327: Object-oriented programming

Lecture 17

11/3/2021

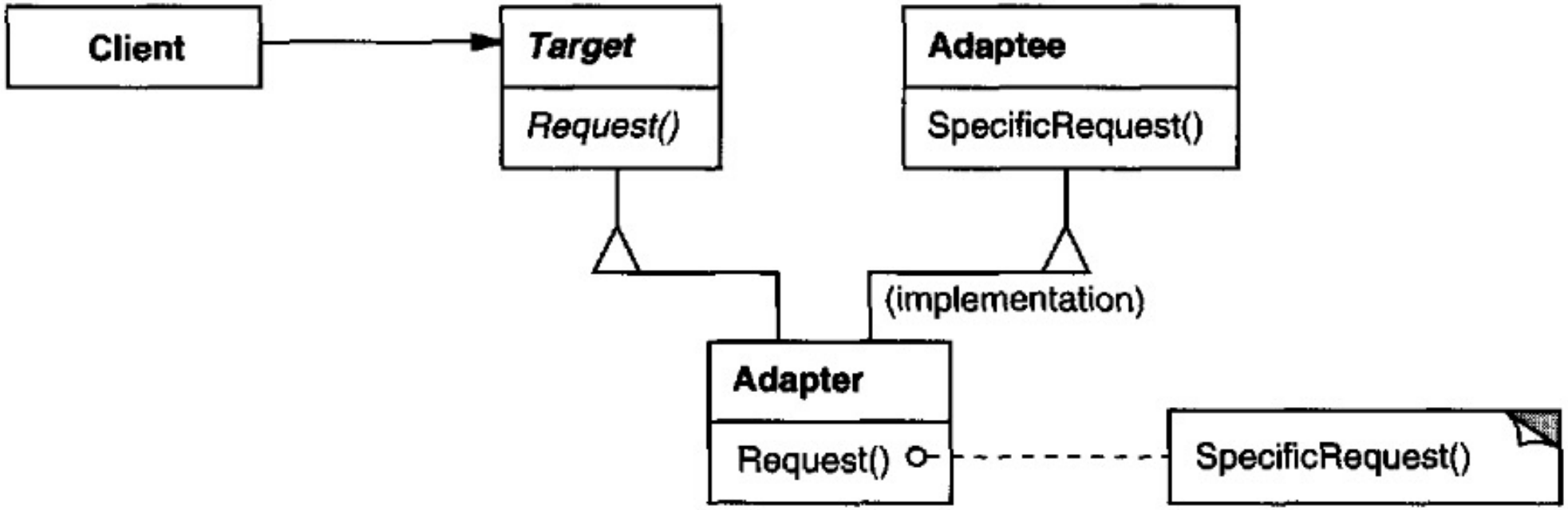
Professor Barron

Adapter pattern

- Structural pattern
- Convert an interface into another interface
- Allows existing objects to work together without modifying their rigid public interfaces
- Also called a wrapper
  - kind of an overloaded term
  - a decorator is also a wrapper, but it doesn't change the interface

