

Bridge Pattern

Decouples an abstraction from its implementation so that the two can vary independently.*

*GoF

Bridge Pattern

Decouples an abstraction from its implementation
so that the two can vary independently.* **

*GoF

**wtf?

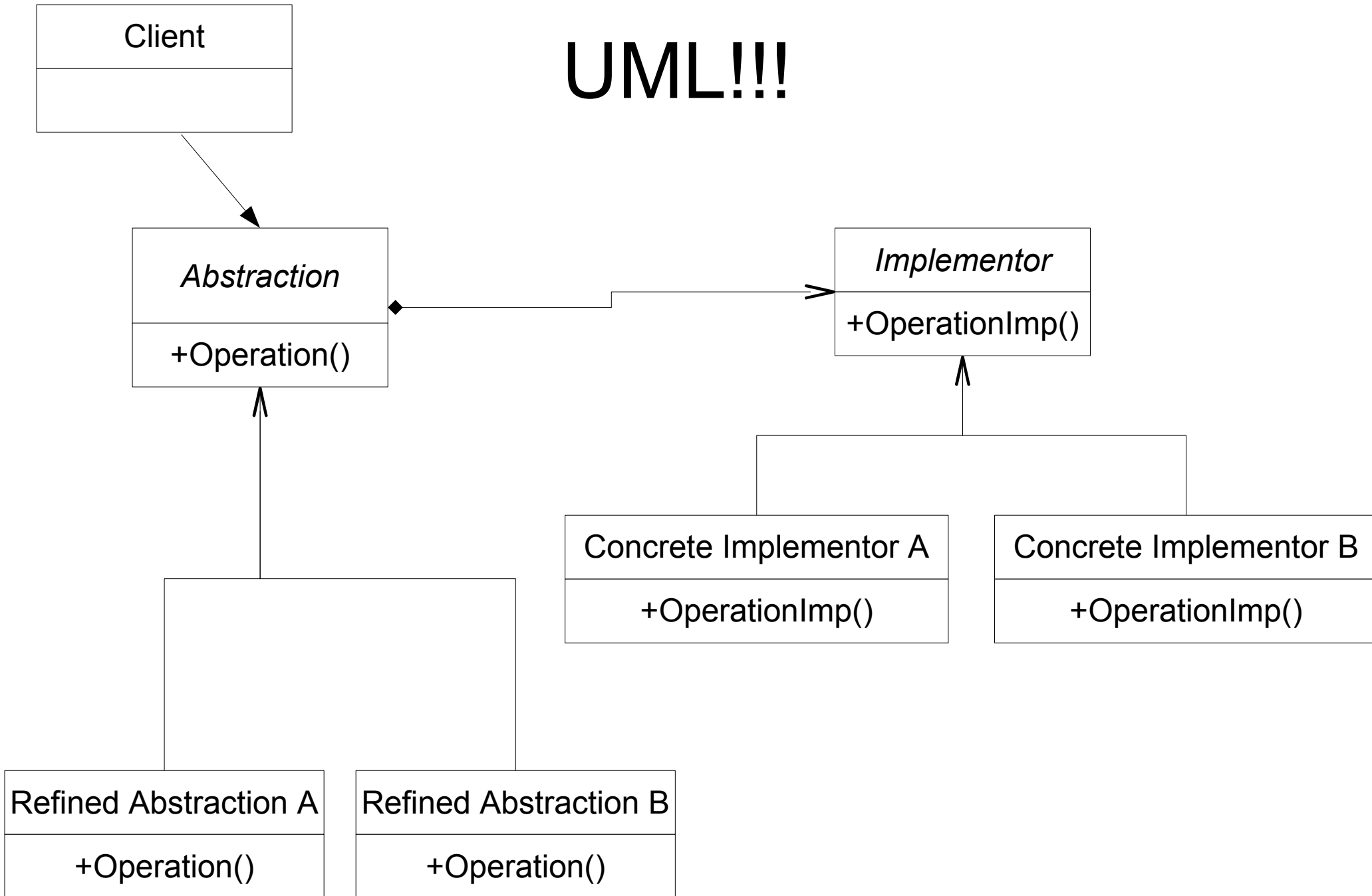
Bridge Pattern

Separate a varying entity from a varying behavior (separate an "abstraction" from its "implementation"), so that these issues can vary independently.

