

System Analysis and Design

L15. Dynamic Object Modeling
(UML Interaction Diagrams)

Topics

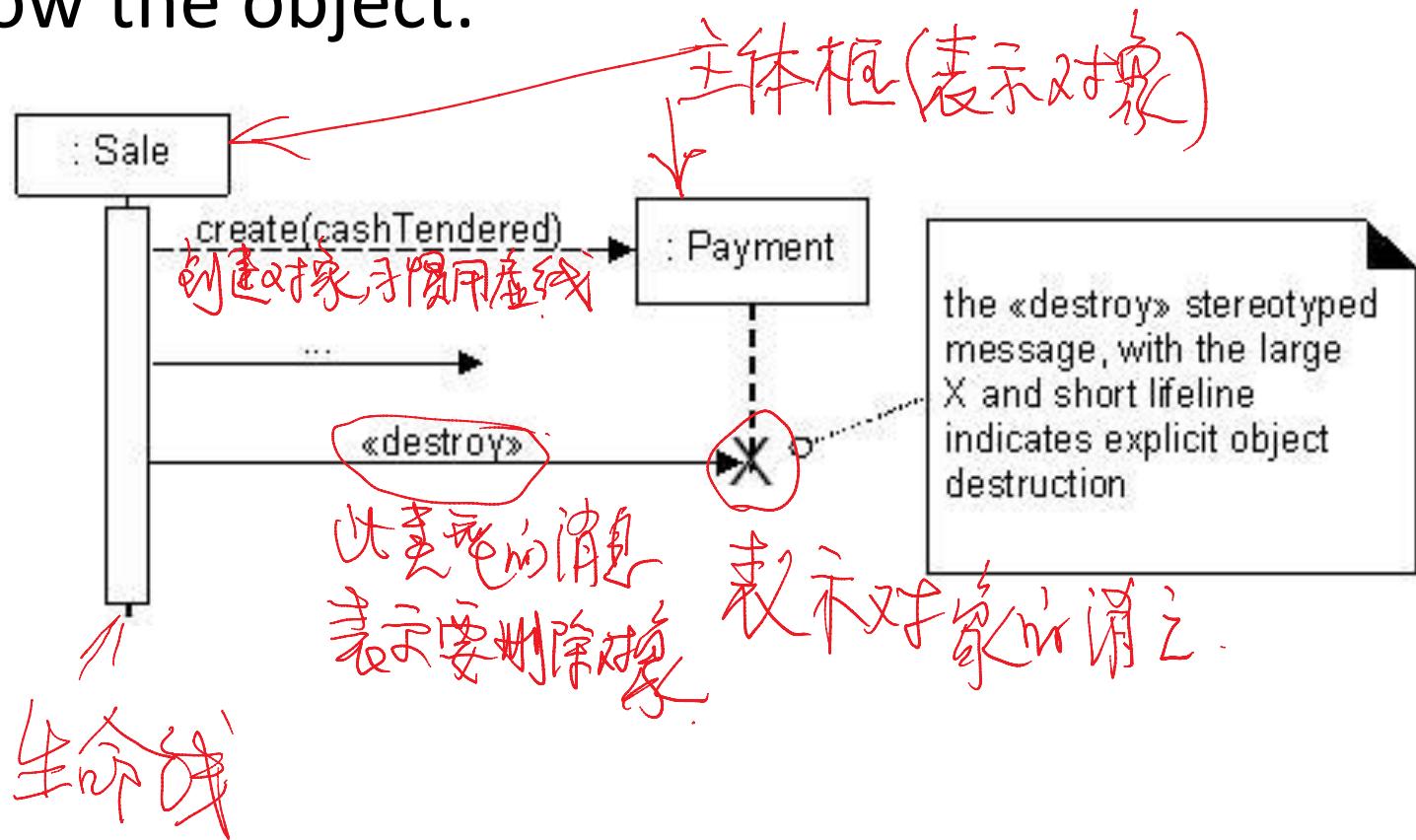
- UML Interaction Diagrams
- Common UML Interaction Diagram Notation
- Basic Sequence Diagram Notation
- Basic Communication Diagram Notation

BASIC SEQUENCE DIAGRAM NOTATION

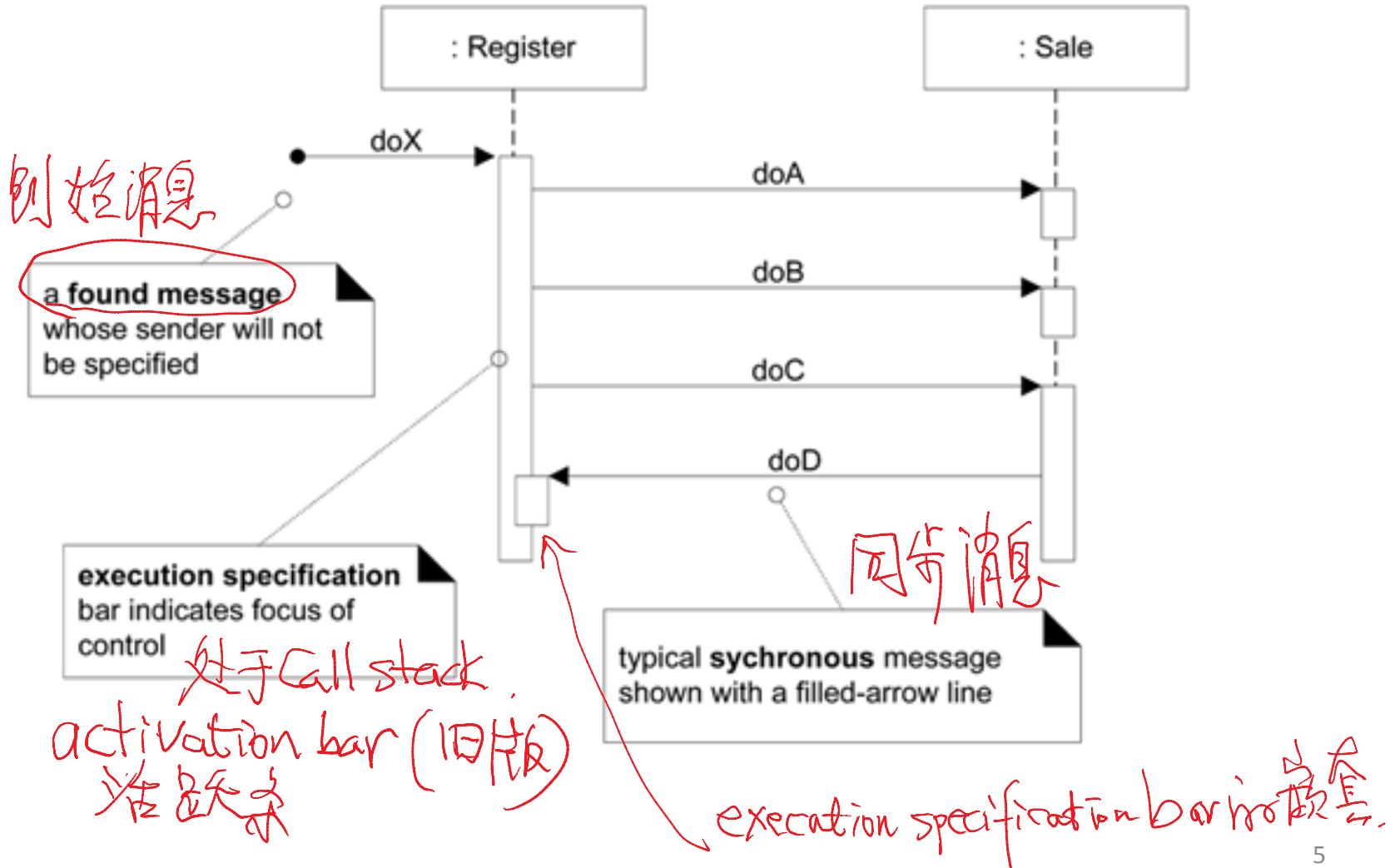
顺序图的标记符号 .

Lifeline Boxes

- Lifeline is a dashed (or solid, in UML 2) line below the object.



Messages



Focus of Control

- The bar on the left is an Execution Specification bar. Indicates focus of control
being in its call stack!

两种方法表示返回消息
Reply

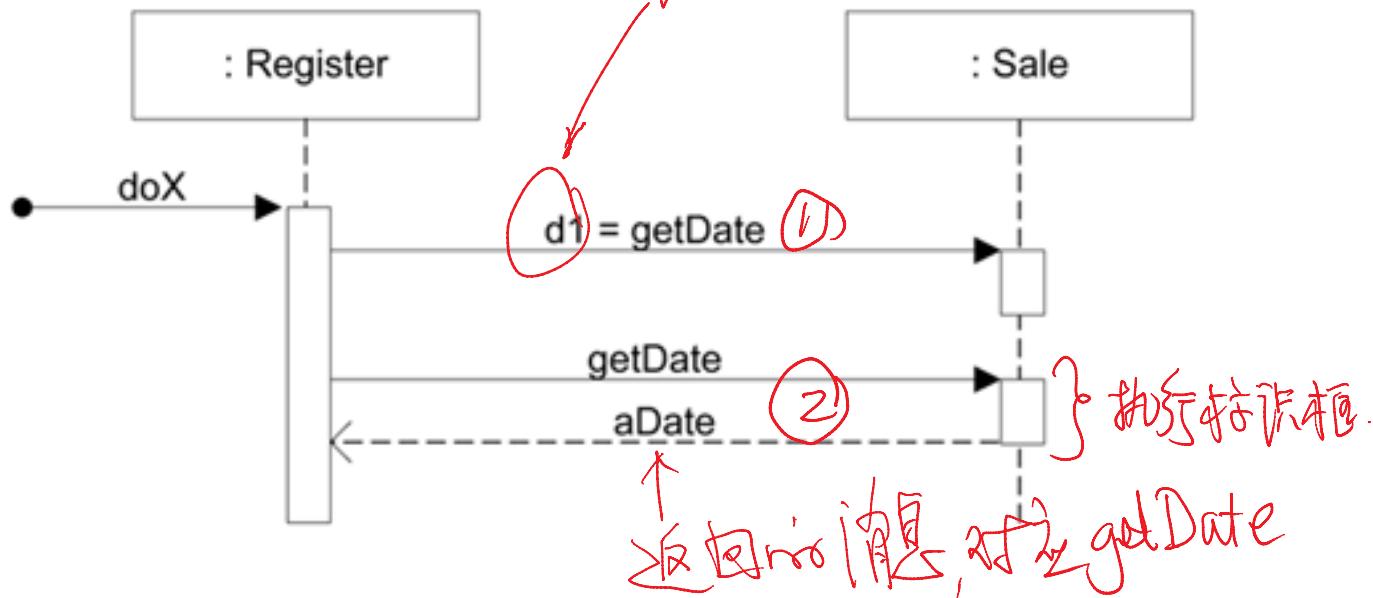
Reply or Returns

- Use the message syntax

① returnVar=Message(parameter)

