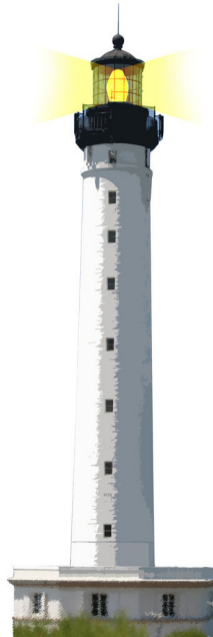


Libraries versus Frameworks



Outline

- What are libraries?
- What are frameworks?
- Some guidelines to create a new framework



What are Libraries?

- a *library* is a set of code entities (classes and methods)
- the *client* of a library sends *requests* to the library
- the library returns *responses*
- the main code (dark blue) is in the client

```
/Users/ducasse/Workspace/FirstCircle/MyBooks/Bk-Writing/Adva
```

What are Frameworks?

- a *framework* is a set of code entities (classes and methods), just like in libraries
- the framework has *abstractions* (i.e., holes) to be implemented
- the *client* implements the abstractions to parameterize the framework
- the main code (dark blue) is in the framework

/Users/ducasse/Workspace/FirstCircle/MyBooks/Bk-Writing/Advanced

Libraries versus Frameworks

- Library
 - You call it
 - You use callbacks to extend it
- Framework
 - *Hollywood principle*: Don't call me, I will call you



Libraries versus Frameworks

Library	Framework
The client instantiates the classes	The framework instantiates the classes
The client invokes library functions	The framework invokes code in the client
The client is responsible for the flow	The framework is responsible for the flow

Inheritance as Parameterization

- a client parameterizes a framework by overriding operations
- the *template method* design pattern is often used

/Users/ducasse/Workspace/FirstCircle/MyBooks/Bk-Writing/Advanced

Framework Design

- Needs at least 3 clients
- "*A Pattern Language for Developing Object-Oriented Frameworks*", by Don Roberts and Ralph Johnson
- Framework design often relies on whitebox abstractions
 - framework are extended by inheritance
 - blackbox abstraction is also possible: usage of composition

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/>