Sometimes we say it this way: separate what something is from what it does, where both of these things vary for different reasons.*

* Design Patterns Explained: A New Perspective on Object-Oriented Design by Alan Shalloway and James R. Trott

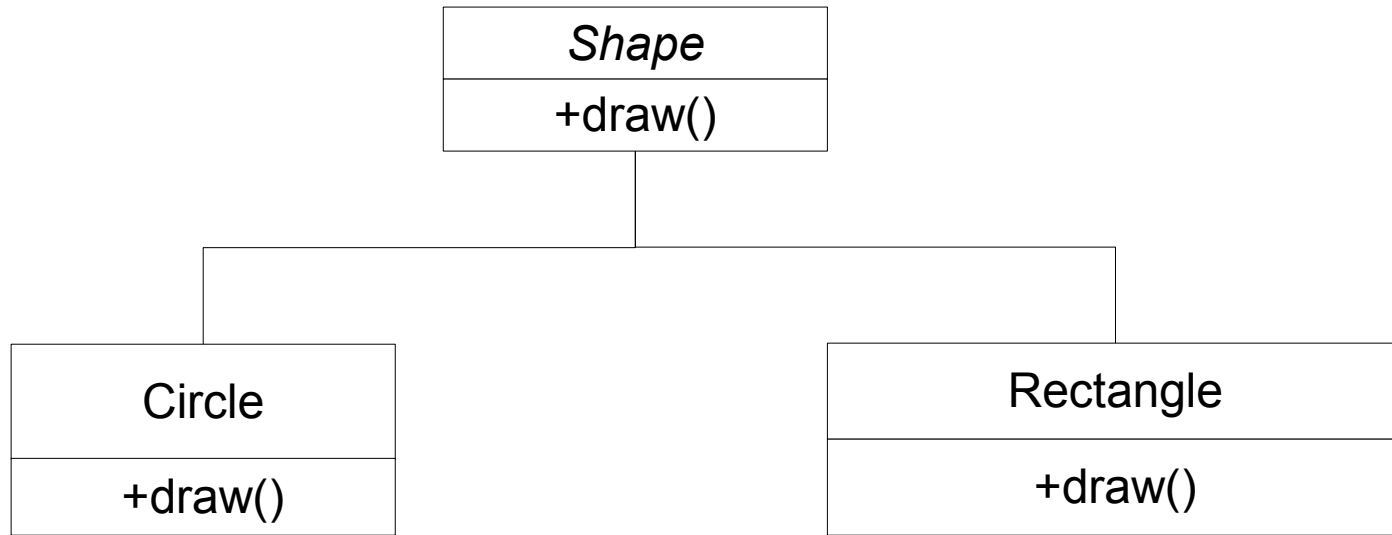
UML!!!