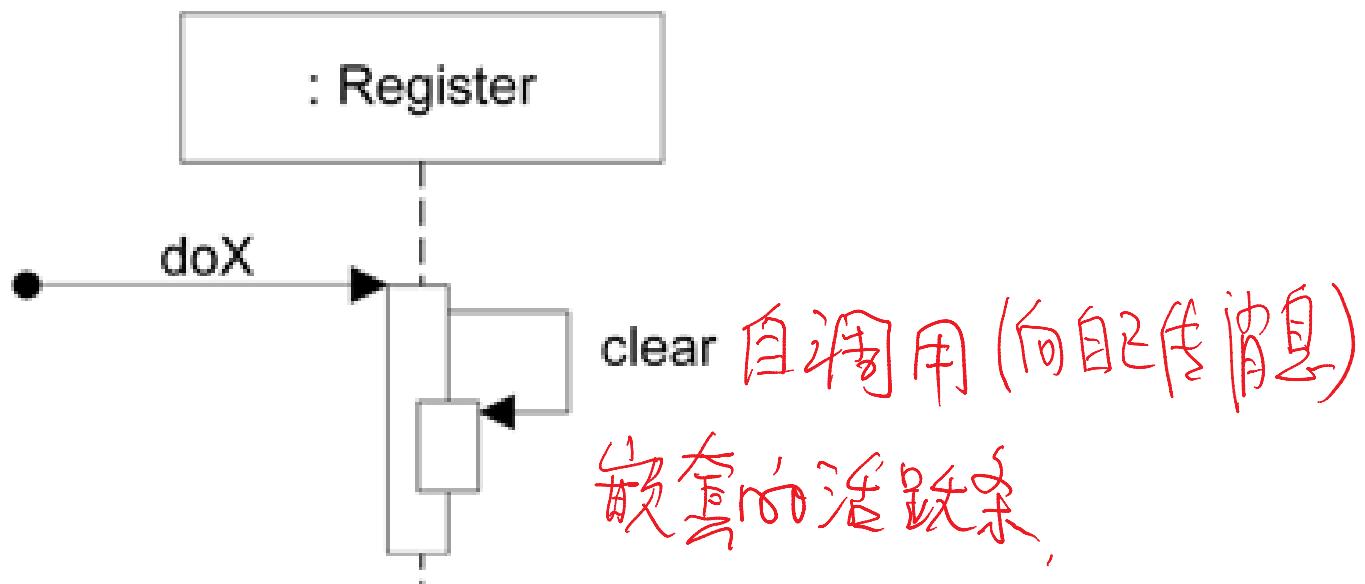
 - Using a reply or return message at the end of an activation bar

② 執行回傳(用法)



Messages to Self

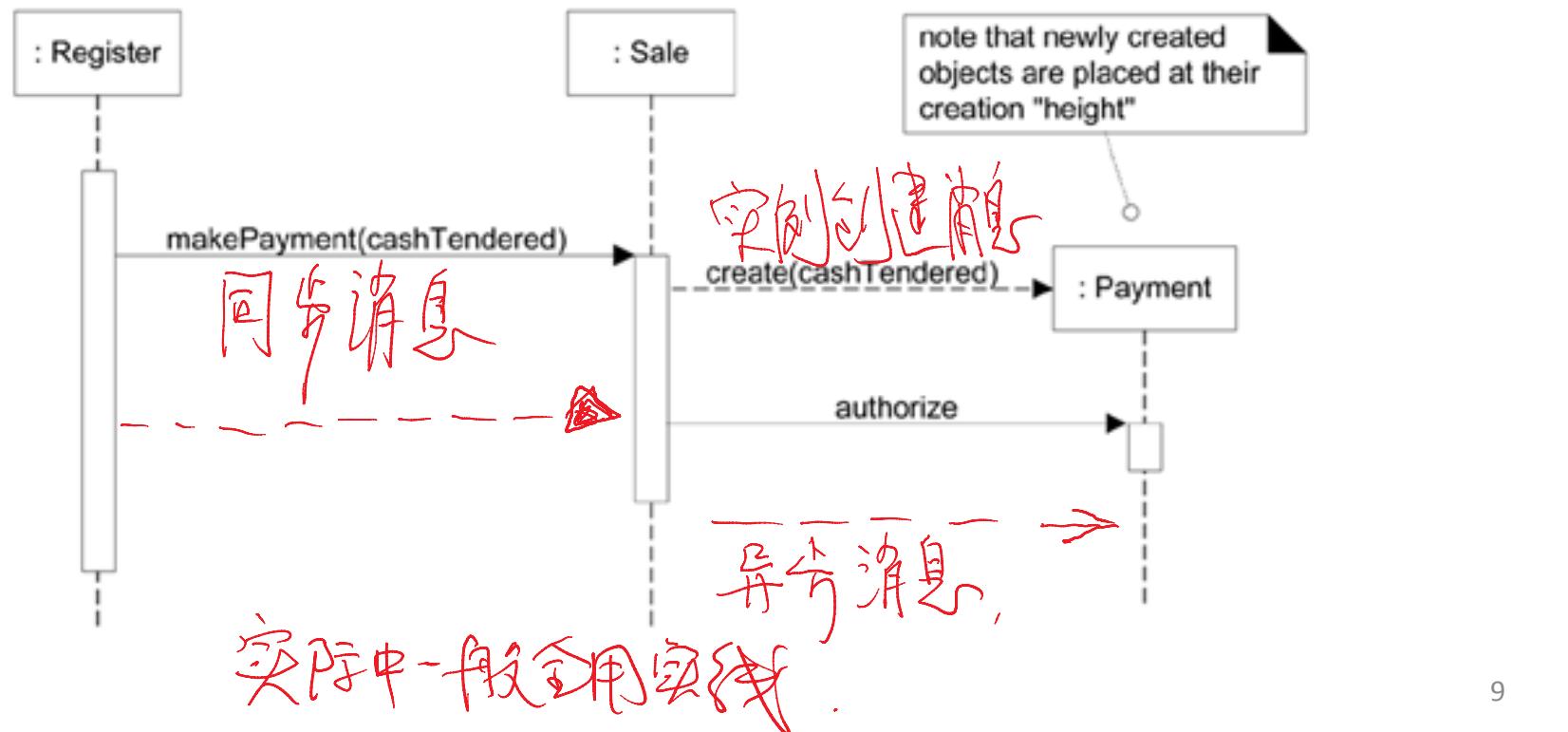
- Message sent to itself with a nested activation bar



实例创建的表示.

Creation of Instances

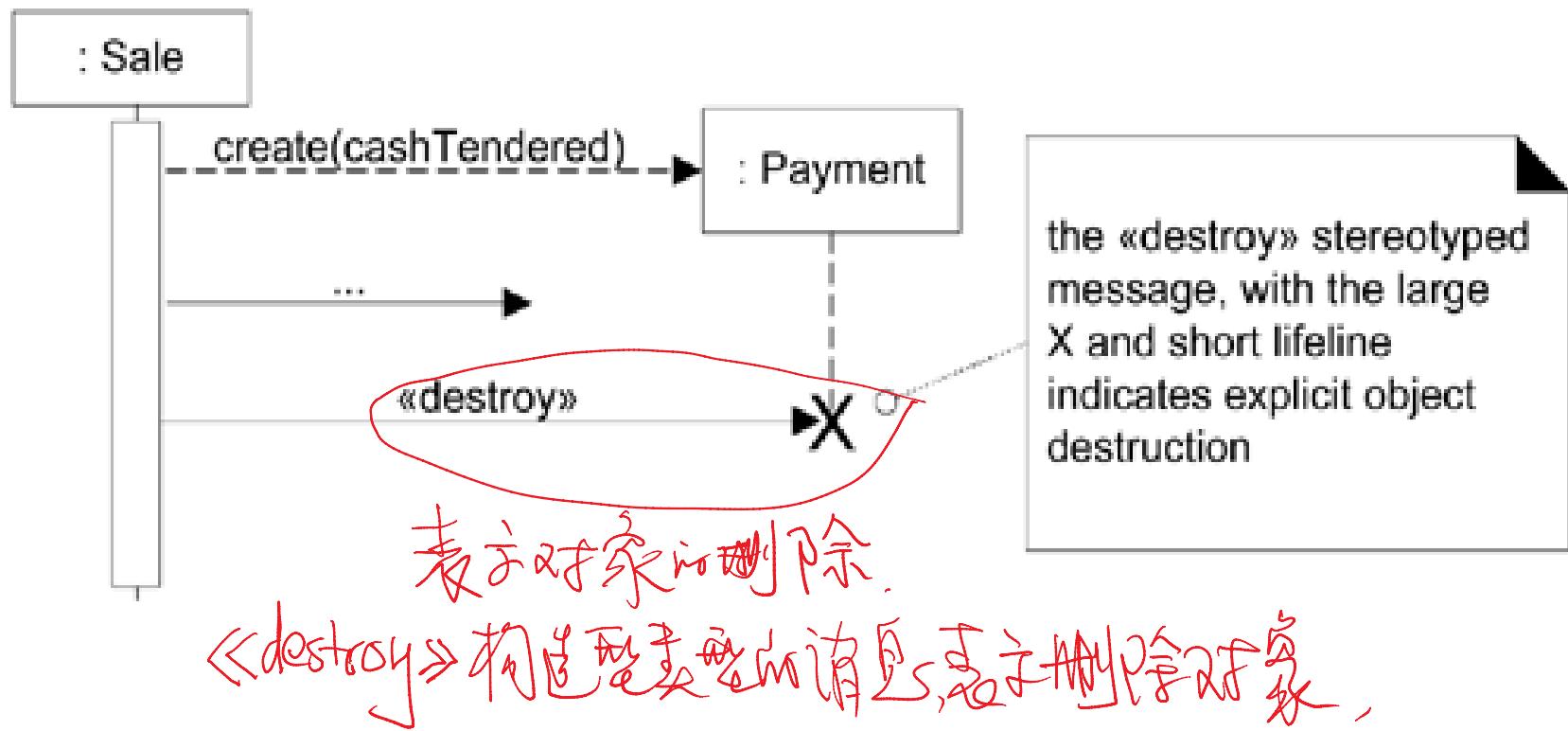
- Dashed line with filled arrow for synchronous message, open if async. 同步消息与异步消息
- Name “create” is not required; it’s a UML idiom 实例创造



