

Design Patterns

Façade

COMP3607

Object Oriented Programming II

Week 9

Outline

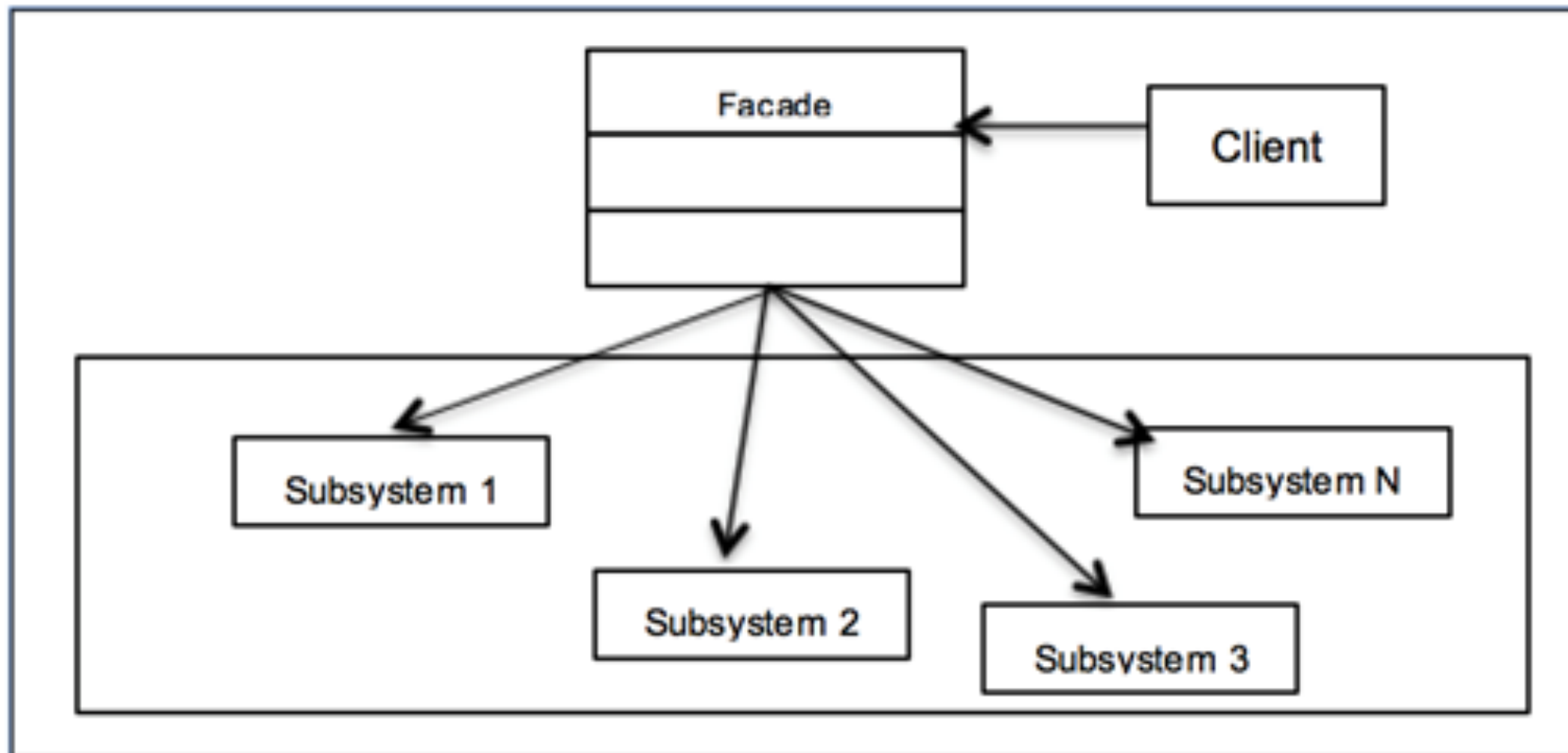
- Design Patterns
 - Façade

Façade Pattern: Intent

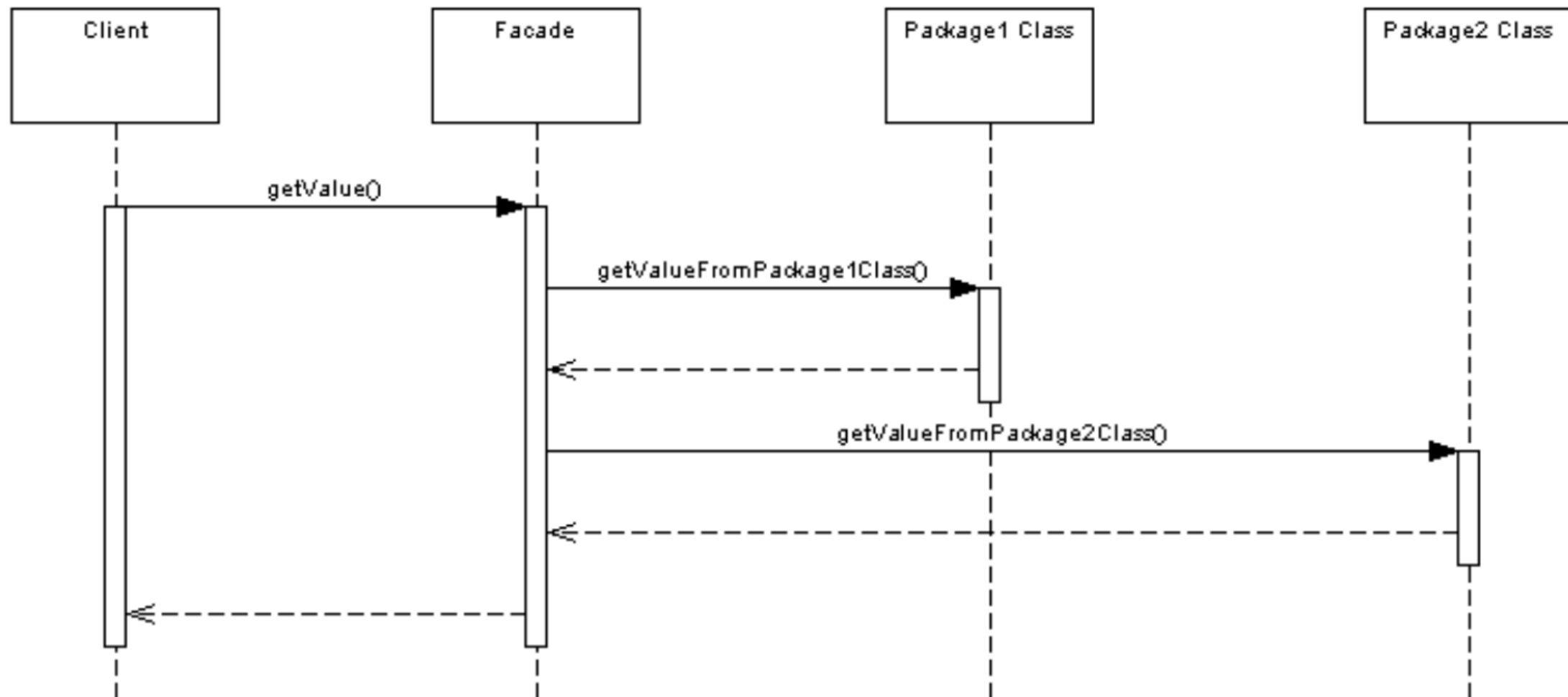
Facade defines a higher-level interface that makes the subsystem easier to use.

- Provides a unified interface to a set of interfaces in a subsystem.
- Wraps a complicated subsystem with a simpler interface.

