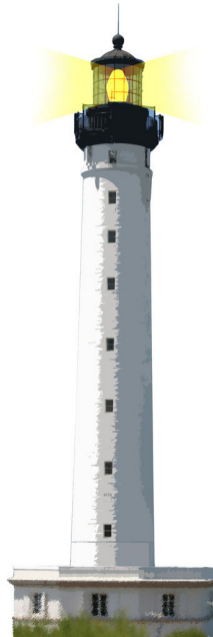


The two interfaces

about programming deltas

Stéphane Ducasse



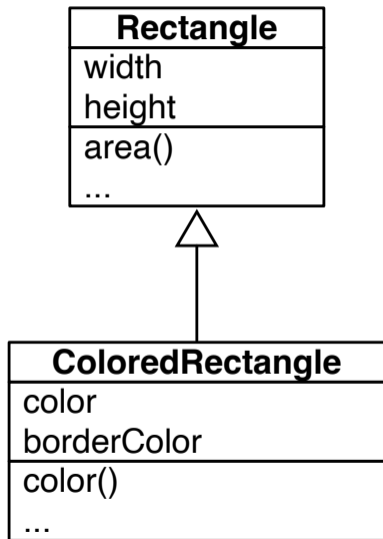
Outline

- Reminders
- Some questions
- Two different clients!



Back to the roots: Inheritance

- Often we want small adaptations to existing classes
- Do not want to reimplement everything: We want to reuse existing behavior
- Solution: **class inheritance**
- A class **extends** the definition of its superclass
 - add state
 - **extend** / **specialize** behavior



Inheritance: expressing delta

Inheritance is a reuse mechanism

- Do not reimplement the code of the superclasses in subclass
- **Extend** or **Specialize** superclass behavior

A subclass expresses a **delta**

- Only specify the differences to the superclasses



What are the consequences of

A

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

Consequences

- Cannot access `x` from clients - ok sounds good
- Cannot access `x` from subclasses - no ok!

How can I express a good delta?

- 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?

Fields should be private



You cannot predict the future

- You are not the Kwisatz Haderach!
- 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**

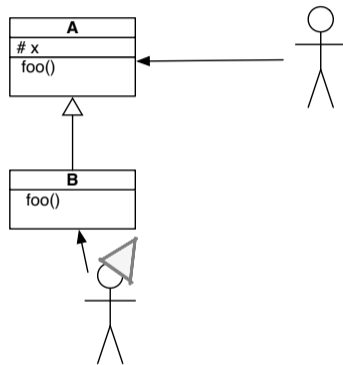
Favor protected

- Support encapsulation AND extension



Benefits

- Your clients cannot access your state
- 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**



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/>