对象被删除

Lifelines and Object Destruction

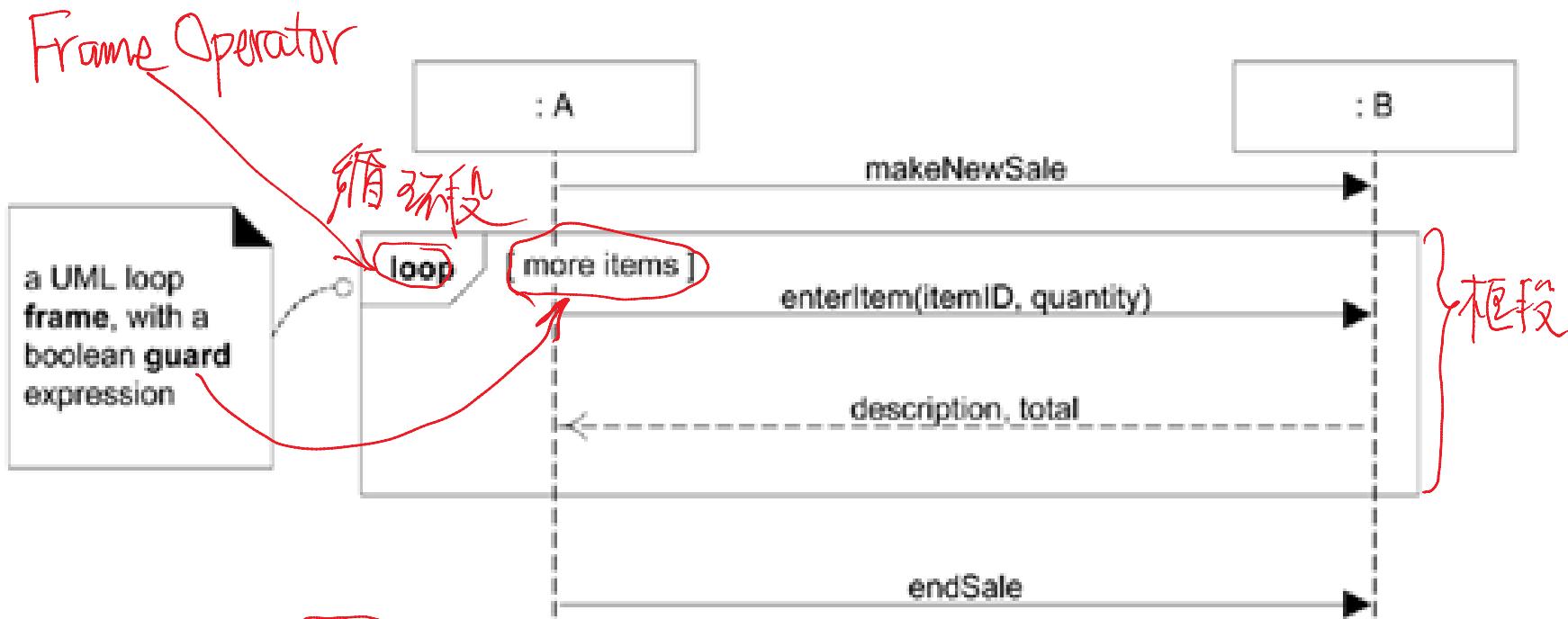
- Show explicit destruction of an object, such as when a database connection is closed



图框(段)

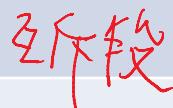
Diagram Frames

- These support conditionals and looping

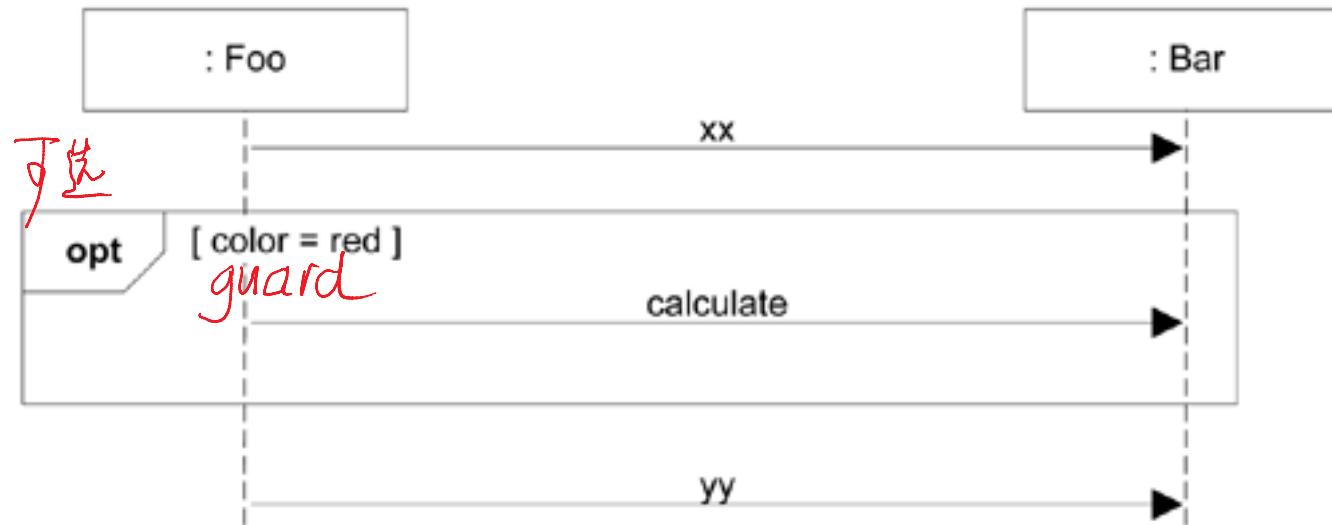


图框内的消息传递一直停留在guard失败

Frame Operators

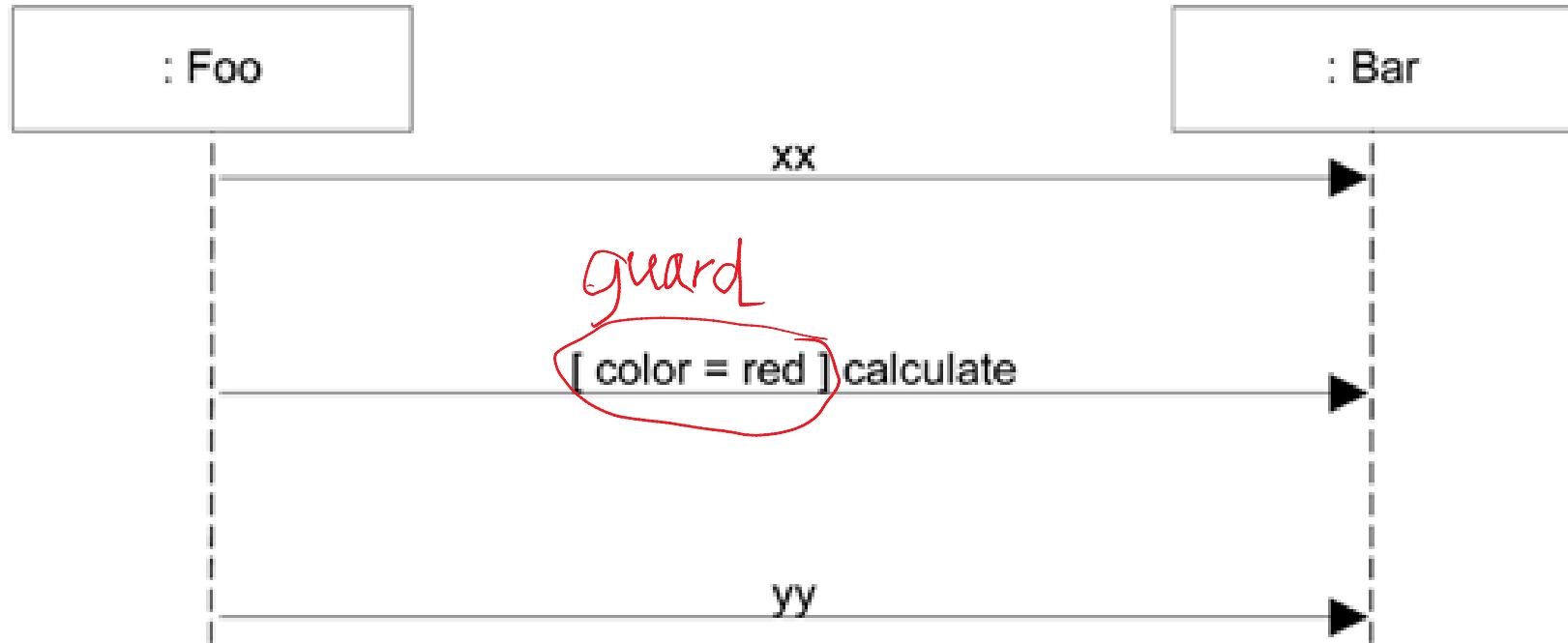
Frame Operator	Meaning
Alt 	Alternative fragment for mutual exclusion conditional logic expressed in the guards
Loop 	Loop fragment while guard is true. Can also write loop(n) to indicate looping n times. Can be enhanced to define for loop
Opt 	Optional fragment that executes if guard is true
Par 	Parallel fragments that execute in parallel
Region 	Critical region within which only one thread can run A “guard” is a Boolean test in brackets over the line to which it belongs