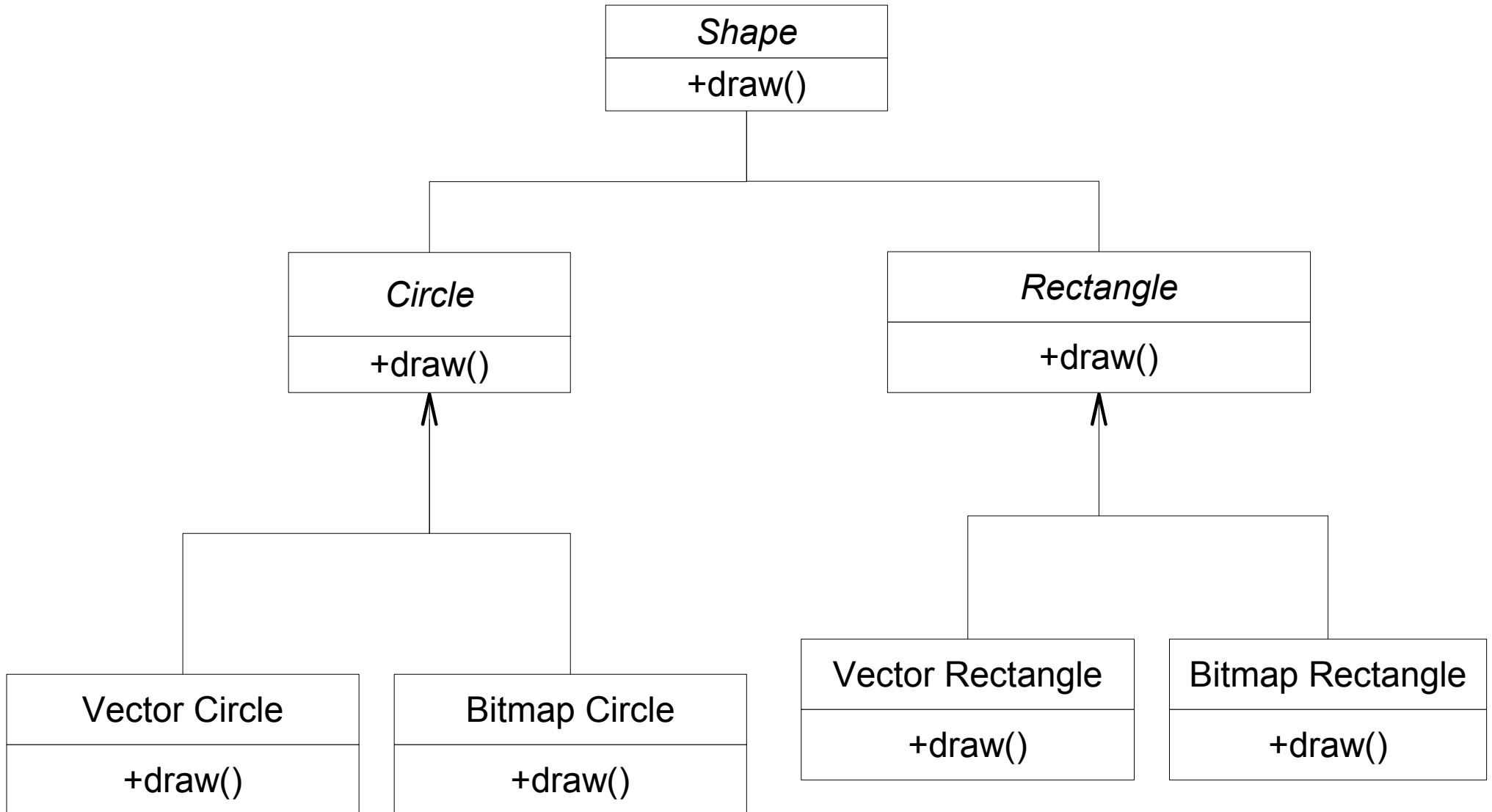
An Example

Circle
+draw()

