



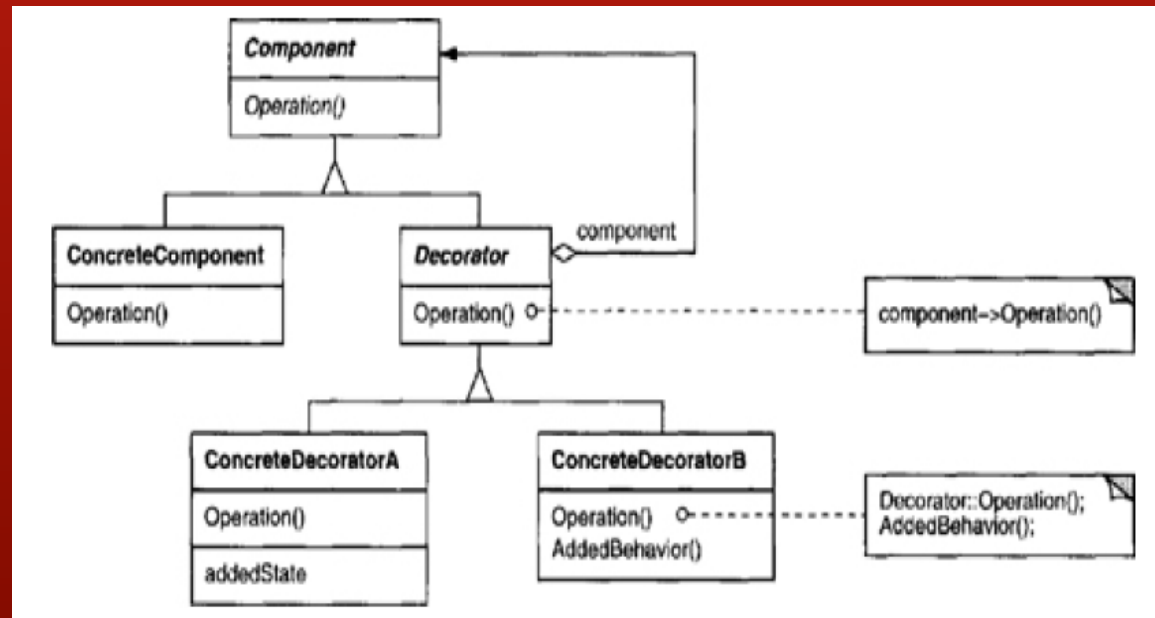
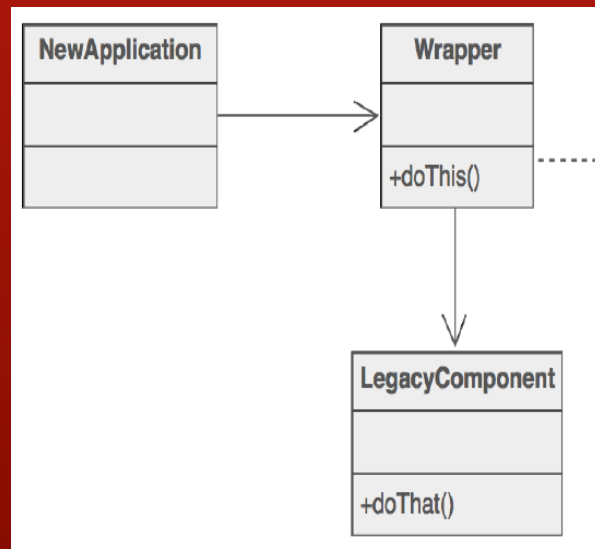
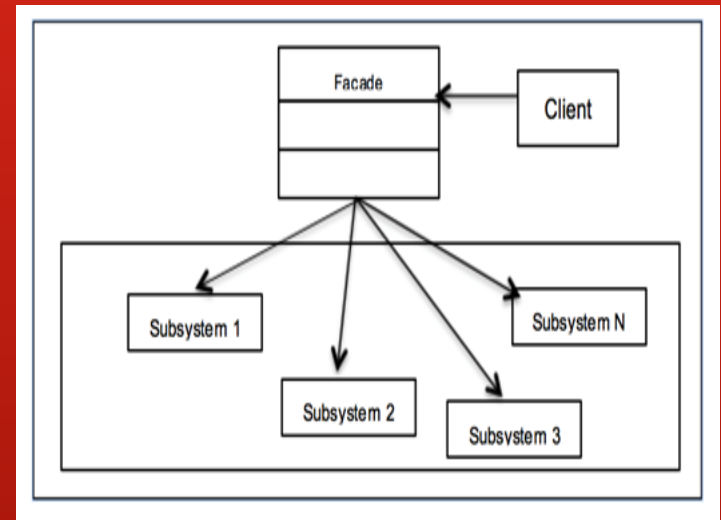
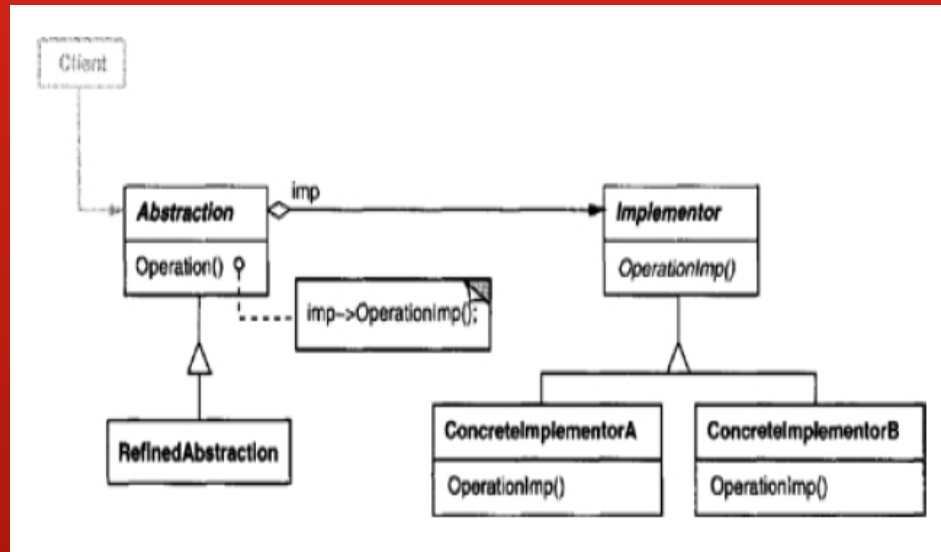
Com S 362

Object-Oriented Analysis & Design

Spring 2018 – Friday Apr 20, 2018

Adaptive Patterns

- Facade:
 - new face on existing code
 - defines a *new* interface.
- Adapter:
 - Creates connector between two existing pieces of code who are “almost” compatible.
- Bridge:
 - Usually a design-time creation.
 - Creates stable abstract interface to multiple, apt to increase in number concrete implementations – that may not always implement abstract interface cleanly.
- Decorator
 - Exposes super set of existing interface
- Proxy
 - Exposes unchanged interface



Factory-like (instancings) patterns

- Abstract Factory
 - Client requests specific object
 - returns single object, might be initialized
- Builder
 - Client submits specification of desired structure
 - Builder returns a completely assembled struture
- Clone
 - Instance source owns list of "master" instances.
 - Client gets a "dedicated" copy of the "master"
- Flyweight
 - Instance source owns a list of sharable instances
 - Client gets ref to shared instance ... must maintain non-sharable state externally.