Conditional Messages 条件消息



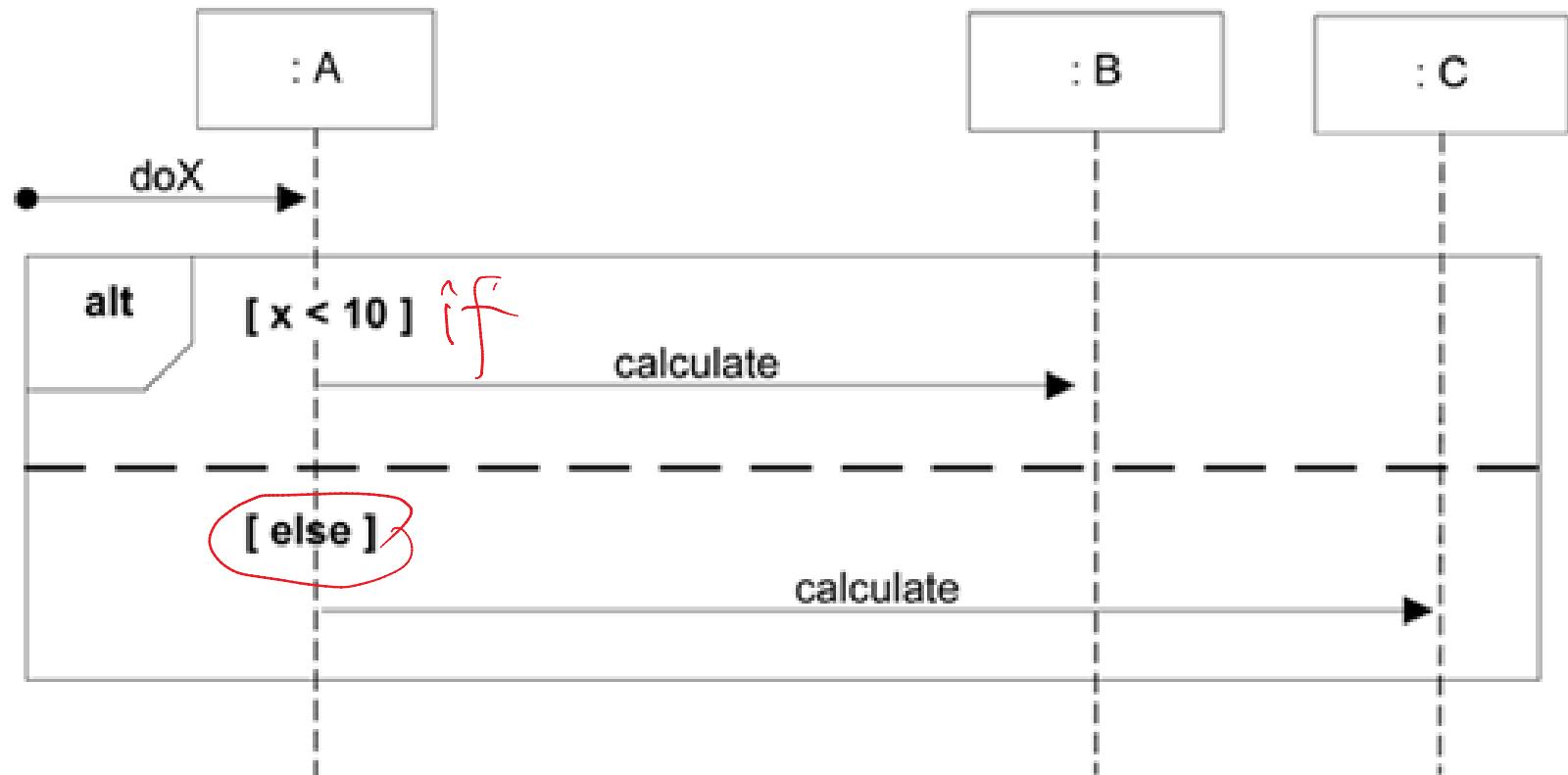
若 guard 条件为真，则传递 calculate

UML 1.x Conditionals



旧版的写法，表示条件性的选择。

Conditional If...Else



条件分支用虚线隔开两个分支

Code to Iterate Over Collection

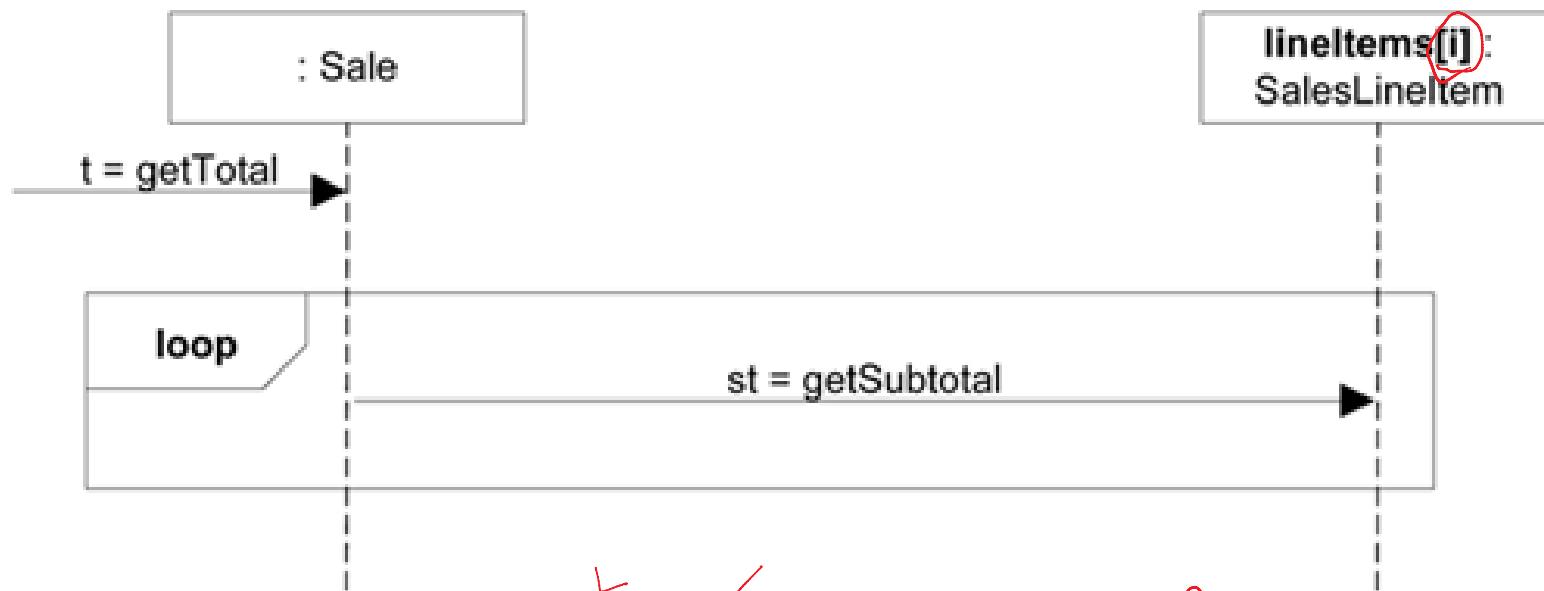
```
public class Sale {  
    private List<SalesLineItem> lineItems = new ArrayList<SalesLineItem>();  
    public Money getTotal() {  
        Money total = new Money();  
        Money subtotal = null;  
        for ( SalesLineItem lineItem : lineItems ) {  
            subtotal = lineItem.getSubtotal();  
            total.add( subtotal );  
        }  
        return total;  
    }  
    // ...  
}
```

表示组织结构为 Collection .

iterator

Implicit Iteration

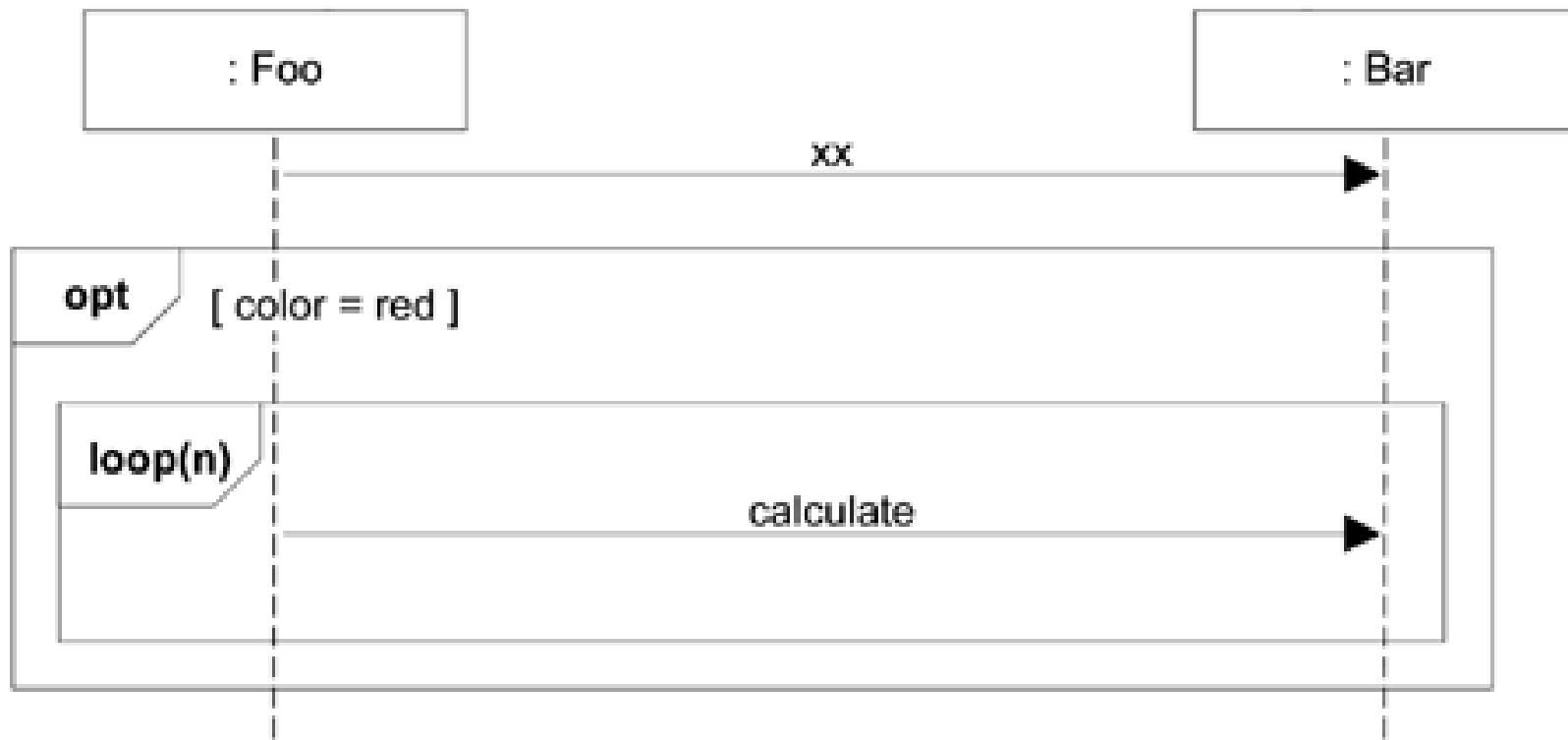
隐喻的迭代



表示循环调用直到再无新元素。
(省略了 guard)

嵌套的图框技.

