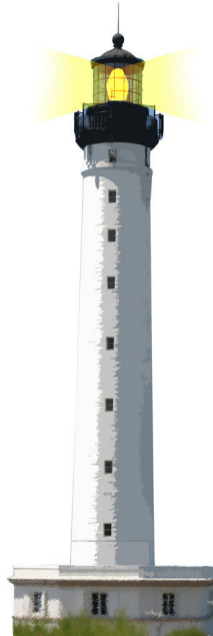


Advanced Object-Oriented Design

About Facade



<http://www.pharo.org>



Outline

- Facade
- Avoid it
- Visitor variations

Facade intent

A facade is an object that provides a simplified interface to a larger body of code, such as a class library.

"Expected" consequences

A facade can:

- make a software library easier to use, understand and test, since the facade has convenient methods for common tasks;
- make the library more readable, for the same reason;
- reduce dependencies of outside code on the inner workings of a library, since most code uses the facade, thus allowing more flexibility in developing the system;
- wrap a poorly designed collection of APIs with a single well-designed API.



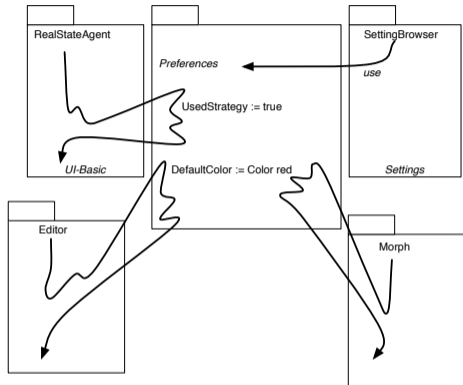
Seen consequences

Often people do not use Facade well

- Procedural design
 - all the clients refers to a single point: often a singleton
- Hampering modularity
 - Difficult to layer application
 - Difficulty to unload the Facade

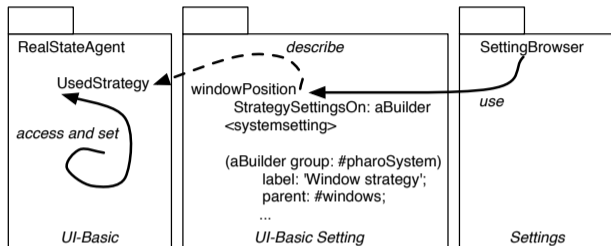


Studying Preferences Facade



- Control flow is not local
- 180 system wide preferences
- Impossible to remove Preferences packages
- Preferences link to too many package (default PngReader)

Solution for preferences



UI-Basic has customization points.

UI-Basic Setting describes them.

Setting collects system settings and populate setting browser.

- Each element defines its preferences
- The flow is local to the element
- The local preferences are described
- Layered architecture: we can remove Preferences

Conclusion

- Facade may work
- Usually this is a plague



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/>