

OO Design “Part II” unit:

- ▶ General introduction to DP; Strategy (Monday)
- ▶ State (Wednesday)
- ▶ Strategy revisited (lab Thursday)
- ▶ Adapter and Decorator (**today**)
- ▶ (Begin pseudo-assembly unit next week Monday)
- ▶ State revisited (lab next week Thursday)

Today:

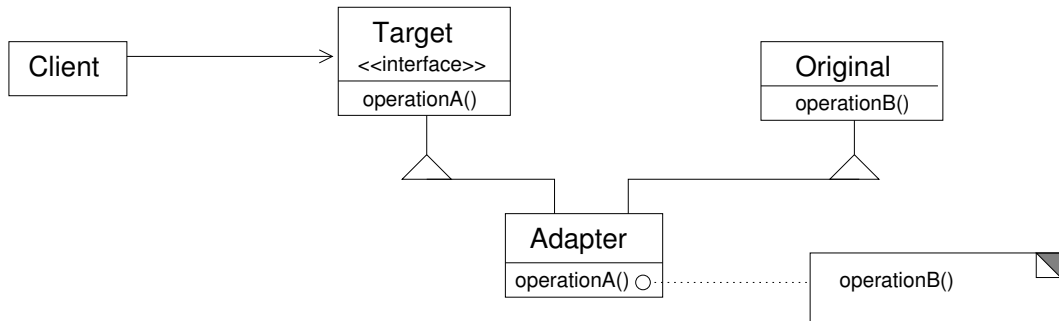
- ▶ The Adapter pattern
 - ▶ Motivating examples
 - ▶ Intent and structure
 - ▶ Class adapter
 - ▶ Object adapter
- ▶ The Decorator pattern
 - ▶ Examples
 - ▶ Intent and structure
 - ▶ Comparison with Adapter and Strategy
 - ▶ Pred-prey example



Posted by Carlos Monoz at <http://nintendoage.com/forum/messageview.cfm?catid=5&threadid=56851> on March 17, 2011; read Nov 30, 2015.

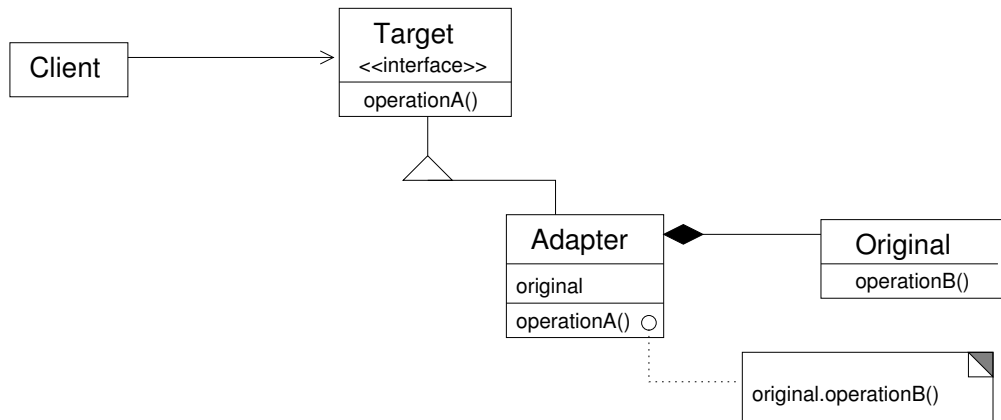
Adapter. Convert the interface of a class into another interface that clients expect. Adapter lets classes work together that couldn't otherwise because of incompatible interfaces. [Gamma et al, *Design Patterns*, pg 139.]

Class adapter:



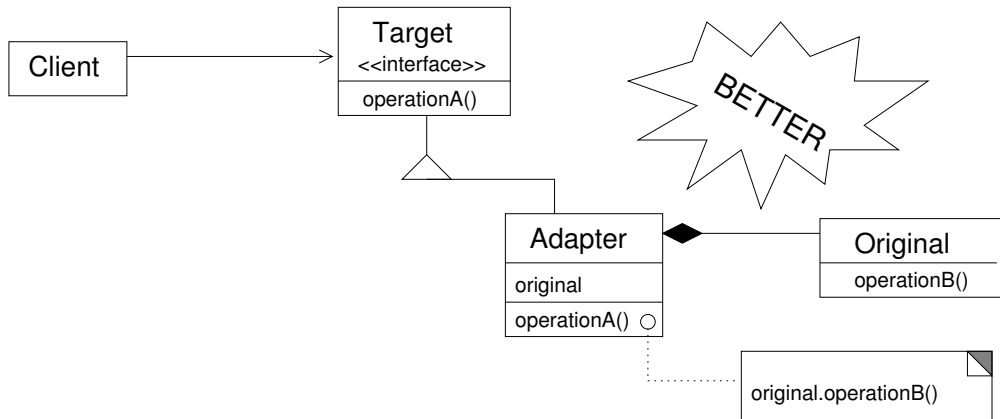
Adapter. Convert the interface of a class into another interface that clients expect. Adapter lets classes work together that couldn't otherwise because of incompatible interfaces. [Gamma et al, *Design Patterns*, pg 139.]

Object adapter:

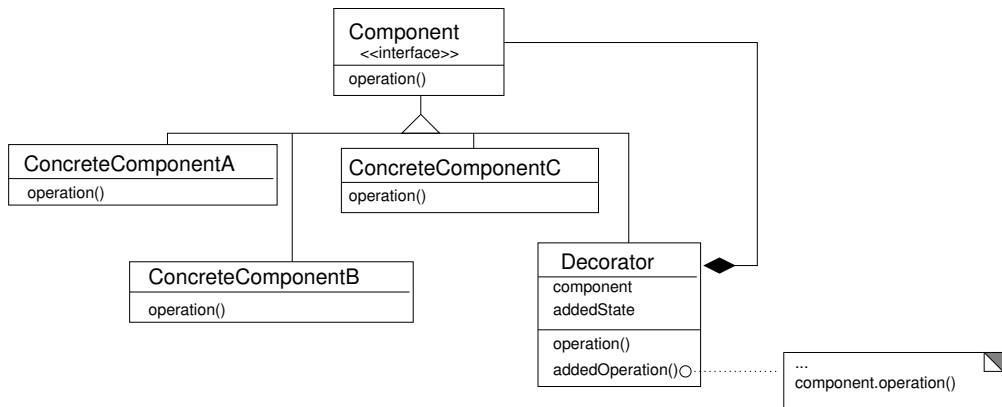


Adapter. Convert the interface of a class into another interface that clients expect. Adapter lets classes work together that couldn't otherwise because of incompatible interfaces. [Gamma et al, *Design Patterns*, pg 139.]

Object adapter:



Decorator. Attach additional responsibilities to an object dynamically. Decorators provide a flexible alternative to subclassing for extending functionality. [Gamma et al, *Design Patterns*, pg 175.]



How do Adaptor and Decorator compare and contrast?

- ▶ Adaptor changes the *interface*, not the responsibility or functionality (although it may hide some functionality).
- ▶ Decorator changes *responsibility*, at most adding to the interface.

How do Decorator and Strategy compare and contrast?

- ▶ The Strategy pattern changes the *guts* of an object. The object adding new functionality is a component of the basic object.
- ▶ The Decorator pattern changes the *skin* of an object. The basic object is a component of the object adding new functionality.

Coming up:

- ▶ **Due Wed, Apr 22.** *Do Project 7, no-ifs calculator*