Nesting Frames

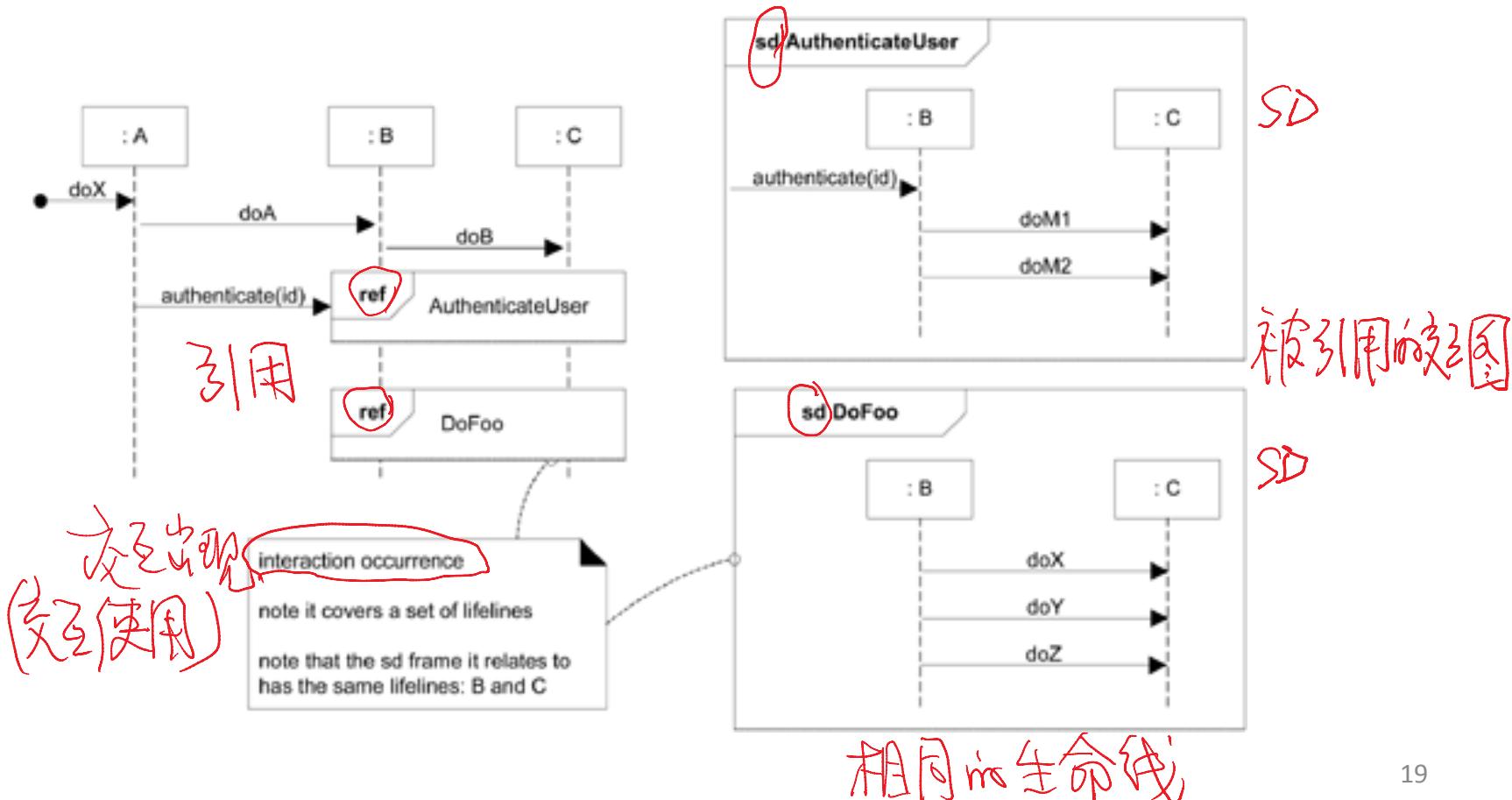


在子段中包括另一个子段.

相关的交互图之间的关联

Relating Interaction Diagrams

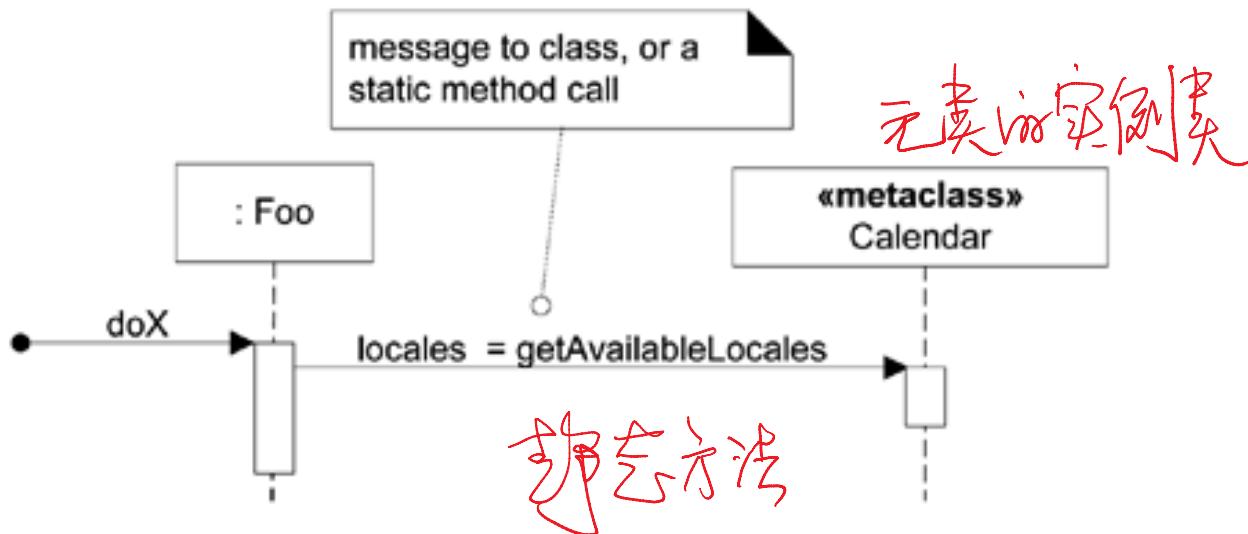
- Interaction Occurrence (or Use) is a reference to an interaction within another interaction.



调用静态方法 (Class wide methods)

Invoking Static Methods

- Metaclasses or static classes are instances of themselves.
- Calling a static method uses the name of the class, not the name of the instance.



```
// static method call on class Calendar  
Locale[] locales = Calendar.getAvailableLocales();
```

使用表达式

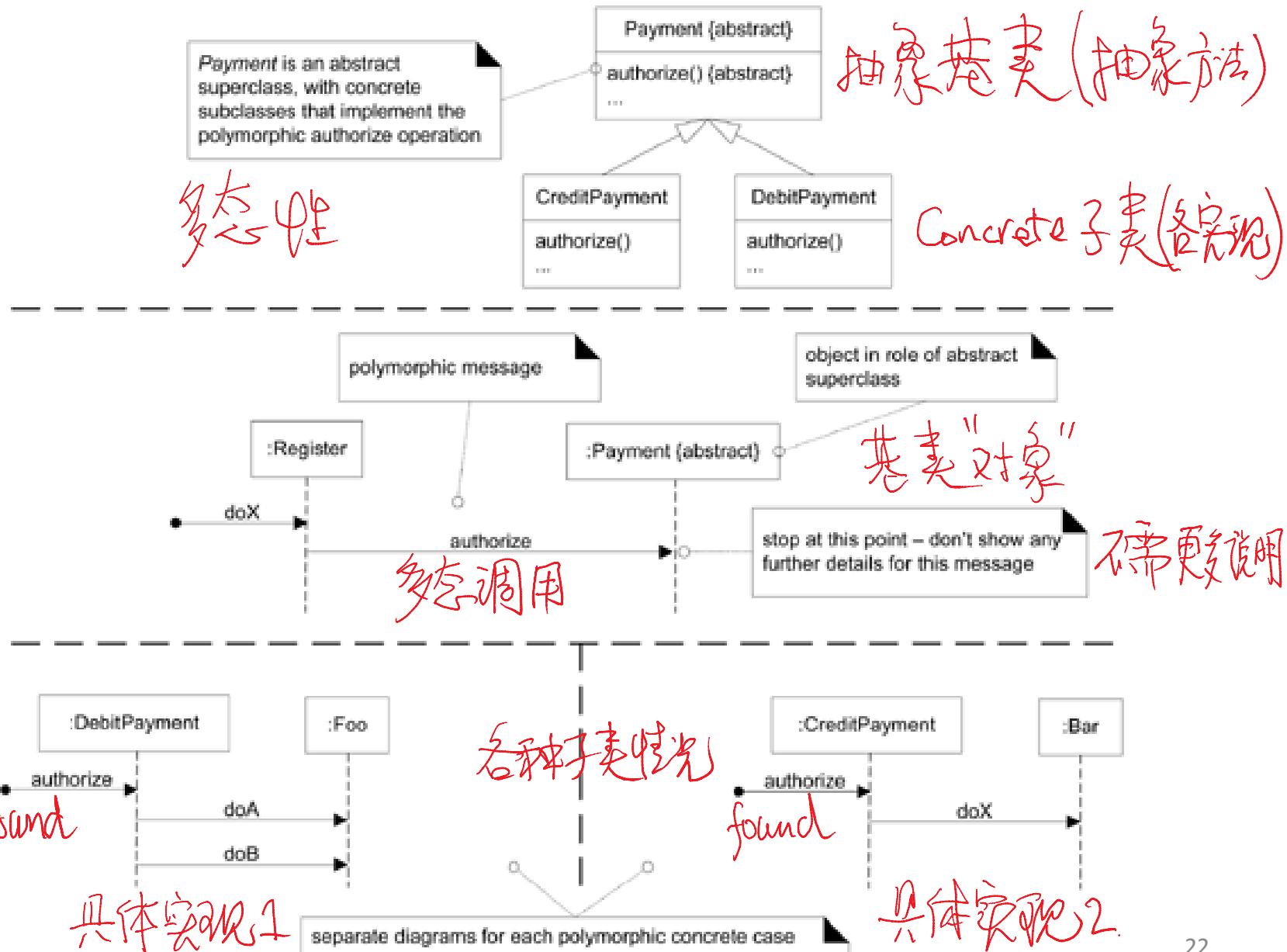
多态消息 (回想多态性)

Polymorphic Messages

如何在交互图中表示多态调用.

- Multiple sequence diagrams: 用多个顺序图表示
 - One to the superclass,
 - One to each case, each starting with a found polymorphic message

Polymorphic Messages



异步和同步调用

Async and Sync Calls

- Async (non-blocking) calls are a stick (unfilled) arrow, sync are a filled arrow
- This can be used for multi-threading

```
Public class ClockStarter {  
    Public void startClock {  
        Thread t = new Thread(new Clock) ;  
        t.start();  
        System.runFinalization(); //  
    }  
}
```

Async Calls and Active Objects

a stick arrow in UML implies an asynchronous call

a filled arrow is the more common synchronous call

In Java, for example, an asynchronous call may occur as follows:

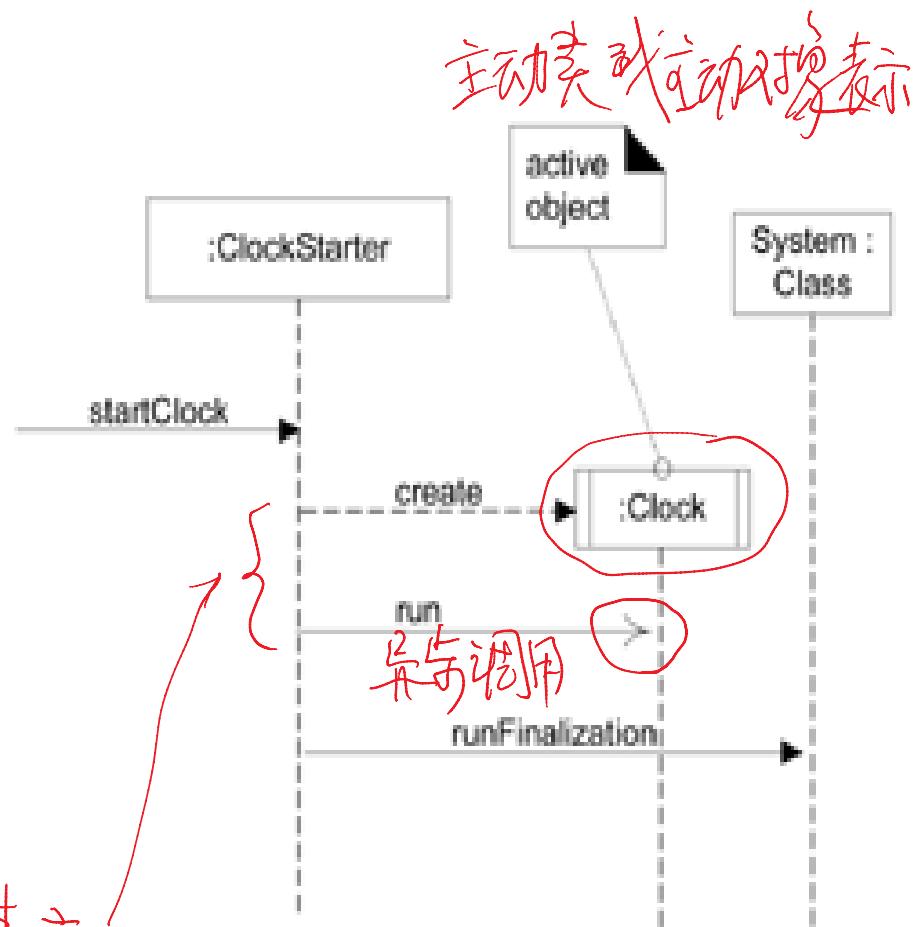
```
// Clock implements the Runnable interface  
Thread t = new Thread( new Clock() );  
t.start();
```

