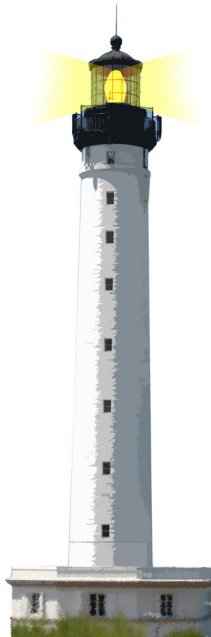
Inheritance and Lookup

101 on lookup

S. Ducasse and L. Fabresse



Goal

- Understanding
  - message sending
  - method lookup
  - semantics of self

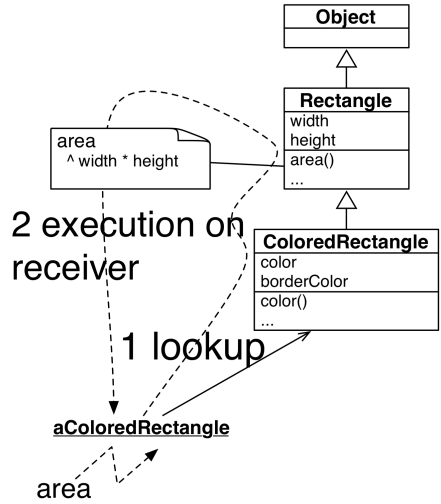
Remeber inheritance

- Inheritance of **state** is **static** (done at compile time)
- Inheritance of **behavior** is **dynamic**
- In this lecture we focus on the behavior part

Message sending

Sending a **message** is a two-step process:

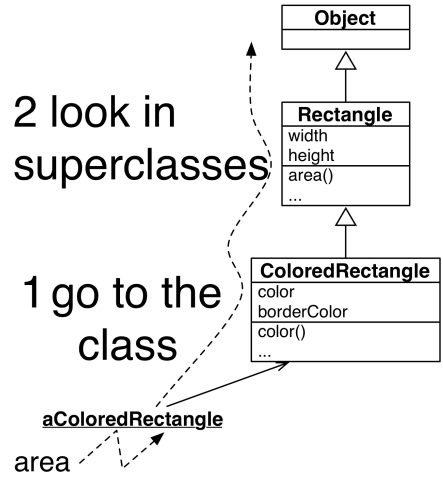
1. **look up** the **method** matching the message
2. execute this method on the **receiver**



Method lookup

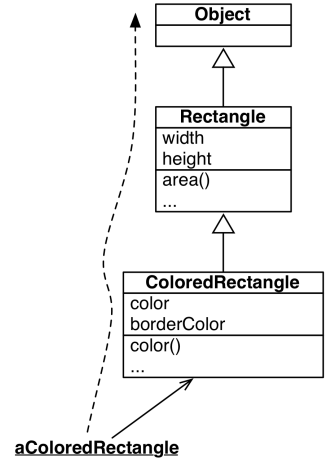
The lookup starts in the **class** of the **receiver** then:

- if the method is defined in the class, it is returned
- otherwise the search continues in the superclass



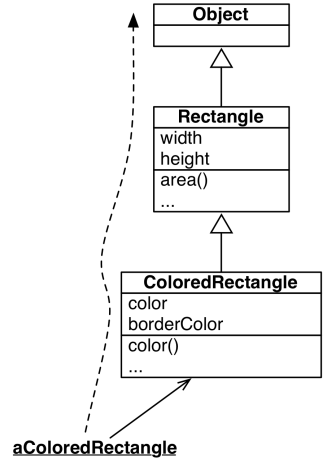
Some lookup cases

Sending the message color to aColoredRectangle

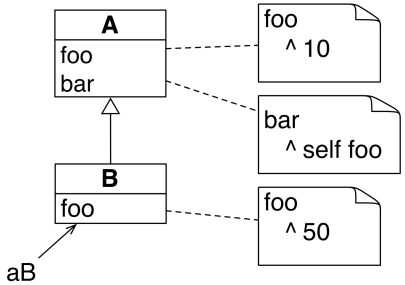


Some lookup cases

Sending the message area to aColoredRectangle



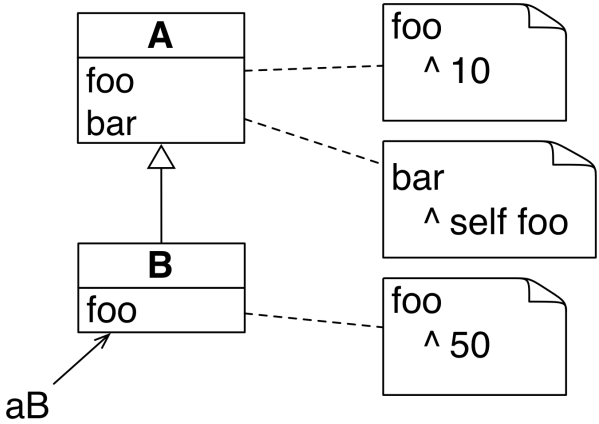
What is self/this?



Take 5 min and write the definition of self (this in Java).

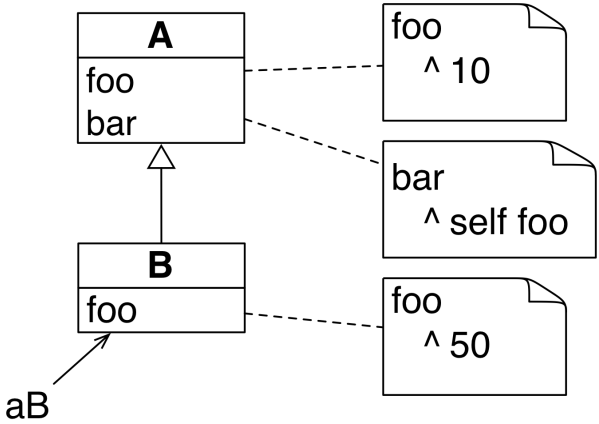
- your definition should have two points:
  - what does self represent?
  - how is a method looked up when a message is sent to self?