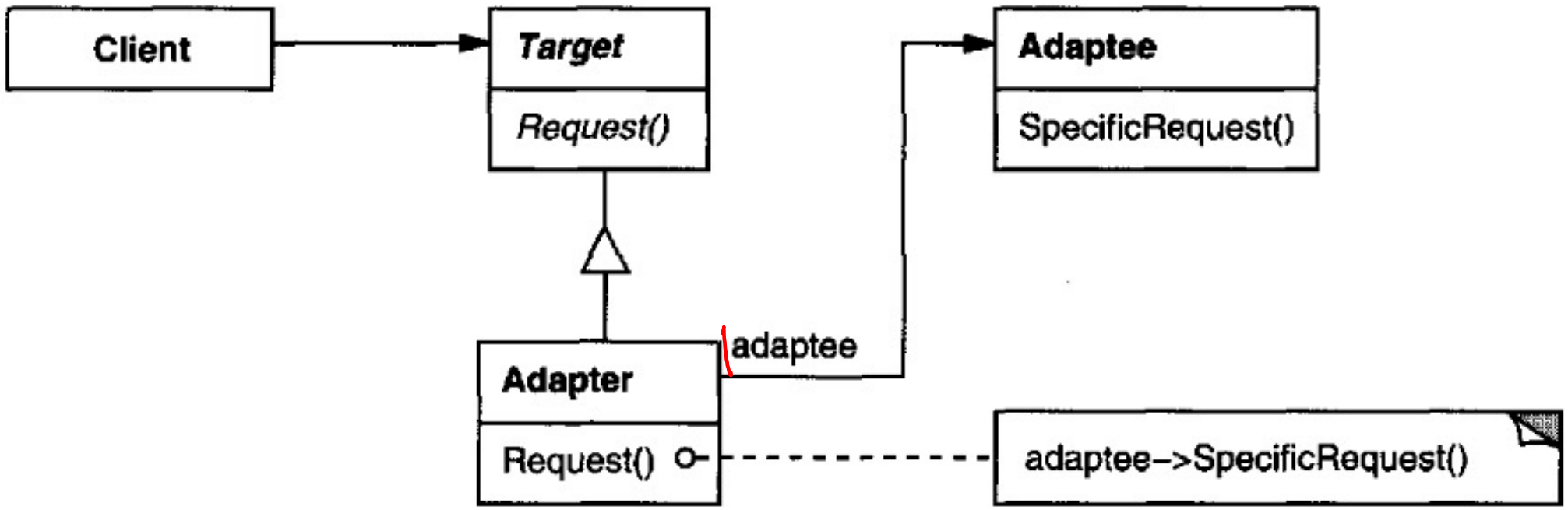
Class diagram

- class adapter uses multiple inheritance

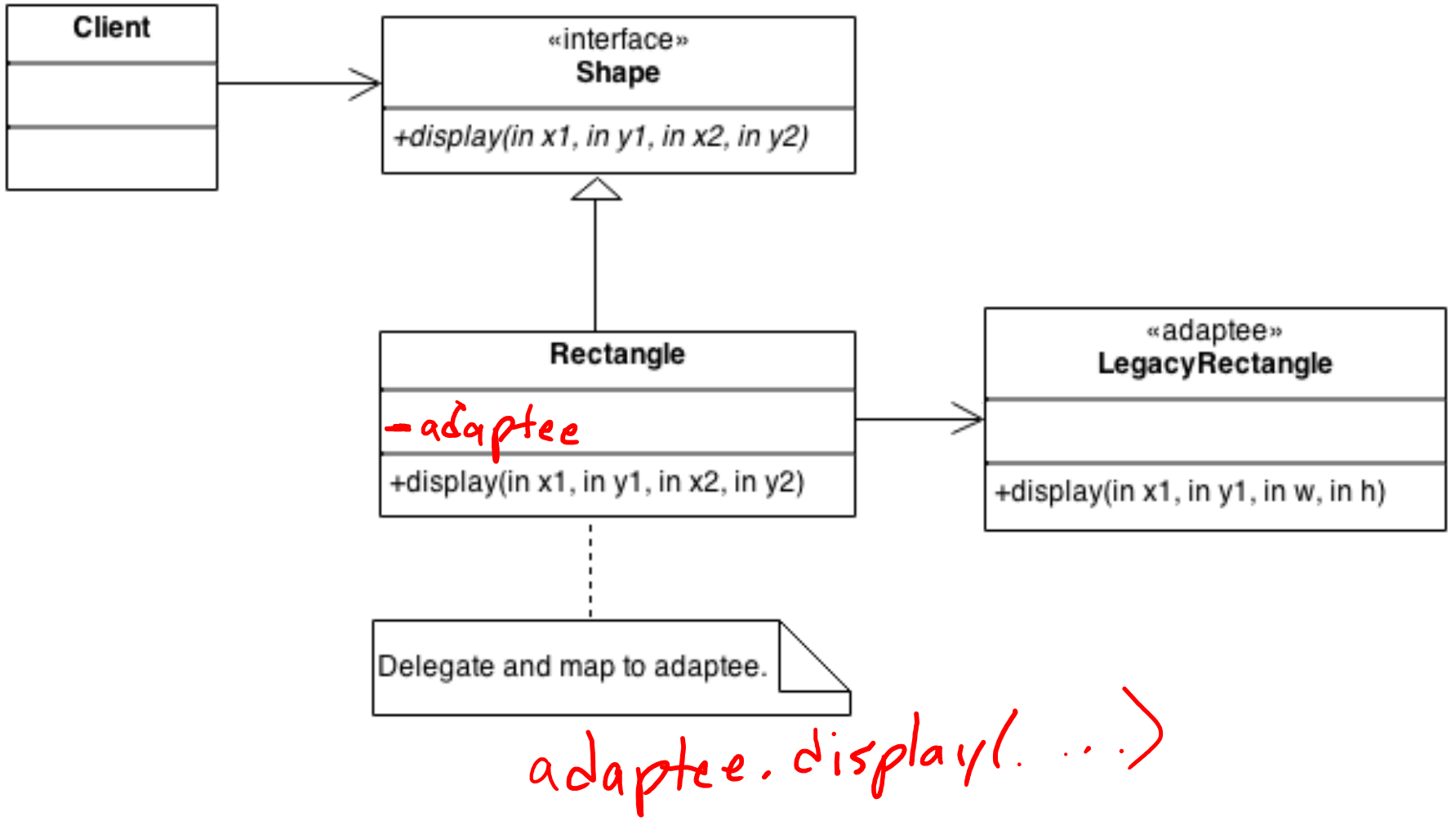


Class diagram

- object adapter uses an object reference to delegate



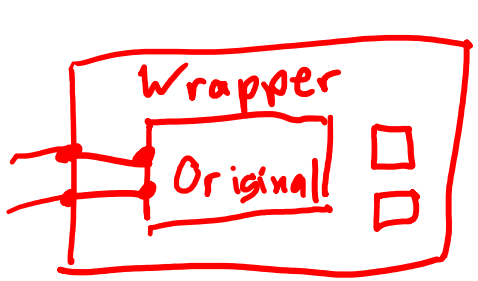
Example

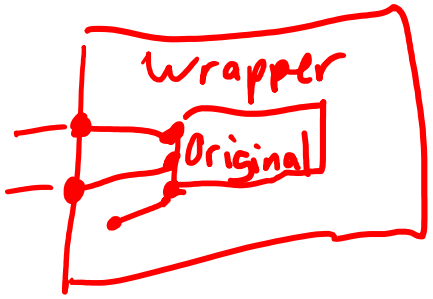

Client
«interface»
Shape
+display(in x1, in y1, in x2, in y2)
Rectangle
-adaptee
+display(in x1, in y1, in x2, in y2)
«adaptee»
LegacyRectangle
+display(in x1, in y1, in w, in h)
Delegate and map to adaptee.
adaptee.display(...)

Potential translations in adapter

- Convert arguments to a different format
- Rearrange the order of arguments
- Call a differently named method
- Supplying default arguments
- *No new behavior*