异步

the asynchronous start call always invokes the run method on the Runnable (Clock) object

to simplify the UML diagram, the Thread object and the start message may be avoided (they are standard "overhead"); instead, the essential detail of the Clock creation and the run message imply the asynchronous call

省略了 Thread 对象, 使用了简写方法

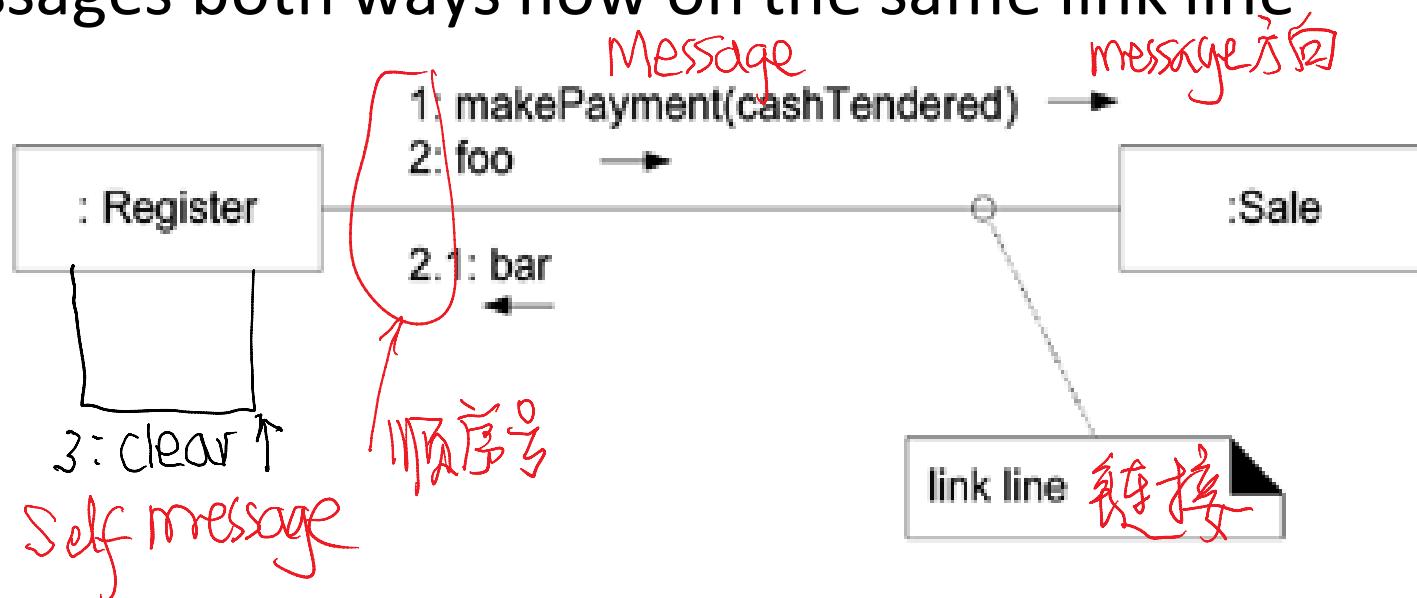


Active objects: each instance runs on and controls its own thread of execution

Basic Communication Diagram Notation

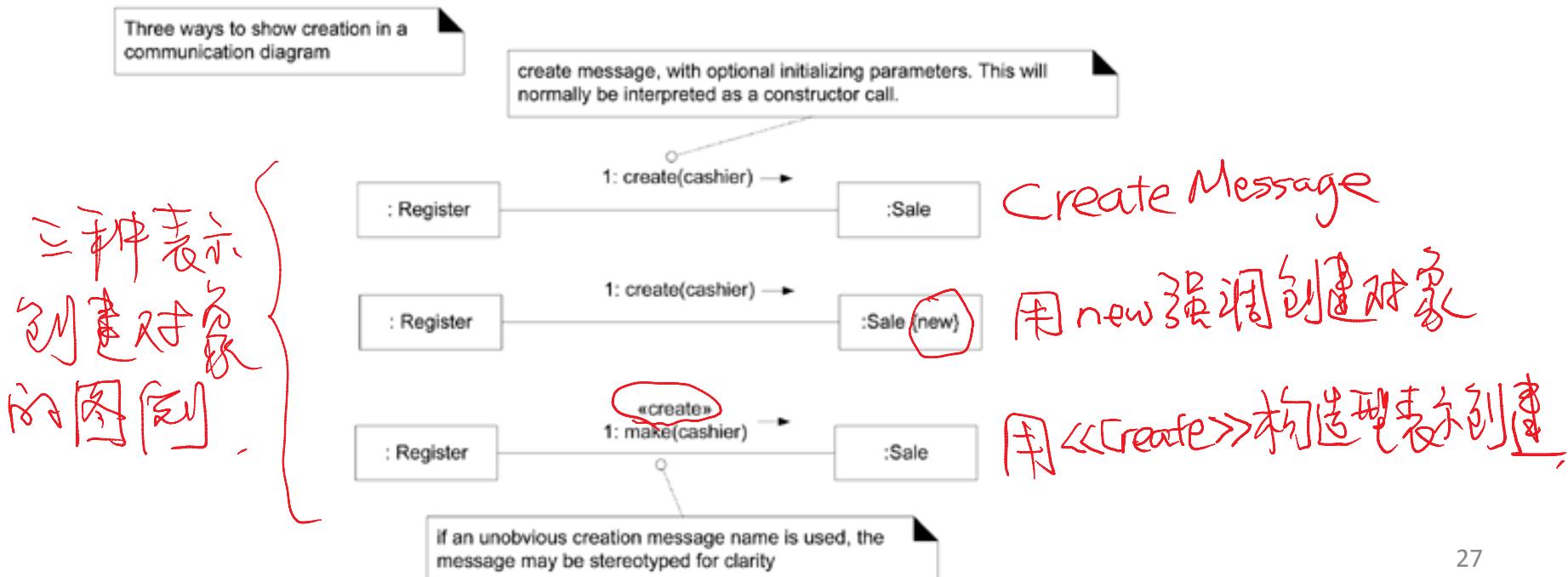
Communication Diagram: Links and Messages

- Link is a Connection path between objects.
 - Navigation and visibility between objects
- Formally it is an Instance of an association.
- Multiple messages flow on the same link line
- Messages both ways flow on the same link line



Creation of Instances

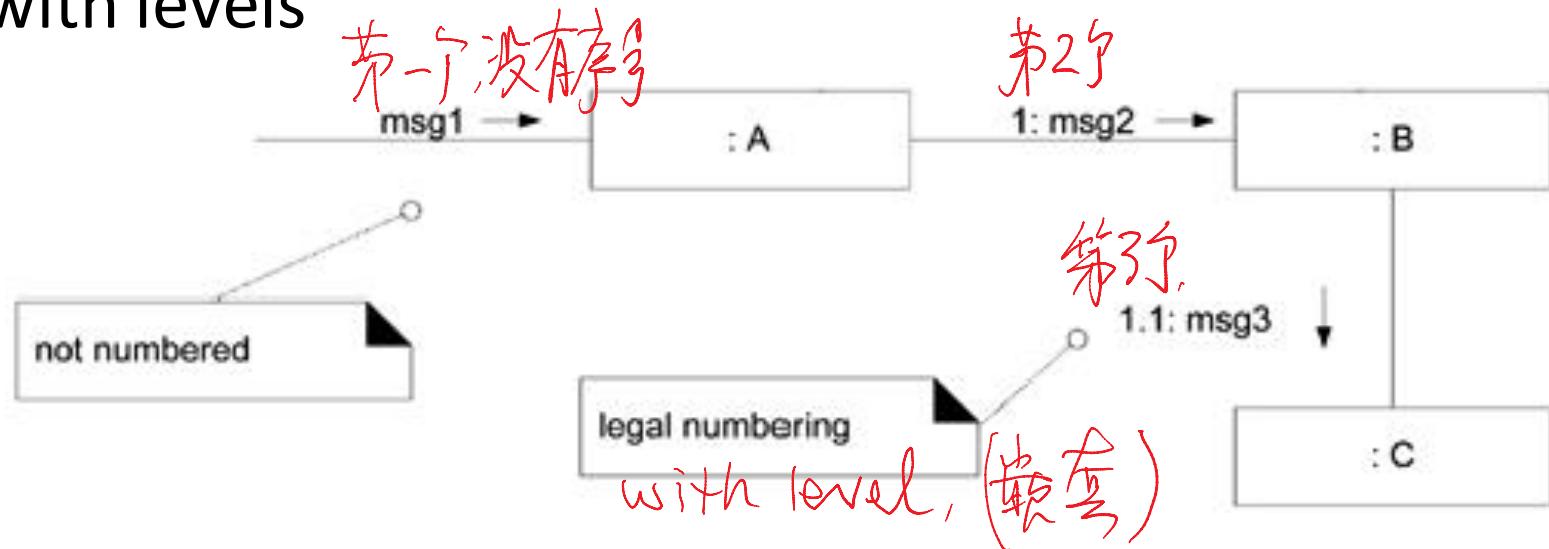
- Message named **Create**.
- If some other name is used, use the UML stereotype **<<Create>>** to indicate.
- UML tagged value **{new}** to indicate creation
- Note that Message can include parameters.



