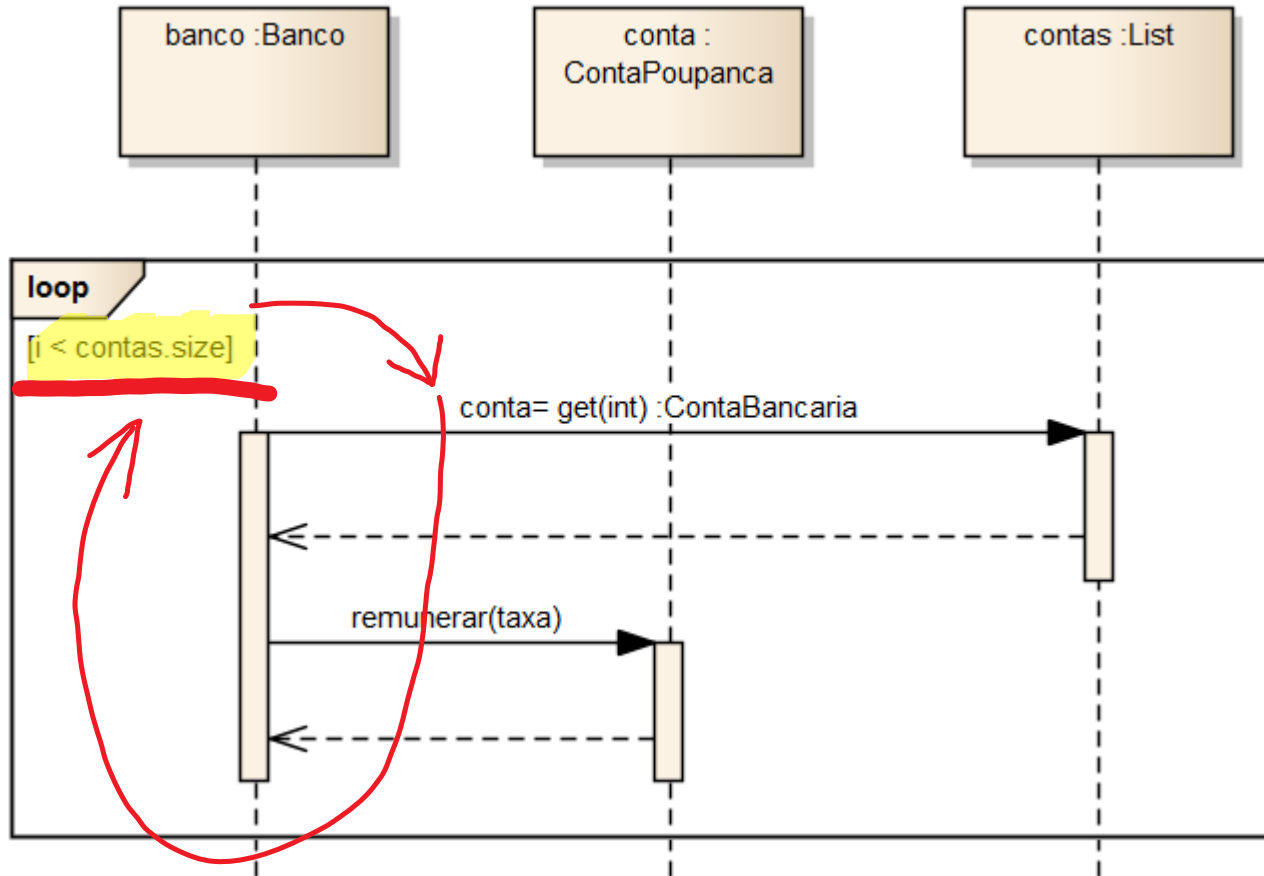
Fragmento "alt" (alternativas)