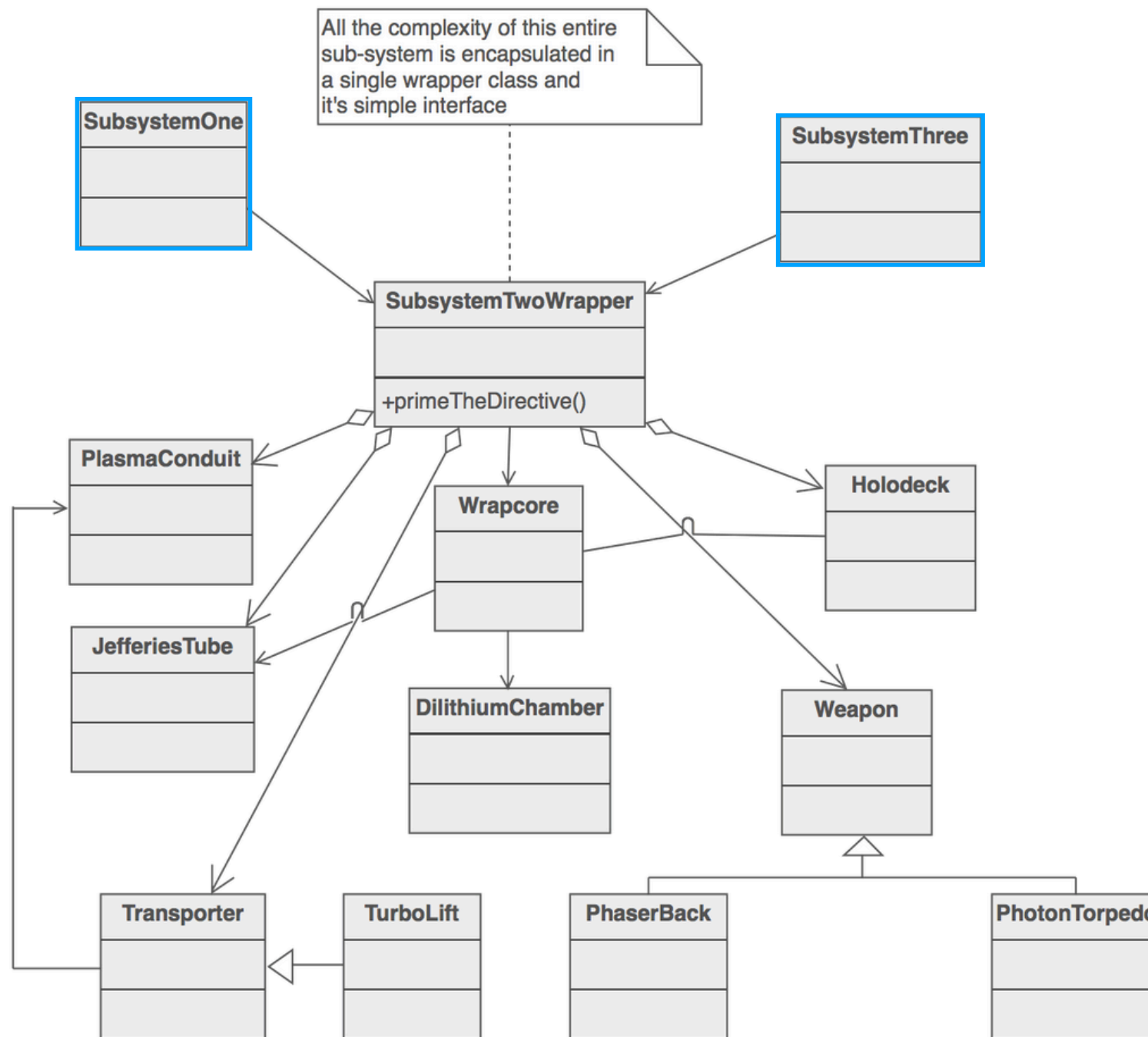
Façade Pattern



Façade Pattern



Façade Pattern: Example



SubsystemOne and SubsystemThree do not interact with the internal components of SubsystemTwo.

They use the SubsystemTwoWrapper "facade" (i.e. the higher level abstraction).