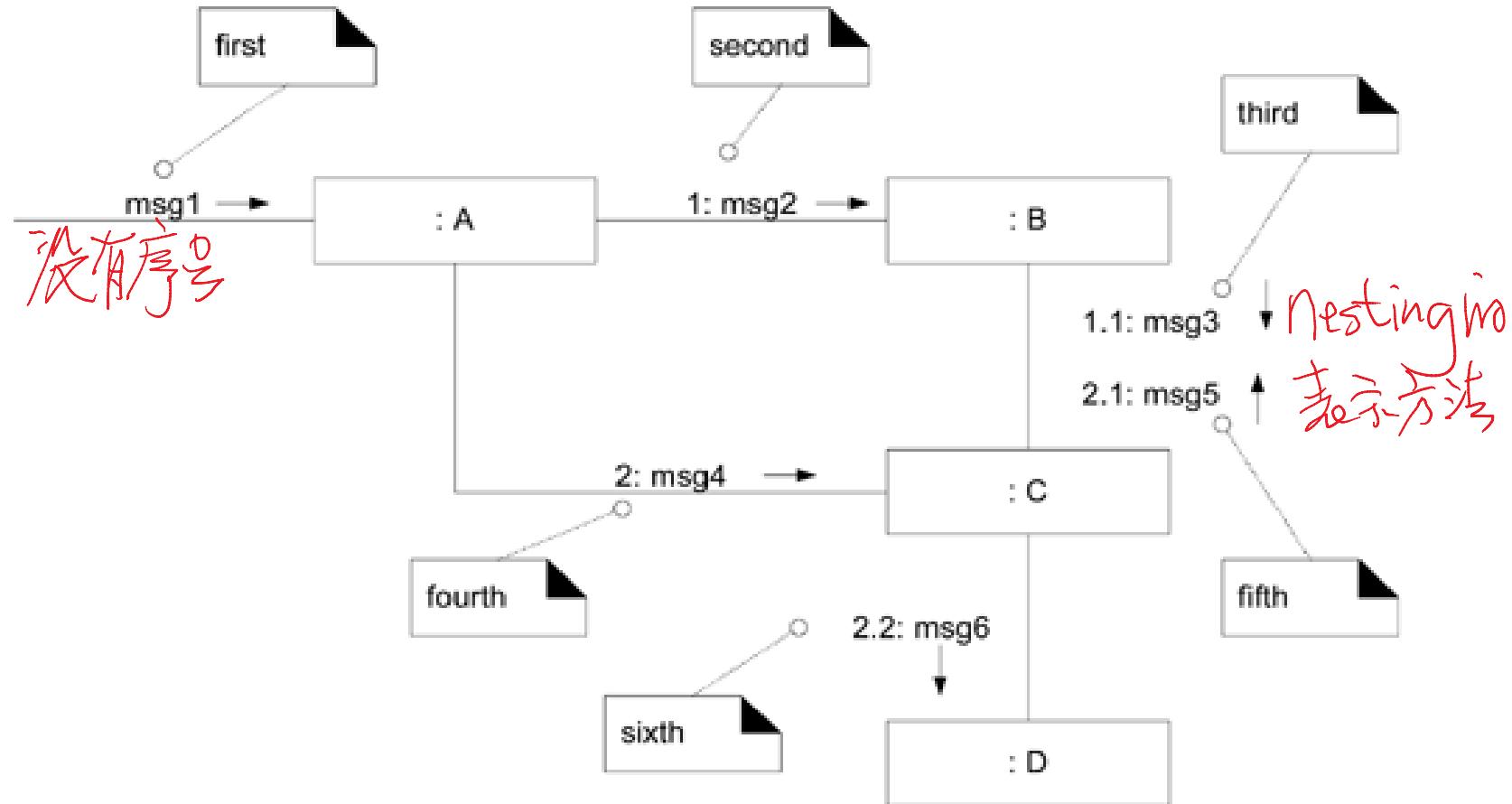
消息序号

Message Number Sequencing

- If ordering is important, indicate this with **sequence numbers**.
消息顺序号
- The first message is not numbered (usually) 第一个一般不赋顺序号
- Order and nesting of subsequent numbers is shown with levels



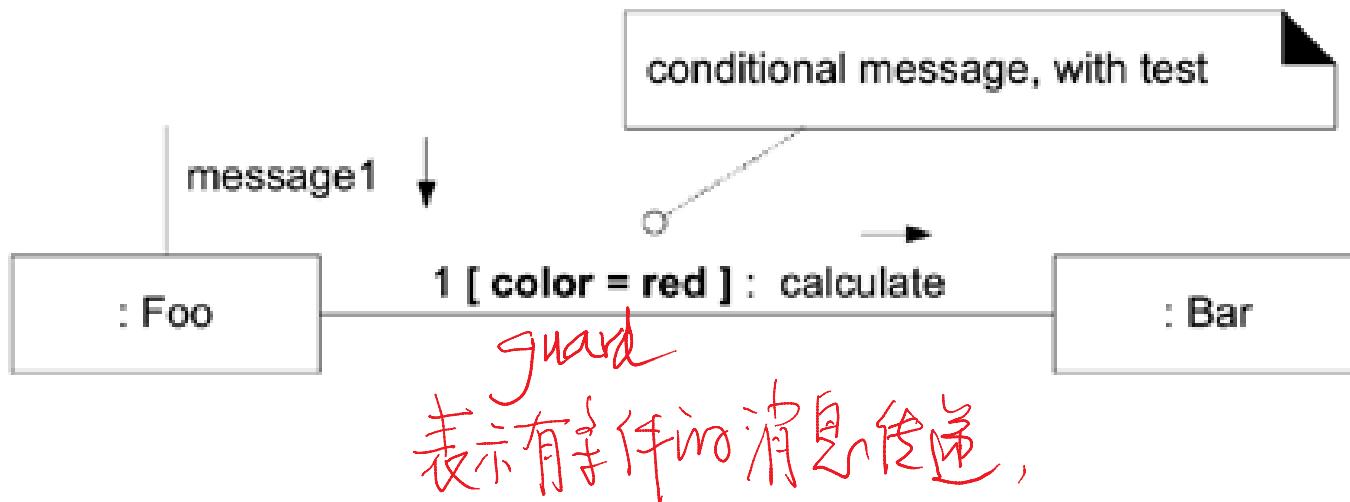
Complex sequence Sequencing



You denote nesting by prepending the incoming message number to the outgoing message number.

Conditional Messages

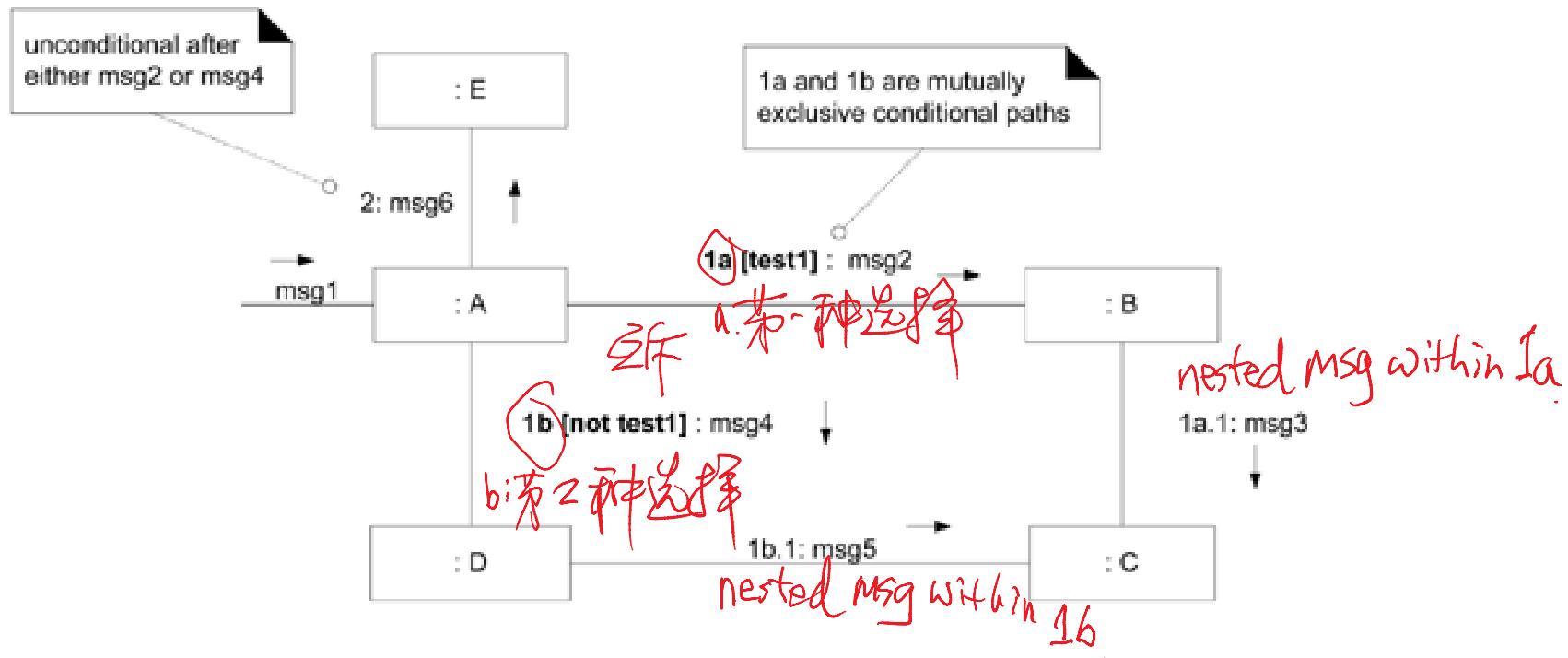
- Show this with the condition in brackets



互斥条件的表示

Mutually Exclusive Conditionals

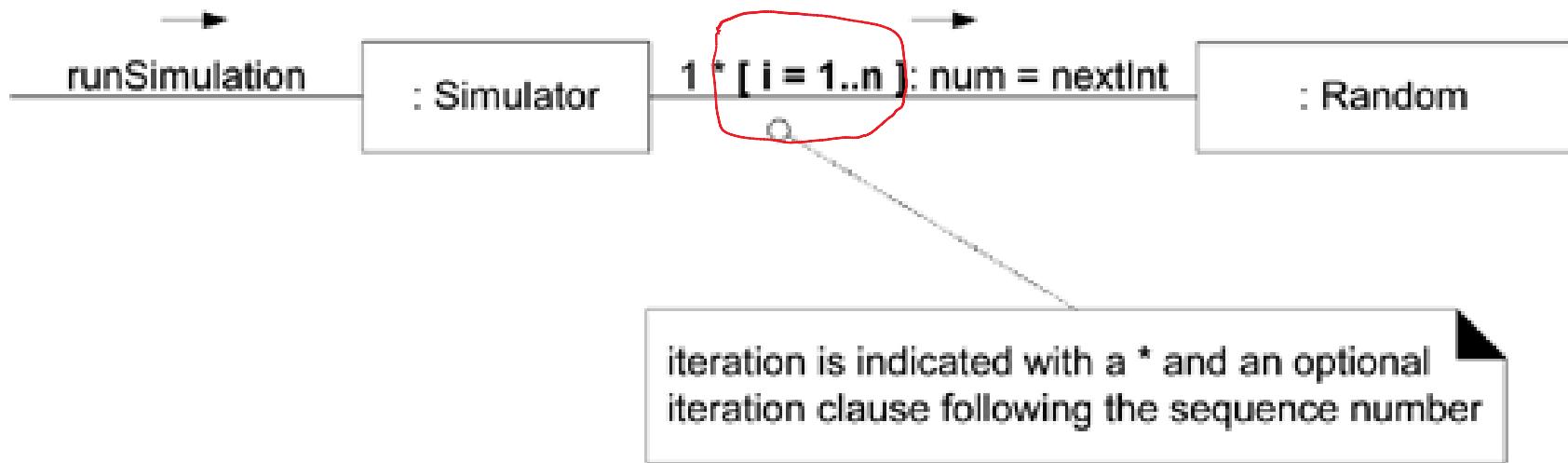
- Equivalent of If...else. Modify the sequence expression with a letter:
expression with a letter: 序列附加了字母表示一种选择.



