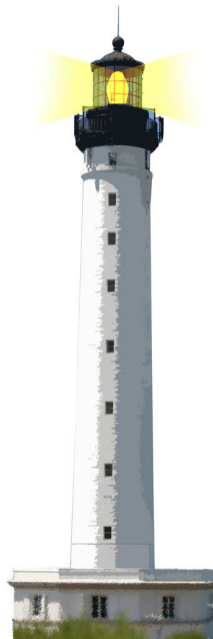


# About Advanced Object-Oriented Design

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



<http://www.pharo.org>



# Objectives

- Revisit basic elements such as encapsulation, delegation
- Think about essential aspects of OOP
- Learn key Design Patterns



# Outline

- Test and test-driven
- Essential aspects of OOP
- Basic and Universal Principles
- Double dispatch
- Case studies in Pharo
- Some design patterns
- Some coding idioms



# About Pharo

All the code samples and examples are taken using and from Pharo (<http://www.pharo.org>)

The screenshot displays the Pharo IDE interface with several tool windows open:

- Repositories:** A table listing local repositories.
- Branch browser:** A table showing commit history.
- Code editor:** Shows a stack trace and source code for a method.
- Inspector:** Shows the current object's state.

Name	Current branch	Loaded version	Status
iceberg	dev-0.4	f9c63ee (15 May 2017)	Up to date
libgit2-pharo-bindings	master	1f582be (18 April 2017)	Up to date

Date & time	Id	User	Parents
2017-05-13 18:16	ec0389e	Esteban Lorenzano	64dd53e
2017-05-13 16:22	64dd53e	Esteban Lorenzano	ee52967
2017-05-12 22:47	ee52967	Esteban Lorenzano	883763a
2017-05-12 17:11	883763a	Esteban Lorenzano	9880fc8
2017-05-12 17:11	9880fc8	Esteban Lorenzano	fefadb2
2017-05-12 14:24	fefadb2	Esteban Lorenzano	3f3e063, 7969cdc
2017-05-12 09:40	7969cdc	GitHub	5382135, 535238f
2017-05-12 09:32	535238f	Max Leske	5382135
2017-05-11 15:18	5382135	Esteban Lorenzano	127aa13, 432d664
2017-05-10 10:47	003688e	Esteban Lorenzano	

```
doOneCycleNowFor: world
"Immediately do one cycle of the interaction loop.
This should not be called directly, but only via doOneCycleNowFor:"

DisplayScreen checkForNewScreenTake.

"process user input events"
LastCycleTime := Time allTimeClockValue.
self handle: [ :k |
    self activate: k.
    a processEvent.
    self activate: nil.
]
```

# Why Pharo?

- Pharo is pure and powerful object-oriented language (read more at <http://www.pharo.org>)
- With only objects, messages, and lambdas, Pharo helps us to focus on important ideas!
- Complement the excellent Pharo mooc (<http://mooc.pharo.org>)



# About Vocabulary: The case of Polymorphism

- To avoid some misunderstandings
- The word "Polymorphism" is overloaded
- Literally, it is strange: having multiple forms :)
- It could mean subtyping: when a class is a subclass of another one.



# Our stance

"If it walks like a duck, swims like a duck, and quacks like a duck, then it probably is a duck."

We use the terms **polymorphic** or **polymorphism** in the sense of:

- exposing the same API,
- being substitutable,
- an object can be replaced by another one because both present the same API (set of messages),
- polymorphic objects do not have to be subtypes



# Polymorphism

- Looking at object API
- To produce reusable and extensible programs
- Not much as class structural relation (subtyping)





# A unique lecture

- More than 60 years of experience in OO design and practices
- Concrete cases
- Knowledge actionable in many languages



A course by

S. Ducasse, G. Polito, and Pablo Tesone



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/>