Inheritance and Lookup

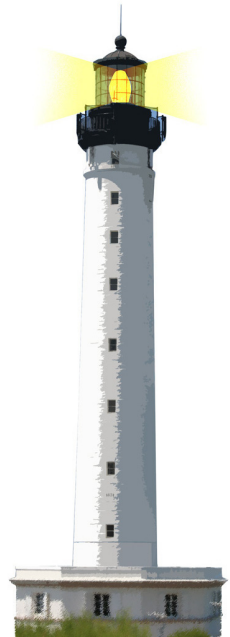
3: super

Damien Cassou, Stéphane Ducasse and Luc Fabresse

W4S03

Pharo

http://www.pharo.org



Goal

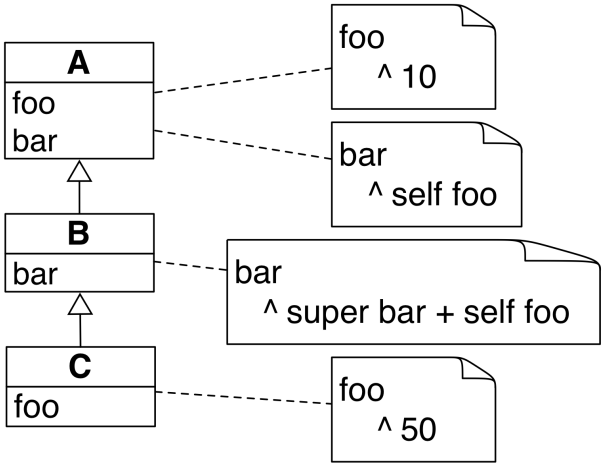
- Sending a message
- Method lookup
- super semantics and the differences with self

What is super?

Take 5 min and write the definition of super

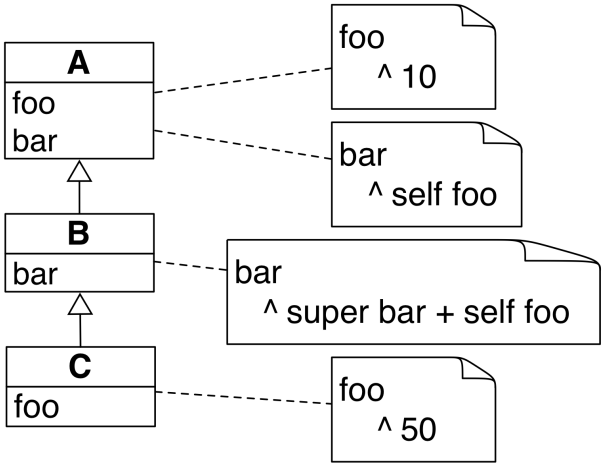
- your definition should have two points:
  - what does super represent?
  - how is a method looked up when a message is sent to super?

Challenge Yourself With super!



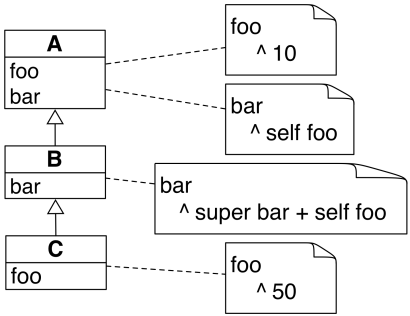

```
A new bar
> ...
B new bar
> ...
C new bar
> ...
```

Challenge Yourself With super!




```
A new bar
> 10
B new bar
> 20
C new bar
> 100
```

super Changes Where the Lookup Starts



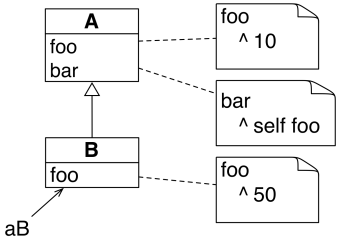
Evaluation of aC bar

1. aC's class is C
2. no method bar in C
3. look up in B - bar is found
4. method bar is executed
5. bar is sent to super
6. super is aC but lookup starts in A
7. bar is found in A and executed
8. foo is sent to aC
9. foo is found in C

super Changes Where the Lookup Starts

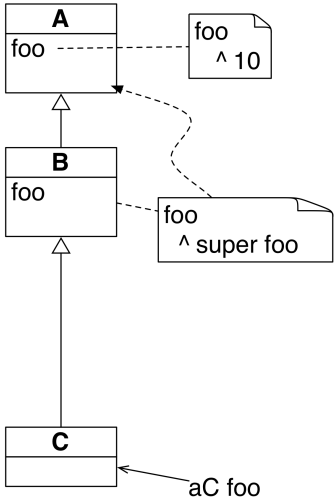
- super refers to the receiver of the message (just like self)
- The method lookup starts in the superclass of **the class containing the** super **expression**

self is Dynamic



We don’t know which foo method bar refers to

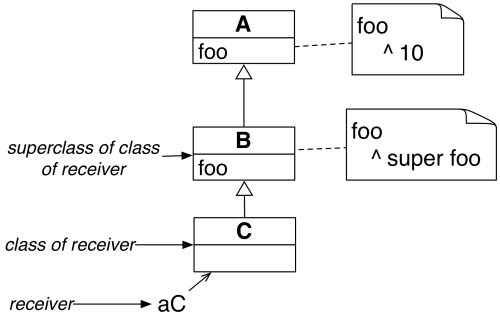
super is Static



- at compilation-time, we know that B>>foo refers to A>>foo
- we should look above the class containing the **method** using super

Even Some Books Got it Wrong

- **Wrong** definition: super looks for the method in the superclass of the ***receiver***'s class
- With this definition, this example would loop forever:



In reality it **does not** loop, the definition is wrong

What You Should Know

- self always represents the receiver
- super always represents the receiver
- super changes the lookup:
  - a super send starts the lookup in the class above it
- self sends act as a hook: code of subclasses may be invoked

A course by

and

in collaboration with