



**Learning Object-Oriented
Programming and Design with TDD**

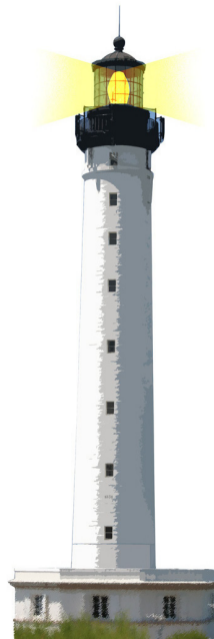
Really Understanding Class Methods

Stéphane Ducasse

<http://stephane.ducasse.free.fr>



W7S03



What You Will Learn

- There is only one lookup algorithm
 - it works for both instance and class methods
- Class methods are not Java-like static methods



There is Only One Lookup

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

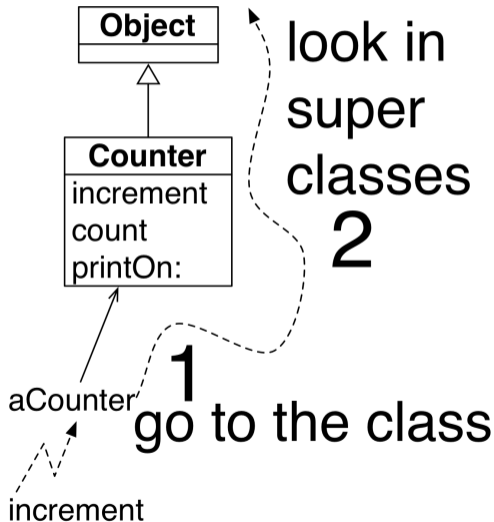
- if the method is defined in the class, it is returned
- otherwise the search continues in the superclass



Sending a Message

Sending a message is a two-step process:

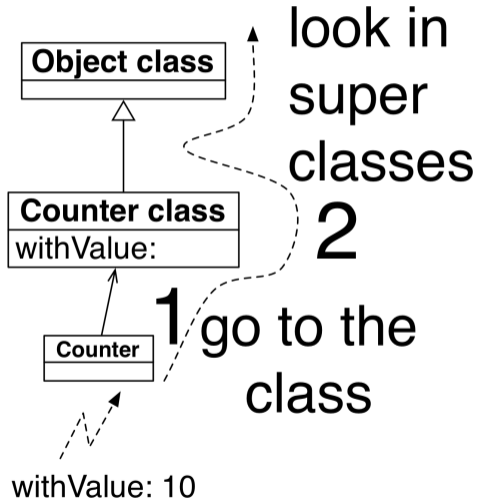
1. look up the method matching the message
2. execute this method on the receiver



Sending a Message to a Class

Sending a message to Counter (a class)

- go to the class of the class (Counter class)
- follow the inheritance chain
- apply the found method to the receiver: the class Counter



A Class is an Instance

A class is an instance of another class (called a metaclass)

- Counter is the unique instance of the class Counter class
- A metaclass is just one class whose instances are classes
- Counter class is automatically created after Counter
- The class of class XXX is named XXX class

Browsing two Classes

The browser can edit two classes: a class and its class

The diagram illustrates the relationship between two classes: **Counter** and **aCounter**. **aCounter** inherits from **Counter** (solid arrow) and has a **withValue: 10** association (dashed arrow). **aCounter** also has an **increment** association with **Counter** (dashed arrow).

The IDE screenshots show the following details:

- Counter class>>#withValue:** Shows the `withValue:` association with the code `^ self new count: anInteger`.
- Counter>>#increment:** Shows the `increment` association with the code `count := count + 1`.

```
classDiagram
    class Counter
    class aCounter
    Counter <|-- aCounter
    Counter ..> aCounter : withValue: 10
    Counter ..> aCounter : increment
```

What You Should Know

- A class is an object
- A class can receive messages
- A class is an instance of a class, called a metaclass
- Method lookup works the same
- More during the lecture *Understanding Metaclasses*



Resources

- Pharo mooc - Videos W5S02: <http://mooc.pharo.org>
- Pharo by Example: <http://books.pharo.org>



A course by Stéphane Ducasse
<http://stephane.ducasse.free.fr>

Reusing some parts of the Pharo Mocc by

Damien Cassou, Stéphane Ducasse, Luc Fabresse
<http://mocc.pharo.org>



Except where otherwise noted, this work is licensed under CC BY-NC-ND 3.0 France
<https://creativecommons.org/licenses/by-nc-nd/3.0/fr/>