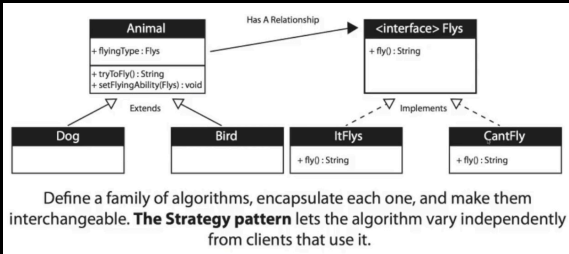


Strategy 算法=具封装

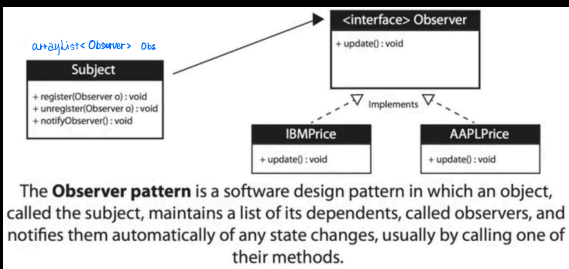


encapsulated
interchangeable
algorithms

When to Use the Strategy Pattern

- Often reduces long lists of conditionals
- Avoids duplicate code
- Keeps class changes from forcing other class changes
- Can hide complicated / secret code from the user
- Negative: Increased Number of Objects / Classes

Observer



△ use thread to update repeatedly

When to Use the Observer Pattern

- When you need many other objects to receive an update when another object changes
- Stock market with thousands of stocks needs to send updates to objects representing individual stocks
- The Subject (publisher) sends many stocks to the Observers
- The Observers (subscribers) takes the ones they want and use them

When to Use the Observer Pattern

- Loose coupling is a benefit
- The Subject (publisher) doesn't need to know anything about the Observers (subscribers)
- Negatives: The Subject (publisher) may send updates that don't matter to the Observer (subscriber)