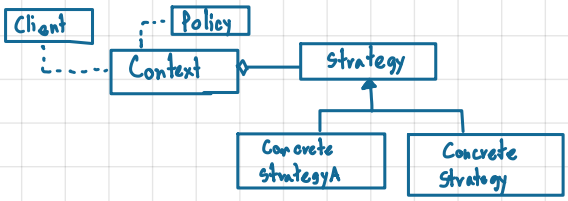
Implementing design patterns

Strategy is a behavioral design pattern that let you define a family of algo, put each of them into a seperate class, and make their objects interchangeable.

- Name
  - Strategy (aka Policy)
- Applicability
  - many related classes differ only in their behavior
  - many different variants of algorithm
  - need to encapsulate algo information

Structure



Strategy

- You want to
  - use different algorithms depending upon the context
  - avoid having to change the context or client
- *Strategy*
  - decouples interface from implementation
  - shields client from implementations
  - Context is not aware which strategy is being used; Client configures the Context
  - strategies can be substituted at runtime
  - example: interface to wired and wireless networks

Observer (Event - Subscriber, Listener)

Observer is a behavioral design pattern that lets you define a subscription mechanism to notify multiple objects about any events that happen to the object they're observing.