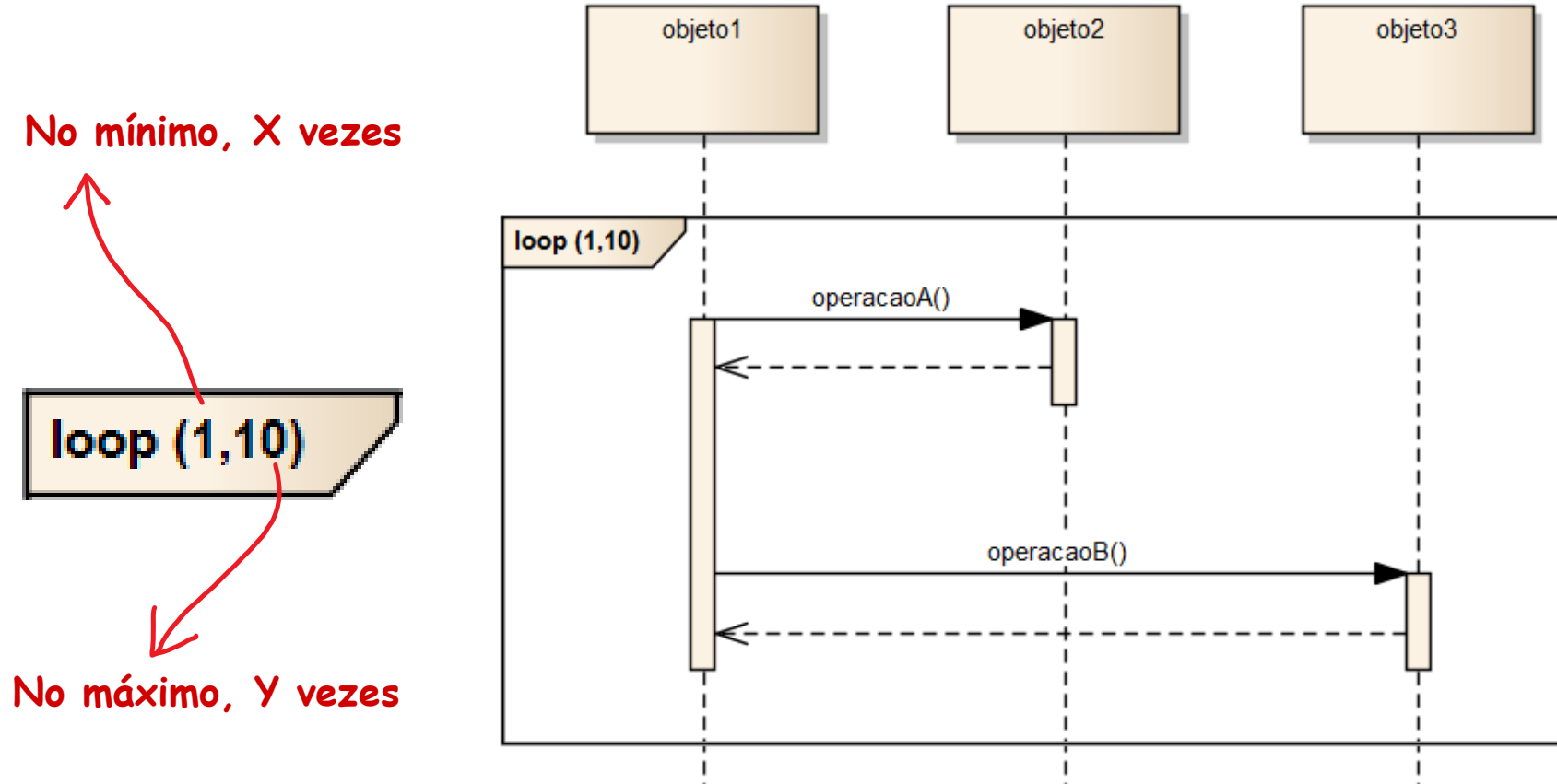
Fragmento "loop" (laços)