11

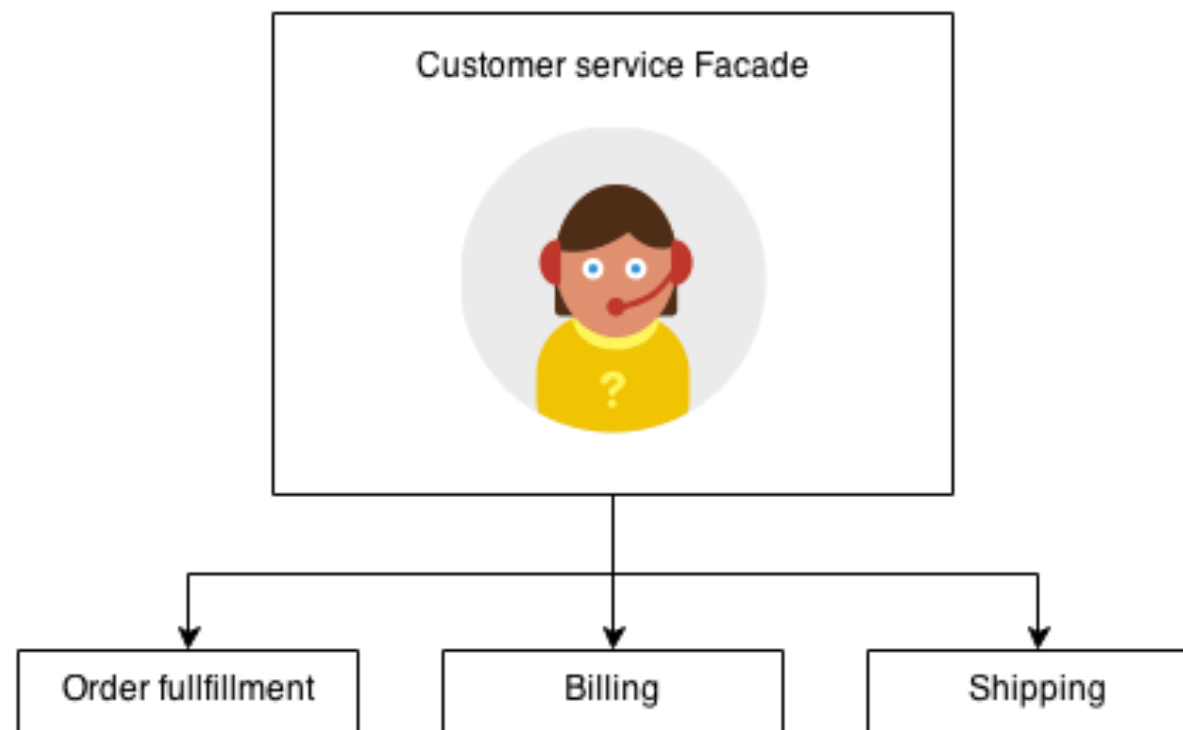
Façade Design Pattern

<https://www.youtube.com/watch?v=B1Y8fcYrz5o>

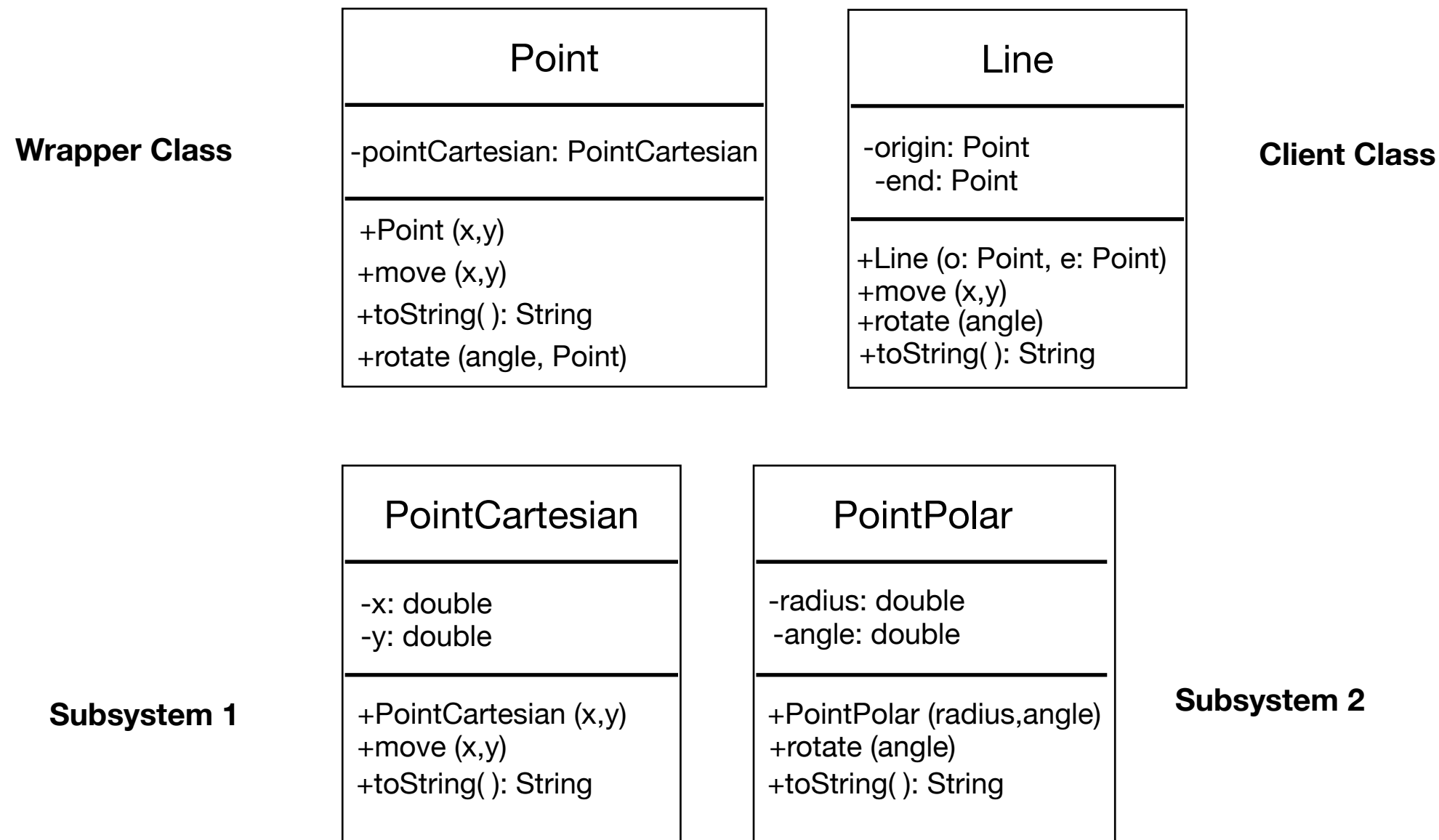
(example code in Java) -Derek Banas

Example

The Facade defines a unified, higher level interface to a subsystem that makes it easier to use. Consumers encounter a Facade when ordering from a catalog. The consumer calls one number and speaks with a customer service representative. The customer service representative acts as a Facade, providing an interface to the order fulfillment department, the billing department, and the shipping department.



Example



Applicability:

The Façade pattern is used:

- To simplify an interface
 - For example, service oriented architectures make use of the facade pattern.
 - A web service might provide access to a number of smaller services that have been hidden from the caller by the facade.
 - A typical pattern in OSGi bundles is to provide an interface package that is exposed to users of the bundle. All other packages are hidden from the user.

Adapter vs Façade

- Facade defines a new interface, whereas Adapter reuses an old interface.
- Remember that Adapter makes two existing interfaces work together as opposed to defining an entirely new one.

Adapter vs Façade

- Adapter and Facade are both wrappers; but they are different kinds of wrappers.
- The intent of Facade is to produce a simpler interface, and the intent of Adapter is to design to an existing interface.
- Facade routinely wraps multiple objects and Adapter wraps a single object.
- Facade could front-end a single complex object and Adapter could wrap several legacy objects.

Question:

So the way to tell the difference between the Adapter pattern and the Facade pattern is that the Adapter wraps one class and the Facade may represent many classes?

Answer:

No! Remember, the Adapter pattern changes the interface of one or more classes into one interface that a client is expecting.

While most textbook examples show the adapter adapting one class, you may need to adapt many classes to provide the interface a client is coded to.

Likewise, a Facade may provide a simplified interface to a single class with a very complex interface.

The difference between the two is not in terms of how many classes they "wrap", it is in their intent.

References

Design Patterns: online reading resources

- Facade
 - https://sourcemaking.com/design_patterns/facade
 - <https://dzone.com/articles/design-patterns-uncovered-1>