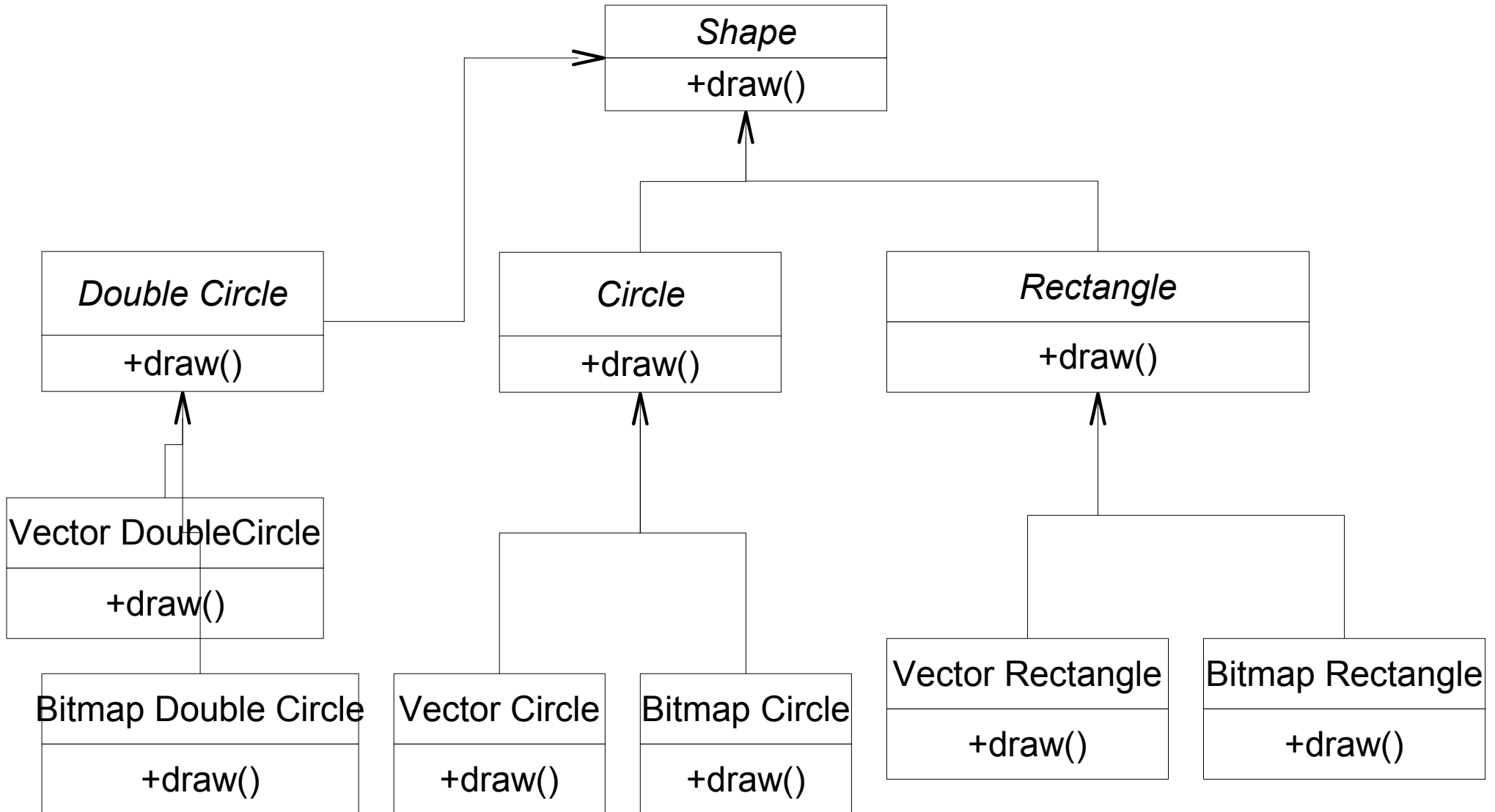
An Example



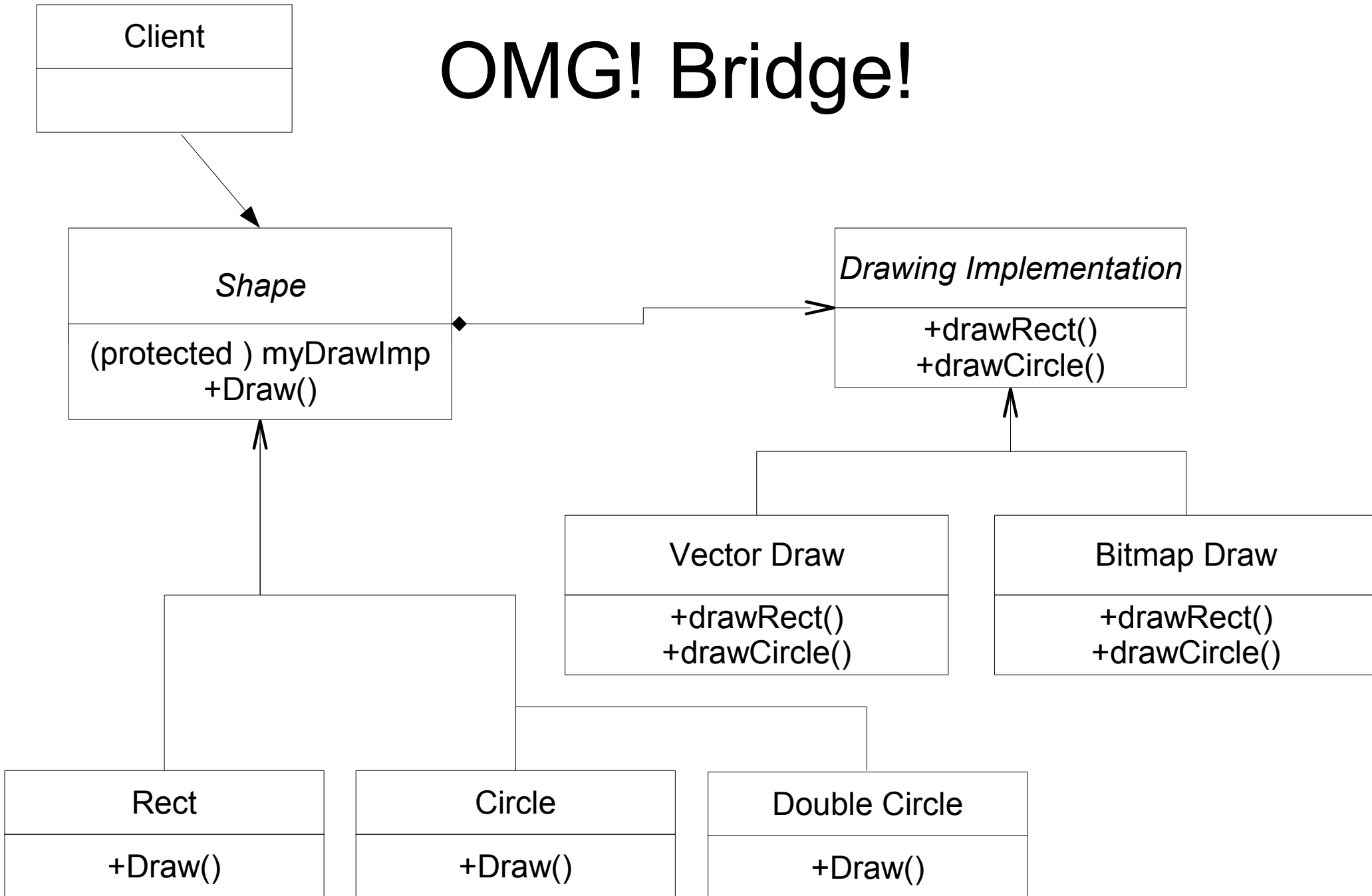
An Example



An Example



OMG! Bridge!



Similar Patterns

- What patterns is it similar to?
 - Strategy – but it supports varying the abstraction as well as the implementation (behavior)
 - Template – the abstraction can be handled like a template
 - It's kind of a combination of Strategy and Template
- What patterns does it frequently use?
 - Adapter – often times the supported implementations will have different interfaces