Iteration or looping

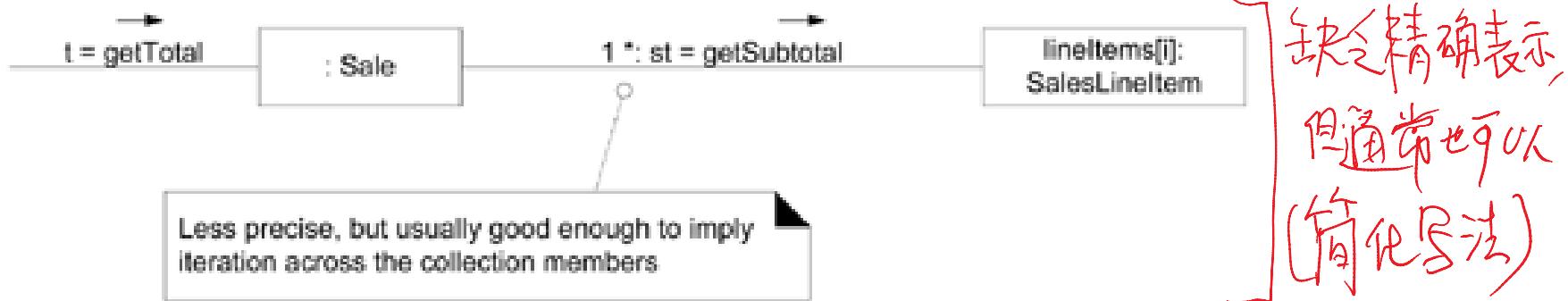
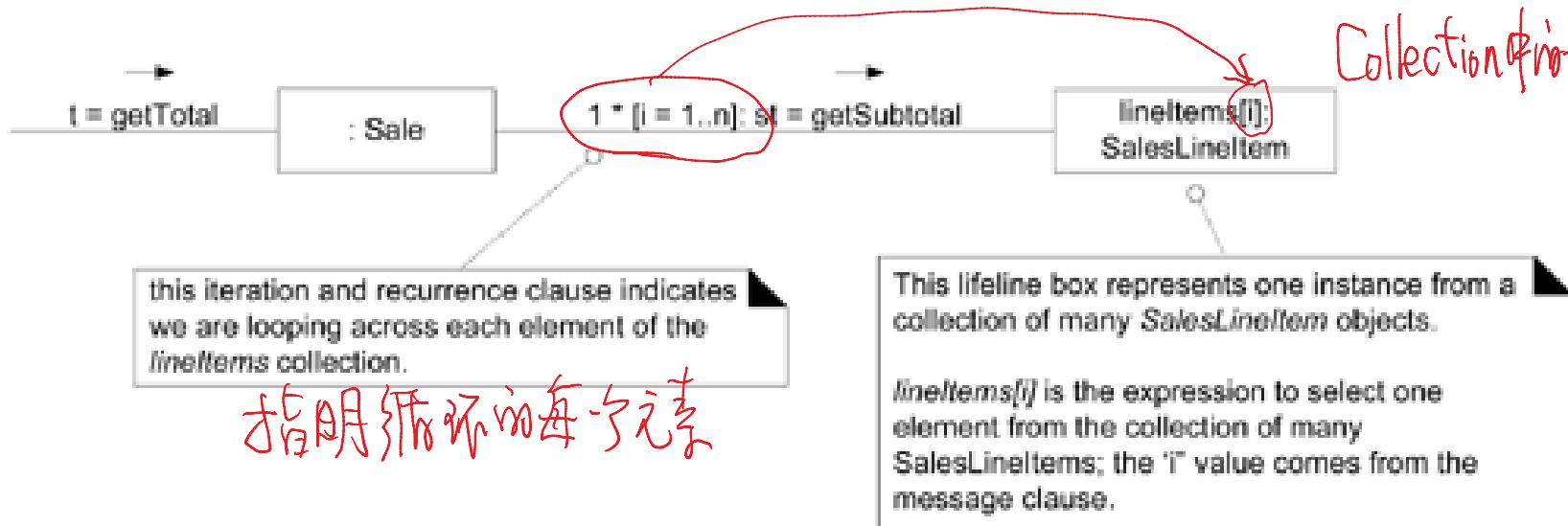
循環表示

表示迭代(循环), 循环子句可选.



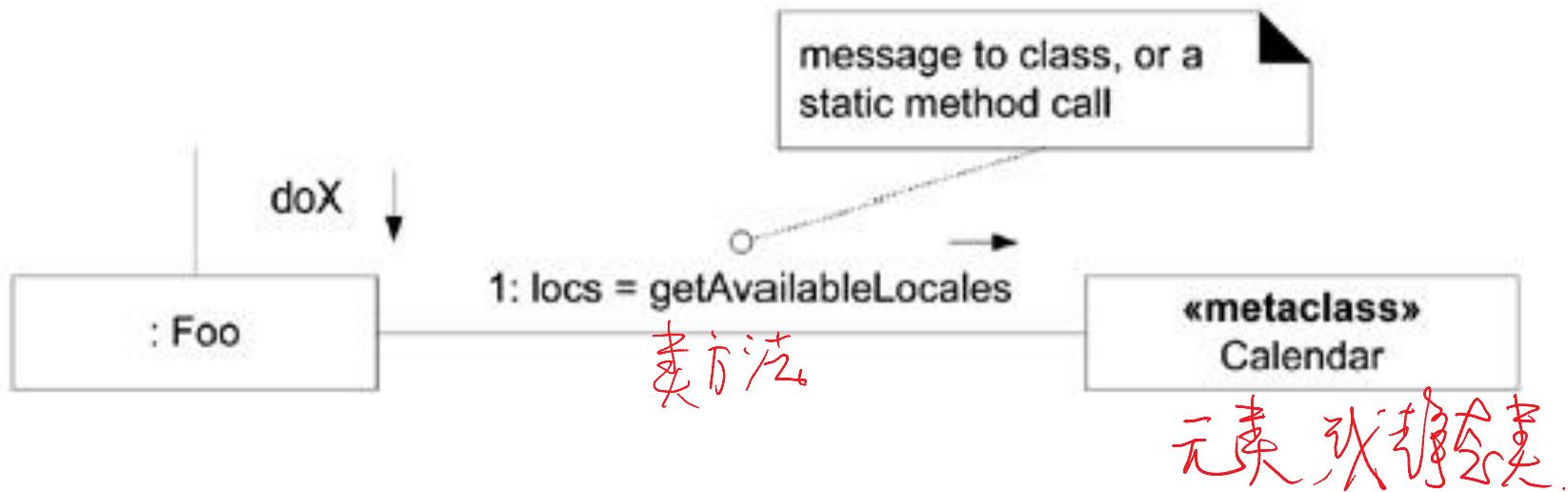
* [i = 1..n]

例: Iteration Over a Collection



静态方法调用

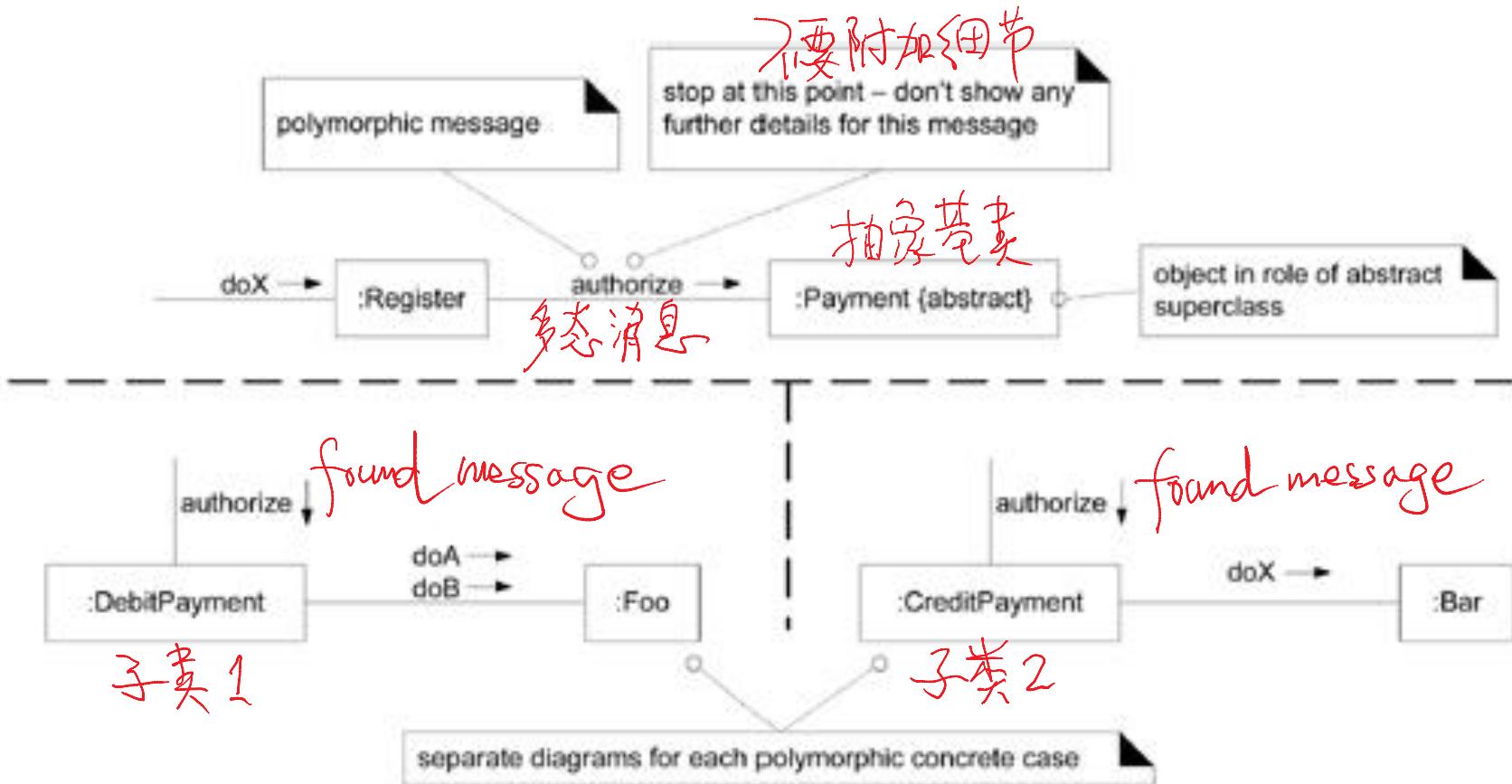
Invoke Static (Class) Methods



参见 Sequence 图中的表示方法

多态性消息的建模

Modeling Polymorphic Messages



Asynchronous call in a communication diagram