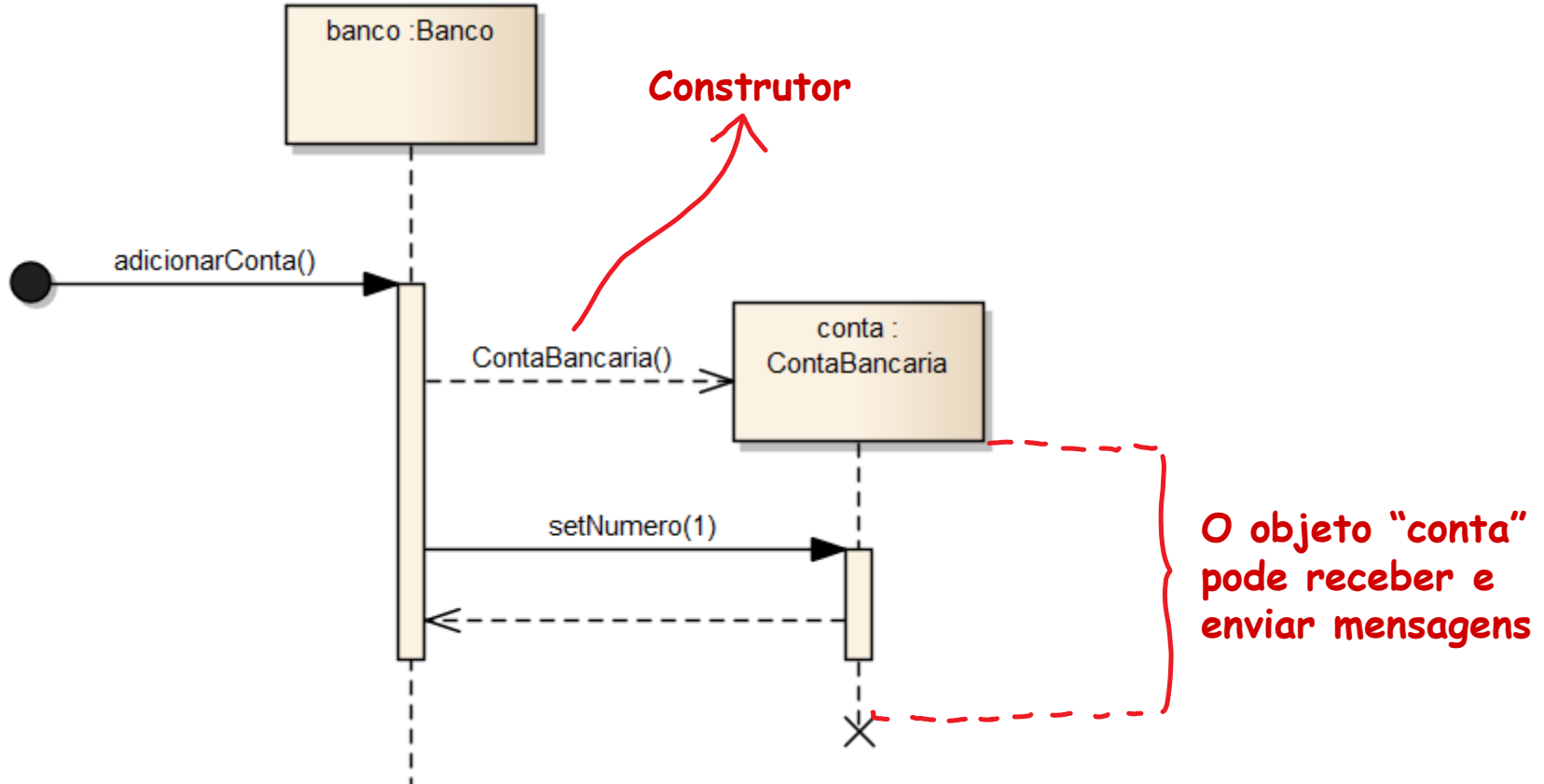
Fragmento "loop" (laços)



Fragmento "loop" (laços)



Criação de objetos



Referências

50

- ❑ Grady Booch, James Rumbaugh, and Ivar Jacobson. **The Unified Modeling Language User Guide**. 2nd ed. Addison-Wesley, 2005.
- ❑ Ricardo Pereira e Silva. **UML 2 em Modelagem Orientada a Objetos**. Visual Books, 2007.
- ❑ OMG (Object Management Group), **OMG Unified Modeling Language v2.5**, 2012.
 - ❑ <http://www.omg.org/spec/UML/2.5/Beta1/PDF/>



Clássico!!
Excelente
referência!



Padrão
UML!

Referências



- Martin Fowler. **UML Distilled: A Brief Guide to the Standard Object Modeling Language**. 3th ed. Addison-Wesley, 2003.

PROGRAMAÇÃO ORIENTADA A OBJETOS

Diagrama de Sequência - UML

