

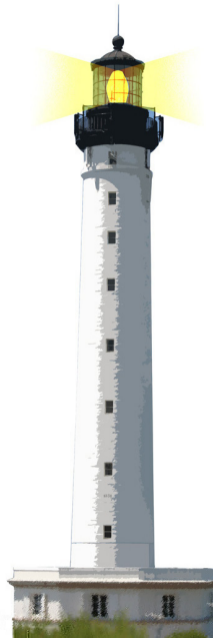


**Learning Object-Oriented
Programming and Design with TDD**

The Two Interfaces of OOI

Stéphane Ducasse

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



Outline

- Some questions
- Two different clients!

What are the consequences of

A

```
{ private x;  
void foo(){ ... x ...}  
}
```

Consequences

- Cannot access `x` from clients
- Cannot replace `x` by something else
- Cannot extend from subclasses

Cannot even copy and paste the body of `foo(){ ...}` in subclasses to extend it manually!

Clients?

What are the clients of a class?

- Its users (e.g., Person is client of Address)
- But also its **subclasses**

What do you think about this

Some tutorials mention: Fields should be private

You cannot predict the future

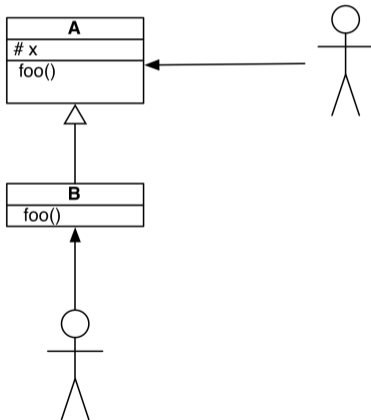
- You are not the Kwisatz Haderach (Yes Dune)!
- You cannot predict how your classes **MUST** be extended in 5 years from now!
- Think about your extenders!
- No final no private, use protected!

The correct idiom is...

- Fields should be private AND class should provide protected accessors
- OR
- Fields should be protected

Benefits

- Your clients cannot access your fields
- And your subclasses are empowered
 - A subclass can extend/refine the behavior of the superclass



OOP is about encapsulation AND extension

A class has always two clients:

- Its users
- Its extenders

Remember: **Late binding** is the core of OOP



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/>