Let us explore a bit



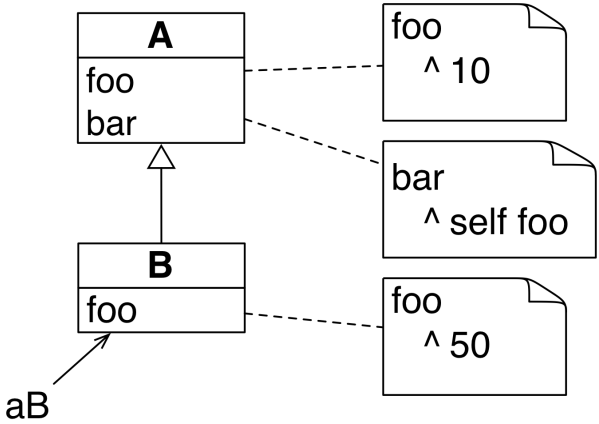

```
A new foo
> ...
B new foo
> ...
```

self always represents the receiver



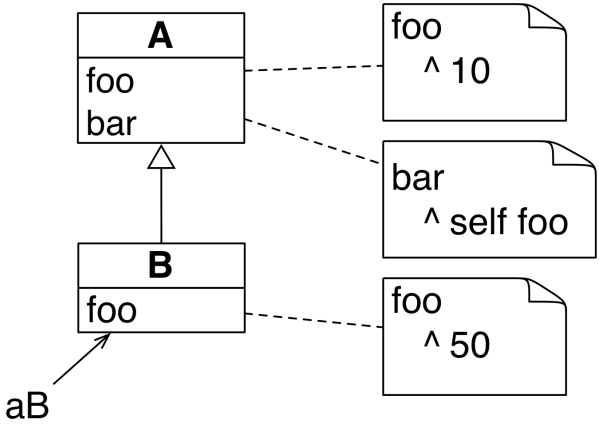

```
A new foo
> 10
B new foo
> 50
```

self always represents the receiver



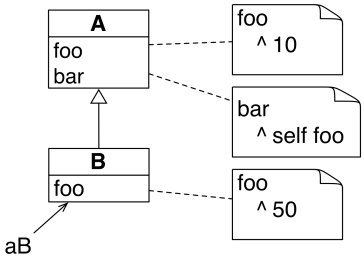

```
A new bar
> ...
B new bar
> ...
```

self always represents the receiver




```
A new bar
> 10
B new bar
> 50
```

Following message lookup and execution




```
B new bar
> 50
```

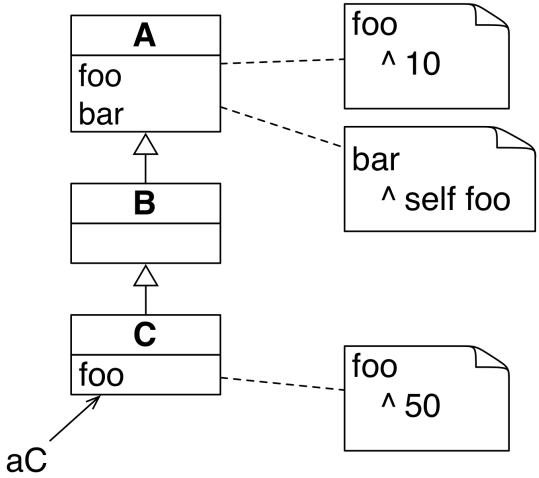
Evaluation of aB bar

1. aB's class is B
2. no method bar in B
3. look up in A - bar is found
4. method bar is executed
5. self refers to the receiver aB
6. foo is sent to self
7. look up foo in the aB's class: B
8. foo is found there and executed

self/this in two sentences

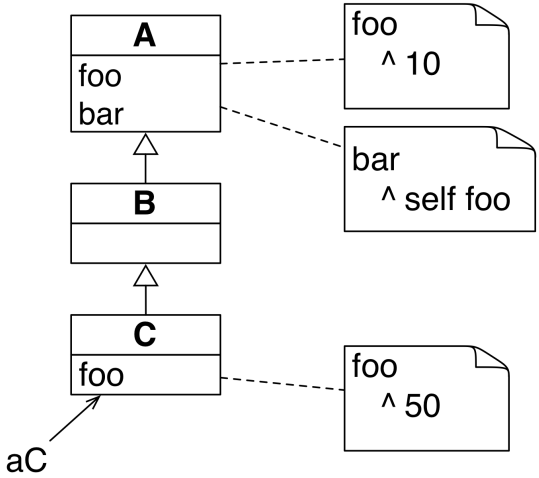
- self represents the **receiver** of the message
  - self in Pharo, this in Java
- The method lookup **starts in the class of the receiver**

self always represents the receiver




```
A new bar
> ...
B new bar
> ...
C new bar
> ...
```

self always represents the receiver




```
A new bar
> 10
B new bar
> 10
C new bar
> 50
```

What you should know

- self always represents the receiver
- Sending a message is a **two-step** process:
  1. **Look up** the method matching the message
  2. Execute this method **on the receiver**
- Method lookup maps a message to a method
- Method lookup starts in the **class of the receiver**
  - ...and goes up in the hierarchy

A course by

S. Ducasse, L. Fabresse, G. Polito, and Pablo Tesone