vs Decorator

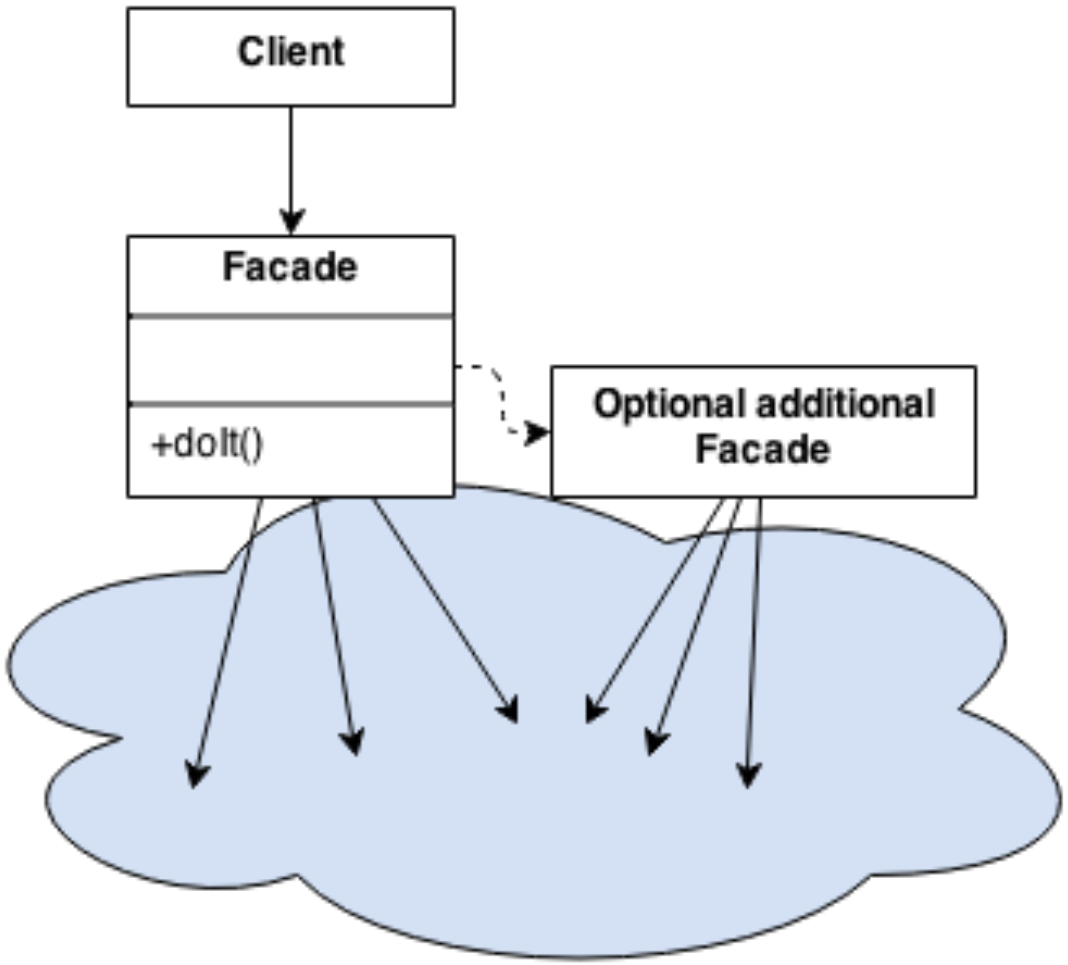




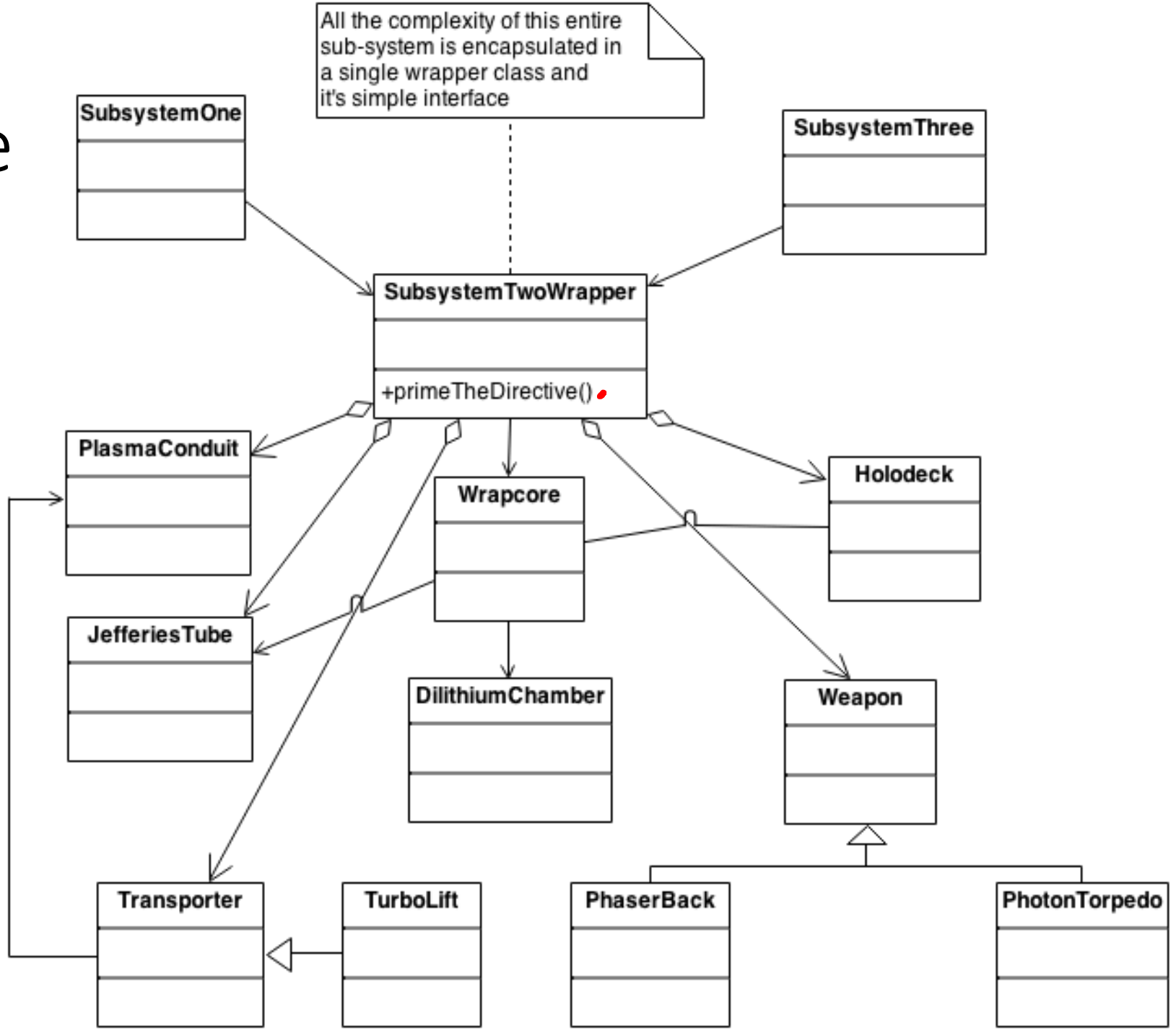
Facade pattern

- Structural pattern
- Provides a simpler or higher-level interface to some complicated code
- A type of wrapper like adapter and decorator
  - no new functionality

Diagram



Example


All the complexity of this entire sub-system is encapsulated in a single wrapper class and it's simple interface
SubsystemOne
SubsystemThree
SubsystemTwoWrapper
+primeTheDirective()
PlasmaConduit
Wrapcore
Holodeck
JefferiesTube
DilithiumChamber
Weapon
Transporter
TurboLift
PhaserBack
PhotonTorpedo

Facades in Python

- So commonly used that it’s not always referred to by name
- for loops are a façade over iterators
- defaultdict is a façade over dicts to reduce reoccurring code when keys do not exist
- requests library is a façade over lower-level HTTP libraries
  - HTTP libraries are a façade over managing text-based messages over sockets