



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

# Class Methods in Pharo

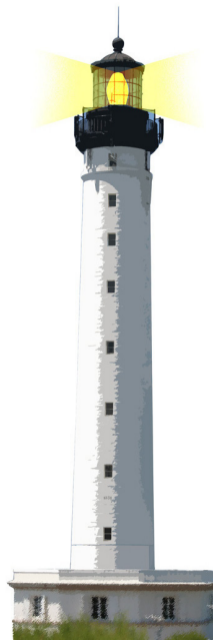
Stéphane Ducasse

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



<http://www.pharo.org>

**W6S03**



# Class Methods

1. In Pharo, everything is an object
2. Objects can receive messages
3. Classes are objects too

Classes can receive messages



# Examples

`Time` now

```
>>> 9:18:36.304688 pm
```

The message `now` is sent to the class `Time`

`Date` today

```
>>> 29 July 2015
```

The message `today` is sent to the class `Date`

# Examples

`FileLocator` `workingDirectory`

`ZnEasy` `getPng`: `'http://pharo.org/web/files/pharo.png'`

`ZnServer` `startDefaultOn`: 8080



# Class Methods are Defined on Class Side

Note the Class button pressed!

The screenshot shows an IDE window titled "Date class>>#today". The interface is divided into several panes:

- Left Pane:** A tree view showing a hierarchy of classes. "Chronology" is selected, and its sub-items include "Classes", "Copying", "Exceptions", "Messaging", "Methods", and "Models".
- Middle Pane:** A list of classes under "Date class". "Date" is selected. Other classes include "DateParser", "DosTimestamp", "Duration", "Stopwatch", "Time", "Timespan", and "Month".
- Right Pane (History Navigator):** A list of methods. "today" is selected under the "starting:" category. Other methods include "readFrom:", "readFrom:pattern:", "tomorrow", "week:day:", "year:day:", "year:month:day:", and "year:week:day:". The "today" method is highlighted in blue.
- Bottom Pane:** The source code for the "today" method, showing a single line: `^ self current`.

At the bottom of the IDE, there is a status bar with "1/3 [1]" on the left and "Format as you read W +L" on the right.

# Common Mistake

```
Counter class >> withValue: anInteger  
self new  
  value: anInteger;  
  yourself
```

Counter withValue: 10 returns the class Counter instead of a new instance



# Why?

```
Counter class >> withValue: anInteger  
  self new  
    value: anInteger;  
  yourself
```

is equivalent to

```
Counter class >> withValue: anInteger  
  self new  
    value: anInteger;  
  yourself.  
  ^ self
```

self here is the class Counter (the receiver of the message)



# Solution

```
Counter class >> withValue: anInteger  
  ^ self new  
    value: anInteger;  
    yourself
```



# Summary

- Classes are objects
- Messages can be sent to classes too
- Class-side methods are no different from other methods
- Most class-side methods create new instances
- To define a class-side method, press the `class` button



# Resources

- Pharo mooc - Videos W3S06: <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/>